# Django test AJAX autocomplete Documentation

***Release 0.2.1***

**pvergain@gmail.com**

**Jul 16, 2017**

# Contents

**See also:**

- https://django-test-autocomplete.readthedocs.io/en/latest/
- https://github.com/pvergain/django-test-autocomplete.git
- https://en.wikipedia.org/wiki/Ajax_(programming)
- https://en.wikipedia.org/wiki/JSON

**Dango test AJAX autocomplete**

**Projet** Dango test AJAX autocomplete

**Release** 0.2.1

**Date** Jul 16, 2017

**Authors** **pvergain**

- genindex
- search
- *Glossary*

Thanks https://readthedocs.org/

- https://readthedocs.org/projects/django-test-autocomplete/

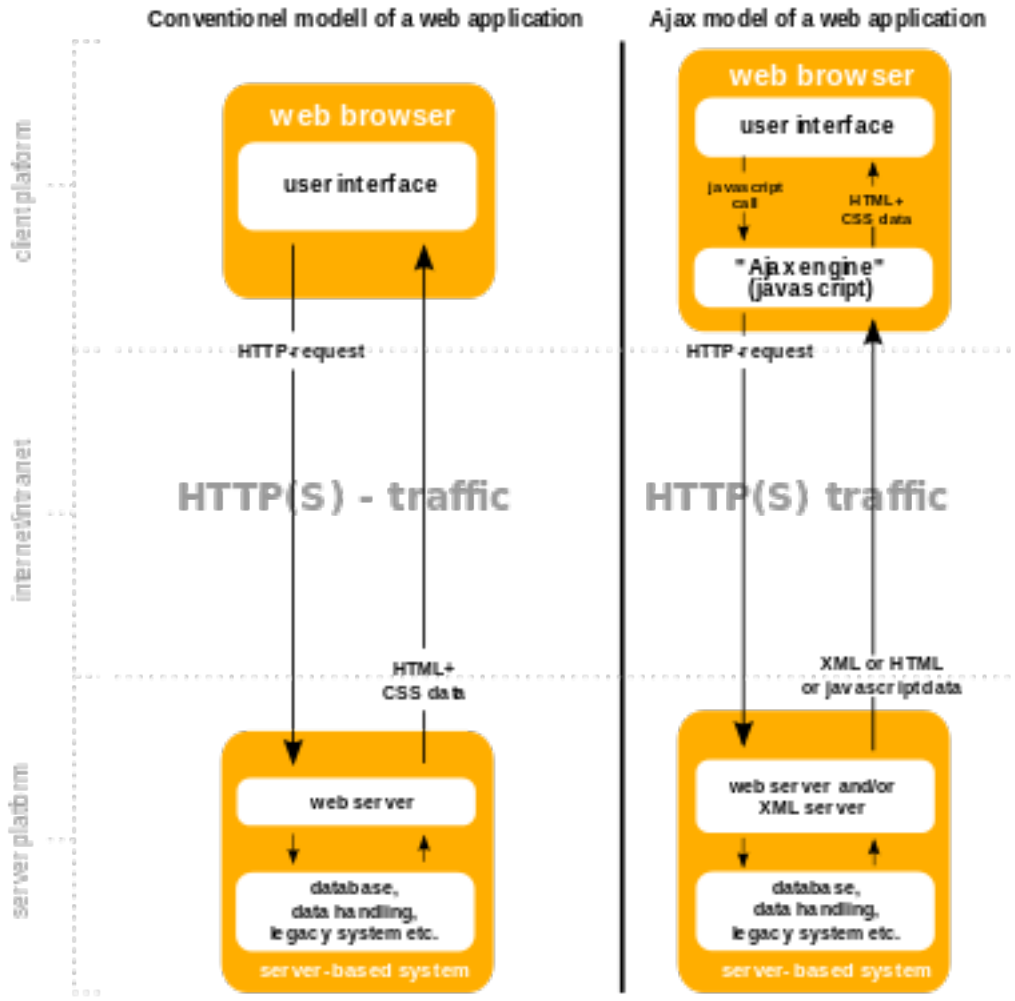



Fig. 1: *AJAX* source: https://en.wikipedia.org/wiki/Ajax_(programming)

Fig. 2: *JSON* logo (https://en.wikipedia.org/wiki/JSON)


Fig. 3: httpie logo (https://httpie.org/)

CHAPTER 1

# Autocomplete introduction

**Contents**



## Definitions

### Definition 1

**See also:**

- http://blog.appliedinformaticsinc.com/autocomplete-input-field-in-django-template-with-jquery-ui/

**Auto-complete is one of the most important aspects of modern web interface**.

Auto-complete feature is used to provide auto suggestion for users while entering input. We can create auto-complete using an AJAX call to make a list and display the list using javascript. Creating auto-complete with jquery and jquery-ui is the most efficient way of creating it.

### Definition 2

**See also:**

- http://api.jqueryui.com/autocomplete/

- http://api.jqueryui.com/autocomplete/#option-source

You can pull data in from a local or remote source: Local is good for small data sets, e.g., an address book with 50 entries; remote is necessary for big data sets, such as a database with hundreds or millions of entries to select from. To find out more about customizing the data source, see the documentation for the source option.

## Arguments

When using <*select*> you need to retrieve *all* values from the database before displaying the form. This is a potentially expensive operation on the server and delays the time when the user see the form.

CHAPTER 2

# Autocomplete jQuery plugins tests

**Contents**



## Specification

Write a form for selecting 'champion' for a project.

```
#!/usr/bin/python
# -*- coding: utf8 -*-
"""The project's models.

"""
from __future__ import unicode_literals
from django.utils.encoding import python_2_unicode_compatible

from django.db import models
from django.core.urlresolvers import reverse

# https://docs.djangoproject.com/en/dev/ref/contrib/auth/#user-model
from django.contrib.auth.models import User



@python_2_unicode_compatible
class Project(models.Model):
    """A project with a title and a champion which is the foreign key to the
```

```
    auth user.

Documentation
=============

- http://guiqinqian.blogspot.fr/2012/01/using-jquery-auto-complete-in-django.html

"""
title = models.CharField(max_length=200)
champion = models.ForeignKey(User)

def __str__(self):
    return "{} {}".format(self.title, self.champion)

def get_absolute_url(self):
    """
    https://docs.djangoproject.com/en/dev/ref/class-based-views/generic-editing/
    """
    return reverse('projects:project_update',
                   kwargs={'pk': self.pk})
```

# Test the different jQuery plugins

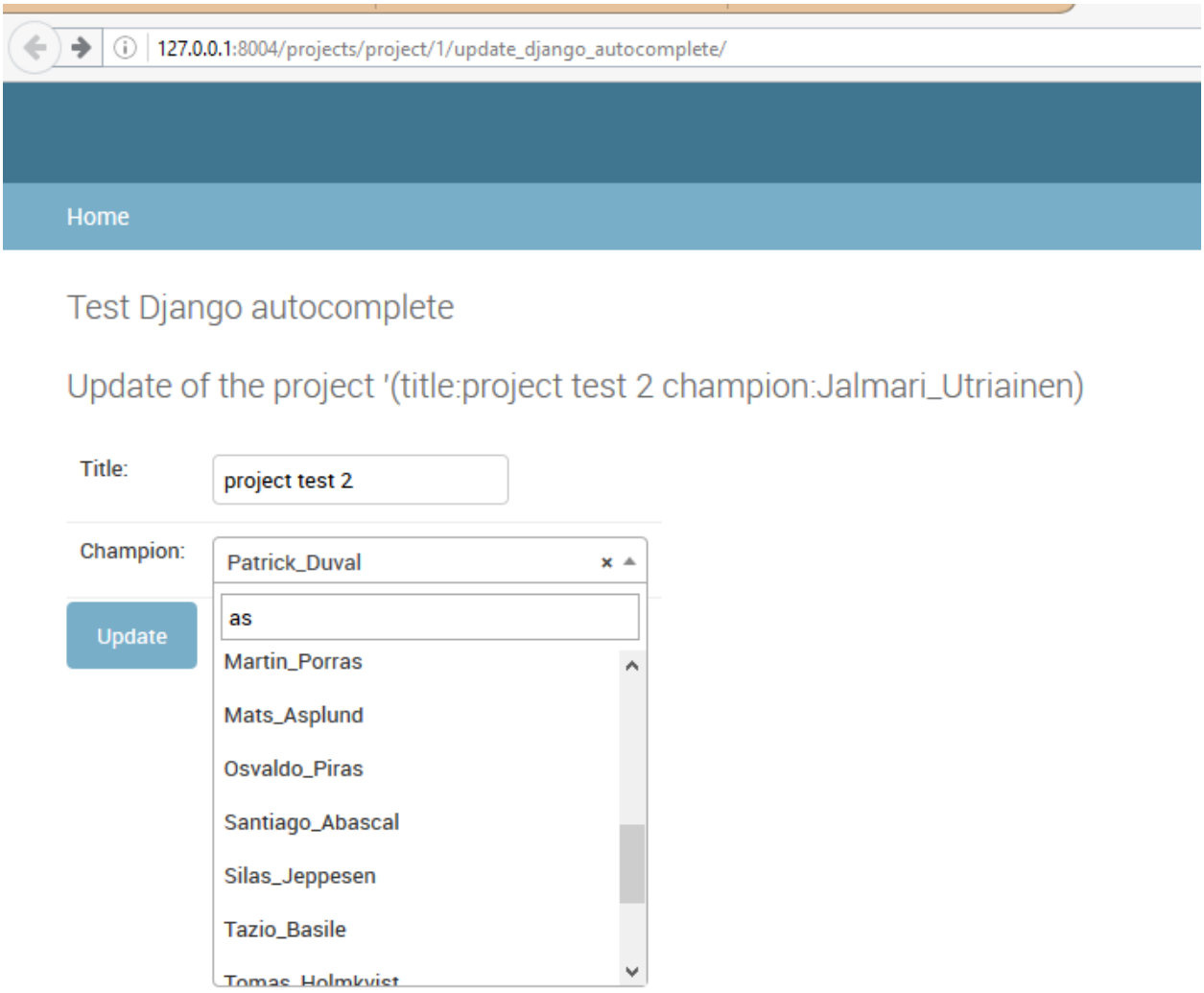
## django-autocomplete-light test

**See also:**

- https://github.com/yourlabs/django-autocomplete-light.git
- https://github.com/select2/select2.git
- https://github.com/django/django/blob/master/django/contrib/admin/templates/admin/base.html

**Contents**

        * *Django template*
    - *Test URL http://127.0.0.1:8004/projects/project/1/update_django_autocomplete*
    - *Passing options to select2*




### install

```
pip install django-autocomplete-light
```

### update the settings.py module

Add the 2 following applications:

- dal
- dal-select2

```
# Applications definition
INSTALLED_APPS = [
    'dal',
    'dal_select2',
```

### tree

```
C:\PROJECTS_ID3\DJANGO-TEST-AUTOCOMPLETE\ANNEXES\DJANGO-AUTOCOMPLETE-LIGHT
|   .gitignore
|   .gitmodules
|   .travis.yml
|   AUTHORS
|   CHANGELOG
|   CHANGES.txt
|   LICENSE
|   MANIFEST.in
|   pytest.ini
|   README
|   README.rst
|   setup.py
|   tox.ini
|
+---.openshift
|   |   README.md
|   |
|   +---action_hooks
|   |       deploy
|   |       README.md
|   |
|   \---cron
|       |   README.cron
|       |
|       \---hourly
|               reset_database
|
+---docs
|   |   api.rst
|   |   conf.py
|   |   creation.rst
|   |   genericm2m.rst
|   |   gfk.rst
|   |   gm2m.rst
|   |   index.rst
|   |   install.rst
|   |   Makefile
|   |   requirements.txt
|   |   tagging.rst
|   |   taggit.rst
|   |   tutorial.rst
|   |
|   +---img
|   |       all.png
|   |       autocomplete.png
|   |       created_option.png
|   |       create_option.png
|   |       mine.png
```

```
|   |       view.png
|   |
|   \---_ext
|           djangodocs.py
|
+---src
|   |   __init__.py
|   |
|   +---dal
|   |   |   autocomplete.py
|   |   |   forms.py
|   |   |   forward.py
|   |   |   views.py
|   |   |   widgets.py
|   |   |   __init__.py
|   |   |
|   |   +---static
|   |   |   +---admin
|   |   |   |   \---js
|   |   |   |           jquery.init.js
|   |   |   |
|   |   |   \---autocomplete_light
|   |   |           autocomplete.init.js
|   |   |           forward.js
|   |   |           jquery.init.js
|   |   |
|   |   \---test
|   |           case.py
|   |           stories.py
|   |           utils.py
|   |           __init__.py
|   |
|   +---dal_contenttypes
|   |       fields.py
|   |       __init__.py
|   |
|   +---dal_genericm2m
|   |       fields.py
|   |       __init__.py
|   |
|   +---dal_genericm2m_queryset_sequence
|   |       fields.py
|   |       __init__.py
|   |
|   +---dal_gm2m
|   |       fields.py
|   |       __init__.py
|   |
|   +---dal_gm2m_queryset_sequence
|   |       fields.py
|   |       __init__.py
|   |
|   +---dal_queryset_sequence
|   |   |   fields.py
|   |   |   views.py
|   |   |   widgets.py
|   |   |   __init__.py
|   |   |
```

```
|   |   \---tests
|   |           test_views.py
|   |
|   +---dal_select2
|   |   |   apps.py
|   |   |   checks.py
|   |   |   fields.py
|   |   |   models.py
|   |   |   test.py
|   |   |   views.py
|   |   |   widgets.py
|   |   |   __init__.py
|   |   |
|   |   \---static
|   |       \---autocomplete_light
|   |           |   select2.css
|   |           |   select2.js
|   |           |
|   |           \---vendor
|   |               \---select2
|   +---dal_select2_queryset_sequence
|   |       views.py
|   |       widgets.py
|   |       __init__.py
|   |
|   +---dal_select2_tagging
|   |       widgets.py
|   |       __init__.py
|   |
|   \---dal_select2_taggit
|           widgets.py
|           __init__.py
|
\---test_project
    |   .coveragerc
    |   db.sqlite3
    |   manage.py
    |   requirements.txt
    |   reset.json
    |   urls.py
    |   views.py
    |   wsgi.py
    |
    +---linked_data
    |   |   admin.py
    |   |   apps.py
    |   |   forms.py
    |   |   models.py
    |   |   test_forms.py
    |   |   test_functional.py
    |   |   urls.py
    |   |   __init__.py
    |   |
    |   +---migrations
    |   |       0001_initial.py
    |   |       __init__.py
    |   |
    |   \---static
```

```
    |           linked_data.js
    |
    +---rename_forward
    |   |   admin.py
    |   |   apps.py
    |   |   forms.py
    |   |   models.py
    |   |   test_forms.py
    |   |   test_functional.py
    |   |   urls.py
    |   |   __init__.py
    |   |
    |   +---migrations
    |   |       0001_initial.py
    |   |       __init__.py
    |   |
    |   \---static
    |           linked_data.js
    |
    +---secure_data
    |   |   admin.py
    |   |   apps.py
    |   |   forms.py
    |   |   models.py
    |   |   test_functional.py
    |   |   urls.py
    |   |   views.py
    |   |   __init__.py
    |   |
    |   \---migrations
    |           0001_initial.py
    |           __init__.py
    |
    +---select2_foreign_key
    |   |   admin.py
    |   |   apps.py
    |   |   forms.py
    |   |   models.py
    |   |   test_functional.py
    |   |   urls.py
    |   |   __init__.py
    |   |
    |   \---migrations
    |           0001_initial.py
    |           __init__.py
    |
    +---select2_generic_foreign_key
    |   |   admin.py
    |   |   apps.py
    |   |   forms.py
    |   |   models.py
    |   |   test_forms.py
    |   |   test_functional.py
    |   |   urls.py
    |   |   views.py
    |   |   __init__.py
    |   |
    |   \---migrations
```

```
    |           0001_initial.py
    |           __init__.py
    |
    +---select2_generic_m2m
    |   |   admin.py
    |   |   apps.py
    |   |   forms.py
    |   |   models.py
    |   |   test_forms.py
    |   |   test_functional.py
    |   |   urls.py
    |   |   views.py
    |   |   __init__.py
    |   |
    |   \---migrations
    |           0001_initial.py
    |           __init__.py
    |
    +---select2_gm2m
    |   |   admin.py
    |   |   apps.py
    |   |   forms.py
    |   |   models.py
    |   |   test_forms.py
    |   |   test_functional.py
    |   |   urls.py
    |   |   views.py
    |   |   __init__.py
    |   |
    |   \---migrations
    |           0001_initial.py
    |           __init__.py
    |
    +---select2_list
    |   |   admin.py
    |   |   forms.py
    |   |   models.py
    |   |   test_fields.py
    |   |   test_functional.py
    |   |   test_views.py
    |   |   urls.py
    |   |   views.py
    |   |   __init__.py
    |   |
    |   \---migrations
    |           0001_initial.py
    |           __init__.py
    |
    +---select2_many_to_many
    |   |   admin.py
    |   |   apps.py
    |   |   forms.py
    |   |   models.py
    |   |   test_functional.py
    |   |   urls.py
    |   |   __init__.py
    |   |
    |   \---migrations
```

```
    |           0001_initial.py
    |           __init__.py
    |
    +---select2_one_to_one
    |   |   admin.py
    |   |   apps.py
    |   |   forms.py
    |   |   models.py
    |   |   test_functional.py
    |   |   urls.py
    |   |   __init__.py
    |   |
    |   \---migrations
    |           0001_initial.py
    |           __init__.py
    |
    +---select2_outside_admin
    |   |   urls.py
    |   |   views.py
    |   |   __init__.py
    |   |
    |   \---templates
    |           select2_outside_admin.html
    |
    +---select2_tagging
    |   |   admin.py
    |   |   forms.py
    |   |   models.py
    |   |   test_forms.py
    |   |   test_functional.py
    |   |   urls.py
    |   |   __init__.py
    |   |
    |   \---migrations
    |           0001_initial.py
    |           0002_testmodel_test.py
    |           __init__.py
    |
    +---select2_taggit
    |   |   admin.py
    |   |   forms.py
    |   |   models.py
    |   |   test_forms.py
    |   |   test_functional.py
    |   |   urls.py
    |   |   __init__.py
    |   |
    |   \---migrations
    |           0001_initial.py
    |           __init__.py
    |
    +---settings
    |       base.py
    |       __init__.py
    |
    +---templates
    |   |   base.html
    |   |
```

```
    |   \---admin
    |           login.html
    |
    \---tests
            admin.py
            models.py
            test_widgets.py
            __init__.py
```

### projects/views_django_autocomplete_light.py

```
from django.contrib.auth.models import User

from django.db.models import Q
from dal import autocomplete



class ApiUserDjangoAutocompleteLight(autocomplete.Select2QuerySetView):
    """https://django-autocomplete-light.readthedocs.io/en/master/tutorial.html"""
    def get_queryset(self):
        # Don't forget to filter out results depending on the visitor !
        users = User.objects.all()

        if self.q:
            users = User.objects.filter(Q(username__icontains=self.q)
                                        | Q(email__icontains=self.q)).order_by(
↪'username')

        return users
```

### projects/api/get_users/?q (projects/urls.py)

```
url(r'^api/get_users/$',
    ApiUserDjangoAutocompleteLight.as_view(),
    name='api_get_users'),
```

### test

```
{"pagination": {"more": true}, "results": [{"text": "Aaron_De Angelis", "id": 119},
{"text": "Abdul_Parker", "id": 177}, {"text": "Adolf_Wahlberg", "id": 338},
{"text": "Adrian_Jenkins", "id": 184}, {"text": "Aim\u00e9_Roussel", "id": 42},
{"text": "Alejandro_Hoyos", "id": 240}, {"text": "Alexander_Dahlberg", "id": 335},
{"text": "Alexandria_Weiss", "id": 29}, {"text": "Alicia_Pareja", "id": 149},
{"text": "Alighieri_Coppola", "id": 216}]}
```

### projects/forms_django_autocomplete_light.py

```
from django import forms
```

```
from .models import Project

from dal import autocomplete


class ProjectFormDjangoAutocomplete(forms.ModelForm):
    """https://django-autocomplete-light.readthedocs.io/en/master/tutorial.html"""
    class Meta:
        model = Project
        fields = ('__all__')
        widgets = {
            'champion': autocomplete.ModelSelect2(url='projects:api_get_users')
        }
```

### projects/admin.py

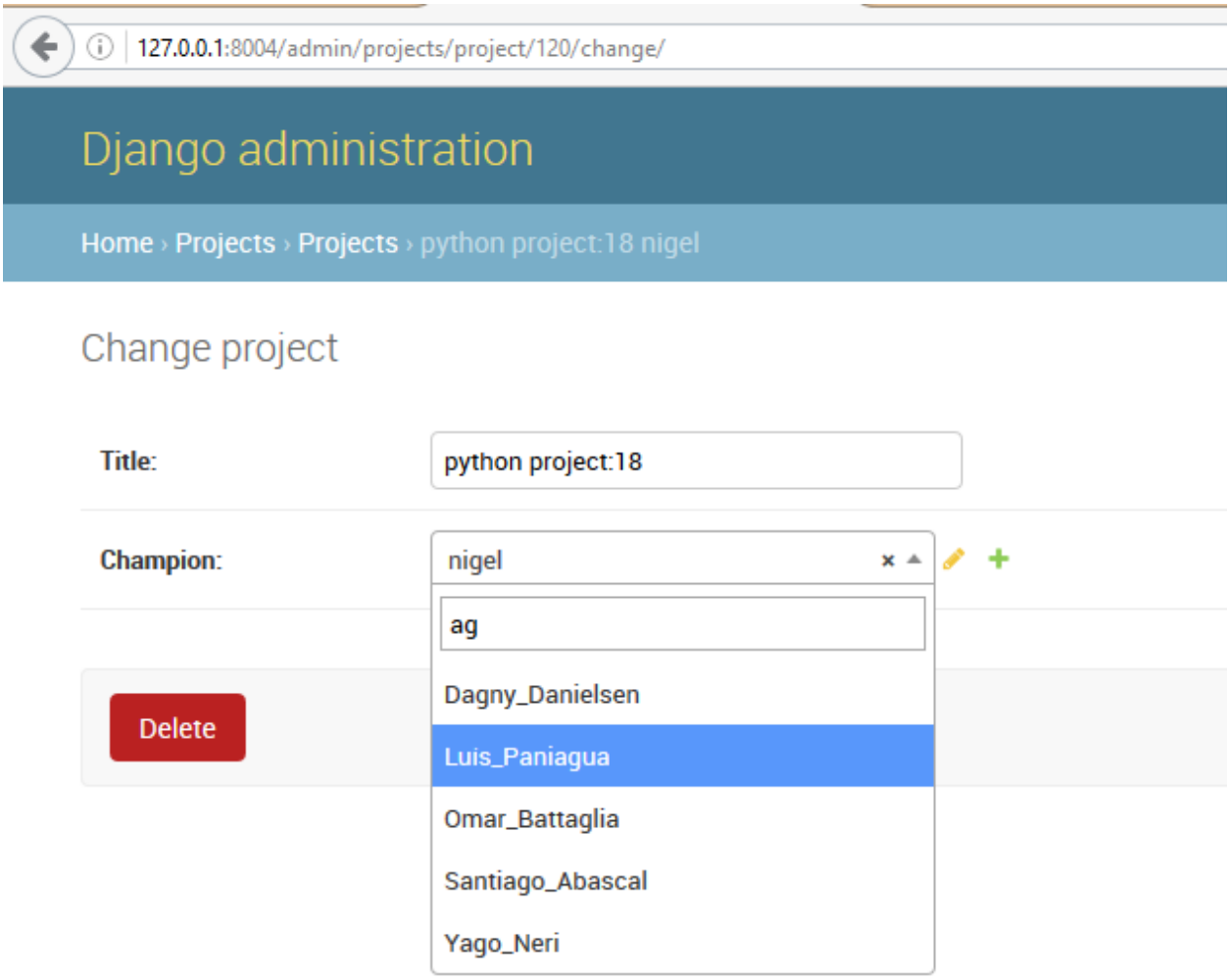
```
#!/usr/bin/python
# -*- coding: UTF-8 -*-
"""Project Administration.

"""

from django.contrib import admin

from .models import Project

from .forms_django_autocomplete_light import ProjectFormDjangoAutocomplete

@admin.register(Project)
class ProjectAdmin(admin.ModelAdmin):
    """Project administration

    Documentation
    =============

    - https://docs.djangoproject.com/en/dev/ref/contrib/admin/#modeladmin-objects

    """
    form = ProjectFormDjangoAutocomplete
    list_display = ('title', 'champion')
    search_fields = ('title', 'champion')
    list_filter = ('title', 'champion')
```

### Using autocompletes in the admin : nice !

### Using autocompletes outside the admin

### projects/views_django_autocomplete_light.py

```
from django.contrib.auth.models import User

from django.db.models import Q
from django.views.generic.edit import UpdateView
```


127.0.0.1:8004/admin/projects/project/120/change/
Django administration
Home › Projects › Projects › python project:18 nigel
Change project
Title:
python project:18
Champion:
nigel
ag
Dagny_Danielsen
Luis_Paniagua
Omar_Battaglia
Santiago_Abascal
Yago_Neri
Delete

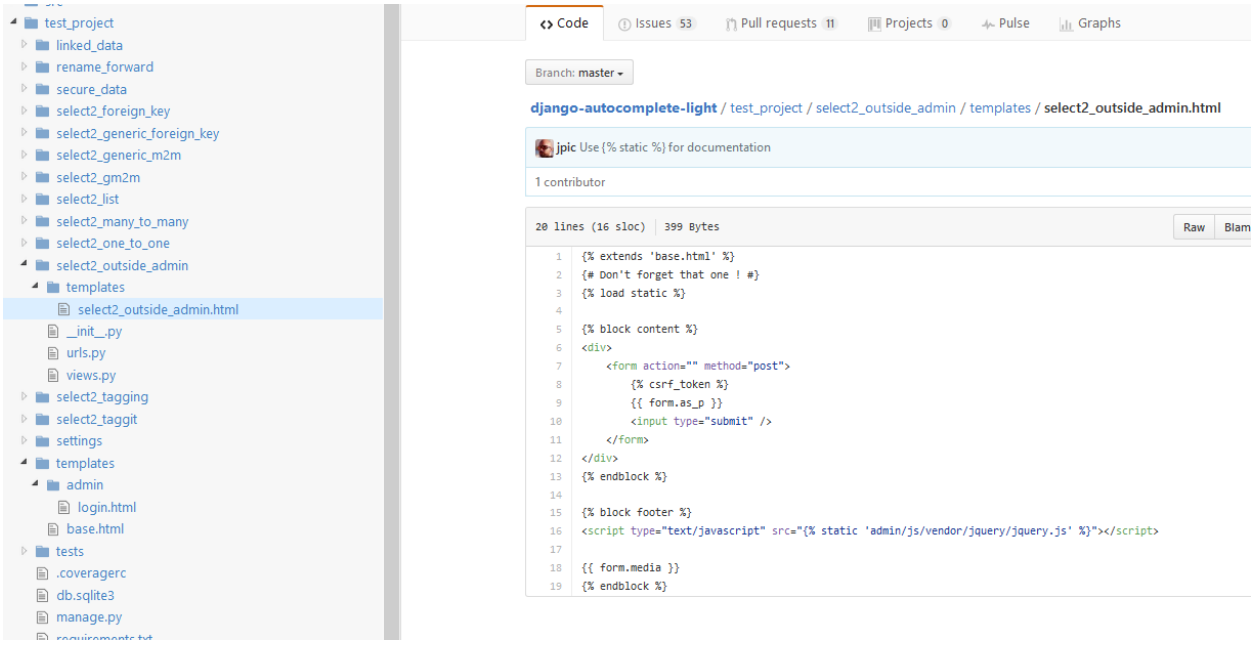
```
from dal import autocomplete

from .models import Project

from .forms_django_autocomplete_light import ProjectFormDjangoAutocomplete

# Get an instance of a logger
logger = logging.getLogger(__name__)


class ApiUserDjangoAutocompleteLight(autocomplete.Select2QuerySetView):
    """https://django-autocomplete-light.readthedocs.io/en/master/tutorial.html"""
    def get_queryset(self):
        # Don't forget to filter out results depending on the visitor !
        users = User.objects.all()
        if self.q:
            users = User.objects.filter(Q(username__icontains=self.q)
                                        | Q(email__icontains=self.q)).order_by(
↪'username')

        return users


class ProjectDjangoAutoCompleteUpdateView(UpdateView):
    """Update the view with the jQuery UI Autocomplete plugin.

    Documentation:

    - http://ccbv.co.uk/projects/Django/1.10/django.views.generic.edit/UpdateView/

    """
    model = Project
    form_class = ProjectFormDjangoAutocomplete
    context_object_name = 'project'
    template_name = 'projects/project/update_django_autocomplete_light.html'

    def get_object(self, queryset=None):
        """Pour mémoriser self.demande_article"""
        self.object = super(ProjectDjangoAutoCompleteUpdateView, self).get_
↪object(queryset)
        return self.object

    def post(self, request, *args, **kwargs):
        logger.warning("Hello from ProjectDjangoAutoCompleteUpdateView !")
        return super(ProjectDjangoAutoCompleteUpdateView, self).post(request, *args,␣
↪**kwargs)
```

### Django template

**See also:**

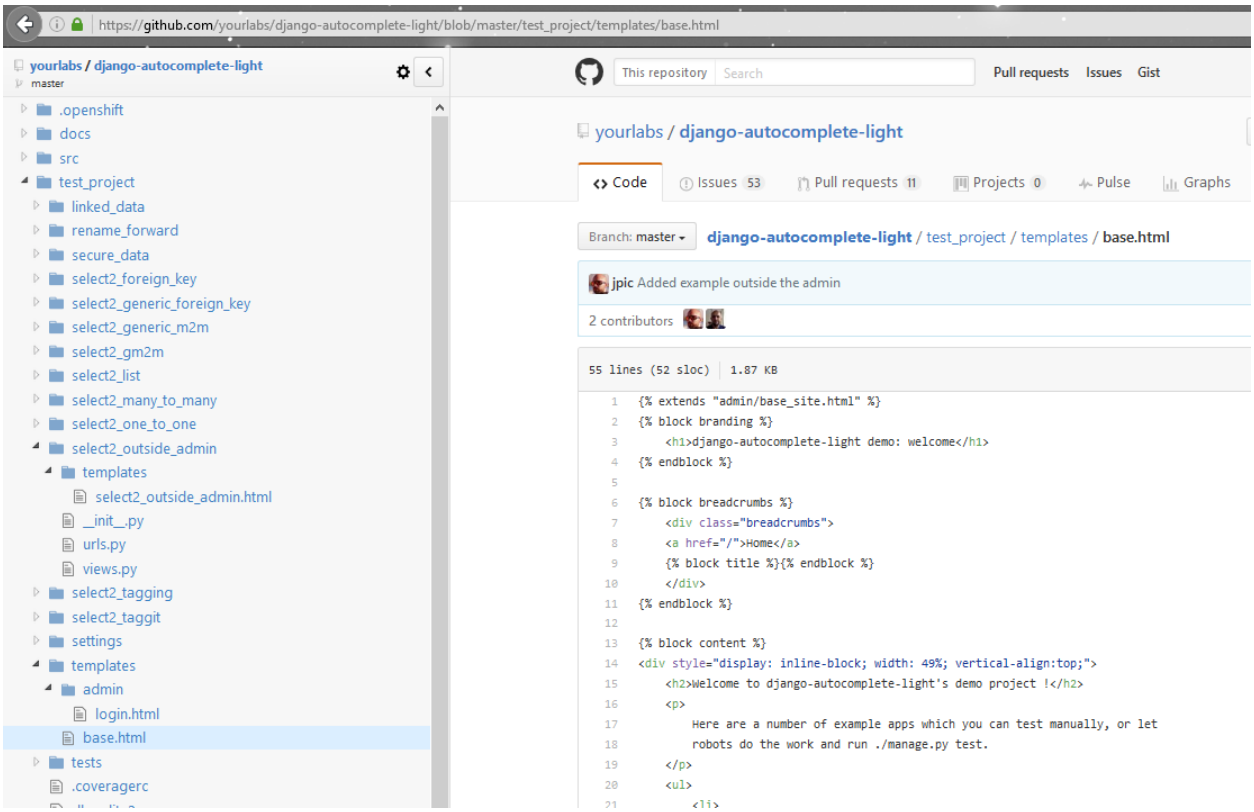
- https://github.com/django/django/blob/master/django/contrib/admin/templates/admin/base.html

```
{# inherit from the django admin base.html file #}
{##############################################}
{% extends "admin/base_site.html" %}
```


test_project
linked_data
rename_forward
secure_data
select2_foreign_key
select2_generic_foreign_key
select2_generic_m2m
select2_gm2m
select2_list
select2_many_to_many
select2_one_to_one
select2_outside_admin
templates
select2_outside_admin.html
__init__.py
urls.py
views.py
select2_tagging
select2_taggit
settings
templates
admin
login.html
base.html
tests
.coveragerc
db.sqlite3
manage.py
Code
Issues 53
Pull requests 11
Projects 0
Pulse
Graphs
Branch: master
django-autocomplete-light / test_project / select2_outside_admin / templates / select2_outside_admin.html
jpic Use {% static %} for documentation
1 contributor
20 lines (16 sloc) 399 Bytes
Raw
Blam
{% extends 'base.html' %}
{# Don't forget that one ! #}
{% load static %}
{% block content %}

<form action="" method="post">
{% csrf_token %}
{{ form.as_p }}
<input type="submit" />
</form>

{% endblock %}
{% block footer %}
<script type="text/javascript" src="{% static 'admin/js/vendor/jquery/jquery.js' %}"></script>
{{ form.media }}
{% endblock %}


https://github.com/yourlabs/django-autocomplete-light/blob/master/test_project/templates/base.html
yourlabs / django-autocomplete-light
master
This repository
Search
Pull requests
Issues
Gist
.openshift
docs
src
test_project
linked_data
rename_forward
secure_data
select2_foreign_key
select2_generic_foreign_key
select2_generic_m2m
select2_gm2m
select2_list
select2_many_to_many
select2_one_to_one
select2_outside_admin
templates
select2_outside_admin.html
__init__.py
urls.py
views.py
select2_tagging
select2_taggit
settings
templates
admin
login.html
base.html
tests
.coveragerc
yourlabs / django-autocomplete-light
Code
Issues 53
Pull requests 11
Projects 0
Pulse
Graphs
Branch: master
django-autocomplete-light / test_project / templates / base.html
jpic Added example outside the admin
2 contributors
55 lines (52 sloc) 1.87 KB
{% extends "admin/base_site.html" %}
{% block branding %}
<h1>django-autocomplete-light demo: welcome</h1>
{% endblock %}
{% block breadcrumbs %}

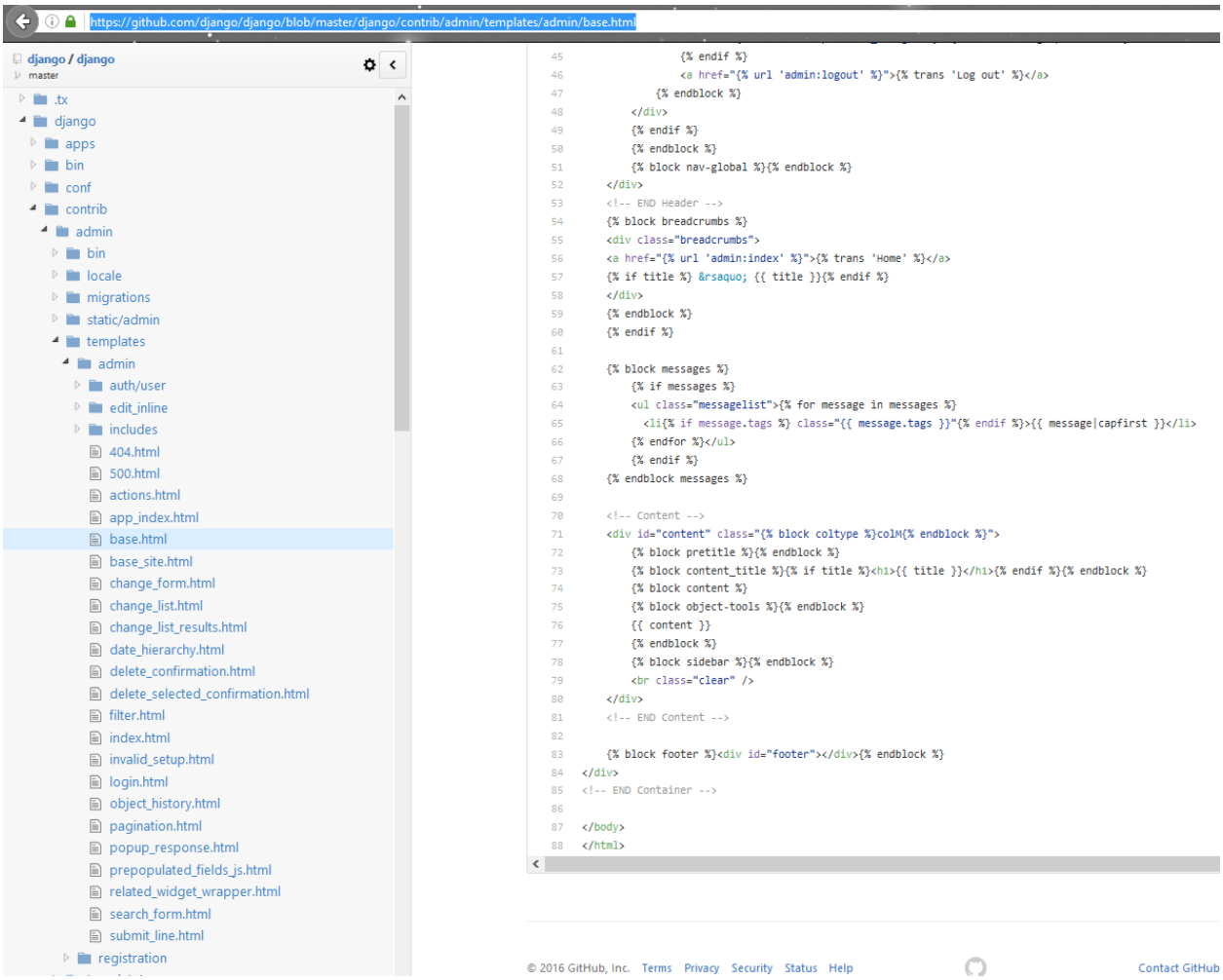
<a href="/">Home</a>
{% block title %}{% endblock %}

{% endblock %}
{% block content %}

<h2>Welcome to django-autocomplete-light's demo project !</h2>

Here are a number of example apps which you can test manually, or let
robots do the work and run ./manage.py test.

<ul>


https://github.com/django/django/blob/master/django/contrib/admin/templates/admin/base.html
django / django
master
.tx
django
apps
bin
conf
contrib
admin
bin
locale
migrations
static/admin
templates
admin
auth/user
edit_inline
includes
404.html
500.html
actions.html
app_index.html
base.html
base_site.html
change_form.html
change_list.html
change_list_results.html
date_hierarchy.html
delete_confirmation.html
delete_selected_confirmation.html
filter.html
index.html
invalid_setup.html
login.html
object_history.html
pagination.html
popup_response.html
prepopulated_fields_js.html
related_widget_wrapper.html
search_form.html
submit_line.html
registration
{% endif %}
<a href="{% url 'admin:logout' %}">{% trans 'Log out' %}</a>
{% endblock %}

{% endif %}
{% endblock %}
{% block nav-global %}{% endblock %}

<!-- END Header -->
{% block breadcrumbs %}

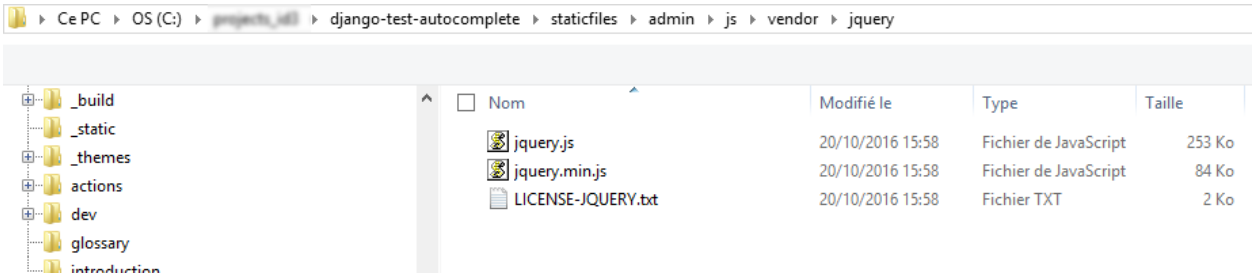
<a href="{% url 'admin:index' %}">{% trans 'Home' %}</a>
{% if title %} › {{ title }}{% endif %}

{% endblock %}
{% endif %}
{% block messages %}
{% if messages %}
<ul class="messagelist">{% for message in messages %}
<li{% if message.tags %} class="{{ message.tags }}"{% endif %}>{{ message|capfirst }}</li>
{% endfor %}</ul>
{% endif %}
{% endblock messages %}
<!-- Content -->

{% block pretitle %}{% endblock %}
{% block content_title %}{% if title %}<h1>{{ title }}</h1>{% endif %}{% endblock %}
{% block content %}
{% block object-tools %}{% endblock %}
{{ content }}
{% endblock %}
{% block sidebar %}{% endblock %}
<br class="clear" />

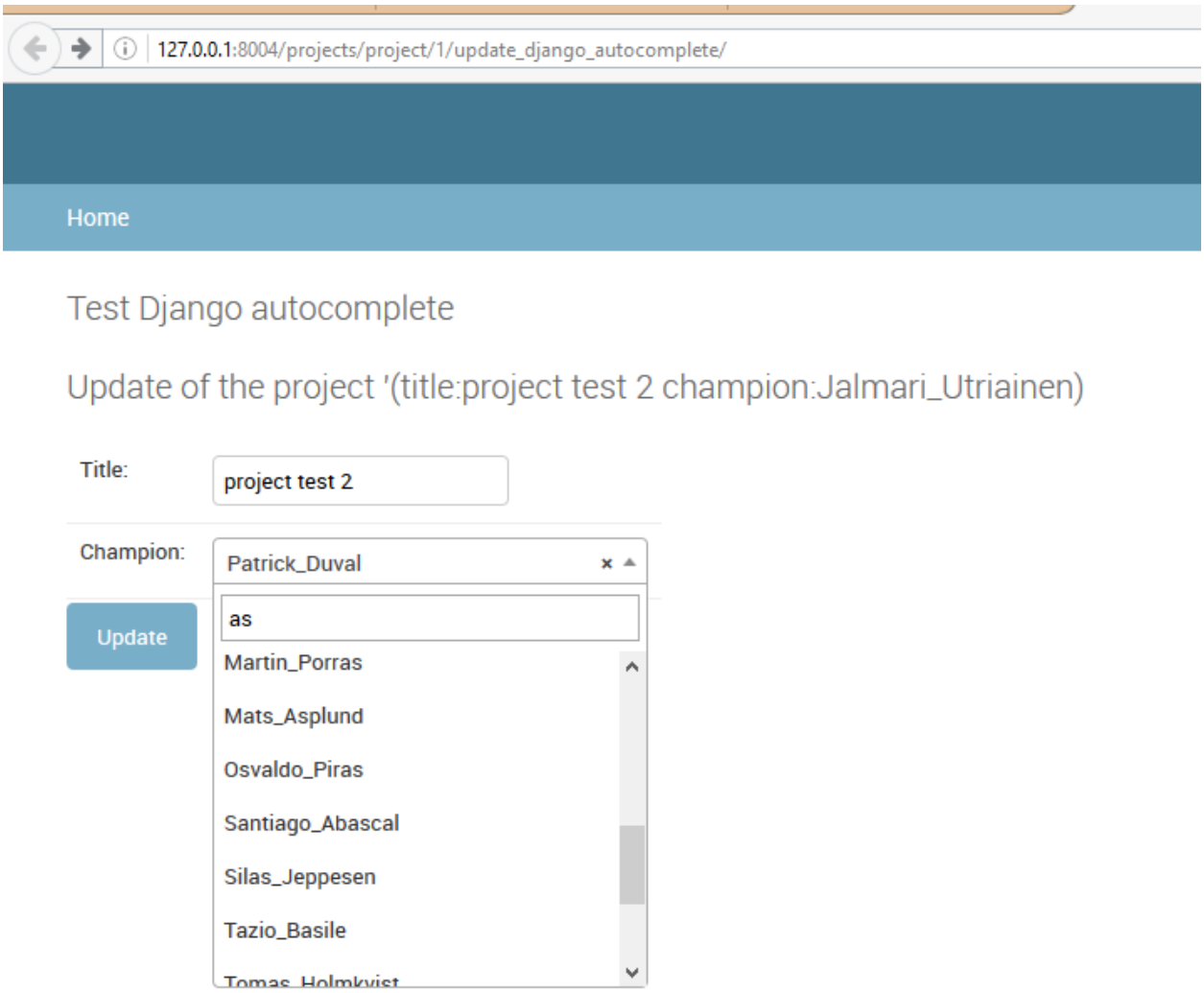
<!-- END Content -->
{% block footer %} {% endblock %}

<!-- END Container -->
</body>
</html>
© 2016 GitHub, Inc. Terms Privacy Security Status Help
Contact GitHub


Ce PC
OS (C:)
django-test-autocomplete
staticfiles
admin
js
vendor
jquery
_build
_static
_themes
actions
dev
glossary
Nom
Modifié le
Type
Taille
jquery.js
20/10/2016 15:58
Fichier de JavaScript
253 Ko
jquery.min.js
20/10/2016 15:58
Fichier de JavaScript
84 Ko
LICENSE-JQUERY.txt
20/10/2016 15:58
Fichier TXT
2 Ko

```
{% load static %}
{% load staticfiles %}

{% block branding %}
        <title>Django autocomplete  example</title>
{% endblock %}


{% block content %}
<!-- STRUCTURE -> HTML5 elements -->
    <h1>Test Django autocomplete</h1>
    <h1>Update of the project '(title:{{ project.title }} champion:{{ project.
↪champion.username }}) </h1>
    <p></p>
    <p></p>
    {# https://docs.djangoproject.com/en/dev/topics/forms/ #}
    <form id="id_form_project" action="{% url 'projects:project_update' project.id %}
↪" method="post">
        {% csrf_token %}
        <div class="forms">
            {{ form.id }}
            {{ form.non_field_errors }}
            {# Include the hidden fields #}
            {% for hidden in form.hidden_fields %}
                {# here we will have the champion filed (which is hidden) #}
                {{ hidden }}
            {% endfor %}
            <table id="id_table" class="table table-hover table-bordered table-
↪condensed">
                <tbody>
                        <tr>
                            <td class="text-right">Title:</td>
                            <td>{{ form.title }}</td>
                        </tr>
                        <tr>
                            <td class="text-right">Champion:</td>
                            <td> {{ form.champion }}  </td>
                        </tr>
                </tbody>
            </table>
        </div>
        <input type="submit" name="btn_update"  value="Update" class="btn btn-success␣
↪btn-block" />
    </form>
<!-- end STRUCTURE-->


{% block footer %}
{# https://github.com/yourlabs/django-autocomplete-light/blob/master/test_project/
↪select2_outside_admin/templates/select2_outside_admin.html #}
<script type="text/javascript" src="{% static 'admin/js/vendor/jquery/jquery.js' %}">
↪</script>

{{ form.media }}
{% endblock footer %}
```

**Test URL http://127.0.0.1:8004/projects/project/1/update_django_autocomplete**




### Passing options to select2

Select2 supports a bunch of options. These options may be set in data-* attributes.

For example:

```
# Instanciate a widget with a bunch of options for select2:
autocomplete.ModelSelect2(
    url='select2_fk',
    attrs={
        # Set some placeholder
        'data-placeholder': 'Autocomplete ...',
        # Only trigger autocompletion after 3 characters have been typed
        'data-minimum-input-length': 3,
    },
)
```

## jQuery EasyAutocomplete test

**Contents**



### Running the local Django web server

```
System check identified no issues (0 silenced).
October 24, 2016 - 10:41:29
Django version 1.10.2, using settings 'projet_ajax.settings'
Starting development server at http://127.0.0.1:8004/
Quit the server with CTRL-BREAK.
```

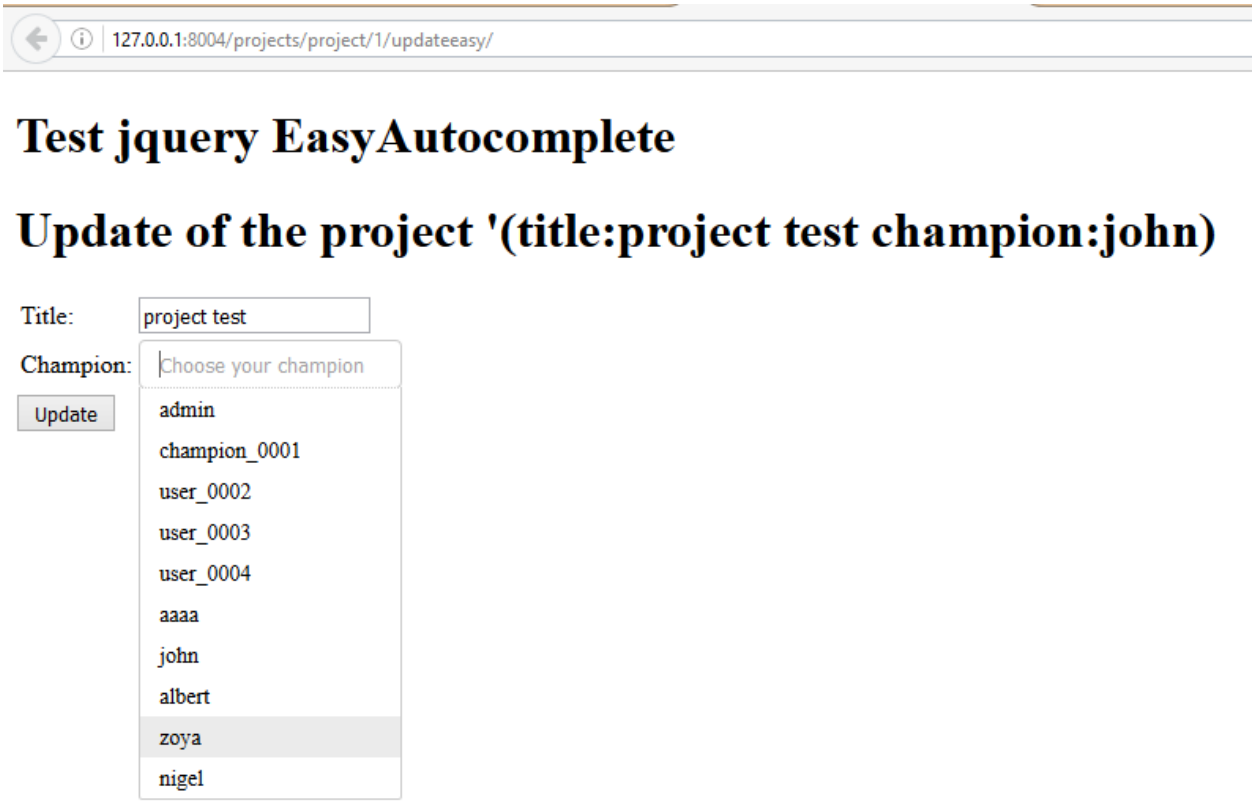
### Testing the JSON API view

```
http://127.0.0.1:8004/projects/api_get_champions/?term=a
```

```
[{"id": 6, "value": "aaaa", "label": "aaaa"},
{"id": 1, "value": "admin", "label": "admin"},
{"id": 8, "value": "albert", "label": "albert"},
{"id": 2, "value": "champion_0001", "label": "champion_0001"},
{"id": 7, "value": "john", "label": "john"},
{"id": 10, "value": "nigel", "label": "nigel"},
{"id": 9, "value": "zoya", "label": "zoya"}]
```

### Testing the form, step1

```
http://127.0.0.1:8004/projects/project/1/updateeasy
```


127.0.0.1:8004/projects/project/1/updateeasy/
Test jquery EasyAutocomplete
Update of the project '(title:project test champion:john)
Title:
project test
Champion:
Choose your champion
Update
admin
champion_0001
user_0002
user_0003
user_0004
aaaa
john
albert
zoya
nigel

### projects/urls.py

```
url(r'^project/(?P<pk>\d+)/updateeasy/$',
    ProjectUpdateViewEasyAutoComplete.as_view(),
    name='project_update_easy'),
```

### The forms.py part

```
#!/usr/bin/python
# -*- coding: utf8 -*-
"""The project's forms.

"""

from django import forms

from .models import Project

class ProjectChampionForm(forms.ModelForm):
    """The champion project form"""
    champions_choice_list = forms.CharField(max_length=100,
                                            help_text='type username or email')

    class Meta:
        model = Project
        fields = ('title',
                  'champions_choice_list', 'champion',)

    def __init__(self, *args, **kwargs):
        super(ProjectChampionForm, self).__init__(*args, **kwargs)
        self.fields['champions_choice_list'].label = "Update the champion"
        self.fields['champion'].widget = forms.HiddenInput()
```

### The HTML and JavaScript part

```
<body>

        <!-- STRUCTURE -> HTML5 elements -->
            <h1>Test jquery EasyAutocomplete</h1>
            <h1>Update of the project '(title:{{ project.title }} champion:{{ project.
↪champion }}) </h1>
            <p></p>
            <p></p>
            {# https://docs.djangoproject.com/en/dev/topics/forms/ #}
            <form id="id_form_project_update_easy" action="{% url 'projects:project_
↪update' project.id %}" method="post">
                {% csrf_token %}
                <div class="forms">
                    {{ form.id }}
                    {{ form.non_field_errors }}
                    {# Include the hidden fields #}
                    {% for hidden in form.hidden_fields %}
                        {# here we will have the champion filed (which is hidden) #}
```

```
                        {{ hidden }}
                    {% endfor %}
                    <table id="id_table" class="table table-hover table-bordered␣
↪table-condensed">
                        <tbody>
                                <tr>
                                    <td class="text-right">Title:</td>
                                    <td>{{ form.title }}</td>
                                </tr>
                                <tr>
                                    <td class="text-right">Champion:</td>
                                    <td> {{ form.champions_choice_list }}  </td>
                                </tr>
                        </tbody>
                    </table>
                </div>
                <input type="submit" name="btn_update"  value="Update" class="btn btn-
↪success btn-block" />
            </form>
        <!-- end STRUCTURE-->

        <!--BEHAVIOR -> Javascript scripts-->
            <!-- Using jQuery with a CDN -->
            <script src="//code.jquery.com/jquery-1.11.2.js"></script>
            <script src="{% static 'easyautocomplete/js/jquery.easy-autocomplete.js'
↪%}" type="text/javascript"></script>

            <script>
                var options_easy_autocomplete_champions = {
                    {# we have to build this URL: http://127.0.0.1:8004/projects/api_
↪get_champions/?term=a #}
                    url: function(term) {
                        return "{% url 'projects:api_get_champions' %}" + "?term=" +␣
↪term;
                    },
                    getValue: "value",
                    list: {
                        maxNumberOfElements: 200,
                        onSelectItemEvent: function() {
                            var champion_id = $("#id_champions_choice_list").
↪getSelectedItemData().id;

                            {# put the id in the champion field #}
                            $("#id_champion")
                                .val(champion_id)
                                .trigger("change");
                        },
                        match: {
                            enabled: true
                        },
                    },
                    placeholder: "Choose your champion"
                };

                {# build the autocomplete list for the champions_choice_list #}
                $("#id_champions_choice_list").easyAutocomplete(options_easy_
↪autocomplete_champions);
```

```
            </script>
        <!-- end BEHAVIOR -->

    </body>
```

### Step2 : initialize the easyAutocomplete placeholder with the value of champion


Add these jQuery lines:

```
{# build the autocomplete list for the champions_choice_list #}
$("#id_champions_choice_list").easyAutocomplete(options_easy_autocomplete_champions);
{# Get the value of the former champion (from database) #}
{# Thanks http://stackoverflow.com/questions/1643227/get-selected-text-from-a-drop-
↪down-list-select-box-using-jquery #}
var champion_name = $("#id_champion option:selected").text();
{# replace the placeholder by the value coming from database #}
$("#id_champions_choice_list").attr('placeholder', champion_name);
```

### Step3 : hide the select champion form field

### Before the update

in the HTML file:

```
<td class="text-right">Champion value:</td>
<td> <select id="id_champion" name="champion" required>
<option value="">---------</option>
<option value="1">admin</option>
<option value="2">champion_0001</option>
<option value="3">user_0002</option>
<option value="4">user_0003</option>
<option value="5">user_0004</option>
<option value="6">aaaa</option>
<option value="7">john</option>
<option value="8">albert</option>
<option value="9" selected="selected">zoya</option>
```

```
<option value="10">nigel</option>
</select>
```

in the projects/forms.py file:

```
def __init__(self, *args, **kwargs):
    super(ProjectChampionForm, self).__init__(*args, **kwargs)
    self.fields['champions_choice_list'].label = "Update the champion"
    # self.fields['champion'].widget = forms.HiddenInput()
```

### After the update

in the projects/forms.py file:

```
def __init__(self, *args, **kwargs):
    super(ProjectChampionForm, self).__init__(*args, **kwargs)
    self.fields['champions_choice_list'].label = "Update the champion"
    self.fields['champion'].widget = forms.HiddenInput()
```

New jQuery lines:

```
{# build the autocomplete list for the champions_choice_list #}
$("#id_champions_choice_list").easyAutocomplete(options_easy_autocomplete_champions);

var champion_name = "{{ project.champion.username }}"
$("#id_champions_choice_list").attr('placeholder', champion_name);
```


## jQuery UI autocomplete test

**See also:**

- https://jqueryui.com/autocomplete/
- https://github.com/jquery/jquery-ui/tree/master/demos
- https://github.com/jquery/jquery-ui/blob/master/demos/autocomplete/maxheight.html
- https://www.npmjs.com/package/jquery-ui.autocomplete.scroll

- https://github.com/anseki/jquery-ui-autocomplete-scroll

**Contents**



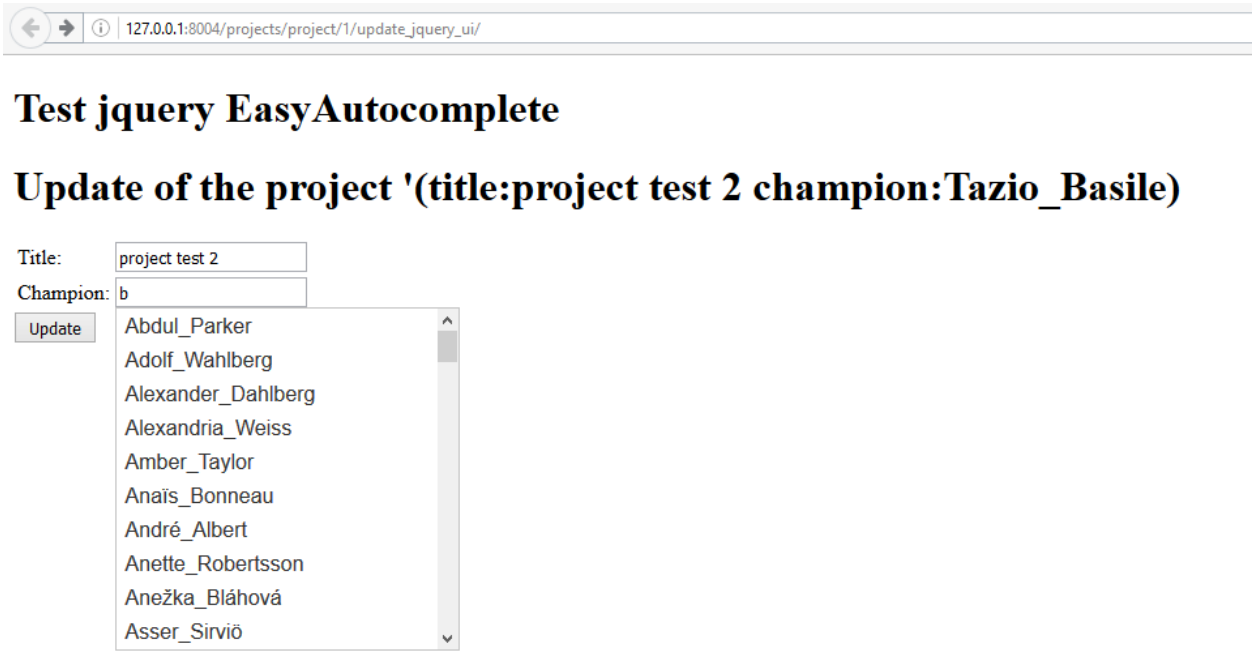



Fig. 2.1: jQuery UI with scrollbar (thanks https://github.com/anseki)

### Running the local Django web server

```
System check identified no issues (0 silenced).
October 26, 2016 - 08:54:53
Django version 1.10.2, using settings 'projet_ajax.settings'
Starting development server at http://127.0.0.1:8004/
Quit the server with CTRL-BREAK.
```

### The projects/forms.py module

```
#!/usr/bin/python
# -*- coding: utf8 -*-
"""The project's forms.

"""

from django import forms

from .models import Project

class ProjectChampionForm(forms.ModelForm):
    """The champion project form"""
    champions_choice_list = forms.CharField(max_length=100,
                                            help_text='type username or email')

    class Meta:
        model = Project
        fields = ('title',
                  'champions_choice_list', 'champion',)

    def __init__(self, *args, **kwargs):
        super(ProjectChampionForm, self).__init__(*args, **kwargs)
        self.fields['champions_choice_list'].label = "Update the champion"
        self.fields['champion'].widget = forms.HiddenInput()
```

### The HTML and JavaScript part (the Django Template)

```
{% load static %}
{% load staticfiles %}

<!DOCTYPE html>
<html lang="en">
    <head>
        <title>jQuery UI complete simple example</title>

        <!-- STYLE -> CSS -->
        {# https://jqueryui.com/autocomplete/ #}
        <link rel="stylesheet" href="//code.jquery.com/ui/1.12.1/themes/base/jquery-
↪ui.css" type="text/css">
        <!-- end STYLE-->

    </head>
    <body>

        <!-- STRUCTURE -> HTML5 elements -->
            <h1>Test jquery EasyAutocomplete</h1>
            <h1>Update of the project '(title:{{ project.title }} champion:{{ project.
↪champion.username }}) </h1>
            <p></p>
            <p></p>
            {# https://docs.djangoproject.com/en/dev/topics/forms/ #}
            <form id="id_form_project_update_easy" action="{% url 'projects:project_
↪update' project.id %}" method="post">
                {% csrf_token %}
```

```
                <div class="forms">
                    {{ form.id }}
                    {{ form.non_field_errors }}
                    {# Include the hidden fields #}
                    {% for hidden in form.hidden_fields %}
                        {# here we will have the champion filed (which is hidden) #}
                        {{ hidden }}
                    {% endfor %}
                    <table id="id_table" class="table table-hover table-bordered␣
↪table-condensed">
                        <tbody>
                                <tr>
                                    <td class="text-right">Title:</td>
                                    <td>{{ form.title }}</td>
                                </tr>
                                <tr>
                                    <td class="text-right">Champion:</td>
                                    <td> {{ form.champions_choice_list }}  </td>
                                </tr>
                        </tbody>
                    </table>
                </div>
                <input type="submit" name="btn_update"  value="Update" class="btn btn-
↪success btn-block" />
            </form>
        <!-- end STRUCTURE-->

        <!--BEHAVIOR -> Javascript scripts-->
            <!-- Using jQuery with a CDN -->
            <script src="//code.jquery.com/jquery-1.11.2.js"></script>
            <script src="//cdnjs.cloudflare.com/ajax/libs/jqueryui/1.12.1/jquery-ui.js
↪"></script>
            {# https://github.com/anseki/jquery-ui-autocomplete-scroll #}
            <script src="{% static 'jquery_ui/js/jquery.ui.autocomplete.scroll.min.js
↪' %}" type="text/javascript"></script>
            <script>
                var options_jquery_ui_autocomplete_champions = {
                    {# we have to build this URL: http://127.0.0.1:8004/projects/api_
↪get_champions/?term=a #}
                    {# calling the JSON view (this is the very famous AJAX call) the␣
↪+ "?term=" + term; is added by autocomplete #}
                    maxShowItems: 10,
                    source: "{% url 'projects:api_get_champions' %}",
                    minLength: 1,
                    select: function( event, ui ) {
                        {# put the id in the champion field #}
                        let champion_id = ui.item.id;
                        $("#id_champion")
                            .val(champion_id)
                            .trigger("change");
                        console.log( "Selected: " + ui.item.value + " aka " + ui.item.
↪id );
                    }
                };

                {# build the autocomplete list for the champions_choice_list #}
                $("#id_champions_choice_list").autocomplete(options_jquery_ui_
↪autocomplete_champions);
```

```
                var champion_name = "{{ project.champion.username }}"
                $("#id_champions_choice_list").attr('placeholder', champion_name);

            </script>
        <!-- end BEHAVIOR -->

    </body>

</html>
```

### The projects/views.py module

```
class ProjectUpdateViewJQueryUIAutoComplete(UpdateView):
    """Update the view with the jQuery UI Autocomplete plugin.

    Documentation:

    - http://ccbv.co.uk/projects/Django/1.10/django.views.generic.edit/UpdateView/

    """
    model = Project
    form_class = ProjectChampionForm
    context_object_name = 'project'
    template_name = 'projects/project/update_jquery_ui_autocomplete.html'

    def get_object(self, queryset=None):
        """Pour mémoriser self.demande_article"""
        self.object = super(ProjectUpdateViewJQueryUIAutoComplete, self).get_
↪object(queryset)
        return self.object

    def post(self, request, *args, **kwargs):
        logger.warning("Hello from ProjectUpdateViewJQueryUIAutoComplete !")
        return super(ProjectUpdateViewJQueryUIAutoComplete, self).post(request, *args,
↪ **kwargs)
```

### The projects/urls.py module

```
urlpatterns = [
     url(r'^project/(?P<pk>\d+)/update/$',
         ProjectUpdateView.as_view(),
         name='project_update'),

    url(r'^project/(?P<pk>\d+)/updateeasy/$',
        ProjectUpdateViewEasyAutoComplete.as_view(),
        name='project_update_easy'),

    url(r'^project/(?P<pk>\d+)/update_jquery_ui/$',
        ProjectUpdateViewJQueryUIAutoComplete.as_view(),
        name='project_update_jquery_ui'),
```

### The jQuery UI scrollbar module

**See also:**

*jquery_ui_autocomplete_scroll plugin*

CHAPTER 3

# Actions

## Actions 2016

### 2016-10-25 installing gitsome

**See also:**

- *django-autocomplete-light test*
- https://github.com/donnemartin/gitsome
- https://github.com/integrations/gitsome
- https://twitter.com/donne_martin

#### install

```
(django_test_autocomplete_35_64) C:\projects_id3\django-test-autocomplete>pip install␣
↪gitsome


Collecting gitsome
  Downloading gitsome-0.6.0.tar.gz (308kB)
    100% |################################| 317kB 819kB/s
Collecting numpydoc<1.0,>=0.5 (from gitsome)
  Downloading numpydoc-0.6.0.tar.gz
Collecting ply<4.0,>=3.4 (from gitsome)
  Downloading ply-3.9.tar.gz (150kB)
    100% |################################| 153kB 1.3MB/s
Requirement already satisfied (use --upgrade to upgrade): prompt-toolkit<1.1.0,>=1.0.
↪0 in c:\project\python_envs\django_test_autocomplete_35_64\lib\site-packages (from␣
↪gitsome)
Requirement already satisfied (use --upgrade to upgrade): requests<3.0.0,>=2.8.1 in␣
↪c:\project\python_envs\django_test_autocomplete_35_64\lib\site-packages (from␣
↪gitsome)
```

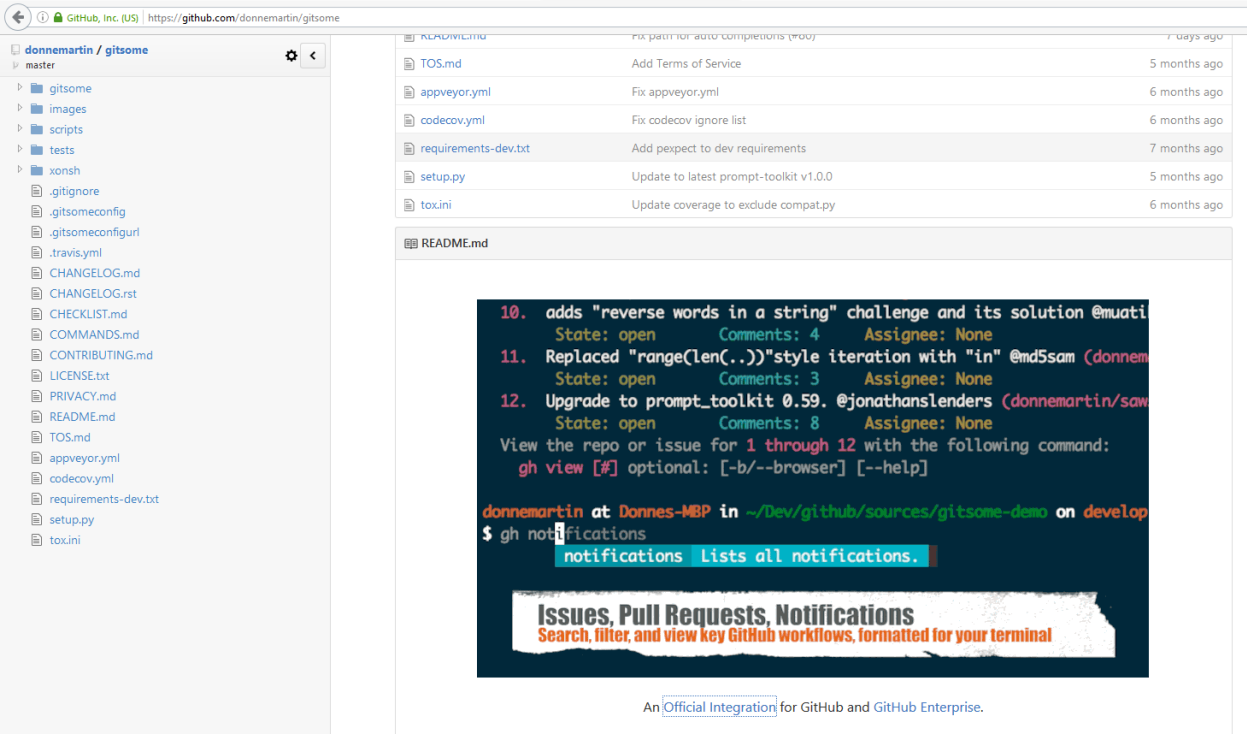



Fig. 3.1: An Official Integration for GitHub and GitHub Enterprise.

```
Requirement already satisfied (use --upgrade to upgrade): colorama<1.0.0,>=0.3.3 in␣
↪c:\project\python_envs\django_test_autocomplete_35_64\lib\site-packages (from␣
↪gitsome)
Collecting click<7.0,>=5.1 (from gitsome)
Requirement already satisfied (use --upgrade to upgrade): pygments<3.0.0,>=2.0.2 in␣
↪c:\project\python_envs\django_test_autocomplete_35_64\lib\site-packages (from␣
↪gitsome)
Collecting feedparser<6.0.0,>=5.2.1 (from gitsome)
  Downloading feedparser-5.2.1.zip (1.2MB)
    100% |################################| 1.2MB 655kB/s
Requirement already satisfied (use --upgrade to upgrade): pytz<2017.0,>=2016.3 in␣
↪c:\project\python_envs\django_test_autocomplete_35_64\lib\site-packages (from␣
↪gitsome)
Collecting docopt<1.0.0,>=0.6.2 (from gitsome)
Collecting uritemplate.py<1.0.0,>=0.2.0 (from gitsome)
  Downloading uritemplate.py-0.3.0.tar.gz
Requirement already satisfied (use --upgrade to upgrade): wcwidth in␣
↪c:\project\python_envs\django_test_autocomplete_35_64\lib\site-packages (from␣
↪prompt-toolkit<1.1.0,>=1.0.0->gitsome)
Requirement already satisfied (use --upgrade to upgrade): six>=1.9.0 in␣
↪c:\project\python_envs\django_test_autocomplete_35_64\lib\site-packages (from␣
↪prompt-toolkit<1.1.0,>=1.0.0->gitsome)
Building wheels for collected packages: gitsome, numpydoc, ply, feedparser,␣
↪uritemplate.py
  Running setup.py bdist_wheel for gitsome ... done
  Stored in directory:␣
↪C:\Users\pvergain\AppData\Local\pip\Cache\wheels\c8\4f\5a\128527ed75a5c07780bcef517c1be9a031b0f4672
  Running setup.py bdist_wheel for numpydoc ... done
  Stored in directory:␣
↪C:\Users\pvergain\AppData\Local\pip\Cache\wheels\9d\ca\8c\9be286b195791bbbfebfd407899de13eb12725a2c
```

```
  Running setup.py bdist_wheel for ply ... done
  Stored in directory:␣
↪C:\Users\pvergain\AppData\Local\pip\Cache\wheels\c1\0c\bd\306a63396decbe8353a4a056fcba97a092be0e035
  Running setup.py bdist_wheel for feedparser ... done
  Stored in directory:␣
↪C:\Users\pvergain\AppData\Local\pip\Cache\wheels\15\ce\10\b500f745822ea6db6ea8ed225c06b15c000d71016
  Running setup.py bdist_wheel for uritemplate.py ... done
  Stored in directory:␣
↪C:\Users\pvergain\AppData\Local\pip\Cache\wheels\f0\28\64\52c28cc4428d1c79eb7e923cade677f5e63ae1d2f
Successfully built gitsome numpydoc ply feedparser uritemplate.py
Installing collected packages: numpydoc, ply, click, feedparser, docopt, uritemplate.
↪py, gitsome
Successfully installed click-6.6 docopt-0.6.2 feedparser-5.2.1 gitsome-0.6.0 numpydoc-
↪0.6.0 ply-3.9 uritemplate.py-0.3.0
```

## 2016-10-25 testing jquery-ui autocomplete and django-autocomplete-light

**See also:**

- *django-autocomplete-light test*

### Conclusion

We adopt *django-autocomplete-light*.

---

**Note:** With jqueru-ui autocomplete we have discovered https://github.com/anseki/jquery-ui-autocomplete-scroll and https://github.com/anseki who has a deep knowledge of JavaScript.

---

## 2016-10-25 Building fake users with the Python faker module

**See also:**

- *Python faker module*
- http://blog.districtdatalabs.com/a-practical-guide-to-anonymizing-datasets-with-python-faker

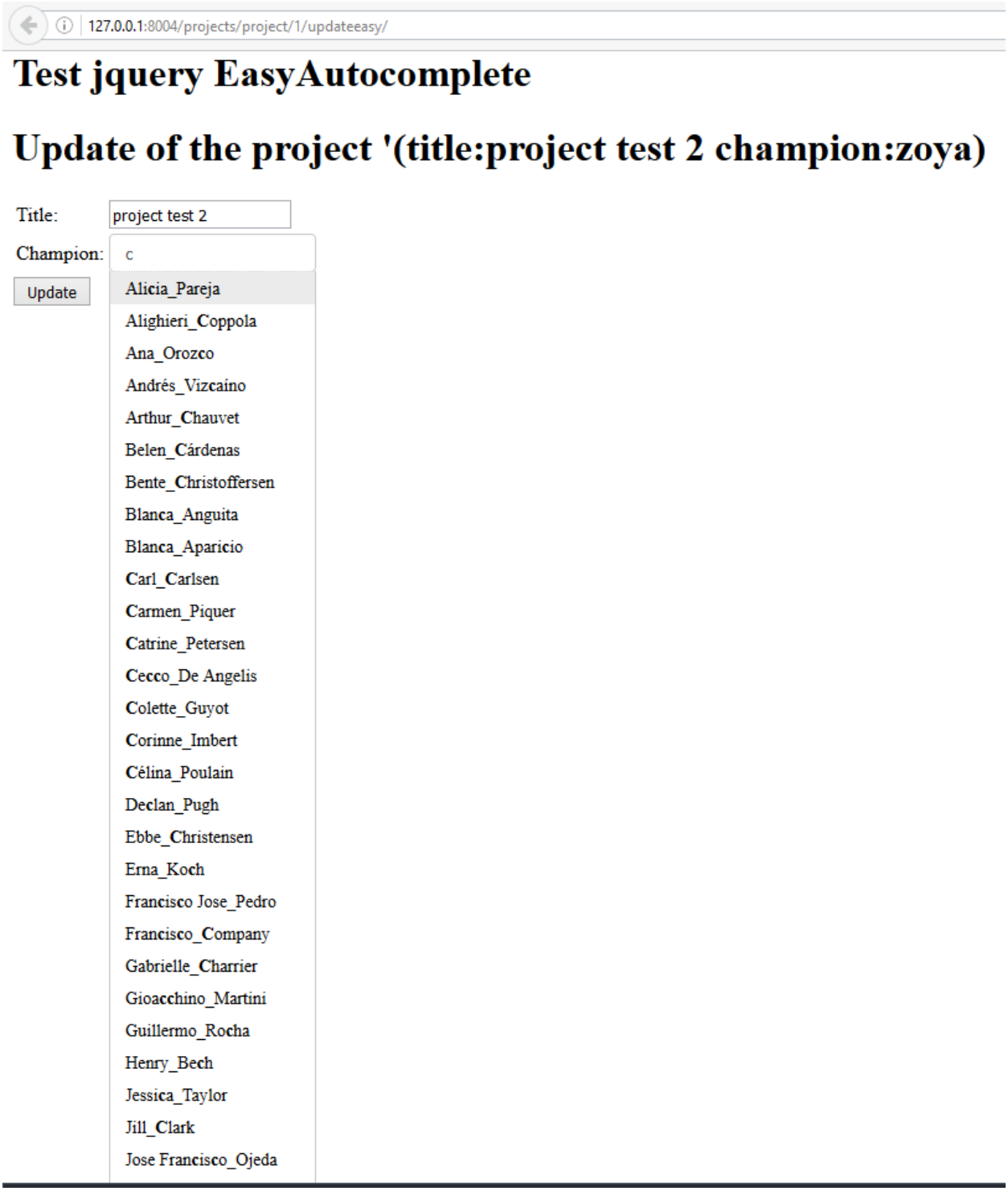
### Motivation

We want to test the autocomplete test with a large number of users. So, we want to create *fake* users.

## 2016-10-24 Learning jQuery 101

**See also:**

- *Step2 : initialize the easyAutocomplete placeholder with the value of champion*
- *2016-10-21 Receive 2 new excellent books about jQuery: jQuery In Action and jquery UI in Action*


127.0.0.1:8004/projects/project/1/updateeasy/
Test jquery EasyAutocomplete
Update of the project '(title:project test 2 champion:zoya)
Title:
project test 2
Champion:
c
Update
Alicia_Pareja
Alighieri_Coppola
Ana_Orozco
Andrés_Vizcaíno
Arthur_Chauvet
Belen_Cárdenas
Bente_Christoffersen
Blanca_Anguita
Blanca_Aparicio
Carl_Carlsen
Carmen_Piquer
Catrine_Petersen
Cecco_De Angelis
Colette_Guyot
Corinne_Imbert
Célina_Poulain
Declan_Pugh
Ebbe_Christensen
Erna_Koch
Francisco Jose_Pedro
Francisco_Company
Gabrielle_Charrier
Gioacchino_Martini
Guillermo_Rocha
Henry_Bech
Jessica_Taylor
Jill_Clark
Jose Francisco_Ojeda

## 2016-10-21 Receive 2 new excellent books about jQuery: `jQuery In Action` and `jquery UI in Action`

**See also:**

- *2016-10-21 Receive 2 new excellent books about jQuery: jQuery In Action and jquery UI in Action*

## 2016-10-21 testing the jquery EasyAutocomplete (EAC) options for an AJAX call

**Contents**



### Renaming views

Rename champion_get_json into api_get_champions

### Add a great number of projects into the 'project' table

**See also:**

- https://github.com/fcurella/django-fakery
- http://blog.districtdatalabs.com/a-practical-guide-to-anonymizing-datasets-with-python-faker

We can create a Django command or a simple loop in the command line interface.

```
user = User.objects.last()
```

```
for i in range(20):
    title='python project:{}'.format(i)
    p = Project.objects.create(champion=user, title=title)
    print(p)
```

### Write an API view/urls to select projects

### The view

```
class ApiGetProjectsView(FormView):
    """
    Documentation
    =============

    - https://ccbv.co.uk/projects/Django/1.10/django.views.generic.edit/FormView/

    """
    def get(self, request, *args, **kwargs):
        """term is sent by the jquery-ui autocomplete widget.

        For the jquery-ui autocomplete widget we have to return 3 fields:

        - id
        - label
        - value

        For the jquery EasyAutocomplete we can return what we want.


        """
        term = request.GET.get("term")
        if term:
            projects = Project.objects.filter(title__icontains=term)
        else:
            projects = Project.objects.all()[:50]

        results = []
        for project in projects:
            project_json = {}
            project_json['id'] = project.id
            project_json['label'] = project.title
            project_json['value'] = project.title
            results.append(project_json)

        data = json.dumps(results)
        mimetype = 'application/json'
        return HttpResponse(data, mimetype)
```

### The URL

```
# calls by jquery EasyAutocomplete (EAC)
# http://127.0.0.1:8004/projects/api_get_projects/?term=a
url(r'^api_get_projects/$',
```

```
    ApiEACGetProjectsView.as_view(),
    name='api_get_projects'),
```

### Tests with httpie

```
http http://127.0.0.1:8004/projects/api_get_projects/?term=a
```

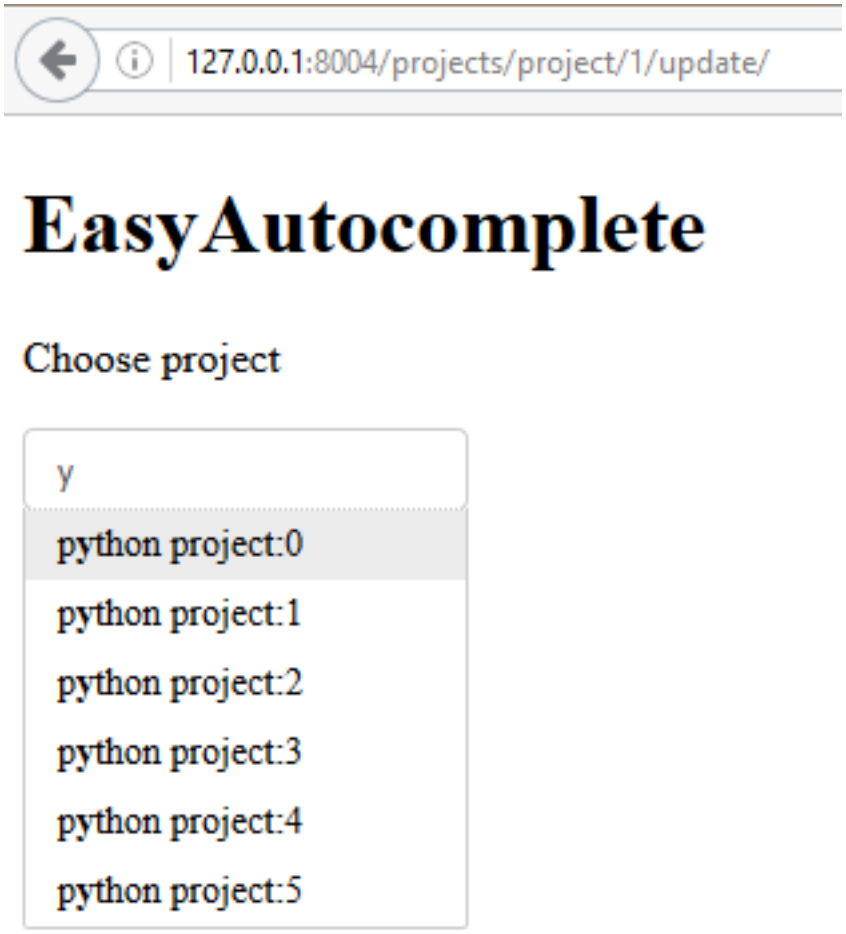
```
HTTP/1.0 200 OK
Content-Type: application/json
Date: Fri, 21 Oct 2016 13:03:31 GMT
Server: WSGIServer/0.2 CPython/3.5.2
X-Frame-Options: SAMEORIGIN

[
    {
        "id": 52,
        "label": "an other projec:0",
        "value": "an other projec:0"
    },
    {
        "id": 53,
        "label": "an other projec:1",
        "value": "an other projec:1"
    },
    {
        "id": 54,
        "label": "an other projec:2",
        "value": "an other projec:2"
    },
```

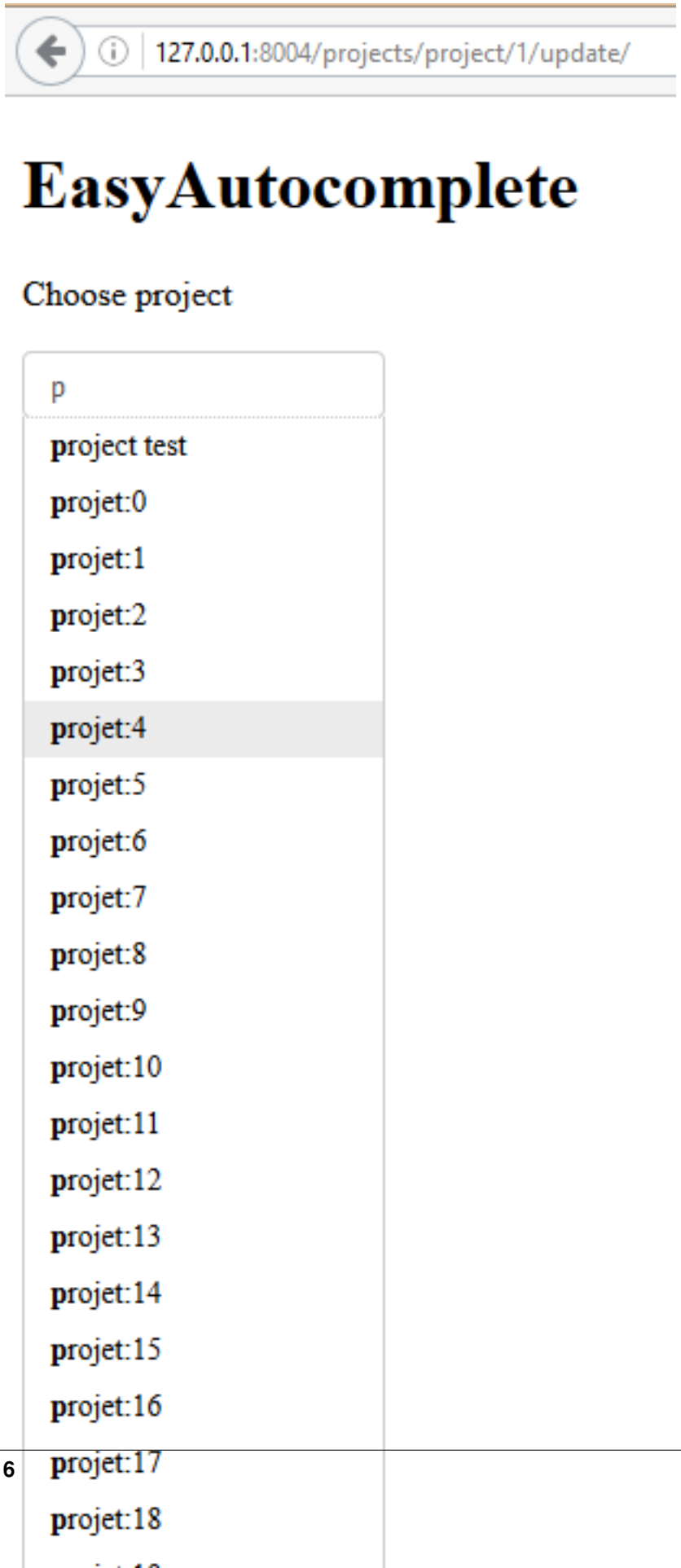
### Update the Django template file in order to call this view

```
<script>
    var options_easy_autocomplete_ajax = {
        {# we have to build this URL: http://127.0.0.1:8004/projects/api_get_projects/
↪?term=a #}
        url: function(term) {
            return "{% url 'projects:api_get_projects' %}" + "?term=" + term;
        },
        getValue: "value",

    };
    $("#data-ajax").easyAutocomplete(options_easy_autocomplete_ajax);

</script>
```

### 6 elements is not enough

OK, no slider


127.0.0.1:8004/projects/project/1/update/
EasyAutocomplete
Choose project
y
python project:0
python project:1
python project:2
python project:3
python project:4
python project:5


127.0.0.1:8004/projects/project/1/update/
EasyAutocomplete
Choose project
p
project test
projet:0
projet:1
projet:2
projet:3
projet:4
projet:5
projet:6
projet:7
projet:8
projet:9
projet:10
projet:11
projet:12
projet:13
projet:14
projet:15
projet:16
projet:17
projet:18

### Ajout placeholder

```
<!-- STRUCTURE -> HTML5 elements -->
    <h1>EasyAutocomplete</h1>

    <input id="data-ajax" placeholder="Choose the project" size="50" />
<!-- end STRUCTURE-->
```


### list['onSelectItemEvent']

**See also:**

http://easyautocomplete.com/guide#sec-function-selected-data

### HTML5

```
<!-- STRUCTURE -> HTML5 elements -->
    <h1>EasyAutocomplete</h1>

    <input id="project_choice_list" placeholder="Choose the project" size="50" />
    <input id="projet_holder" type="text" value="" size="50">
<!-- end STRUCTURE-->
```

### Javascript

```
var options_easy_autocomplete_ajax = {
    {# we have to build this URL: http://127.0.0.1:8004/projects/api_get_projects/?
↪term=a #}
    url: function(term) {
        return "{% url 'projects:api_get_projects' %}" + "?term=" + term;
    },
    getValue: "title",
    list: {
        maxNumberOfElements: 200,
        onSelectItemEvent: function() {
            var project_id = $("#project_choice_list").getSelectedItemData().id;
```

```
            {# put the project.id in the projet_holder field #}
            $("#projet_holder").val(project_id).trigger("change");
        },
        match: {
            enabled: true
        },
    },
};
$("#project_choice_list").easyAutocomplete(options_easy_autocomplete_ajax);
```

### list['showAnimation']

### TODO : tester categories

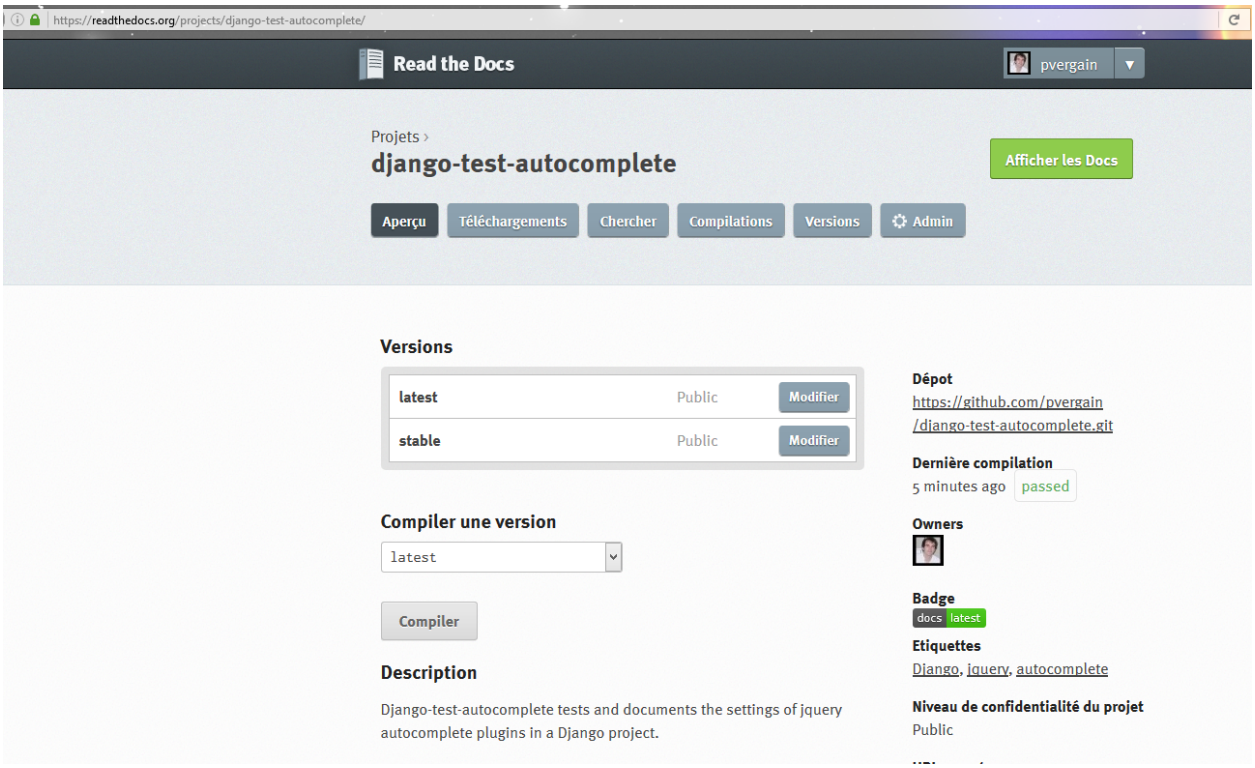
**See also:**

- http://easyautocomplete.com/guide#sec-categories

## 2016-10-21 pushing documentation on readthedocs

**See also:**

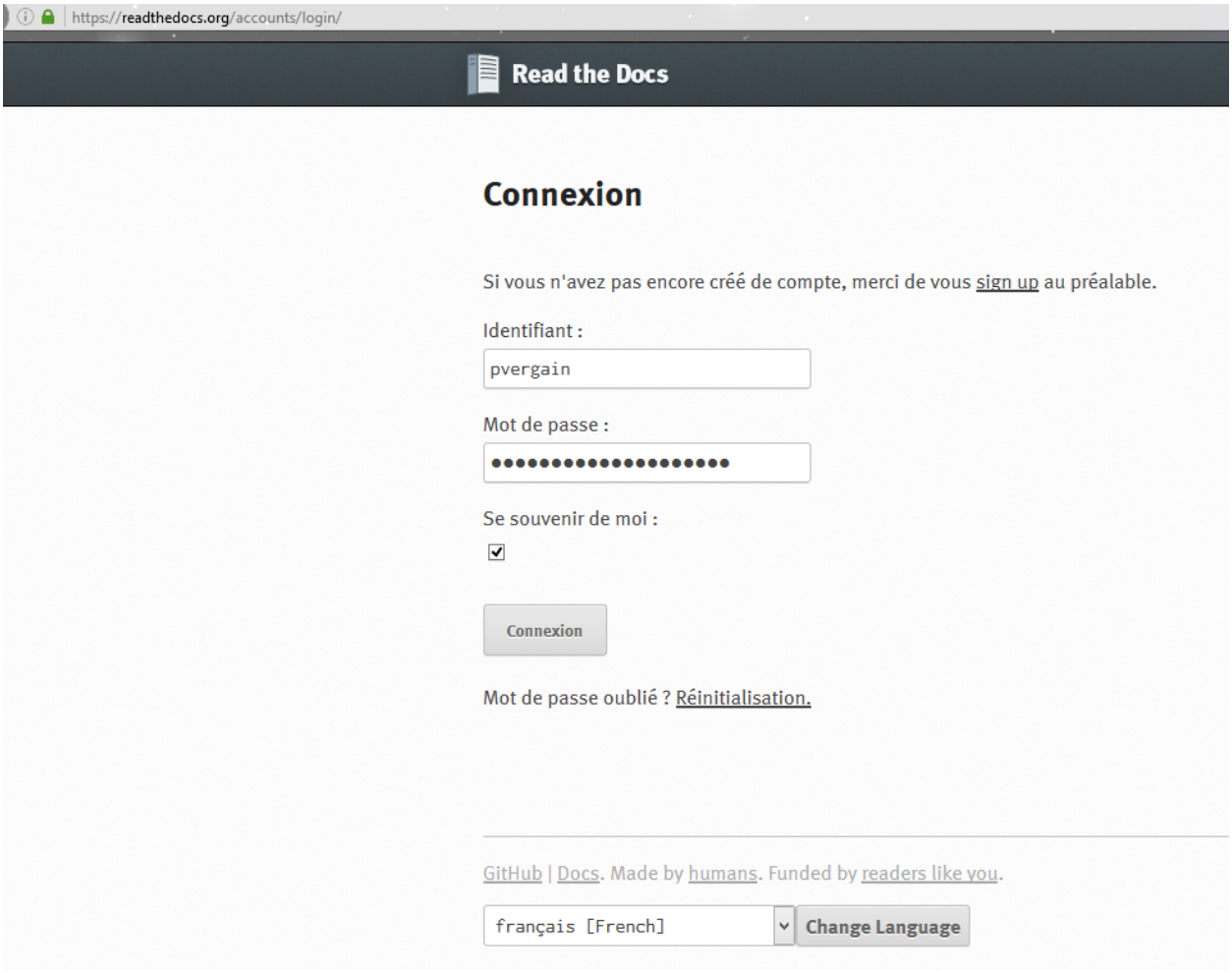
- https://readthedocs.org/
- https://readthedocs.org/accounts/login/
- https://django-test-autocomplete.readthedocs.io/en/latest/
- https://readthedocs.org/projects/django-test-autocomplete/



### Connect to readthedocs

**See also:**

- https://readthedocs.org/accounts/login/




### Import a projet

**See also:**

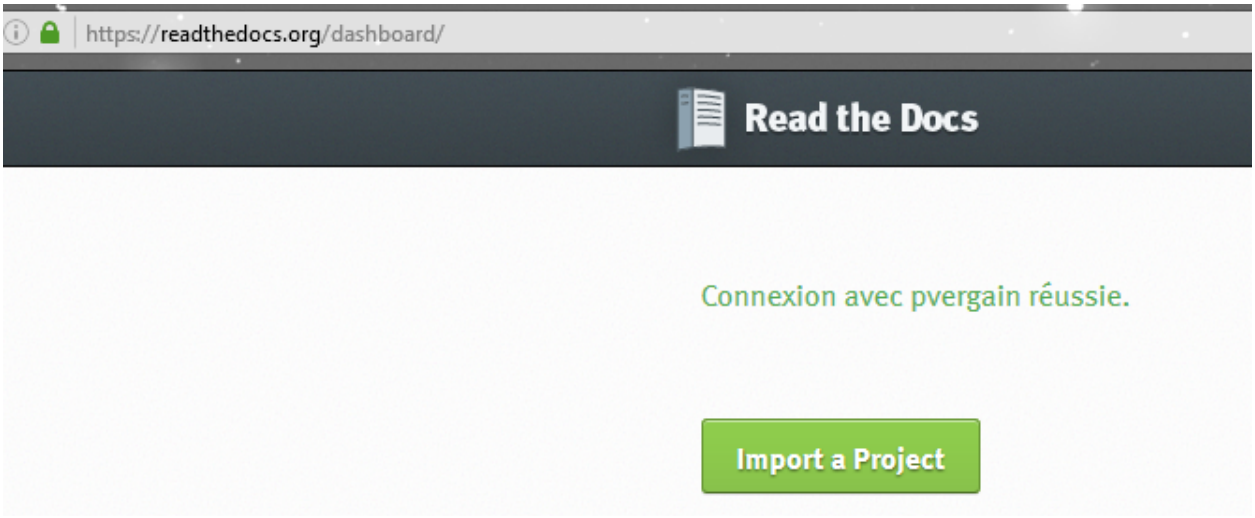
- https://docs.readthedocs.io/en/latest/getting_started.html#import-docs

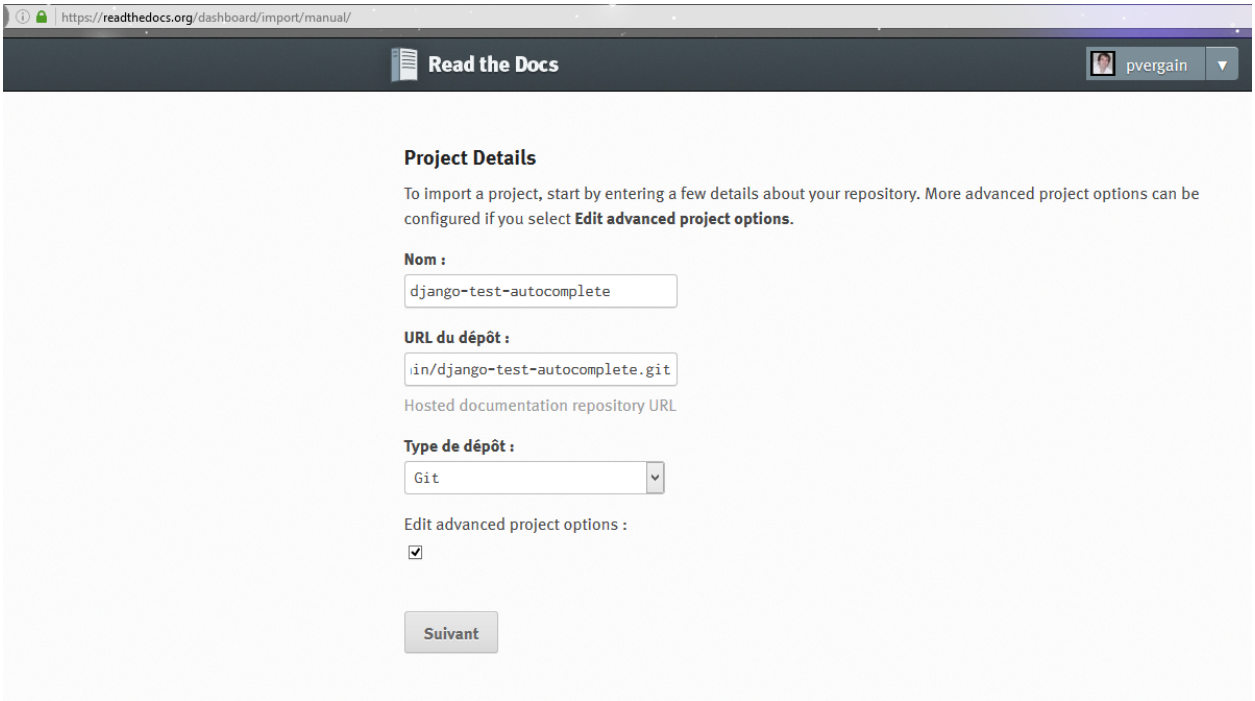
### Advanced parameters

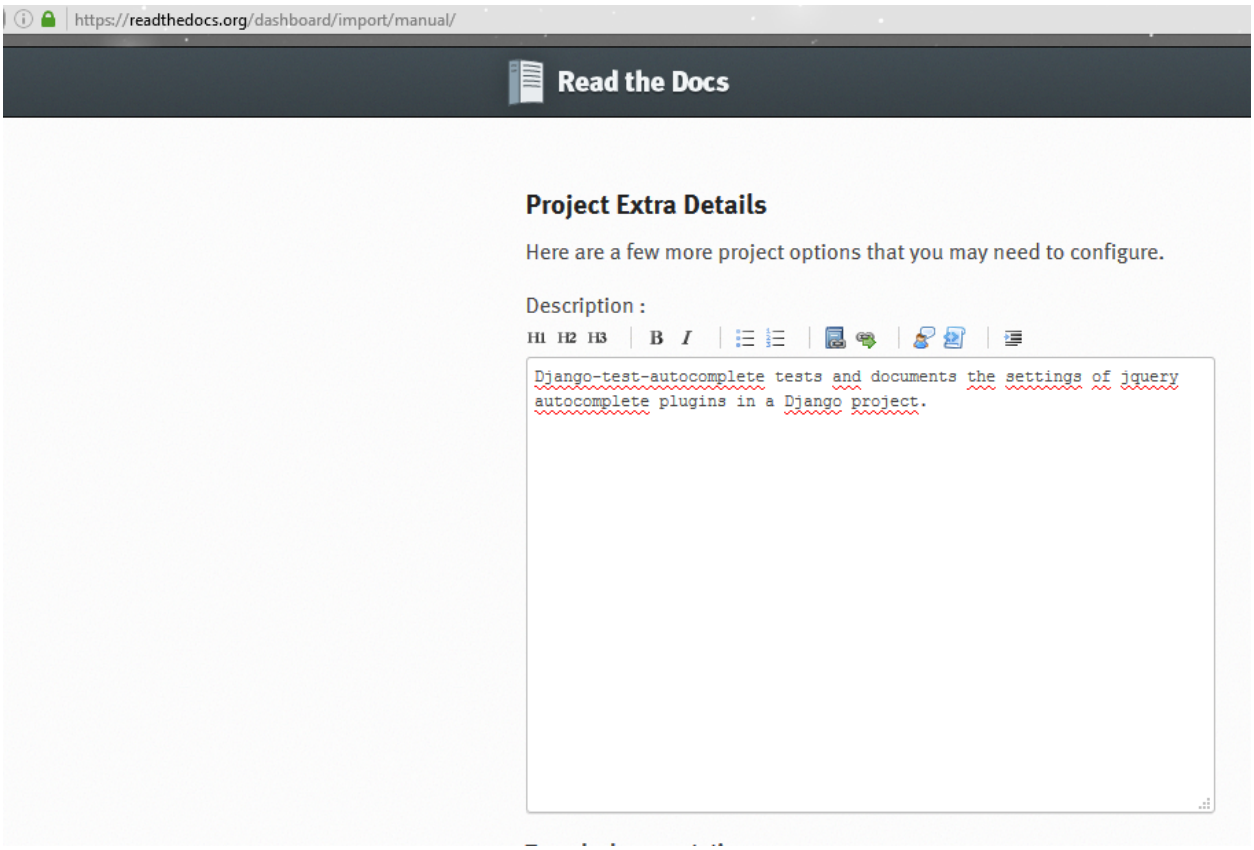
## 2016-10-20 2016-10-21 First step: using the jquery EasyAutocomplete plugin with Django

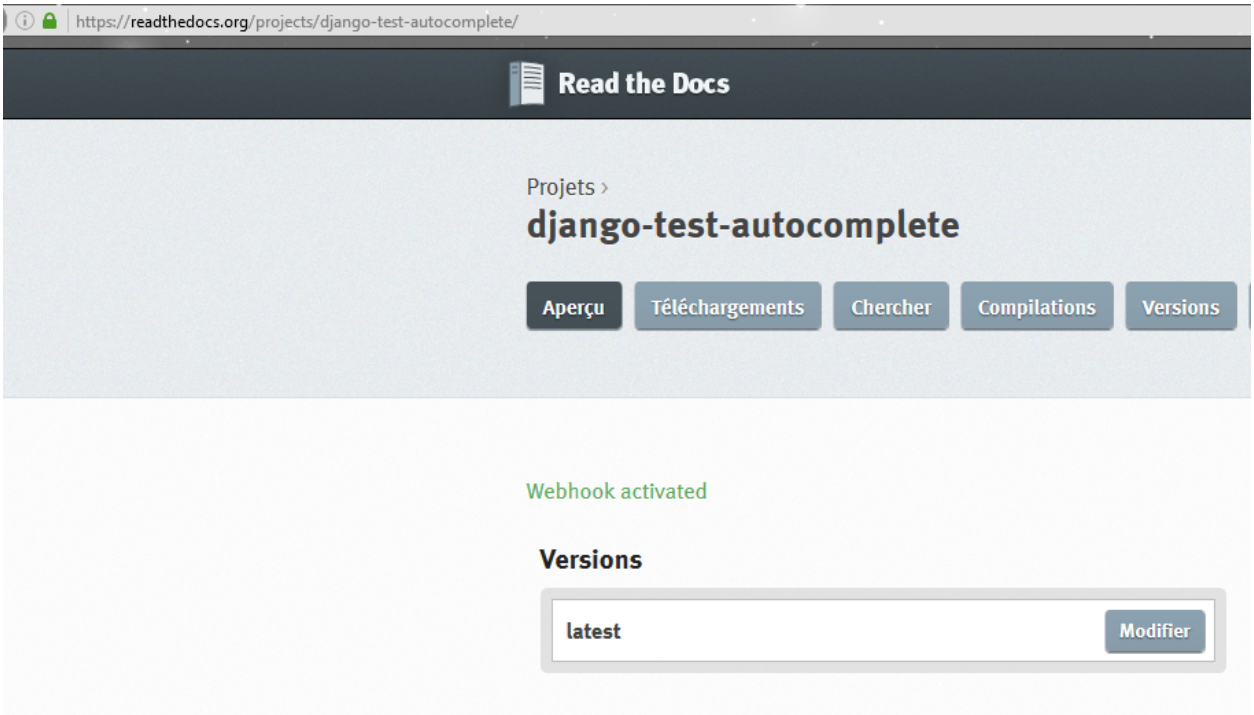
**See also:**

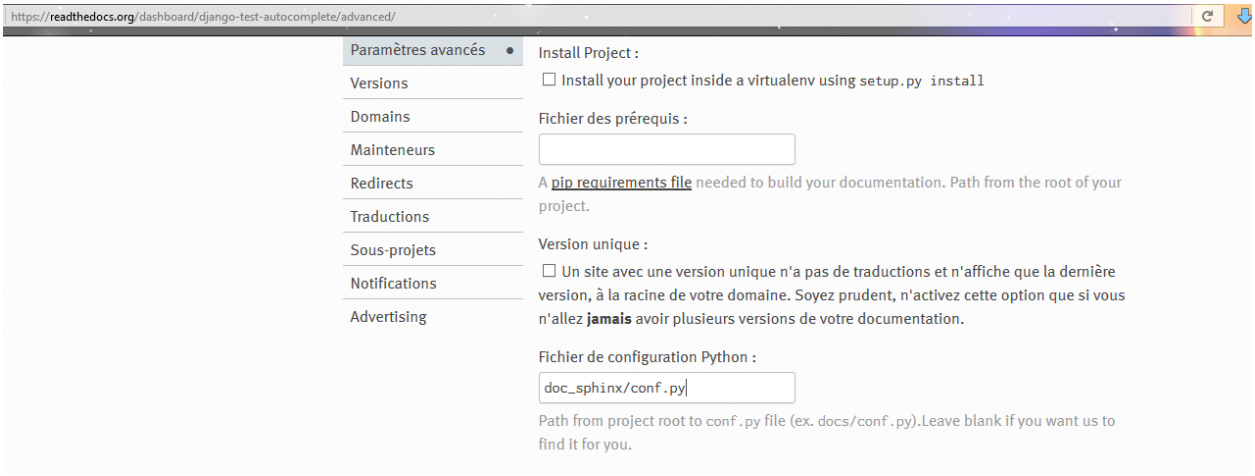
- http://easyautocomplete.com/
- http://easyautocomplete.com/guide

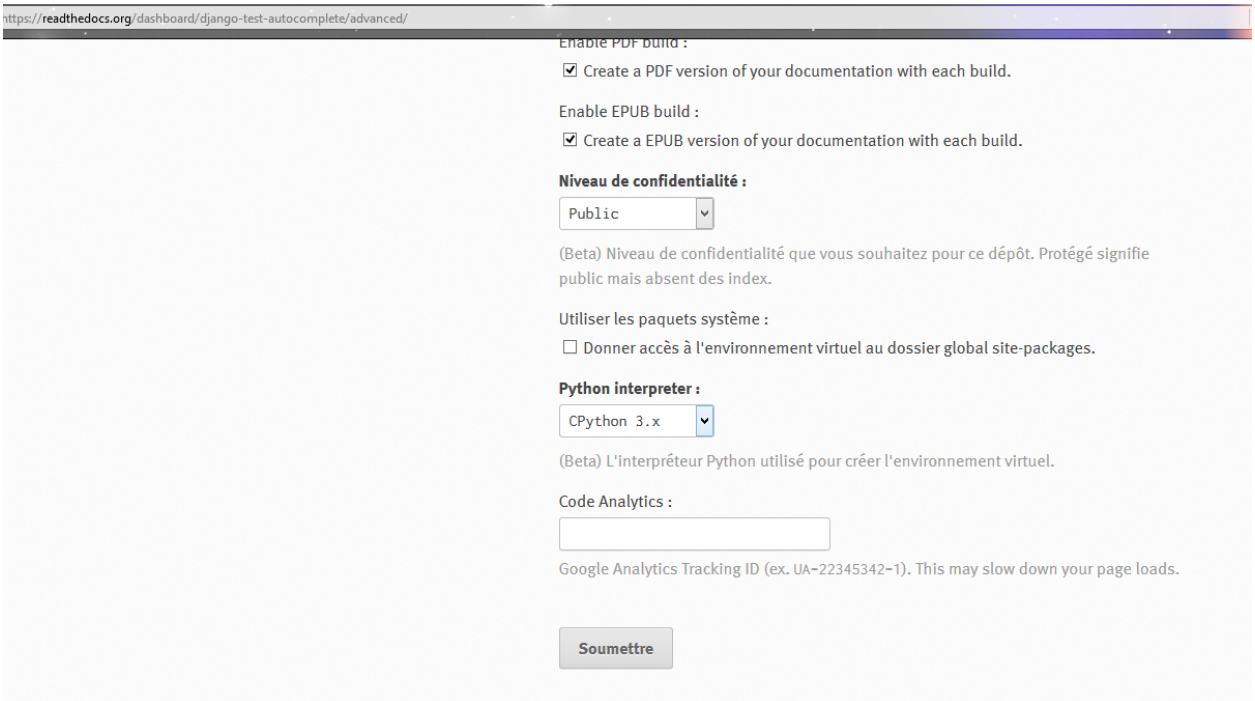

https://readthedocs.org/dashboard/
Read the Docs
Connexion avec pvergain réussie.
Import a Project


https://readthedocs.org/dashboard/import/manual/
Read the Docs
pvergain
Project Details
To import a project, start by entering a few details about your repository. More advanced project options can be configured if you select Edit advanced project options.
Nom :
django-test-autocomplete
URL du dépôt :
in/django-test-autocomplete.git
Hosted documentation repository URL
Type de dépôt :
Git
Edit advanced project options :
Suivant


https://readthedocs.org/dashboard/import/manual/
Read the Docs
Project Extra Details
Here are a few more project options that you may need to configure.
Description :
H1 H2 H3
Django-test-autocomplete tests and documents the settings of jquery autocomplete plugins in a Django project.


https://readthedocs.org/projects/django-test-autocomplete/
Read the Docs
Projets ›
django-test-autocomplete
Aperçu
Téléchargements
Chercher
Compilations
Versions
Webhook activated
Versions
latest
Modifier


https://readthedocs.org/dashboard/django-test-autocomplete/advanced/
Paramètres avancés
Versions
Domains
Mainteneurs
Redirects
Traductions
Sous-projets
Notifications
Advertising
Install Project :
Install your project inside a virtualenv using setup.py install
Fichier des prérequis :
A pip requirements file needed to build your documentation. Path from the root of your project.
Version unique :
Un site avec une version unique n'a pas de traductions et n'affiche que la dernière version, à la racine de votre domaine. Soyez prudent, n'activez cette option que si vous n'allez jamais avoir plusieurs versions de votre documentation.
Fichier de configuration Python :
doc_sphinx/conf.py
Path from project root to conf.py file (ex. docs/conf.py).Leave blank if you want us to find it for you.


https://readthedocs.org/dashboard/django-test-autocomplete/advanced/
Enable PDF build :
Create a PDF version of your documentation with each build.
Enable EPUB build :
Create a EPUB version of your documentation with each build.
Niveau de confidentialité :
Public
(Beta) Niveau de confidentialité que vous souhaitez pour ce dépôt. Protégé signifie public mais absent des index.
Utiliser les paquets système :
Donner accès à l'environnement virtuel au dossier global site-packages.
Python interpreter :
CPython 3.x
(Beta) L'interpréteur Python utilisé pour créer l'environnement virtuel.
Code Analytics :
Google Analytics Tracking ID (ex. UA-22345342-1). This may slow down your page loads.
Soumettre

- http://easyautocomplete.com/guide#sec-data-providers
- https://github.com/pawelczak/EasyAutocomplete

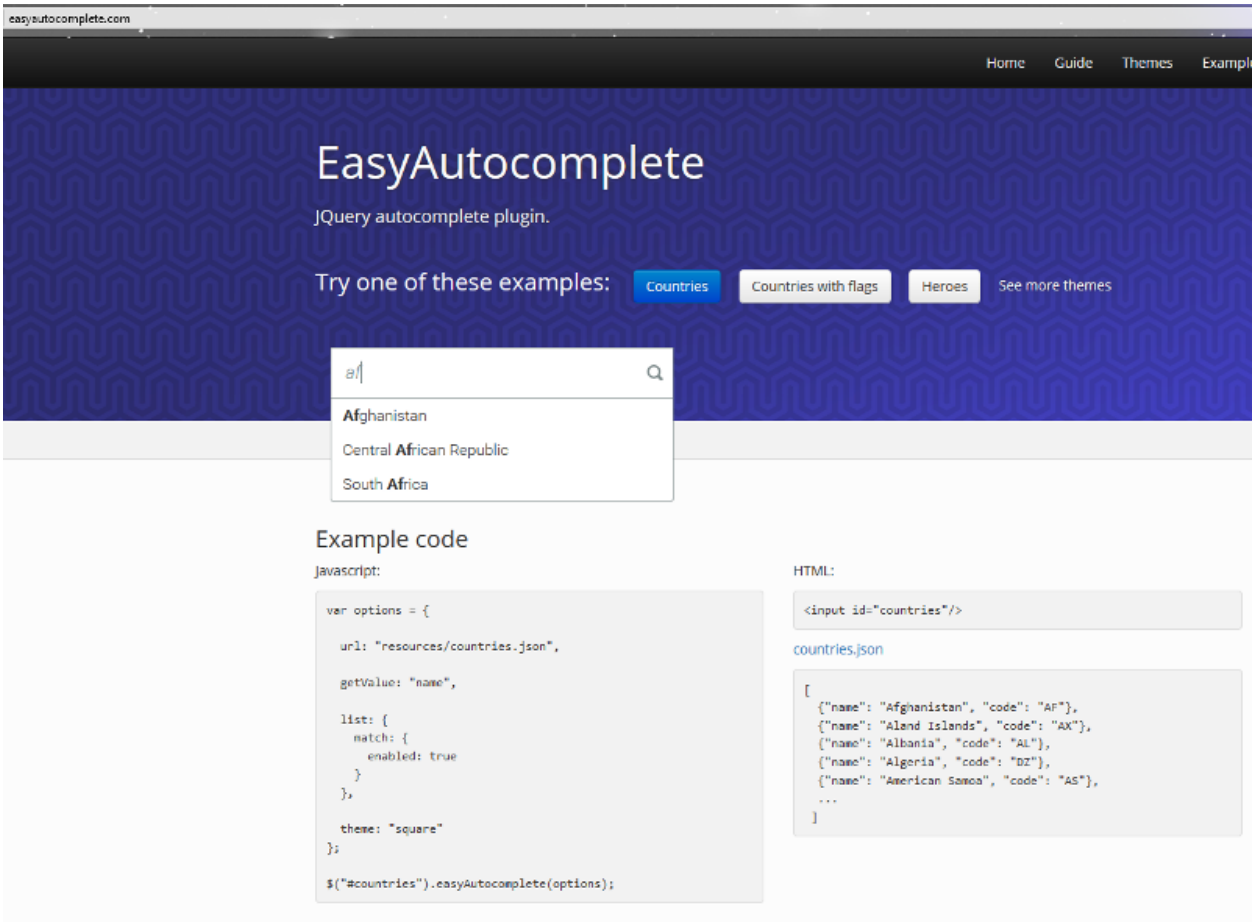
**Contents**

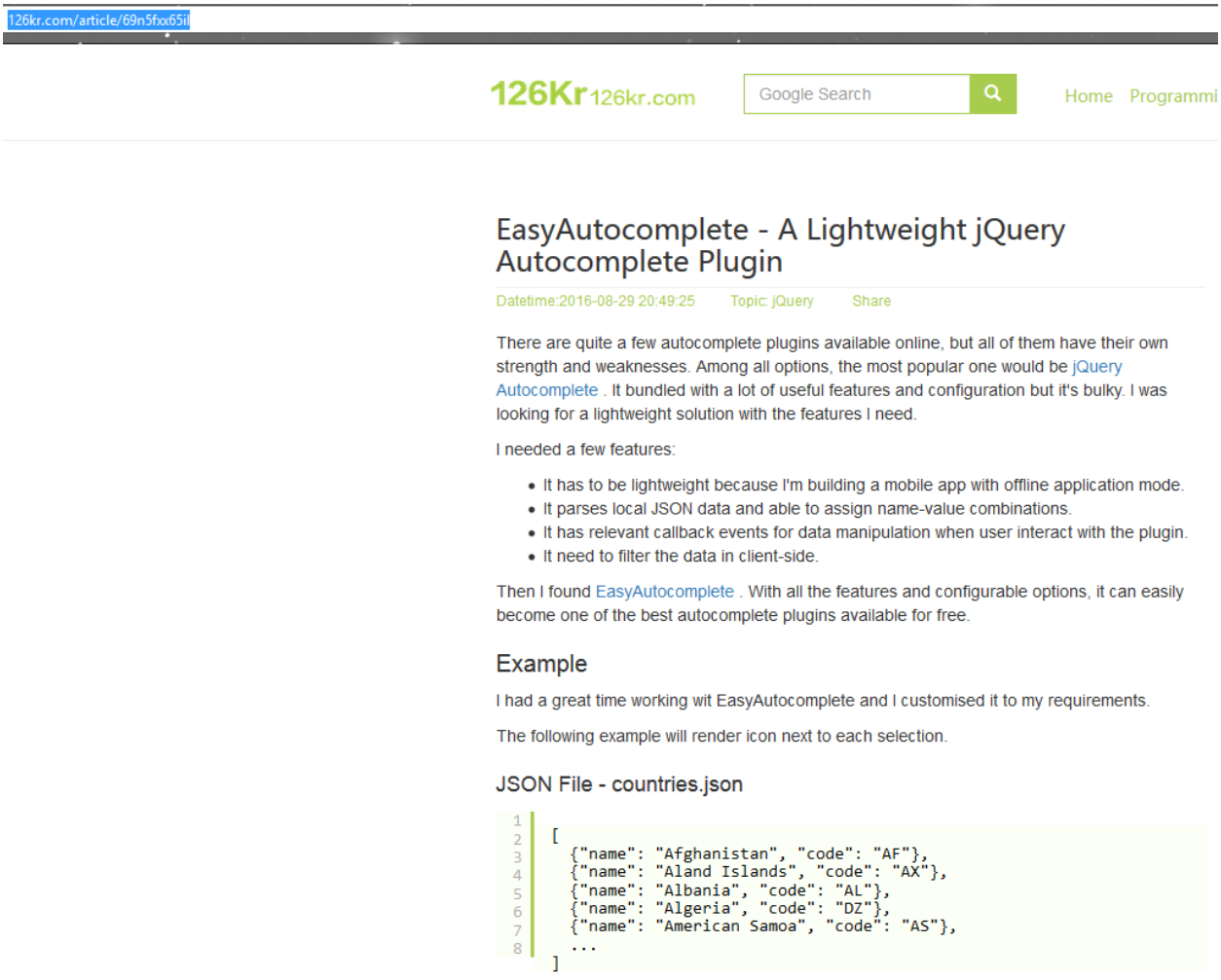

### Why ? EasyAutocomplete can easily become one of the best autocomplete plugins available for free

**See also:**

http://126kr.com/article/69n5fxx65il

**Because jquery Easyautocomplete can easily become one of the best autocomplete plugins available for free..**


easyautocomplete.com
Home
Guide
Themes
EasyAutocomplete
JQuery autocomplete plugin.
Try one of these examples:
Countries
Countries with flags
Heroes
See more themes
af
Afghanistan
Central African Republic
South Africa
Example code
Javascript:
var options = {
url: "resources/countries.json",
getValue: "name",
list: {
match: {
enabled: true
}
},
theme: "square"
};
$("#countries").easyAutocomplete(options);
HTML:
<input id="countries"/>
countries.json
[
{"name": "Afghanistan", "code": "AF"},
{"name": "Aland Islands", "code": "AX"},
{"name": "Albania", "code": "AL"},
{"name": "Algeria", "code": "DZ"},
{"name": "American Samoa", "code": "AS"},
...
]


126kr.com/article/69n5fxx65il
126Kr 126kr.com
Google Search
Home Programmi
EasyAutocomplete - A Lightweight jQuery Autocomplete Plugin
Datetime:2016-08-29 20:49:25 Topic: jQuery Share
There are quite a few autocomplete plugins available online, but all of them have their own strength and weaknesses. Among all options, the most popular one would be jQuery Autocomplete . It bundled with a lot of useful features and configuration but it's bulky. I was looking for a lightweight solution with the features I need.
I needed a few features:
It has to be lightweight because I'm building a mobile app with offline application mode.
It parses local JSON data and able to assign name-value combinations.
It has relevant callback events for data manipulation when user interact with the plugin.
It need to filter the data in client-side.
Then I found EasyAutocomplete . With all the features and configurable options, it can easily become one of the best autocomplete plugins available for free.
Example
I had a great time working wit EasyAutocomplete and I customised it to my requirements.
The following example will render icon next to each selection.
JSON File - countries.json
[
{"name": "Afghanistan", "code": "AF"},
{"name": "Aland Islands", "code": "AX"},
{"name": "Albania", "code": "AL"},
{"name": "Algeria", "code": "DZ"},
{"name": "American Samoa", "code": "AS"},
...
]

### EasyAutocomplete files tree (Javascript, CSS, JSON, PHP files)

```
C:.
|   .gitattributes
|   .gitignore
|   .jscsrc
|   .jshintrc
|   authors.txt
|   bower.json
|   Gruntfile.js
|   LICENSE.txt
|   package.json
|   README.md
|   tree_a_f.txt
|
+---demo
|   |   example_categories.html
|   |   example_custom_template.html
|   |   example_duckduckgo.html
|   |   example_email.html
|   |   example_flags.html
|   |   example_json.html
|   |   example_remote.html
|   |   example_simple.html
|   |   example_static_link.html
|   |   example_theme.html
|   |   example_theme_funky.html
|   |   example_theme_square.html
|   |   example_xml.html
|   |
|   +---api
|   |       countries.php
|   |       countrySearch.php
|   |
|   \---resources
|           categories.json
|           countries.json
|           countries.xml
|           flags.css
|           flags.png
|           icon_search.png
|           people.json
|           site.json
|
+---dist
|   |   easy-autocomplete.css
|   |   easy-autocomplete.min.css
|   |   easy-autocomplete.themes.css
|   |   easy-autocomplete.themes.min.css
|   |   jquery.easy-autocomplete.js
|   |   jquery.easy-autocomplete.min.js
|   |
|   \---maps
|           easy-autocomplete.css.map
|           easy-autocomplete.min.css.map
|           easy-autocomplete.themes.css.map
|           easy-autocomplete.themes.min.css.map
|
```

```
+---lib
|       jquery-1.11.2.min.js
|       jquery-2.1.3.min.js
|
+---src
|   |   configuration.js
|   |   constans.js
|   |   core.js
|   |   listBuilder.js
|   |   logger.js
|   |   proccessData.js
|   |   template.js
|   |
|   \---sass
|           easy-autocomplete.scss
|           easy-autocomplete.themes.scss
|
\---test
    |   configuration.js
    |   listBuilder.js
    |   modules.html
    |   proccessData.js
    |   template.js
    |
    +---core
    |   |   build.html
    |   |   build.js
    |   |   categories.html
    |   |   categories.js
    |   |   event.html
    |   |   event.js
    |   |   features.html
    |   |   features.js
    |   |   functions.html
    |   |   functions.js
    |   |   handles.html
    |   |   handles.js
    |   |   plugin.html
    |   |   plugin.js
    |   |   response.html
    |   |   response.js
    |   |   response_json.js
    |   |   response_remote.html
    |   |   response_remote.js
    |   |   response_static.js
    |   |   response_xml.js
    |   |   template.html
    |   |   template.js
    |   |
    |   +---remote
    |   |       countries.php
    |   |       countrySelectService.php
    |   |
    |   \---resources
    |       |   categories.json
    |       |   colors.json
    |       |   colors.xml
    |       |   colors_caps_string.json
```

```
    |       |   colors_object.json
    |       |   colors_object.xml
    |       |   colors_string.json
    |       |   countries.json
    |       |   duckduckgo.json
    |       |   response.json
    |       |
    |       \---categories
    |               fruits.json
    |               fruits.xml
    |               otherFruits.xml
    |
    \---qunit
            qunit.css
            qunit.js
```

### Javascript SCSS and CSS EasyAutocomplete files

### EasyAutocomplete-master/src/sass/easy-autocomplete.scss

Listing 3.1: EasyAutocomplete-master/src/sass/easy-autocomplete.scss

```
1
2  $blue: #5A91CB;
3  $blue-light: rgba(102, 175, 233, 1);
4  $blue-lighter: rgba(102, 175, 233, 0.6);
5  $green-light: #41DB00;
6  $green-lighter: rgba(146, 237, 107, 0.6);
7  $red-light: #ff5b5b;
8  $red-lighter: rgba(255, 90, 90, 0.6);
9  $yellow-light: #ffdb00;
10 $yellow-lighter: rgba(255, 231, 84, 0.6);
11 $dark-light: #333;
12 $dark-lighter: rgba(55, 55, 55, 0.6);
13 $dark-glass: rgba(0, 0, 0, 0.8);
14 $dark: #333;
15 $yellow: rgba(255, 212, 100, 1);
16 $purple: #c7c0de;
17
18 @mixin placeholder {
19   &::-webkit-input-placeholder {@content}
20   &:-moz-placeholder           {@content}
21   &::-moz-placeholder          {@content}
22   &:-ms-input-placeholder      {@content}
23 }
24
25 .easy-autocomplete {
26     position: relative;
27
28     input {
29         border: {
30             color: #ccc;
31             radius: 4px;
32             style: solid;
33             width: 1px;
34         }
```

```
        box-shadow: 0 1px 2px rgba(0, 0, 0, 0.1) inset;
        color: #555;
        float: none;
        padding: 6px 12px;

        &:hover, &:focus {
            box-shadow: none;
        }
    }

    a {
        display: block;
    }


    &.eac-blue-light {

        input {

            &:hover, &:focus {
                border-color: $blue-light;
                box-shadow: 0 1px 1px rgba(0, 0, 0, 0.075) inset, 0 0 8px $blue-
↪lighter;
            }
        }

        ul {
            border-color: $blue-light;
            box-shadow: 0 1px 1px rgba(0, 0, 0, 0.075) inset, 0 0 8px $blue-lighter;

            li, .eac-category {
                border-color: $blue-light;

                &.selected {
                    background-color: lighten($blue-light, 30%);
                }

            }
        }
    }

    &.eac-green-light {

        input {
            &:hover, &:focus {
                border-color: $green-light;
                box-shadow: 0 1px 1px rgba(0, 0, 0, 0.075) inset, 0 0 8px $green-
↪lighter;
            }
        }

        ul {
            border-color: $green-light;
            box-shadow: 0 1px 1px rgba(0, 0, 0, 0.075) inset, 0 0 8px $green-lighter;

            li, .eac-category {
                border-color: $green-light;

```

```
                &.selected {
                    background-color: lighten($green-light, 30%);
                }

            }
        }
    }

    &.eac-red-light {

        input {
            &:hover, &:focus {
                border-color: $red-light;
                box-shadow: 0 1px 1px rgba(0, 0, 0, 0.075) inset, 0 0 8px $red-
↪lighter;
            }
        }

        ul {
            border-color: $red-light;
            box-shadow: 0 1px 1px rgba(0, 0, 0, 0.075) inset, 0 0 8px $red-lighter;

            li, .eac-category {
                border-color: $red-light;

                &.selected {
                    background-color: lighten($red-light, 10%);
                }

            }
        }
    }

    &.eac-yellow-light {

        input {
            &:hover, &:focus {
                border-color: $yellow-light;
                box-shadow: 0 1px 1px rgba(0, 0, 0, 0.075) inset, 0 0 8px $yellow-
↪lighter;
            }
        }

        ul {
            border-color: $yellow-light;
            box-shadow: 0 1px 1px rgba(0, 0, 0, 0.075) inset, 0 0 8px $yellow-lighter;

            li, .eac-category {
                border-color: $yellow-light;

                &.selected {
                    background-color: lighten($yellow-light, 10%);
                }

            }
        }
    }

```

```
147     &.eac-dark-light {
148
149         input {
150             &:hover, &:focus {
151                 border-color: $dark-light;
152                 box-shadow: 0 1px 1px rgba(0, 0, 0, 0.075) inset, 0 0 8px $dark-
↪lighter;
153             }
154         }
155
156         ul {
157             border-color: $dark-light;
158             box-shadow: 0 1px 1px rgba(0, 0, 0, 0.075) inset, 0 0 8px $dark-lighter;
159
160             li, .eac-category {
161                 border-color: $dark-light;
162
163                 &.selected {
164                     background-color: lighten($dark-light, 10%);
165                     color: #fff;
166                 }
167
168             }
169         }
170     }
171
172     &.eac-dark {
173         color: #fff;
174
175         input {
176             background-color: lighten($dark-light, 5%);
177             border: {
178                 radius: 4px;
179             }
180             box-shadow: 0;
181             color: #f6f6f6;
182
183             &:hover, &:focus {
184                 border-color: $dark;
185                 box-shadow: 0;
186             }
187         }
188
189         ul {
190             border-color: $dark;
191
192             li, .eac-category {
193                 background-color: lighten($dark-light, 5%);
194                 border-color: $dark;
195
196                 &.selected {
197                     background-color: lighten($dark, 25%);
198                     color: #f6f6f6;
199                 }
200
201             }
202         }
203     }
```

```
204
205     &.eac-dark-glass {
206         color: #fff;
207
208         input {
209             background-color: $dark-glass;
210             border: {
211                 radius: 4px;
212             }
213             box-shadow: 0;
214             color: #f6f6f6;
215
216             &:hover, &:focus {
217                 border-color: $dark-glass;
218                 box-shadow: 0;
219             }
220         }
221
222         ul {
223             border-color: $dark-glass;
224
225             li, .eac-category {
226                 background-color: $dark-glass;
227                 border-color: $dark-glass;
228
229                 &.selected {
230                     background-color: lighten($dark-glass, 25%);
231                     color: #f6f6f6;
232                 }
233
234                 &:last-child {
235                     border-radius: 0 0 4px 4px;
236                 }
237
238             }
239         }
240     }
241
242
243
244     &.eac-blue {
245         color: #fff;
246
247         input {
248             @include placeholder {
249                 color: #f6f6f6;
250             }
251             background-color: lighten($blue, 5%);
252             border: {
253                 radius: 4px;
254             }
255             box-shadow: 0;
256             color: #f6f6f6;
257
258             &:hover, &:focus {
259                 border-color: $blue;
260                 box-shadow: 0;
261             }
```

```
        }

        ul {
            border-color: $blue;

            li, .eac-category {
                background-color: lighten($blue, 5%);
                border-color: $blue;

                &.selected {
                    background-color: lighten($blue, 15%);
                    color: #f6f6f6;
                }

            }
        }
    }

    &.eac-yellow {
        color: #333;

        input {
            background-color: lighten($yellow, 5%);
            border: {
                color: #333;
                radius: 4px;
            }
            box-shadow: 0;
            color: #333;

            &:hover, &:focus {
                border-color: #333;
                box-shadow: 0;
            }
        }

        ul {
            border-color: #333;

            li, .eac-category {
                background-color: lighten($yellow, 5%);
                border-color: #333;

                &.selected {
                    background-color: lighten($yellow, 15%);
                    color: #333;
                }

            }
        }
    }

    &.eac-purple {
        color: #333;

        input {
            background-color: lighten($purple, 5%);
            border: {
```

```
                color: darken($purple, 5%);
            }
            box-shadow: 0;
            color: #333;

            &:hover, &:focus {
                border-color: #333;
                box-shadow: 0;
            }
        }

        ul {
            border-color: #333;

            li, .eac-category {
                background-color: lighten($purple, 5%);
                border-color: #333;

                &.selected {
                    background-color: lighten($purple, 12%);
                    color: #333;
                }

            }
        }
    }

    &.eac-bootstrap {

        input {
            border-color: #ccc;
            border-radius: 4px;
            border-style: solid;
            border-width: 1px;
            color: #555;
            padding: 6px 12px;
        }
    }


}


.easy-autocomplete-container {
    left: 0;
    position: absolute;
    width: 100%;
    z-index: 2;

    ul {
        background: none repeat scroll 0 0 #ffffff;
        border-top: 1px dotted #ccc;
        display: none;
        margin-top: 0;
        padding-bottom: 0;
        padding-left: 0;
        position: relative;
        top: -1px;
```

```
        li, .eac-category {
            background: inherit;
            border: {
                color: #ccc;
                image: none;
                style: solid;
                width: 0 1px;
            }
            display: block;
            font: {
                size: 14px;
                weight: normal;
            }
            padding: 4px 12px;
        }

        li {

            &:last-child {
                border: {
                    radius: 0 0 2px 2px;
                    width: 0 1px 1px;
                }
            }

            &.selected {
                background: none repeat scroll 0 0 #ebebeb;
                cursor: pointer;

                div {
                    font-weight: normal;
                }
            }

            div {
                display: block;
                font-weight: normal;
                word-break: break-all;
            }

            b {
                font: {
                    weight: bold;
                }
            }
        }

        .eac-category {
            font: {
                color: #aaa;
                style: italic;
            }
        }

    }
}
```

```
436
437 .eac-description {
438
439     .eac-item {
440
441         span {
442             color: #aaa;
443             font-style: italic;
444             font-size: 0.9em;
445         }
446     }
447 }
448
449 .eac-icon-left {
450
451     .eac-item {
452         img {
453             margin-right: 4px;
454             max-height: 30px;
455         }
456     }
457 }
458
459
460 .eac-icon-right {
461
462     .eac-item {
463         margin-top: 8px;
464         min-height: 24px;
465         position: relative;
466
467         img {
468             margin-left: 4px;
469             max-height: 30px;
470             position: absolute;
471             right: -4px;
472             top: -8px;
473         }
474     }
475 }
```

### EasyAutocomplete-master/dist/easy-autocomplete.css

Listing 3.2: EasyAutocomplete-master/dist/easy-autocomplete.css

```
1  /*
2   * easy-autocomplete
3   * jQuery plugin for autocompletion
4   *
5   * @author Łukasz Pawełczak (http://github.com/pawelczak)
6   * @version 1.3.5
7   * Copyright  License:
8   */
9
10 .easy-autocomplete {
11   position: relative;
```

```
}
.easy-autocomplete input {
  border-color: #ccc;
  border-radius: 4px;
  border-style: solid;
  border-width: 1px;
  box-shadow: 0 1px 2px rgba(0, 0, 0, 0.1) inset;
  color: #555;
  float: none;
  padding: 6px 12px;
}
.easy-autocomplete input:hover, .easy-autocomplete input:focus {
  box-shadow: none;
}
.easy-autocomplete a {
  display: block;
}
.easy-autocomplete.eac-blue-light input:hover, .easy-autocomplete.eac-blue-light␣
↪input:focus {
  border-color: #66afe9;
  box-shadow: 0 1px 1px rgba(0, 0, 0, 0.075) inset, 0 0 8px rgba(102, 175, 233, 0.6);
}
.easy-autocomplete.eac-blue-light ul {
  border-color: #66afe9;
  box-shadow: 0 1px 1px rgba(0, 0, 0, 0.075) inset, 0 0 8px rgba(102, 175, 233, 0.6);
}
.easy-autocomplete.eac-blue-light ul li, .easy-autocomplete.eac-blue-light ul .eac-
↪category {
  border-color: #66afe9;
}
.easy-autocomplete.eac-blue-light ul li.selected, .easy-autocomplete.eac-blue-light␣
↪ul .eac-category.selected {
  background-color: #ecf5fc;
}
.easy-autocomplete.eac-green-light input:hover, .easy-autocomplete.eac-green-light␣
↪input:focus {
  border-color: #41DB00;
  box-shadow: 0 1px 1px rgba(0, 0, 0, 0.075) inset, 0 0 8px rgba(146, 237, 107, 0.6);
}
.easy-autocomplete.eac-green-light ul {
  border-color: #41DB00;
  box-shadow: 0 1px 1px rgba(0, 0, 0, 0.075) inset, 0 0 8px rgba(146, 237, 107, 0.6);
}
.easy-autocomplete.eac-green-light ul li, .easy-autocomplete.eac-green-light ul .eac-
↪category {
  border-color: #41DB00;
}
.easy-autocomplete.eac-green-light ul li.selected, .easy-autocomplete.eac-green-light␣
↪ul .eac-category.selected {
  background-color: #9eff75;
}
.easy-autocomplete.eac-red-light input:hover, .easy-autocomplete.eac-red-light␣
↪input:focus {
  border-color: #ff5b5b;
  box-shadow: 0 1px 1px rgba(0, 0, 0, 0.075) inset, 0 0 8px rgba(255, 90, 90, 0.6);
}
.easy-autocomplete.eac-red-light ul {
  border-color: #ff5b5b;
```

```
  box-shadow: 0 1px 1px rgba(0, 0, 0, 0.075) inset, 0 0 8px rgba(255, 90, 90, 0.6);
}
.easy-autocomplete.eac-red-light ul li, .easy-autocomplete.eac-red-light ul .eac-
↪category {
  border-color: #ff5b5b;
}
.easy-autocomplete.eac-red-light ul li.selected, .easy-autocomplete.eac-red-light ul .
↪eac-category.selected {
  background-color: #ff8e8e;
}
.easy-autocomplete.eac-yellow-light input:hover, .easy-autocomplete.eac-yellow-light␣
↪input:focus {
  border-color: #ffdb00;
  box-shadow: 0 1px 1px rgba(0, 0, 0, 0.075) inset, 0 0 8px rgba(255, 231, 84, 0.6);
}
.easy-autocomplete.eac-yellow-light ul {
  border-color: #ffdb00;
  box-shadow: 0 1px 1px rgba(0, 0, 0, 0.075) inset, 0 0 8px rgba(255, 231, 84, 0.6);
}
.easy-autocomplete.eac-yellow-light ul li, .easy-autocomplete.eac-yellow-light ul .
↪eac-category {
  border-color: #ffdb00;
}
.easy-autocomplete.eac-yellow-light ul li.selected, .easy-autocomplete.eac-yellow-
↪light ul .eac-category.selected {
  background-color: #ffe233;
}
.easy-autocomplete.eac-dark-light input:hover, .easy-autocomplete.eac-dark-light␣
↪input:focus {
  border-color: #333;
  box-shadow: 0 1px 1px rgba(0, 0, 0, 0.075) inset, 0 0 8px rgba(55, 55, 55, 0.6);
}
.easy-autocomplete.eac-dark-light ul {
  border-color: #333;
  box-shadow: 0 1px 1px rgba(0, 0, 0, 0.075) inset, 0 0 8px rgba(55, 55, 55, 0.6);
}
.easy-autocomplete.eac-dark-light ul li, .easy-autocomplete.eac-dark-light ul .eac-
↪category {
  border-color: #333;
}
.easy-autocomplete.eac-dark-light ul li.selected, .easy-autocomplete.eac-dark-light␣
↪ul .eac-category.selected {
  background-color: #4d4d4d;
  color: #fff;
}
.easy-autocomplete.eac-dark {
  color: #fff;
}
.easy-autocomplete.eac-dark input {
  background-color: #404040;
  border-radius: 4px;
  box-shadow: 0;
  color: #f6f6f6;
}
.easy-autocomplete.eac-dark input:hover, .easy-autocomplete.eac-dark input:focus {
  border-color: #333;
  box-shadow: 0;
}
```

```
.easy-autocomplete.eac-dark ul {
  border-color: #333;
}
.easy-autocomplete.eac-dark ul li, .easy-autocomplete.eac-dark ul .eac-category {
  background-color: #404040;
  border-color: #333;
}
.easy-autocomplete.eac-dark ul li.selected, .easy-autocomplete.eac-dark ul .eac-
↪category.selected {
  background-color: #737373;
  color: #f6f6f6;
}
.easy-autocomplete.eac-dark-glass {
  color: #fff;
}
.easy-autocomplete.eac-dark-glass input {
  background-color: rgba(0, 0, 0, 0.8);
  border-radius: 4px;
  box-shadow: 0;
  color: #f6f6f6;
}
.easy-autocomplete.eac-dark-glass input:hover, .easy-autocomplete.eac-dark-glass␣
↪input:focus {
  border-color: rgba(0, 0, 0, 0.8);
  box-shadow: 0;
}
.easy-autocomplete.eac-dark-glass ul {
  border-color: rgba(0, 0, 0, 0.8);
}
.easy-autocomplete.eac-dark-glass ul li, .easy-autocomplete.eac-dark-glass ul .eac-
↪category {
  background-color: rgba(0, 0, 0, 0.8);
  border-color: rgba(0, 0, 0, 0.8);
}
.easy-autocomplete.eac-dark-glass ul li.selected, .easy-autocomplete.eac-dark-glass␣
↪ul .eac-category.selected {
  background-color: rgba(64, 64, 64, 0.8);
  color: #f6f6f6;
}
.easy-autocomplete.eac-dark-glass ul li:last-child, .easy-autocomplete.eac-dark-glass␣
↪ul .eac-category:last-child {
  border-radius: 0 0 4px 4px;
}
.easy-autocomplete.eac-blue {
  color: #fff;
}
.easy-autocomplete.eac-blue input {
  background-color: #6d9ed1;
  border-radius: 4px;
  box-shadow: 0;
  color: #f6f6f6;
}
.easy-autocomplete.eac-blue input::-webkit-input-placeholder {
  color: #f6f6f6;
}
.easy-autocomplete.eac-blue input:-moz-placeholder {
  color: #f6f6f6;
}
```

```
.easy-autocomplete.eac-blue input::-moz-placeholder {
  color: #f6f6f6;
}
.easy-autocomplete.eac-blue input:-ms-input-placeholder {
  color: #f6f6f6;
}
.easy-autocomplete.eac-blue input:hover, .easy-autocomplete.eac-blue input:focus {
  border-color: #5A91CB;
  box-shadow: 0;
}
.easy-autocomplete.eac-blue ul {
  border-color: #5A91CB;
}
.easy-autocomplete.eac-blue ul li, .easy-autocomplete.eac-blue ul .eac-category {
  background-color: #6d9ed1;
  border-color: #5A91CB;
}
.easy-autocomplete.eac-blue ul li.selected, .easy-autocomplete.eac-blue ul .eac-
↪category.selected {
  background-color: #94b8dd;
  color: #f6f6f6;
}
.easy-autocomplete.eac-yellow {
  color: #333;
}
.easy-autocomplete.eac-yellow input {
  background-color: #ffdb7e;
  border-color: #333;
  border-radius: 4px;
  box-shadow: 0;
  color: #333;
}
.easy-autocomplete.eac-yellow input:hover, .easy-autocomplete.eac-yellow input:focus {
  border-color: #333;
  box-shadow: 0;
}
.easy-autocomplete.eac-yellow ul {
  border-color: #333;
}
.easy-autocomplete.eac-yellow ul li, .easy-autocomplete.eac-yellow ul .eac-category {
  background-color: #ffdb7e;
  border-color: #333;
}
.easy-autocomplete.eac-yellow ul li.selected, .easy-autocomplete.eac-yellow ul .eac-
↪category.selected {
  background-color: #ffe9b1;
  color: #333;
}
.easy-autocomplete.eac-purple {
  color: #333;
}
.easy-autocomplete.eac-purple input {
  background-color: #d6d1e7;
  border-color: #b8afd5;
  box-shadow: 0;
  color: #333;
}
.easy-autocomplete.eac-purple input:hover, .easy-autocomplete.eac-purple input:focus {
```

```
  border-color: #333;
  box-shadow: 0;
}
.easy-autocomplete.eac-purple ul {
  border-color: #333;
}
.easy-autocomplete.eac-purple ul li, .easy-autocomplete.eac-purple ul .eac-category {
  background-color: #d6d1e7;
  border-color: #333;
}
.easy-autocomplete.eac-purple ul li.selected, .easy-autocomplete.eac-purple ul .eac-
↪category.selected {
  background-color: #ebe8f3;
  color: #333;
}
.easy-autocomplete.eac-bootstrap input {
  border-color: #ccc;
  border-radius: 4px;
  border-style: solid;
  border-width: 1px;
  color: #555;
  padding: 6px 12px;
}

.easy-autocomplete-container {
  left: 0;
  position: absolute;
  width: 100%;
  z-index: 2;
}
.easy-autocomplete-container ul {
  background: none repeat scroll 0 0 #ffffff;
  border-top: 1px dotted #ccc;
  display: none;
  margin-top: 0;
  padding-bottom: 0;
  padding-left: 0;
  position: relative;
  top: -1px;
}
.easy-autocomplete-container ul li, .easy-autocomplete-container ul .eac-category {
  background: inherit;
  border-color: #ccc;
  border-image: none;
  border-style: solid;
  border-width: 0 1px;
  display: block;
  font-size: 14px;
  font-weight: normal;
  padding: 4px 12px;
}
.easy-autocomplete-container ul li:last-child {
  border-radius: 0 0 2px 2px;
  border-width: 0 1px 1px;
}
.easy-autocomplete-container ul li.selected {
  background: none repeat scroll 0 0 #ebebeb;
  cursor: pointer;
```

```
}
.easy-autocomplete-container ul li.selected div {
  font-weight: normal;
}
.easy-autocomplete-container ul li div {
  display: block;
  font-weight: normal;
  word-break: break-all;
}
.easy-autocomplete-container ul li b {
  font-weight: bold;
}
.easy-autocomplete-container ul .eac-category {
  font-color: #aaa;
  font-style: italic;
}

.eac-description .eac-item span {
  color: #aaa;
  font-style: italic;
  font-size: 0.9em;
}

.eac-icon-left .eac-item img {
  margin-right: 4px;
  max-height: 30px;
}

.eac-icon-right .eac-item {
  margin-top: 8px;
  min-height: 24px;
  position: relative;
}
.eac-icon-right .eac-item img {
  margin-left: 4px;
  max-height: 30px;
  position: absolute;
  right: -4px;
  top: -8px;
}

/*# sourceMappingURL=easy-autocomplete.css.map */
```

### EasyAutocomplete-master/dist/jquery.easy-autocomplete.js

Listing 3.3: EasyAutocomplete-master/dist/jquery.easy-autocomplete.js

```
/*
 * easy-autocomplete
 * jQuery plugin for autocompletion
 *
 * @author Łukasz Pawełczak (http://github.com/pawelczak)
 * @version 1.3.5
 * Copyright  License:
 */

```

```
/*
 * EasyAutocomplete - Configuration
 */
var EasyAutocomplete = (function(scope){

        scope.Configuration = function Configuration(options) {
                var defaults = {
                        data: "list-required",
                        url: "list-required",
                        dataType: "json",

                        listLocation: function(data) {
                                return data;
                        },

                        xmlElementName: "",

                        getValue: function(element) {
                                return element;
                        },

                        autocompleteOff: true,

                        placeholder: false,

                        ajaxCallback: function() {},

                        matchResponseProperty: false,

                        list: {
                                sort: {
                                        enabled: false,
                                        method: function(a, b) {
                                                a = defaults.getValue(a);
                                                b = defaults.getValue(b);
                                                if (a < b) {
                                                        return -1;
                                                }
                                                if (a > b) {
                                                        return 1;
                                                }
                                                return 0;
                                        }
                                },

                                maxNumberOfElements: 6,

                                hideOnEmptyPhrase: true,

                                match: {
                                        enabled: false,
                                        caseSensitive: false,
                                        method: function(element, phrase) {

                                                if (element.search(phrase) > -1) {
                                                        return true;
                                                } else {
                                                        return false;
```

```
                                        }
                                }
                        },

                        showAnimation: {
                                type: "normal", //normal|slide|fade
                                time: 400,
                                callback: function() {}
                        },

                        hideAnimation: {
                                type: "normal",
                                time: 400,
                                callback: function() {}
                        },

                        /* Events */
                        onClickEvent: function() {},
                        onSelectItemEvent: function() {},
                        onLoadEvent: function() {},
                        onChooseEvent: function() {},
                        onKeyEnterEvent: function() {},
                        onMouseOverEvent: function() {},
                        onMouseOutEvent: function() {},
                        onShowListEvent: function() {},
                        onHideListEvent: function() {}
                },

                highlightPhrase: true,

                theme: "",

                cssClasses: "",

                minCharNumber: 0,

                requestDelay: 0,

                adjustWidth: true,

                ajaxSettings: {},

                preparePostData: function(data, inputPhrase) {return data;},

                loggerEnabled: true,

                template: "",

                categoriesAssigned: false,

                categories: [{
                        maxNumberOfElements: 4
                }]

        };

        var externalObjects = ["ajaxSettings", "template"];

```

```
126                 this.get = function(propertyName) {
127                         return defaults[propertyName];
128                 };
129
130                 this.equals = function(name, value) {
131                         if (isAssigned(name)) {
132                                 if (defaults[name] === value) {
133                                         return true;
134                                 }
135                         }
136
137                         return false;
138                 };
139
140                 this.checkDataUrlProperties = function() {
141                         if (defaults.url === "list-required" && defaults.data ===
    ↪"list-required") {
142                                 return false;
143                         }
144                         return true;
145                 };
146                 this.checkRequiredProperties = function() {
147                         for (var propertyName in defaults) {
148                                 if (defaults[propertyName] === "required") {
149                                         logger.error("Option " + propertyName + "␣
    ↪must be defined");
150                                         return false;
151                                 }
152                         }
153                         return true;
154                 };
155
156                 this.printPropertiesThatDoesntExist = function(consol,␣
    ↪optionsToCheck) {
157                         printPropertiesThatDoesntExist(consol, optionsToCheck);
158                 };
159
160
161                 prepareDefaults();
162
163                 mergeOptions();
164
165                 if (defaults.loggerEnabled === true) {
166                         printPropertiesThatDoesntExist(console, options);
167                 }
168
169                 addAjaxSettings();
170
171                 processAfterMerge();
172                 function prepareDefaults() {
173
174                         if (options.dataType === "xml") {
175
176                                 if (!options.getValue) {
177
178                                         options.getValue = function(element) {
179                                                 return $(element).text();
180                                         };
```

```
                                }


                                if (!options.list) {

                                        options.list = {};
                                }

                                if (!options.list.sort) {
                                        options.list.sort = {};
                                }


                                options.list.sort.method = function(a, b) {
                                        a = options.getValue(a);
                                        b = options.getValue(b);
                                        if (a < b) {
                                                return -1;
                                        }
                                        if (a > b) {
                                                return 1;
                                        }
                                        return 0;
                                };

                                if (!options.list.match) {
                                        options.list.match = {};
                                }

                                options.list.match.method = function(element, phrase)
↪{

                                        if (element.search(phrase) > -1) {
                                                return true;
                                        } else {
                                                return false;
                                        }
                                };

                        }
                        if (options.categories !== undefined && options.categories␣
↪instanceof Array) {

                                var categories = [];

                                for (var i = 0, length = options.categories.length; i
↪< length; i += 1) {

                                        var category = options.categories[i];

                                        for (var property in defaults.categories[0]) {

                                                if (category[property] === undefined)
↪{
                                                        category[property] = defaults.
↪categories[0][property];
                                                }
                                        }
```

```
234
235                                     categories.push(category);
236                                 }
237
238                                 options.categories = categories;
239                             }
240                         }
241
242                         function mergeOptions() {
243
244                             defaults = mergeObjects(defaults, options);
245
246                             function mergeObjects(source, target) {
247                                 var mergedObject = source || {};
248
249                                 for (var propertyName in source) {
250                                     if (target[propertyName] !== undefined &&␣
↪target[propertyName] !== null) {
251
252                                         if (typeof target[propertyName] !==
↪"object" ||
253                                                 target[propertyName]␣
↪instanceof Array) {
254                                             mergedObject[propertyName] =␣
↪target[propertyName];
255                                         } else {
256                                             mergeObjects(source[propertyName],
↪ target[propertyName]);
257                                         }
258                                     }
259                                 }
260
261                                 /* If data is an object */
262                                 if (target.data !== undefined && target.data !== null␣
↪&& typeof target.data === "object") {
263                                     mergedObject.data = target.data;
264                                 }
265
266                                 return mergedObject;
267                             }
268                         }
269
270
271                         function processAfterMerge() {
272
273                             if (defaults.url !== "list-required" && typeof defaults.url !
↪== "function") {
274                                 var defaultUrl = defaults.url;
275                                 defaults.url = function() {
276                                     return defaultUrl;
277                                 };
278                             }
279
280                             if (defaults.ajaxSettings.url !== undefined && typeof␣
↪defaults.ajaxSettings.url !== "function") {
281                                 var defaultUrl = defaults.ajaxSettings.url;
282                                 defaults.ajaxSettings.url = function() {
283                                     return defaultUrl;
```

```
                                };
                        }

                        if (typeof defaults.listLocation === "string") {
                                var defaultlistLocation = defaults.listLocation;

                                if (defaults.dataType.toUpperCase() === "XML") {
                                        defaults.listLocation = function(data) {
                                                return $(data).
↪find(defaultlistLocation);
                                        };
                                } else {
                                        defaults.listLocation = function(data) {
                                                return data[defaultlistLocation];
                                        };
                                }
                        }

                        if (typeof defaults.getValue === "string") {
                                var defaultsGetValue = defaults.getValue;
                                defaults.getValue = function(element) {
                                        return element[defaultsGetValue];
                                };
                        }

                        if (options.categories !== undefined) {
                                defaults.categoriesAssigned = true;
                        }

                }

                function addAjaxSettings() {

                        if (options.ajaxSettings !== undefined && typeof options.
↪ajaxSettings === "object") {
                                defaults.ajaxSettings = options.ajaxSettings;
                        } else {
                                defaults.ajaxSettings = {};
                        }

                }

                function isAssigned(name) {
                        if (defaults[name] !== undefined && defaults[name] !== null) {
                                return true;
                        } else {
                                return false;
                        }
                }
                function printPropertiesThatDoesntExist(consol, optionsToCheck) {

                        checkPropertiesIfExist(defaults, optionsToCheck);

                        function checkPropertiesIfExist(source, target) {
                                for(var property in target) {
                                        if (source[property] === undefined) {
                                                consol.log("Property '" + property + "
↪' does not exist in EasyAutocomplete options API.");
```

```
                                        }

                                        if (typeof source[property] === "object" && $.
↪inArray(property, externalObjects) === -1) {
                                                checkPropertiesIfExist(source[property],
↪ target[property]);
                                        }
                                }
                        }
                };

                return scope;

})(EasyAutocomplete || {});


/*
 * EasyAutocomplete - Logger
 */
var EasyAutocomplete = (function(scope){

        scope.Logger = function Logger() {

                this.error = function(message) {
                        console.log("ERROR: " + message);
                };

                this.warning = function(message) {
                        console.log("WARNING: " + message);
                };
        };

        return scope;

})(EasyAutocomplete || {});


/*
 * EasyAutocomplete - Constans
 */
var EasyAutocomplete = (function(scope){

        scope.Constans = function Constans() {
                var constants = {
                        CONTAINER_CLASS: "easy-autocomplete-container",
                        CONTAINER_ID: "eac-container-",

                        WRAPPER_CSS_CLASS: "easy-autocomplete"
                };

                this.getValue = function(propertyName) {
                        return constants[propertyName];
                };

        };

        return scope;
```

```
395
396 })(EasyAutocomplete || {});
397
398 /*
399  * EasyAutocomplete - ListBuilderService
400  *
401  * @author Łukasz Pawełczak
402  *
403  */
404 var EasyAutocomplete = (function(scope) {
405
406         scope.ListBuilderService = function ListBuilderService(configuration,␣
    ↪proccessResponseData) {
407
408
409                 this.init = function(data) {
410                         var listBuilder = [],
411                                 builder = {};
412
413                         builder.data = configuration.get("listLocation")(data);
414                         builder.getValue = configuration.get("getValue");
415                         builder.maxListSize = configuration.get("list").
    ↪maxNumberOfElements;
416
417
418                         listBuilder.push(builder);
419
420                         return listBuilder;
421                 };
422
423                 this.updateCategories = function(listBuilder, data) {
424
425                         if (configuration.get("categoriesAssigned")) {
426
427                                 listBuilder = [];
428
429                                 for(var i = 0; i < configuration.get("categories").
    ↪length; i += 1) {
430
431                                         var builder =␣
    ↪convertToListBuilder(configuration.get("categories")[i], data);
432
433                                         listBuilder.push(builder);
434                                 }
435
436                         }
437
438                         return listBuilder;
439                 };
440
441                 this.convertXml = function(listBuilder) {
442                         if(configuration.get("dataType").toUpperCase() === "XML") {
443
444                                 for(var i = 0; i < listBuilder.length; i += 1) {
445                                         listBuilder[i].data =␣
    ↪convertXmlToList(listBuilder[i]);
446                                 }
447                         }
```

```
448
449                     return listBuilder;
450             };
451
452             this.processData = function(listBuilder, inputPhrase) {
453
454                     for(var i = 0, length = listBuilder.length; i < length; i+=1)
    ↪{
455                             listBuilder[i].data =␣
    ↪proccessResponseData(configuration, listBuilder[i], inputPhrase);
456                     }
457
458                     return listBuilder;
459             };
460
461             this.checkIfDataExists = function(listBuilders) {
462
463                     for(var i = 0, length = listBuilders.length; i < length; i +=␣
    ↪1) {
464
465                             if (listBuilders[i].data !== undefined &&␣
    ↪listBuilders[i].data instanceof Array) {
466                                     if (listBuilders[i].data.length > 0) {
467                                             return true;
468                                     }
469                             }
470                     }
471
472                     return false;
473             };
474
475
476             function convertToListBuilder(category, data) {
477
478                     var builder = {};
479
480                     if(configuration.get("dataType").toUpperCase() === "XML") {
481
482                             builder = convertXmlToListBuilder();
483                     } else {
484
485                             builder = convertDataToListBuilder();
486                     }
487
488
489                     if (category.header !== undefined) {
490                             builder.header = category.header;
491                     }
492
493                     if (category.maxNumberOfElements !== undefined) {
494                             builder.maxNumberOfElements = category.
    ↪maxNumberOfElements;
495                     }
496
497                     if (configuration.get("list").maxNumberOfElements !==␣
    ↪undefined) {
498
499                             builder.maxListSize = configuration.get("list").
    ↪maxNumberOfElements;
```

```
500                         }
501
502                         if (category.getValue !== undefined) {
503
504                                 if (typeof category.getValue === "string") {
505                                         var defaultsGetValue = category.getValue;
506                                         builder.getValue = function(element) {
507                                                 return element[defaultsGetValue];
508                                         };
509                                 } else if (typeof category.getValue === "function") {
510                                         builder.getValue = category.getValue;
511                                 }
512
513                         } else {
514                                 builder.getValue = configuration.get("getValue");
515                         }
516
517
518                         return builder;
519
520
521                         function convertXmlToListBuilder() {
522
523                                 var builder = {},
524                                         listLocation;
525
526                                 if (category.xmlElementName !== undefined) {
527                                         builder.xmlElementName = category.
    ↪xmlElementName;
528                                 }
529
530                                 if (category.listLocation !== undefined) {
531
532                                         listLocation = category.listLocation;
533                                 } else if (configuration.get("listLocation") !==␣
    ↪undefined) {
534
535                                         listLocation = configuration.get("listLocation
    ↪");
536                                 }
537
538                                 if (listLocation !== undefined) {
539                                         if (typeof listLocation === "string") {
540                                                 builder.data = $(data).
    ↪find(listLocation);
541                                         } else if (typeof listLocation === "function
    ↪") {
542
543                                                 builder.data = listLocation(data);
544                                         }
545                                 } else {
546
547                                         builder.data = data;
548                                 }
549
550                                 return builder;
551                         }
552
```

```
553
554                         function convertDataToListBuilder() {
555
556                                 var builder = {};
557
558                                 if (category.listLocation !== undefined) {
559
560                                         if (typeof category.listLocation === "string
    ↪") {
561                                                 builder.data = data[category.
    ↪listLocation];
562                                         } else if (typeof category.listLocation ===
    ↪"function") {
563                                                 builder.data = category.
    ↪listLocation(data);
564                                         }
565                                 } else {
566                                         builder.data = data;
567                                 }
568
569                                 return builder;
570                         }
571                 }
572
573                 function convertXmlToList(builder) {
574                         var simpleList = [];
575
576                         if (builder.xmlElementName === undefined) {
577                                 builder.xmlElementName = configuration.get(
    ↪"xmlElementName");
578                         }
579
580
581                         $(builder.data).find(builder.xmlElementName).each(function() {
582                                 simpleList.push(this);
583                         });
584
585                         return simpleList;
586                 }
587
588         };
589
590         return scope;
591
592 })(EasyAutocomplete || {});
593
594
595 /*
596  * EasyAutocomplete - Data proccess module
597  *
598  * Process list to display:
599  * - sort
600  * - decrease number to specific number
601  * - show only matching list
602  *
603  */
604 var EasyAutocomplete = (function(scope) {
605
```

```
606         scope.proccess = function proccessData(config, listBuilder, phrase) {
607
608                 scope.proccess.match = match;
609
610                 var list = listBuilder.data,
611                         inputPhrase = phrase;//TODO REFACTOR
612
613                 list = findMatch(list, inputPhrase);
614                 list = reduceElementsInList(list);
615                 list = sort(list);
616
617                 return list;
618
619
620                 function findMatch(list, phrase) {
621                         var preparedList = [],
622                                 value = "";
623
624                         if (config.get("list").match.enabled) {
625
626                                 for(var i = 0, length = list.length; i < length; i +=
↪1) {
627
628                                         value = config.get("getValue")(list[i]);
629
630                                         if (match(value, phrase)) {
631                                                 preparedList.push(list[i]);
632                                         }
633
634                                 }
635
636                         } else {
637                                 preparedList = list;
638                         }
639
640                         return preparedList;
641                 }
642
643                 function match(value, phrase) {
644
645                         if (!config.get("list").match.caseSensitive) {
646
647                                 if (typeof value === "string") {
648                                         value = value.toLowerCase();
649                                 }
650
651                                 phrase = phrase.toLowerCase();
652                         }
653                         if (config.get("list").match.method(value, phrase)) {
654                                 return true;
655                         } else {
656                                 return false;
657                         }
658                 }
659
660                 function reduceElementsInList(list) {
661                         if (listBuilder.maxNumberOfElements !== undefined && list.
↪length > listBuilder.maxNumberOfElements) {
```

```
662                                 list = list.slice(0, listBuilder.maxNumberOfElements);
663                         }
664
665                         return list;
666                 }
667
668                 function sort(list) {
669                         if (config.get("list").sort.enabled) {
670                                 list.sort(config.get("list").sort.method);
671                         }
672
673                         return list;
674                 }
675
676         };
677
678
679         return scope;
680
681
682 })(EasyAutocomplete || {});
683
684
685 /*
686  * EasyAutocomplete - Template
687  *
688  *
689  *
690  */
691 var EasyAutocomplete = (function(scope){
692
693         scope.Template = function Template(options) {
694
695
696                 var genericTemplates = {
697                         basic: {
698                                 type: "basic",
699                                 method: function(element) { return element; },
700                                 cssClass: ""
701                         },
702                         description: {
703                                 type: "description",
704                                 fields: {
705                                         description: "description"
706                                 },
707                                 method: function(element) {         return element + "␣
    ↪- description"; },
708                                 cssClass: "eac-description"
709                         },
710                         iconLeft: {
711                                 type: "iconLeft",
712                                 fields: {
713                                         icon: ""
714                                 },
715                                 method: function(element) {
716                                         return element;
717                                 },
718                                 cssClass: "eac-icon-left"
```

```
719                         },
720                         iconRight: {
721                                 type: "iconRight",
722                                 fields: {
723                                         iconSrc: ""
724                                 },
725                                 method: function(element) {
726                                         return element;
727                                 },
728                                 cssClass: "eac-icon-right"
729                         },
730                         links: {
731                                 type: "links",
732                                 fields: {
733                                         link: ""
734                                 },
735                                 method: function(element) {
736                                         return element;
737                                 },
738                                 cssClass: ""
739                         },
740                         custom: {
741                                 type: "custom",
742                                 method: function() {},
743                                 cssClass: ""
744                         }
745                 },
746
747
748
749                 /*
750                  * Converts method with {{text}} to function
751                  */
752                 convertTemplateToMethod = function(template) {
753
754
755                         var _fields = template.fields,
756                                 buildMethod;
757
758                         if (template.type === "description") {
759
760                                 buildMethod = genericTemplates.description.method;
761
762                                 if (typeof _fields.description === "string") {
763                                         buildMethod = function(elementValue, element)
↪{
764                                                 return elementValue + " - <span>" +␣
↪element[_fields.description] + "</span>";
765                                         };
766                                 } else if (typeof _fields.description === "function")
↪{
767                                         buildMethod = function(elementValue, element)
↪{
768                                                 return elementValue + " - <span>" + _
↪fields.description(element) + "</span>";
769                                         };
770                                 }
771
```

```
772                                 return buildMethod;
773                         }
774
775                         if (template.type === "iconRight") {
776
777                                 if (typeof _fields.iconSrc === "string") {
778                                         buildMethod = function(elementValue, element)
↪{
779                                                 return elementValue + "<img class=
↪'eac-icon' src='" + element[_fields.iconSrc] + "' />" ;
780                                         };
781                                 } else if (typeof _fields.iconSrc === "function") {
782                                         buildMethod = function(elementValue, element)
↪{
783                                                 return elementValue + "<img class=
↪'eac-icon' src='" + _fields.iconSrc(element) + "' />" ;
784                                         };
785                                 }
786
787                                 return buildMethod;
788                         }
789
790
791                         if (template.type === "iconLeft") {
792
793                                 if (typeof _fields.iconSrc === "string") {
794                                         buildMethod = function(elementValue, element)
↪{
795                                                 return "<img class='eac-icon' src='"␣
↪+ element[_fields.iconSrc] + "' />" + elementValue;
796                                         };
797                                 } else if (typeof _fields.iconSrc === "function") {
798                                         buildMethod = function(elementValue, element)
↪{
799                                                 return "<img class='eac-icon' src='"␣
↪+ _fields.iconSrc(element) + "' />" + elementValue;
800                                         };
801                                 }
802
803                                 return buildMethod;
804                         }
805
806                         if(template.type === "links") {
807
808                                 if (typeof _fields.link === "string") {
809                                         buildMethod = function(elementValue, element)
↪{
810                                                 return "<a href='" + element[_fields.
↪link] + "' >" + elementValue + "</a>";
811                                         };
812                                 } else if (typeof _fields.link === "function") {
813                                         buildMethod = function(elementValue, element)
↪{
814                                                 return "<a href='" + _fields.
↪link(element) + "' >" + elementValue + "</a>";
815                                         };
816                                 }
817
```

```
818                                 return buildMethod;
819                         }
820
821
822                         if (template.type === "custom") {
823
824                                 return template.method;
825                         }
826
827                         return genericTemplates.basic.method;
828
829                 },
830
831
832                 prepareBuildMethod = function(options) {
833                         if (!options || !options.type) {
834
835                                 return genericTemplates.basic.method;
836                         }
837
838                         if (options.type && genericTemplates[options.type]) {
839
840                                 return convertTemplateToMethod(options);
841                         } else {
842
843                                 return genericTemplates.basic.method;
844                         }
845
846                 },
847
848                 templateClass = function(options) {
849                         var emptyStringFunction = function() {return "";};
850
851                         if (!options || !options.type) {
852
853                                 return emptyStringFunction;
854                         }
855
856                         if (options.type && genericTemplates[options.type]) {
857                                 return (function () {
858                                         var _cssClass = genericTemplates[options.
↪type].cssClass;
859                                         return function() { return _cssClass;};
860                                 })();
861                         } else {
862                                 return emptyStringFunction;
863                         }
864                 };
865
866
867                 this.getTemplateClass = templateClass(options);
868
869                 this.build = prepareBuildMethod(options);
870
871
872         };
873
874         return scope;
```

```
875
876 })(EasyAutocomplete || {});
877
878
879 /*
880  * EasyAutocomplete - jQuery plugin for autocompletion
881  *
882  */
883 var EasyAutocomplete = (function(scope) {
884
885
886         scope.main = function Core($input, options) {
887
888                 var module = {
889                                 name: "EasyAutocomplete",
890                                 shortcut: "eac"
891                         };
892
893                 var consts = new scope.Constans(),
894                         config = new scope.Configuration(options),
895                         logger = new scope.Logger(),
896                         template = new scope.Template(options.template),
897                         listBuilderService = new scope.ListBuilderService(config,␣
    ↪scope.proccess),
898                         checkParam = config.equals,
899
900                         $field = $input,
901                         $container = "",
902                         elementsList = [],
903                         selectedElement = -1,
904                         requestDelayTimeoutId;
905
906                 scope.consts = consts;
907
908                 this.getConstants = function() {
909                         return consts;
910                 };
911
912                 this.getConfiguration = function() {
913                         return config;
914                 };
915
916                 this.getContainer = function() {
917                         return $container;
918                 };
919
920                 this.getSelectedItemIndex = function() {
921                         return selectedElement;
922                 };
923
924                 this.getItems = function () {
925                         return elementsList;
926                 };
927
928                 this.getItemData = function(index) {
929
930                         if (elementsList.length < index || elementsList[index] ===␣
    ↪undefined) {
```

```
931                                 return -1;
932                         } else {
933                                 return elementsList[index];
934                         }
935                 };
936
937                 this.getSelectedItemData = function() {
938                         return this.getItemData(selectedElement);
939                 };
940
941                 this.build = function() {
942                         prepareField();
943                 };
944
945                 this.init = function() {
946                         init();
947                 };
948                 function init() {
949
950                         if ($field.length === 0) {
951                                 logger.error("Input field doesn't exist.");
952                                 return;
953                         }
954
955                         if (!config.checkDataUrlProperties()) {
956                                 logger.error("One of options variables 'data' or 'url
↪' must be defined.");
957                                 return;
958                         }
959
960                         if (!config.checkRequiredProperties()) {
961                                 logger.error("Will not work without mentioned␣
↪properties.");
962                                 return;
963                         }
964
965
966                         prepareField();
967                         bindEvents();
968
969                 }
970                 function prepareField() {
971
972
973                         if ($field.parent().hasClass(consts.getValue("WRAPPER_CSS_
↪CLASS"))) {
974                                 removeContainer();
975                                 removeWrapper();
976                         }
977
978                         createWrapper();
979                         createContainer();
980
981                         $container = $("#" + getContainerId());
982                         if (config.get("placeholder")) {
983                                 $field.attr("placeholder", config.get("placeholder"));
984                         }
985
```

```
986
987                         function createWrapper() {
988                                 var $wrapper = $("<div>"),
989                                         classes = consts.getValue("WRAPPER_CSS_CLASS
↪");
990
991
992                                 if (config.get("theme") && config.get("theme") !== "
↪") {
993                                         classes += " eac-" + config.get("theme");
994                                 }
995
996                                 if (config.get("cssClasses") && config.get("cssClasses
↪") !== "") {
997                                         classes += " " + config.get("cssClasses");
998                                 }
999
1000                                if (template.getTemplateClass() !== "") {
1001                                        classes += " " + template.getTemplateClass();
1002                                }
1003
1004
1005                                $wrapper
1006                                        .addClass(classes);
1007                                $field.wrap($wrapper);
1008
1009
1010                                if (config.get("adjustWidth") === true) {
1011                                        adjustWrapperWidth();
1012                                }
1013
1014
1015                        }
1016
1017                        function adjustWrapperWidth() {
1018                                var fieldWidth = $field.outerWidth();
1019
1020                                $field.parent().css("width", fieldWidth);
1021                        }
1022
1023                        function removeWrapper() {
1024                                $field.unwrap();
1025                        }
1026
1027                        function createContainer() {
1028                                var $elements_container = $("<div>").addClass(consts.
↪getValue("CONTAINER_CLASS"));
1029
1030                                $elements_container
1031                                                .attr("id", getContainerId())
1032                                                .prepend($("<ul>"));
1033
1034
1035                                (function() {
1036
1037                                        $elements_container
1038                                                /* List show animation */
1039                                                .on("show.eac", function() {
```

```
                                                    switch(config.get("list").
↪showAnimation.type) {

                                                        case "slide":
                                                            var␣
↪animationTime = config.get("list").showAnimation.time,
                                                                                callback␣
↪= config.get("list").showAnimation.callback;

                                                            $elements_
↪container.find("ul").slideDown(animationTime, callback);
                                                        break;

                                                        case "fade":
                                                            var␣
↪animationTime = config.get("list").showAnimation.time,
                                                                                callback␣
↪= config.get("list").showAnimation.callback;

                                                            $elements_
↪container.find("ul").fadeIn(animationTime), callback;
                                                        break;

                                                        default:
                                                            $elements_
↪container.find("ul").show();
                                                        break;
                                                    }

                                                    config.get("list").
↪onShowListEvent();

                                                })
                                                /* List hide animation */
                                                .on("hide.eac", function() {

                                                    switch(config.get("list").
↪hideAnimation.type) {

                                                        case "slide":
                                                            var␣
↪animationTime = config.get("list").hideAnimation.time,
                                                                                callback␣
↪= config.get("list").hideAnimation.callback;

                                                            $elements_
↪container.find("ul").slideUp(animationTime, callback);
                                                        break;

                                                        case "fade":
                                                            var␣
↪animationTime = config.get("list").hideAnimation.time,
                                                                                callback␣
↪= config.get("list").hideAnimation.callback;

                                                            $elements_
↪container.find("ul").fadeOut(animationTime, callback);
```

```
                                                                    break;

                                                                default:
                                                                        $elements_
↪container.find("ul").hide();
                                                                    break;
                                                            }

                                                            config.get("list").
↪onHideListEvent();

                                                    })
                                                    .on("selectElement.eac", function() {
                                                            $elements_container.find("ul␣
↪li").removeClass("selected");
                                                            $elements_container.find("ul␣
↪li").eq(selectedElement).addClass("selected");

                                                            config.get("list").
↪onSelectItemEvent();
                                                    })
                                                    .on("loadElements.eac",␣
↪function(event, listBuilders, phrase) {


                                                            var $item = "",
                                                                    $listContainer =
↪$elements_container.find("ul");

                                                            $listContainer
                                                                    .empty()
                                                                    .detach();

                                                            elementsList = [];
                                                            var counter = 0;
                                                            for(var builderIndex = 0,␣
↪listBuildersLength = listBuilders.length; builderIndex < listBuildersLength;␣
↪builderIndex += 1) {

                                                                    var listData =␣
↪listBuilders[builderIndex].data;

                                                                    if (listData.length␣
↪=== 0) {
                                                                            continue;
                                                                    }

                                                                    if␣
↪(listBuilders[builderIndex].header !== undefined && listBuilders[builderIndex].
↪header.length > 0) {

↪$listContainer.append("<div class='eac-category' >" + listBuilders[builderIndex].
↪header + "</div>");
                                                                    }

                                                                    for(var i = 0,␣
↪listDataLength = listData.length; i < listDataLength && counter <␣
↪listBuilders[builderIndex].maxListSize; i += 1) {
```

```
                                                                    $item = $("
↪<li><div class='eac-item'></div></li>");


                                                                    (function() {
                                                                        var j␣
↪= i,
                                                                    itemCounter␣
↪= counter,
                                                                    elementsValue
↪= listBuilders[builderIndex].getValue(listData[j]);

                                                                    $item.
↪find(" > div")
                                                                    .
↪on("click", function() {

↪$field.val(elementsValue).trigger("change");

                                                                        selec
↪= itemCounter;
                                                                        selec
↪

                                                                        confi
↪get("list").onClickEvent();
                                                                        confi
↪get("list").onChooseEvent();
                                                                    }
↪)
                                                                    .
↪mouseover(function() {

                                                                        selec
↪= itemCounter;
                                                                        selec
↪

                                                                        confi
↪get("list").onMouseOverEvent();
                                                                    }
↪)
                                                                    .
↪mouseout(function() {
                                                                        confi
↪get("list").onMouseOutEvent();
                                                                    }
↪)
                                                                    .
↪html(template.build(highlight(elementsValue, phrase), listData[j]));
                                                                    })();


↪$listContainer.append($item);
                                                                    elementsList.
↪push(listData[i]);
                                                                    counter += 1;
```

```
                                                                }
                                                        }

                                                        $elements_container.append(
↪$listContainer);

                                                        config.get("list").
↪onLoadEvent();
                                                });

                                })();

                                $field.after($elements_container);
                        }

                        function removeContainer() {
                                $field.next("." + consts.getValue("CONTAINER_CLASS")).
↪remove();
                        }

                        function highlight(string, phrase) {

                                if(config.get("highlightPhrase") && phrase !== "") {
                                        return highlightPhrase(string, phrase);
                                } else {
                                        return string;
                                }

                        }

                        function escapeRegExp(str) {
                                return str.replace(/[\-\[\]\/\{\}\(\)\*\+\?\.\\\^\
↪$\|]/g, "\\$&");
                         }

                        function highlightPhrase(string, phrase) {
                                var escapedPhrase = escapeRegExp(phrase);
                                return (string + "").replace(new RegExp("(" +␣
↪escapedPhrase + ")", "gi") , "<b>$1</b>");
                                }



                }
                function getContainerId() {

                        var elementId = $field.attr("id");

                        elementId = consts.getValue("CONTAINER_ID") + elementId;

                        return elementId;
                }
                function bindEvents() {

                        bindAllEvents();


                        function bindAllEvents() {
```

```
                                if (checkParam("autocompleteOff", true)) {
                                        removeAutocomplete();
                                }

                                bindFocusOut();
                                bindKeyup();
                                bindKeydown();
                                bindKeypress();
                                bindFocus();
                                bindBlur();
                        }

                        function bindFocusOut() {
                                $field.focusout(function () {

                                        var fieldValue = $field.val(),
                                                phrase;

                                        if (!config.get("list").match.caseSensitive) {
                                                fieldValue = fieldValue.toLowerCase();
                                        }

                                        for (var i = 0, length = elementsList.length;
↪i < length; i += 1) {

                                                phrase = config.get("getValue
↪")(elementsList[i]);
                                                if (!config.get("list").match.
↪caseSensitive) {
                                                        phrase = phrase.toLowerCase();
                                                }

                                                if (phrase === fieldValue) {
                                                        selectedElement = i;
                                                        selectElement(selectedElement);
↪
                                                        return;
                                                }
                                        }
                                });
                        }

                        function bindKeyup() {
                                $field
                                .off("keyup")
                                .keyup(function(event) {

                                        switch(event.keyCode) {

                                                case 27:

                                                        hideContainer();
                                                        loseFieldFocus();
                                                break;

                                                case 38:

                                                        event.preventDefault();
```

```
                                                    if(elementsList.length > 0 &&␣
↪selectedElement > 0) {
                                                            selectedElement -= 1;
                                                            $field.val(config.get(
↪"getValue")(elementsList[selectedElement]));
                                                            selectElement(selectedElement);
↪
                                                    }
                                            break;
                                            case 40:
                                                    event.preventDefault();
                                                    if(elementsList.length > 0 &&␣
↪selectedElement < elementsList.length - 1) {
                                                            selectedElement += 1;
                                                            $field.val(config.get(
↪"getValue")(elementsList[selectedElement]));
                                                            selectElement(selectedElement);
↪
                                                    }
                                            break;
                                            default:
                                                    if (event.keyCode > 40 ||␣
↪event.keyCode === 8) {
                                                            var inputPhrase =
↪$field.val();
                                                            if (!(config.get("list
↪").hideOnEmptyPhrase === true && event.keyCode === 8 && inputPhrase === "")) {
                                                                    if (config.
↪get("requestDelay") > 0) {
                                                                            if␣
↪(requestDelayTimeoutId !== undefined) {
                                                                                    clearTimeout
↪
                                                                            }
                                                                            requestDelayTimeoutId
↪= setTimeout(function () { loadData(inputPhrase);}, config.get("requestDelay"));
                                                                    } else {
                                                                            loadData(inputPhrase)
↪
```

```
                                                                }

                                                        } else {
                                                                hideContainer();
↪
                                                        }

                                                }


                                        break;
                                }


                                function loadData(inputPhrase) {


                                        if (inputPhrase.length < config.get(
↪"minCharNumber")) {
                                                return;
                                        }


                                        if (config.get("data") !== "list-
↪required") {

                                                var data = config.get("data");

                                                var listBuilders =␣
↪listBuilderService.init(data);

                                                listBuilders =␣
↪listBuilderService.updateCategories(listBuilders, data);

                                                listBuilders =␣
↪listBuilderService.processData(listBuilders, inputPhrase);

                                                loadElements(listBuilders,␣
↪inputPhrase);

                                                if ($field.parent().find("li
↪").length > 0) {
                                                        showContainer();
                                                } else {
                                                        hideContainer();
                                                }

                                        }

                                        var settings = createAjaxSettings();

                                        if (settings.url === undefined ||␣
↪settings.url === "") {
                                                settings.url = config.get("url
↪");
                                        }

                                        if (settings.dataType === undefined␣
↪|| settings.dataType === "") {
```

```
                                                    settings.dataType = config.
↪get("dataType");
                                            }


                                            if (settings.url !== undefined &&␣
↪settings.url !== "list-required") {

                                                    settings.url = settings.
↪url(inputPhrase);

                                                    settings.data = config.get(
↪"preparePostData")(settings.data, inputPhrase);

                                                    $.ajax(settings)
                                                            .done(function(data) {

                                                                    var␣
↪listBuilders = listBuilderService.init(data);

                                                                    listBuilders␣
↪= listBuilderService.updateCategories(listBuilders, data);

                                                                    listBuilders␣
↪= listBuilderService.convertXml(listBuilders);
                                                                    if␣
↪(checkInputPhraseMatchResponse(inputPhrase, data)) {

                                                                            listBuilders␣
↪= listBuilderService.processData(listBuilders, inputPhrase);

                                                                            loadElements(listBui
↪ inputPhrase);

                                                                    }

                                                                    if␣
↪(listBuilderService.checkIfDataExists(listBuilders) && $field.parent().find("li").
↪length > 0) {
                                                                            showContainer();
↪
                                                                    } else {
                                                                            hideContainer();
↪
                                                                    }

                                                                    config.get(
↪"ajaxCallback")();

                                                            })
                                                            .fail(function() {
                                                                    logger.
↪warning("Fail to load response data");
                                                            })
                                                            .always(function() {

                                                            });
                                            }
```

```
1399
1400
1401
1402                                                function createAjaxSettings() {
1403
1404                                                        var settings = {},
1405                                                                ajaxSettings = config.
    ↪get("ajaxSettings") || {};
1406
1407                                                        for (var set in ajaxSettings)
    ↪{
1408                                                                settings[set] =␣
    ↪ajaxSettings[set];
1409                                                        }
1410
1411                                                        return settings;
1412                                                }
1413
1414                                                function␣
    ↪checkInputPhraseMatchResponse(inputPhrase, data) {
1415
1416                                                        if (config.get(
    ↪"matchResponseProperty") !== false) {
1417                                                                if (typeof config.get(
    ↪"matchResponseProperty") === "string") {
1418                                                                        return␣
    ↪(data[config.get("matchResponseProperty")] === inputPhrase);
1419                                                                }
1420
1421                                                                if (typeof config.get(
    ↪"matchResponseProperty") === "function") {
1422                                                                        return␣
    ↪(config.get("matchResponseProperty")(data) === inputPhrase);
1423                                                                }
1424
1425                                                                return true;
1426                                                        } else {
1427                                                                return true;
1428                                                        }
1429
1430                                                }
1431
1432                                        }
1433
1434
1435                                });
1436                        }
1437
1438                        function bindKeydown() {
1439                                $field
1440                                        .on("keydown", function(evt) {
1441                                    evt = evt || window.event;
1442                                    var keyCode = evt.keyCode;
1443                                    if (keyCode === 38) {
1444                                        suppressKeypress = true;
1445                                        return false;
1446                                    }
1447                                })
```

```
                                        .keydown(function(event) {

                                                if (event.keyCode === 13 &&␣
↪selectedElement > -1) {

                                                        $field.val(config.get(
↪"getValue")(elementsList[selectedElement]));

                                                        config.get("list").
↪onKeyEnterEvent();
                                                        config.get("list").
↪onChooseEvent();

                                                        selectedElement = -1;
                                                        hideContainer();

                                                        event.preventDefault();
                                                }
                                        });
                        }

                        function bindKeypress() {
                                $field
                                .off("keypress");
                        }

                        function bindFocus() {
                                $field.focus(function() {

                                        if ($field.val() !== "" && elementsList.
↪length > 0) {

                                                selectedElement = -1;
                                                showContainer();
                                        }

                                });
                        }

                        function bindBlur() {
                                $field.blur(function() {
                                        setTimeout(function() {

                                                selectedElement = -1;
                                                hideContainer();
                                        }, 250);
                                });
                        }

                        function removeAutocomplete() {
                                $field.attr("autocomplete","off");
                        }

                }

                function showContainer() {
                        $container.trigger("show.eac");
                }
```

```
1501
1502                function hideContainer() {
1503                        $container.trigger("hide.eac");
1504                }
1505
1506                function selectElement(index) {
1507
1508                        $container.trigger("selectElement.eac", index);
1509                }
1510
1511                function loadElements(list, phrase) {
1512                        $container.trigger("loadElements.eac", [list, phrase]);
1513                }
1514
1515                function loseFieldFocus() {
1516                        $field.trigger("blur");
1517                }
1518
1519
1520        };
1521        scope.eacHandles = [];
1522
1523        scope.getHandle = function(id) {
1524                return scope.eacHandles[id];
1525        };
1526
1527        scope.inputHasId = function(input) {
1528
1529                if($(input).attr("id") !== undefined && $(input).attr("id").length >
↪0) {
1530                        return true;
1531                } else {
1532                        return false;
1533                }
1534
1535        };
1536
1537        scope.assignRandomId = function(input) {
1538
1539                var fieldId = "";
1540
1541                do {
1542                        fieldId = "eac-" + Math.floor(Math.random() * 10000);
1543                } while ($("#" + fieldId).length !== 0);
1544
1545                elementId = scope.consts.getValue("CONTAINER_ID") + fieldId;
1546
1547                $(input).attr("id", fieldId);
1548
1549        };
1550
1551        scope.setHandle = function(handle, id) {
1552                scope.eacHandles[id] = handle;
1553        };
1554
1555
1556        return scope;
1557
```

```
})(EasyAutocomplete || {});

(function($) {

        $.fn.easyAutocomplete = function(options) {

                return this.each(function() {
                        var $this = $(this),
                                eacHandle = new EasyAutocomplete.main($this, options);

                        if (!EasyAutocomplete.inputHasId($this)) {
                                EasyAutocomplete.assignRandomId($this);
                        }

                        eacHandle.init();

                        EasyAutocomplete.setHandle(eacHandle, $this.attr("id"));

                });
        };

        $.fn.getSelectedItemIndex = function() {

                var inputId = $(this).attr("id");

                if (inputId !== undefined) {
                        return EasyAutocomplete.getHandle(inputId).
↪getSelectedItemIndex();
                }

                return -1;
        };

        $.fn.getItems = function () {

                var inputId = $(this).attr("id");

                if (inputId !== undefined) {
                        return EasyAutocomplete.getHandle(inputId).getItems();
                }

                return -1;
        };

        $.fn.getItemData = function(index) {

                var inputId = $(this).attr("id");

                if (inputId !== undefined && index > -1) {
                        return EasyAutocomplete.getHandle(inputId).getItemData(index);
                }

                return -1;
        };

        $.fn.getSelectedItemData = function() {

                var inputId = $(this).attr("id");
```

```
                if (inputId !== undefined) {
                        return EasyAutocomplete.getHandle(inputId).
↪getSelectedItemData();
                }

                return -1;
        };

})(jQuery);
```

### EasyAutocomplete simple example


### EasyAutocomplete-master/demo/example_simple.html

Listing 3.4: EasyAutocomplete-master/demo/example_simple.html

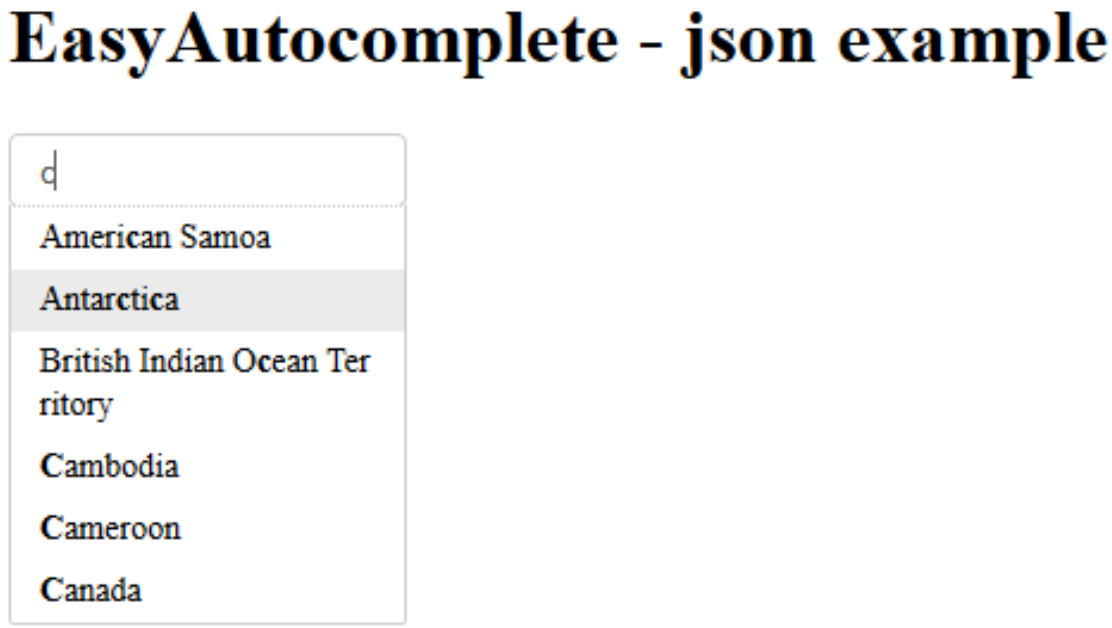
```
<!DOCTYPE html>
<html lang="en">
    <head>
        <title>EasyAutocomplete simple example</title>

        <!-- STYLE -> CSS -->
            <link href="../dist/easy-autocomplete.min.css" rel="stylesheet" type=
↪"text/css">
        <!-- end STYLE-->

    </head>
    <body>

        <!-- STRUCTURE -> HTML5 elements -->
            <h1>EasyAutocomplete - easy example</h1>
            <input id="simple" />
        <!-- end STRUCTURE-->

        <!--BEHAVIOR -> Javascript scripts-->
```

```
            <script src="../lib/jquery-1.11.2.min.js"></script>
            <script src="../dist/jquery.easy-autocomplete.min.js" type="text/
↪javascript" ></script>
            <script>

                var options_easy_autocomplete = {
                    data: ["blue", "green", "pink", "red", "yellow"],

                };

                $("#simple").easyAutocomplete(options_easy_autocomplete);

            </script>
        <!-- end BEHAVIOR -->

    </body>

</html>
```

#### EasyAutocomplete JSON example

**See also:**

*JSON*




#### EasyAutocomplete-master/demo/example_json.html

Listing 3.5: EasyAutocomplete-master/demo/example_json.html

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <title>EasyAutocomplete json example</title>

        <!-- STYLE -> CSS -->
            <link href="../dist/easy-autocomplete.min.css" rel="stylesheet" type=
↪"text/css">
        <!-- end STYLE-->
    </head>
    <body>

        <!-- STRUCTURE -> HTML5 elements -->
            <h1>EasyAutocomplete - json example</h1>
            <input id="data-json" />
        <!-- end STRUCTURE-->

        <!--BEHAVIOR -> Javascript scripts-->
            <script src="../lib/jquery-1.11.2.min.js"></script>
            <script src="../dist/jquery.easy-autocomplete.min.js" type="text/
↪javascript" ></script>
            <script>
                let options_easy_autocomplete = {
                    url: "resources/countries.json",

                    getValue: "name",

                    list: {
                        match: {
                            enabled: true
                        }
                    }
                };
                $("#data-json").easyAutocomplete(options_easy_autocomplete);
            </script>
        <!-- end BEHAVIOR -->
    </body>

</html>
```

### EasyAutocomplete-master/demo/resources/countries.json

Listing 3.6: EasyAutocomplete-master/demo/resources/countries.json

```
[
  {"name": "Afghanistan", "code": "AF"},
  {"name": "Albania", "code": "AL"},
  {"name": "Algeria", "code": "DZ"},
  {"name": "American Samoa", "code": "AS"},
  {"name": "AndorrA", "code": "AD"},
  {"name": "Angola", "code": "AO"},
  {"name": "Anguilla", "code": "AI"},
  {"name": "Antarctica", "code": "AQ"},
  {"name": "Antigua and Barbuda", "code": "AG"},
```

```
    {"name": "Argentina", "code": "AR"},
    {"name": "Armenia", "code": "AM"},
    {"name": "Aruba", "code": "AW"},
    {"name": "Australia", "code": "AU"},
    {"name": "Austria", "code": "AT"},
    {"name": "Azerbaijan", "code": "AZ"},
    {"name": "Bahamas", "code": "BS"},
    {"name": "Bahrain", "code": "BH"},
    {"name": "Bangladesh", "code": "BD"},
    {"name": "Barbados", "code": "BB"},
    {"name": "Belarus", "code": "BY"},
    {"name": "Belgium", "code": "BE"},
    {"name": "Belize", "code": "BZ"},
    {"name": "Benin", "code": "BJ"},
    {"name": "Bermuda", "code": "BM"},
    {"name": "Bhutan", "code": "BT"},
    {"name": "Bolivia", "code": "BO"},
    {"name": "Bosnia and Herzegovina", "code": "BA"},
    {"name": "Botswana", "code": "BW"},
    {"name": "Bouvet Island", "code": "BV"},
    {"name": "Brazil", "code": "BR"},
    {"name": "British Indian Ocean Territory", "code": "IO"},
    {"name": "Brunei Darussalam", "code": "BN"},
    {"name": "Bulgaria", "code": "BG"},
    {"name": "Burkina Faso", "code": "BF"},
    {"name": "Burundi", "code": "BI"},
    {"name": "Cambodia", "code": "KH"},
    {"name": "Cameroon", "code": "CM"},
    {"name": "Canada", "code": "CA"},
    {"name": "Cape Verde", "code": "CV"},
    {"name": "Cayman Islands", "code": "KY"},
    {"name": "Central African Republic", "code": "CF"},
    {"name": "Chad", "code": "TD"},
    {"name": "Chile", "code": "CL"},
    {"name": "China", "code": "CN"},
    {"name": "Christmas Island", "code": "CX"},
    {"name": "Cocos (Keeling) Islands", "code": "CC"},
    {"name": "Colombia", "code": "CO"},
    {"name": "Comoros", "code": "KM"},
    {"name": "Congo", "code": "CG"},
    {"name": "Congo, The Democratic Republic of the", "code": "CD"},
    {"name": "Cook Islands", "code": "CK"},
    {"name": "Costa Rica", "code": "CR"},
    {"name": "Cote D\"Ivoire", "code": "CI"},
    {"name": "Croatia", "code": "HR"},
    {"name": "Cuba", "code": "CU"},
    {"name": "Cyprus", "code": "CY"},
    {"name": "Czech Republic", "code": "CZ"},
    {"name": "Denmark", "code": "DK"},
    {"name": "Djibouti", "code": "DJ"},
    {"name": "Dominica", "code": "DM"},
    {"name": "Dominican Republic", "code": "DO"},
    {"name": "Ecuador", "code": "EC"},
    {"name": "Egypt", "code": "EG"},
    {"name": "El Salvador", "code": "SV"},
    {"name": "Equatorial Guinea", "code": "GQ"},
    {"name": "Eritrea", "code": "ER"},
    {"name": "Estonia", "code": "EE"},
```

```
  {"name": "Ethiopia", "code": "ET"},
  {"name": "Falkland Islands (Malvinas)", "code": "FK"},
  {"name": "Faroe Islands", "code": "FO"},
  {"name": "Fiji", "code": "FJ"},
  {"name": "Finland", "code": "FI"},
  {"name": "France", "code": "FR"},
  {"name": "French Guiana", "code": "GF"},
  {"name": "French Polynesia", "code": "PF"},
  {"name": "French Southern Territories", "code": "TF"},
  {"name": "Gabon", "code": "GA"},
  {"name": "Gambia", "code": "GM"},
  {"name": "Georgia", "code": "GE"},
  {"name": "Germany", "code": "DE"},
  {"name": "Ghana", "code": "GH"},
  {"name": "Gibraltar", "code": "GI"},
  {"name": "Greece", "code": "GR"},
  {"name": "Greenland", "code": "GL"},
  {"name": "Grenada", "code": "GD"},
  {"name": "Guadeloupe", "code": "GP"},
  {"name": "Guam", "code": "GU"},
  {"name": "Guatemala", "code": "GT"},
  {"name": "Guernsey", "code": "GG"},
  {"name": "Guinea", "code": "GN"},
  {"name": "Guinea-Bissau", "code": "GW"},
  {"name": "Guyana", "code": "GY"},
  {"name": "Haiti", "code": "HT"},
  {"name": "Heard Island and Mcdonald Islands", "code": "HM"},
  {"name": "Holy See (Vatican City State)", "code": "VA"},
  {"name": "Honduras", "code": "HN"},
  {"name": "Hong Kong", "code": "HK"},
  {"name": "Hungary", "code": "HU"},
  {"name": "Iceland", "code": "IS"},
```

### EasyAutocomplete flags example

### EasyAutocomplete-master/demo/example_flags.html

Listing 3.7: EasyAutocomplete-master/demo/example_flags.html

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <title>EasyAutocomplete flags example</title>

        <!-- STYLE -> CSS -->
            <link href="../dist/easy-autocomplete.min.css" rel="stylesheet" type=
↪"text/css">
            <link href="resources/flags.css"  rel="stylesheet" type="text/css" >
        <!-- end STYLE-->

    </head>
    <body>
        <!-- STRUCTURE -> HTML5 elements -->
            <h1>EasyAutocomplete - country flags example</h1>

            <input id="flags" />
```


```
        <!-- end STRUCTURE-->

        <!--BEHAVIOR -> Javascript scripts-->
            <script src="../lib/jquery-1.11.2.min.js"></script>
            <script src="../dist/jquery.easy-autocomplete.min.js" type="text/
↪javascript" ></script>
            <script>
                var options_easy_autocomplete = {
                    url: "resources/countries.json",

                    getValue: "name",

                    list: {
                        match: {
                            enabled: true
                        },
                        maxNumberOfElements: 10
                    },

                    template: {
                        type: "custom",
                        method: function(value, item) {
                            return "<span class='flag flag-" + (item.code).
↪toLowerCase() + "' ></span>" + value;
                        }
                    }
                };

                $("#flags").easyAutocomplete(options_easy_autocomplete);
            </script>
        <!-- end BEHAVIOR -->
    </body>

```

```
</html>
```

### EasyAutocomplete-master/demo/resources/flags.css

Listing 3.8: EasyAutocomplete-master/demo/resources/flags.css

```
.flag {
        width: 16px;
        height: 11px;
        background:url(flags.png) no-repeat;
        display: inline-block;
        margin-right: 5px;
}

.flag.flag-ad {background-position: -16px 0}
.flag.flag-ae {background-position: -32px 0}
.flag.flag-af {background-position: -48px 0}
.flag.flag-ag {background-position: -64px 0}
.flag.flag-ai {background-position: -80px 0}
.flag.flag-al {background-position: -96px 0}
.flag.flag-am {background-position: -112px 0}
.flag.flag-an {background-position: -128px 0}
.flag.flag-ao {background-position: -144px 0}
.flag.flag-ar {background-position: -160px 0}
.flag.flag-as {background-position: -176px 0}
.flag.flag-at {background-position: -192px 0}
.flag.flag-au {background-position: -208px 0}
.flag.flag-aw {background-position: -224px 0}
.flag.flag-az {background-position: -240px 0}
.flag.flag-ba {background-position: 0 -11px}
.flag.flag-bb {background-position: -16px -11px}
.flag.flag-bd {background-position: -32px -11px}
.flag.flag-be {background-position: -48px -11px}
.flag.flag-bf {background-position: -64px -11px}
.flag.flag-bg {background-position: -80px -11px}
.flag.flag-bh {background-position: -96px -11px}
.flag.flag-bi {background-position: -112px -11px}
.flag.flag-bj {background-position: -128px -11px}
.flag.flag-bm {background-position: -144px -11px}
.flag.flag-bn {background-position: -160px -11px}
.flag.flag-bo {background-position: -176px -11px}
.flag.flag-br {background-position: -192px -11px}
.flag.flag-bs {background-position: -208px -11px}
.flag.flag-bt {background-position: -224px -11px}
.flag.flag-bv {background-position: -240px -11px}
.flag.flag-bw {background-position: 0 -22px}
.flag.flag-by {background-position: -16px -22px}
.flag.flag-bz {background-position: -32px -22px}
.flag.flag-ca {background-position: -48px -22px}
.flag.flag-catalonia {background-position: -64px -22px}
.flag.flag-cd {background-position: -80px -22px}
.flag.flag-cf {background-position: -96px -22px}
.flag.flag-cg {background-position: -112px -22px}
.flag.flag-ch {background-position: -128px -22px}
.flag.flag-ci {background-position: -144px -22px}
.flag.flag-ck {background-position: -160px -22px}
```

```
.flag.flag-cl {background-position: -176px -22px}
.flag.flag-cm {background-position: -192px -22px}
.flag.flag-cn {background-position: -208px -22px}
.flag.flag-co {background-position: -224px -22px}
.flag.flag-cr {background-position: -240px -22px}
.flag.flag-cu {background-position: 0 -33px}
.flag.flag-cv {background-position: -16px -33px}
.flag.flag-cw {background-position: -32px -33px}
.flag.flag-cy {background-position: -48px -33px}
.flag.flag-cz {background-position: -64px -33px}
.flag.flag-de {background-position: -80px -33px}
.flag.flag-dj {background-position: -96px -33px}
.flag.flag-dk {background-position: -112px -33px}
.flag.flag-dm {background-position: -128px -33px}
.flag.flag-do {background-position: -144px -33px}
.flag.flag-dz {background-position: -160px -33px}
.flag.flag-ec {background-position: -176px -33px}
.flag.flag-ee {background-position: -192px -33px}
.flag.flag-eg {background-position: -208px -33px}
.flag.flag-eh {background-position: -224px -33px}
.flag.flag-england {background-position: -240px -33px}
.flag.flag-er {background-position: 0 -44px}
.flag.flag-es {background-position: -16px -44px}
.flag.flag-et {background-position: -32px -44px}
.flag.flag-eu {background-position: -48px -44px}
.flag.flag-fi {background-position: -64px -44px}
.flag.flag-fj {background-position: -80px -44px}
.flag.flag-fk {background-position: -96px -44px}
.flag.flag-fm {background-position: -112px -44px}
.flag.flag-fo {background-position: -128px -44px}
.flag.flag-fr {background-position: -144px -44px}
.flag.flag-ga {background-position: -160px -44px}
.flag.flag-gb {background-position: -176px -44px}
.flag.flag-gd {background-position: -192px -44px}
.flag.flag-ge {background-position: -208px -44px}
.flag.flag-gf {background-position: -224px -44px}
.flag.flag-gg {background-position: -240px -44px}
.flag.flag-gh {background-position: 0 -55px}
.flag.flag-gi {background-position: -16px -55px}
.flag.flag-gl {background-position: -32px -55px}
.flag.flag-gm {background-position: -48px -55px}
.flag.flag-gn {background-position: -64px -55px}
.flag.flag-gp {background-position: -80px -55px}
.flag.flag-gq {background-position: -96px -55px}
.flag.flag-gr {background-position: -112px -55px}
.flag.flag-gs {background-position: -128px -55px}
.flag.flag-gt {background-position: -144px -55px}
.flag.flag-gu {background-position: -160px -55px}
.flag.flag-gw {background-position: -176px -55px}
.flag.flag-gy {background-position: -192px -55px}
.flag.flag-hk {background-position: -208px -55px}
.flag.flag-hm {background-position: -224px -55px}
.flag.flag-hn {background-position: -240px -55px}
.flag.flag-hr {background-position: 0 -66px}
.flag.flag-ht {background-position: -16px -66px}
.flag.flag-hu {background-position: -32px -66px}
.flag.flag-ic {background-position: -48px -66px}
.flag.flag-id {background-position: -64px -66px}
```

```
.flag.flag-ie {background-position: -80px -66px}
.flag.flag-il {background-position: -96px -66px}
.flag.flag-im {background-position: -112px -66px}
.flag.flag-in {background-position: -128px -66px}
.flag.flag-io {background-position: -144px -66px}
.flag.flag-iq {background-position: -160px -66px}
.flag.flag-ir {background-position: -176px -66px}
.flag.flag-is {background-position: -192px -66px}
.flag.flag-it {background-position: -208px -66px}
.flag.flag-je {background-position: -224px -66px}
.flag.flag-jm {background-position: -240px -66px}
.flag.flag-jo {background-position: 0 -77px}
.flag.flag-jp {background-position: -16px -77px}
.flag.flag-ke {background-position: -32px -77px}
.flag.flag-kg {background-position: -48px -77px}
.flag.flag-kh {background-position: -64px -77px}
.flag.flag-ki {background-position: -80px -77px}
.flag.flag-km {background-position: -96px -77px}
.flag.flag-kn {background-position: -112px -77px}
.flag.flag-kp {background-position: -128px -77px}
.flag.flag-kr {background-position: -144px -77px}
.flag.flag-kurdistan {background-position: -160px -77px}
.flag.flag-kw {background-position: -176px -77px}
.flag.flag-ky {background-position: -192px -77px}
.flag.flag-kz {background-position: -208px -77px}
.flag.flag-la {background-position: -224px -77px}
.flag.flag-lb {background-position: -240px -77px}
.flag.flag-lc {background-position: 0 -88px}
.flag.flag-li {background-position: -16px -88px}
.flag.flag-lk {background-position: -32px -88px}
.flag.flag-lr {background-position: -48px -88px}
.flag.flag-ls {background-position: -64px -88px}
.flag.flag-lt {background-position: -80px -88px}
.flag.flag-lu {background-position: -96px -88px}
.flag.flag-lv {background-position: -112px -88px}
.flag.flag-ly {background-position: -128px -88px}
.flag.flag-ma {background-position: -144px -88px}
.flag.flag-mc {background-position: -160px -88px}
.flag.flag-md {background-position: -176px -88px}
.flag.flag-me {background-position: -192px -88px}
.flag.flag-mg {background-position: -208px -88px}
.flag.flag-mh {background-position: -224px -88px}
.flag.flag-mk {background-position: -240px -88px}
.flag.flag-ml {background-position: 0 -99px}
.flag.flag-mm {background-position: -16px -99px}
.flag.flag-mn {background-position: -32px -99px}
.flag.flag-mo {background-position: -48px -99px}
.flag.flag-mp {background-position: -64px -99px}
.flag.flag-mq {background-position: -80px -99px}
.flag.flag-mr {background-position: -96px -99px}
.flag.flag-ms {background-position: -112px -99px}
.flag.flag-mt {background-position: -128px -99px}
.flag.flag-mu {background-position: -144px -99px}
.flag.flag-mv {background-position: -160px -99px}
.flag.flag-mw {background-position: -176px -99px}
.flag.flag-mx {background-position: -192px -99px}
.flag.flag-my {background-position: -208px -99px}
.flag.flag-mz {background-position: -224px -99px}
```

```
.flag.flag-na {background-position: -240px -99px}
.flag.flag-nc {background-position: 0 -110px}
.flag.flag-ne {background-position: -16px -110px}
.flag.flag-nf {background-position: -32px -110px}
.flag.flag-ng {background-position: -48px -110px}
.flag.flag-ni {background-position: -64px -110px}
.flag.flag-nl {background-position: -80px -110px}
.flag.flag-no {background-position: -96px -110px}
.flag.flag-np {background-position: -112px -110px}
.flag.flag-nr {background-position: -128px -110px}
.flag.flag-nu {background-position: -144px -110px}
.flag.flag-nz {background-position: -160px -110px}
.flag.flag-om {background-position: -176px -110px}
.flag.flag-pa {background-position: -192px -110px}
.flag.flag-pe {background-position: -208px -110px}
.flag.flag-pf {background-position: -224px -110px}
.flag.flag-pg {background-position: -240px -110px}
.flag.flag-ph {background-position: 0 -121px}
.flag.flag-pk {background-position: -16px -121px}
.flag.flag-pl {background-position: -32px -121px}
.flag.flag-pm {background-position: -48px -121px}
.flag.flag-pn {background-position: -64px -121px}
.flag.flag-pr {background-position: -80px -121px}
.flag.flag-ps {background-position: -96px -121px}
.flag.flag-pt {background-position: -112px -121px}
.flag.flag-pw {background-position: -128px -121px}
.flag.flag-py {background-position: -144px -121px}
.flag.flag-qa {background-position: -160px -121px}
.flag.flag-re {background-position: -176px -121px}
.flag.flag-ro {background-position: -192px -121px}
.flag.flag-cs {background-position: -208px -121px}
.flag.flag-ru {background-position: -224px -121px}
.flag.flag-rw {background-position: -240px -121px}
.flag.flag-sa {background-position: 0 -132px}
.flag.flag-sb {background-position: -16px -132px}
.flag.flag-sc {background-position: -32px -132px}
.flag.flag-scotland {background-position: -48px -132px}
.flag.flag-sd {background-position: -64px -132px}
.flag.flag-se {background-position: -80px -132px}
.flag.flag-sg {background-position: -96px -132px}
.flag.flag-sh {background-position: -112px -132px}
.flag.flag-si {background-position: -128px -132px}
.flag.flag-sk {background-position: -144px -132px}
.flag.flag-sl {background-position: -160px -132px}
.flag.flag-sm {background-position: -176px -132px}
.flag.flag-sn {background-position: -192px -132px}
.flag.flag-so {background-position: -208px -132px}
.flag.flag-somaliland {background-position: -224px -132px}
.flag.flag-sr {background-position: -240px -132px}
.flag.flag-ss {background-position: 0 -143px}
.flag.flag-st {background-position: -16px -143px}
.flag.flag-sv {background-position: -32px -143px}
.flag.flag-sx {background-position: -48px -143px}
.flag.flag-sy {background-position: -64px -143px}
.flag.flag-sz {background-position: -80px -143px}
.flag.flag-tc {background-position: -96px -143px}
.flag.flag-td {background-position: -112px -143px}
.flag.flag-tf {background-position: -128px -143px}
```

```
.flag.flag-tg {background-position: -144px -143px}
.flag.flag-th {background-position: -160px -143px}
.flag.flag-tj {background-position: -176px -143px}
.flag.flag-tk {background-position: -192px -143px}
.flag.flag-tl {background-position: -208px -143px}
.flag.flag-tm {background-position: -224px -143px}
.flag.flag-tn {background-position: -240px -143px}
.flag.flag-to {background-position: 0 -154px}
.flag.flag-tr {background-position: -16px -154px}
.flag.flag-tt {background-position: -32px -154px}
.flag.flag-tv {background-position: -48px -154px}
.flag.flag-tw {background-position: -64px -154px}
.flag.flag-tz {background-position: -80px -154px}
.flag.flag-ua {background-position: -96px -154px}
.flag.flag-ug {background-position: -112px -154px}
.flag.flag-um {background-position: -128px -154px}
.flag.flag-us {background-position: -144px -154px}
.flag.flag-uy {background-position: -160px -154px}
.flag.flag-uz {background-position: -176px -154px}
.flag.flag-va {background-position: -192px -154px}
.flag.flag-vc {background-position: -208px -154px}
.flag.flag-ve {background-position: -224px -154px}
.flag.flag-vg {background-position: -240px -154px}
.flag.flag-vi {background-position: 0 -165px}
.flag.flag-vn {background-position: -16px -165px}
.flag.flag-vu {background-position: -32px -165px}
.flag.flag-wales {background-position: -48px -165px}
.flag.flag-wf {background-position: -64px -165px}
.flag.flag-ws {background-position: -80px -165px}
.flag.flag-xk {background-position: -96px -165px}
.flag.flag-ye {background-position: -112px -165px}
.flag.flag-yt {background-position: -128px -165px}
.flag.flag-za {background-position: -144px -165px}
.flag.flag-zanzibar {background-position: -160px -165px}
.flag.flag-zm {background-position: -176px -165px}
.flag.flag-zw {background-position: -192px -165px}
```

### EasyAutocomplete-master/demo/resources/flags.png

Fig. 3.2: EasyAutocomplete-master/demo/resources/flags.png

### EasyAutocomplete categories example


### EasyAutocomplete-master/demo/example_categories.html

Listing 3.9: EasyAutocomplete-master/demo/example_categories.html

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <title>EasyAutocomplete categories</title>
        <!-- STYLE -> CSS -->
            <link href="../dist/easy-autocomplete.min.css" rel="stylesheet" type="text/css">
            <link href="../dist/easy-autocomplete.themes.min.css" rel="stylesheet" type="text/css">
            <link href='//fonts.googleapis.com/css?family=Roboto:400,300,300italic,400italic,600,600italic&amp;subset=latin,cyrillic-ext,greek-ext,latin-ext,cyrillic' rel='stylesheet' type='text/css'>
        <!-- end STYLE-->
    </head>
    <body>
        <!-- STRUCTURE -> HTML5 elements -->
            <h1>EasyAutocomplete - categories example</h1>
            <input id="data-categories" placeholder="Fruits and vegetables" />
        <!-- end STRUCTURE-->

        <!--BEHAVIOR -> Javascript scripts-->
            <script src="../lib/jquery-1.11.2.min.js"></script>
            <script src="../dist/jquery.easy-autocomplete.min.js" type="text/javascript" ></script>
            <script>
```

```
                var options_easy_autocomplete = {
                    url: "resources/categories.json",

                    categories: [{
                        listLocation: "fruits",
                        header: "--- FRUITS ---"
                    }, {
                        listLocation: "vegetables",
                        header: "--- VEGETABLES ---"
                    }],

                    list: {
                        match: {
                            enabled: false
                        },
                        maxNumberOfElements: 10
                    },

                    theme: "dark"
                };

                $("#data-categories").easyAutocomplete(options_easy_autocomplete);

            </script>
        <!-- end BEHAVIOR -->
    </body>

</html>
```

### EasyAutocomplete-master/demo/resources/categories.json

Listing 3.10: EasyAutocomplete-master/demo/resources/categories.json

```
{
        "fruits": ["Apple", "Cherry", "Clementine", "Honeydew melon", "Watermelon",
↪"Satsuma"],
        "vegetables": ["Pepper", "Jerusalem artichoke", "Green bean", "Fennel",
↪"Courgette", "Yam"]
}
```

### EasyAutocomplete static_link example

### EasyAutocomplete-master/demo/example_static_link.html

Listing 3.11: EasyAutocomplete-master/demo/example_static_link.html

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <title>EasyAutocomplete static links</title>
        <!-- STYLE -> CSS -->
            <link href="../dist/easy-autocomplete.min.css" rel="stylesheet" type=
↪"text/css">
        <!-- end STYLE-->
```


```
    </head>
    <body>

        <!-- STRUCTURE -> HTML5 elements -->
            <h1>EasyAutocomplete - static links</h1>
            <input id="data-links" />
        <!-- end STRUCTURE-->


        <!--BEHAVIOR -> Javascript scripts-->
            <script src="../lib/jquery-1.11.2.min.js"></script>
            <script src="../dist/jquery.easy-autocomplete.min.js" type="text/
↪javascript" ></script>
            <script>
                var options_easy_autocomplete = {
                    url: "resources/site.json",

                    getValue: "text",

                    template: {
                        type: "links",
                        fields: {
                            link: "site"
                        }
                    },

                    list: {
                        match: {
                            enabled: false
                        }
                    },

                    theme: "plate-dark"
                };

                $("#data-links").easyAutocomplete(options_easy_autocomplete);

            </script>
        <!-- end BEHAVIOR -->
    </body>
```

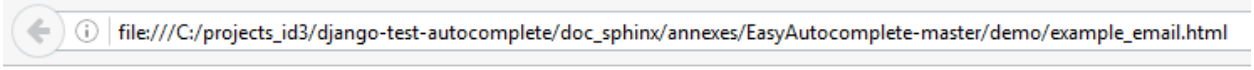
```
</html>
```

### EasyAutocomplete-master/demo/resources/site.json

Listing 3.12: EasyAutocomplete-master/demo/resources/site.json

```
[{
    "text": "Home",
    "site": "http://easyautocomplete.com"
    },
    {
    "text": "Guide",
    "site": "http://easyautocomplete.com/guide"
    },
    {
    "text": "Examples",
    "site": "http://easyautocomplete.com/examples"
    },
    {
    "text": "Themes",
    "site": "http://easyautocomplete.com/themes"
    },
    {
    "text": "Download",
    "site": "http://easyautocomplete.com/download"
    }
]
```

### EasyAutocomplete email example




# EasyAutocomplete - email example


### EasyAutocomplete-master/demo/example_email.html

Listing 3.13: EasyAutocomplete-master/demo/example_email.html

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <title>EasyAutocomplete email example</title>
        <!-- STYLE -> CSS -->
            <link href="../dist/easy-autocomplete.min.css" rel="stylesheet" type=
↪"text/css">
        <!-- end STYLE-->
    </head>
    <body>

        <!-- STRUCTURE -> HTML5 elements -->
            <h1>EasyAutocomplete - email example</h1>
            <input id="data-email" />
        <!-- end STRUCTURE-->

        <!--BEHAVIOR -> Javascript scripts-->
            <script src="../lib/jquery-1.11.2.min.js"></script>
            <script src="../dist/jquery.easy-autocomplete.min.js" type="text/
↪javascript" ></script>
            <script>
                var options_easy_autocomplete = {
                    url: "resources/people.json",

                    getValue: "email",

                    template: {
                        type: "description",
                        fields: {
                            description: "email"
                        }
                    },

                    list: {
                        match: {
                            enabled: true
                        }
                    },

                    theme: "plate-dark"
                };

                $("#data-email").easyAutocomplete(options_easy_autocomplete);
            </script>
        <!-- end BEHAVIOR -->

    </body>

</html>
```

### EasyAutocomplete-master/demo/resources/people.json

Listing 3.14: EasyAutocomplete-master/demo/resources/people.json

```
[    {
        "name": "Declan Haley2 s",
        "email": "Cr12as.lorem.lorem@nonquam.ca"
    },
    {
        "name": "Francis Marsh",
        "email": "ne2que@arcu.edu"
    },
    {
        "name": "Gage Figueroa",
        "email": "Sed.auctor.odio@magnis.ca"
    },
    {
        "name": "Asher Gay",
        "email": "Phasellus@nonsapien.ca"
    },
    {
        "name": "Erasmus Roach",
        "email": "amet@aptent.net"
    },
    {
        "name": "Francis Johnston",
        "email": "Ut.sagittis.lobortis@Donecsollicitudin.com"
    },
    {
        "name": "Dustin Mckinney",
        "email": "velit.Quisque@liberodui.co.uk"
    },
    {
        "name": "Zane Camacho",
        "email": "et.netus@Phasellusvitae.co.uk"
    },
    {
        "name": "Murphy Larsen",
        "email": "sit.amet@felisadipiscing.com"
    },
    {
        "name": "Shad Kemp",
        "email": "justo@laoreetlectus.co.uk"
    },
    {
        "name": "Octavius Wolfe",
        "email": "sociis@aptenttacitisociosqu.org"
    },
    {
        "name": "Cooper Bell",
        "email": "Quisque@apurus.ca"
    },
    {
        "name": "Lev Mccarthy",
        "email": "diam@amet.co.uk"
    },
    {
        "name": "Arthur Mejia",
        "email": "Nulla.dignissim@urna.ca"
    },
```

```
    {
        "name": "Ferris Cameron",
        "email": "vulputate.velit@In.net"
    },
    {
        "name": "Ivor Greer",
        "email": "nisi@Nuncmauriselit.ca"
    },
    {
        "name": "Alan Church",
        "email": "et.ipsum@dolor.co.uk"
    },
    {
        "name": "Walter Berry",
        "email": "neque@lacus.edu"
    },
    {
        "name": "Hyatt Bowman",
        "email": "lobortis.ultrices@mollis.org"
    },
    {
        "name": "Acton Bradley",
        "email": "commodo.tincidunt@ut.com"
    },
    {
        "name": "Duncan Hancock",
        "email": "Donec@sem.net"
    },
    {
        "name": "Macaulay Trujillo",
        "email": "in.sodales.elit@Donec.ca"
    },
    {
        "name": "Jack Estrada",
        "email": "nibh@Quisquetinciduntpede.net"
    },
    {
        "name": "Edward Wilkerson",
        "email": "Quisque.fringilla.euismod@idliberoDonec.co.uk"
    },
    {
        "name": "Clarke Mullen",
        "email": "nunc.Quisque.ornare@leo.com"
    },
    {
        "name": "Steel Rodriquez",
        "email": "a@diam.ca"
    },
    {
        "name": "Jerome Edwards",
        "email": "tincidunt@consectetuer.net"
    },
    {
        "name": "Quentin Blevins",
        "email": "In.scelerisque@idrisus.co.uk"
    },
    {
        "name": "Maxwell Baxter",
```

```
        "email": "lacinia.mattis@erateget.ca"
    },
    {
        "name": "Beau Mcclain",
        "email": "dis.parturient@feugiatnon.co.uk"
    },
    {
        "name": "Jarrod Valenzuela",
        "email": "adipiscing@FuscefeugiatLorem.net"
    },
    {
        "name": "Grant Rosario",
        "email": "feugiat@rutrumFuscedolor.co.uk"
    },
    {
        "name": "Wade Atkinson",
        "email": "et@euerosNam.com"
    },
    {
        "name": "Ryder Lindsey",
        "email": "Quisque.fringilla.euismod@Suspendissealiquetmolestie.edu"
    },
    {
        "name": "Upton Schmidt",
        "email": "arcu.Nunc.mauris@quis.ca"
    },
    {
        "name": "Seth Tate",
        "email": "elit.a.feugiat@turpisnon.edu"
    },
    {
        "name": "Emery Shields",
        "email": "mi.Aliquam.gravida@vestibulummassarutrum.net"
    },
    {
        "name": "Slade Bolton",
        "email": "venenatis.a@fringillami.co.uk"
    },
    {
        "name": "Lucas Yates",
        "email": "purus.in@idnuncinterdum.ca"
    },
    {
        "name": "Dean Rosa",
        "email": "vulputate.ullamcorper@commodoipsum.ca"
    },
    {
        "name": "Ahmed Daniel",
        "email": "ligula@idsapien.ca"
    },
    {
        "name": "Hector Hardin",
        "email": "egestas@Nullam.edu"
    },
    {
        "name": "Byron Dickerson",
        "email": "sociis@Proinultrices.co.uk"
    },
```

```
    {
        "name": "Myles Workman",
        "email": "magnis.dis@odioPhasellusat.co.uk"
    },
    {
        "name": "Jerry Howard",
        "email": "tortor.nibh@sitamet.edu"
    },
    {
        "name": "Igor Chase",
        "email": "sagittis@gravida.edu"
    },
    {
        "name": "Aquila Mcclure",
        "email": "dictum@Phasellusinfelis.co.uk"
    },
    {
        "name": "John Hansen",
        "email": "ut@molestietellus.com"
    },
    {
        "name": "Forrest Goodwin",
        "email": "enim.nec.tempus@massalobortis.com"
    },
    {
        "name": "Griffin Lott",
        "email": "enim.Curabitur.massa@convalliserat.edu"
    },
```

### EasyAutocomplete Django integration

**See also:**

- http://easyautocomplete.com/guide

### Include JS and CSS files from the distribution (static files)

**See also:**

- http://easyautocomplete.com/download
- https://docs.djangoproject.com/en/dev/howto/static-files/

First thing you need to do, is to download EasyAutocomplete plugin files.

More information about it can be found in the Download section.

Javascript and css files(in the actual version) are located in *dist folder* of the project.

---

**Note:** If you want to make any changes, uncompiled javascript files and unprocessed sass files can be found in the src folder.

---

```
|
+---dist
|    |    easy-autocomplete.css
|    |    easy-autocomplete.min.css
|    |    easy-autocomplete.themes.css
|    |    easy-autocomplete.themes.min.css
|    |    jquery.easy-autocomplete.js
|    |    jquery.easy-autocomplete.min.js
|    |
|    \---maps
|              easy-autocomplete.css.map
|              easy-autocomplete.min.css.map
|              easy-autocomplete.themes.css.map
|              easy-autocomplete.themes.min.css.map
|
```

### Exemples of static files in the Django World

### The Django contrib module

```
C:\PROJECT\PYTHON_ENVS\DJANGO_TEST_AUTOCOMPLETE_35_64\LIB\SITE-
↪PACKAGES\DJANGO\CONTRIB\ADMIN\STATIC
\---admin
    +---css
    |       base.css
    |       changelists.css
    |       dashboard.css
    |       fonts.css
    |       forms.css
    |       login.css
    |       rtl.css
    |       widgets.css
    |
    +---fonts
    |       LICENSE.txt
    |       README.txt
    |       Roboto-Bold-webfont.woff
    |       Roboto-Light-webfont.woff
    |       Roboto-Regular-webfont.woff
    |
    +---img
    |   |   calendar-icons.svg
    |   |   icon-addlink.svg
    |   |   icon-alert.svg
    |   |   icon-calendar.svg
    |   |   icon-changelink.svg
    |   |   icon-clock.svg
    |   |   icon-deletelink.svg
    |   |   icon-no.svg
    |   |   icon-unknown-alt.svg
```

```
    |   |   icon-unknown.svg
    |   |   icon-yes.svg
    |   |   inline-delete.svg
    |   |   LICENSE
    |   |   README.txt
    |   |   search.svg
    |   |   selector-icons.svg
    |   |   sorting-icons.svg
    |   |   tooltag-add.svg
    |   |   tooltag-arrowright.svg
    |   |
    |   \---gis
    |           move_vertex_off.svg
    |           move_vertex_on.svg
    |
    \---js
        |   actions.js
        |   actions.min.js
        |   calendar.js
        |   cancel.js
        |   change_form.js
        |   collapse.js
        |   collapse.min.js
        |   core.js
        |   inlines.js
        |   inlines.min.js
        |   jquery.init.js
        |   popup_response.js
        |   prepopulate.js
        |   prepopulate.min.js
        |   prepopulate_init.js
        |   SelectBox.js
        |   SelectFilter2.js
        |   timeparse.js
        |   urlify.js
        |
        +---admin
        |       DateTimeShortcuts.js
        |       RelatedObjectLookups.js
        |
        \---vendor
            +---jquery
            |       jquery.js
            |       jquery.min.js
            |       LICENSE-JQUERY.txt
            |
            \---xregexp
                    LICENSE-XREGEXP.txt
                    xregexp.js
                    xregexp.min.js
```

### Django_By_Example_Code/Chapter 1/mysite/blog

The static directory is in the blog **application directory**.

```
C:\TMP\DJANGO-BY-EXAMPLE-BOOK-MASTER\CHAP_1_BLOG\MYSITE\BLOG
|   admin.py
|   apps.py
|   models.py
|   tests.py
|   urls.py
|   views.py
|   __init__.py
|
+---migrations
|       0001_initial.py
|       __init__.py
|
+---static
|   \---css
|           blog.css
|
\---templates
    |   pagination.html
    |
    \---blog
        |   base.html
        |   pagination.html
        |
        \---post
                detail.html
                list.html
                pagination.html
```

### Django_By_Example_Code/Chapter 8

There are static directories in:

- the courses and shop **applications directory**
- the **project directory**.

```
\---myshop
    |   manage.py
    |
    +---cart
    |   |   admin.py
    |   |   cart.py

    |
    +---myshop
    |       celery.py
    |       db.sqlite3
    |       settings.py
    |       urls.py
    |       wsgi.py
    |       __init__.py
    |
    +---orders
    |   |   admin.py
    |   |   forms.py
    |   |   models.py
```

```
|   |
|   +---static
|   |   \---css
|   |           admin.css
|   |           pdf.css
|   |

|
+---shop
|   |   admin.py
|   |   models.py
|   |   tests.py
|   |   urls.py

|   |
|   +---static
|   |   +---css
|   |   |       base.css
|   |   |
|   |   \---img
|   |           no_image.png
|   |

|
\---static
    +---admin
    |   +---css
    |   |       base.css
    |   |       changelists.css
    |   |       dashboard.css
    |   |       forms.css
    |   |       ie.css
    |   |       login.css
    |   |       rtl.css
    |   |       widgets.css
    |   |
    |   +---img
    |   |   |   changelist-bg.gif
    |   |   |   changelist-bg_rtl.gif
    |   |   |   default-bg-reverse.gif
    |   |   |   default-bg.gif
    |   |   |   deleted-overlay.gif
    |   |   |   icon-no.gif
    |   |   |   icon-unknown.gif
    |   |   |   icon-yes.gif
    |   |   |   icon_addlink.gif
    |   |   |   icon_alert.gif
    |   |   |   icon_calendar.gif
    |   |   |   icon_changelink.gif
    |   |   |   icon_clock.gif
    |   |   |   icon_deletelink.gif
    |   |   |   icon_error.gif
    |   |   |   icon_searchbox.png
    |   |   |   icon_success.gif
    |   |   |   inline-delete-8bit.png
    |   |   |   inline-delete.png
    |   |   |   inline-restore-8bit.png
```

```
|   |   |   inline-restore.png
|   |   |   inline-splitter-bg.gif
|   |   |   nav-bg-grabber.gif
|   |   |   nav-bg-reverse.gif
|   |   |   nav-bg-selected.gif
|   |   |   nav-bg.gif
|   |   |   selector-icons.gif
|   |   |   selector-search.gif
|   |   |   sorting-icons.gif
|   |   |   tooltag-add.png
|   |   |   tooltag-arrowright.png
|   |   |
|   |   \---gis
|   |           move_vertex_off.png
|   |           move_vertex_on.png
|   |
|   \---js
|       |   actions.js
|       |   actions.min.js
|       |   calendar.js
|       |   collapse.js
|       |   collapse.min.js
|       |   core.js
|       |   inlines.js
|       |   inlines.min.js
|       |   jquery.init.js
|       |   jquery.js
|       |   jquery.min.js
|       |   LICENSE-JQUERY.txt
|       |   prepopulate.js
|       |   prepopulate.min.js
|       |   related-widget-wrapper.js
|       |   SelectBox.js
|       |   SelectFilter2.js
|       |   timeparse.js
|       |   urlify.js
|       |
|       \---admin
|               DateTimeShortcuts.js
|               RelatedObjectLookups.js
|
+---css
|       admin.css
|       base.css
|       pdf.css
|
\---img
        no_image.png
```

### Django_By_Example_Code/Chapter 13

There are static directories in:

- the courses **application directory**
- the **project directory**.

```
\---educa
    |   db.sqlite3
    |   manage.py
    |
    +---config
    |       nginx.conf
    |       uwsgi.ini
    |
    +---courses
    |   |   admin.py
    |   |   fields.py
    |   |   forms.py

    |   |
    |   +---static
    |   |   \---css
    |   |           base.css
    |   |



    |
    +---educa
    |   |   db.sqlite3
    |   |   urls.py
    |   |   wsgi.py
    |   |   __init__.py
    |   |
    |   \---settings
    |           base.py
    |           local.py
    |           pro.py
    |           __init__.py
    |
    +---ssl
    |       educa.crt
    |       educa.key
    |
    +---static
    |   +---admin
    |   |   +---css
    |   |   |       base.css
    |   |   |       changelists.css
    |   |   |       dashboard.css
    |   |   |       forms.css
    |   |   |       ie.css
    |   |   |       login.css
    |   |   |       rtl.css
    |   |   |       widgets.css
    |   |   |
    |   |   +---img
    |   |   |   |   changelist-bg.gif
    |   |   |   |   changelist-bg_rtl.gif
    |   |   |   |   default-bg-reverse.gif
    |   |   |   |   default-bg.gif
    |   |   |   |   deleted-overlay.gif
    |   |   |   |   icon-no.gif
    |   |   |   |   icon-unknown.gif
    |   |   |   |   icon-yes.gif
```

```
    |   |   |   |   icon_addlink.gif
    |   |   |   |   icon_alert.gif
    |   |   |   |   icon_calendar.gif
    |   |   |   |   icon_changelink.gif
    |   |   |   |   icon_clock.gif
    |   |   |   |   icon_deletelink.gif
    |   |   |   |   icon_error.gif
    |   |   |   |   icon_searchbox.png
    |   |   |   |   icon_success.gif
    |   |   |   |   inline-delete-8bit.png
    |   |   |   |   inline-delete.png
    |   |   |   |   inline-restore-8bit.png
    |   |   |   |   inline-restore.png
    |   |   |   |   inline-splitter-bg.gif
    |   |   |   |   nav-bg-grabber.gif
    |   |   |   |   nav-bg-reverse.gif
    |   |   |   |   nav-bg-selected.gif
    |   |   |   |   nav-bg.gif
    |   |   |   |   selector-icons.gif
    |   |   |   |   selector-search.gif
    |   |   |   |   sorting-icons.gif
    |   |   |   |   tooltag-add.png
    |   |   |   |   tooltag-arrowright.png
    |   |   |   |
    |   |   |   \---gis
    |   |   |           move_vertex_off.png
    |   |   |           move_vertex_on.png
    |   |   |
    |   |   \---js
    |   |       |   actions.js
    |   |       |   actions.min.js
    |   |       |   calendar.js
    |   |       |   collapse.js
    |   |       |   collapse.min.js
    |   |       |   core.js
    |   |       |   inlines.js
    |   |       |   inlines.min.js
    |   |       |   jquery.init.js
    |   |       |   jquery.js
    |   |       |   jquery.min.js
    |   |       |   LICENSE-JQUERY.txt
    |   |       |   prepopulate.js
    |   |       |   prepopulate.min.js
    |   |       |   related-widget-wrapper.js
    |   |       |   SelectBox.js
    |   |       |   SelectFilter2.js
    |   |       |   timeparse.js
    |   |       |   urlify.js
    |   |       |
    |   |       \---admin
    |   |               DateTimeShortcuts.js
    |   |               RelatedObjectLookups.js
    |   |
    |   +---css
    |   |       base.css
    |   |
    |   \---rest_framework
    |       +---css
```

```
    |       |       bootstrap-tweaks.css
    |       |       bootstrap.min.css
    |       |       default.css
    |       |       prettify.css
    |       |
    |       +---fonts
    |       |       glyphicons-halflings-regular.eot
    |       |       glyphicons-halflings-regular.svg
    |       |       glyphicons-halflings-regular.ttf
    |       |       glyphicons-halflings-regular.woff
    |       |
    |       +---img
    |       |       glyphicons-halflings-white.png
    |       |       glyphicons-halflings.png
    |       |       grid.png
    |       |
    |       \---js
    |               bootstrap.min.js
    |               default.js
    |               jquery-1.8.1-min.js
    |               prettify-min.js
    |
```

### Django settings example

**See also:**

- https://docs.djangoproject.com/en/dev/ref/settings/#settings-staticfiles

```
# Build paths inside the project like this: os.path.join(BASE_DIR, ...)
BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))

# https://docs.djangoproject.com/en/dev/howto/static-files/
STATIC_URL = '/static/'
STATIC_ROOT = os.path.join(BASE_DIR, 'static/')
```

### Integration in our Django test project

```
+---static
|   +---css
|   |       easy-autocomplete.min.css
|   |       easy-autocomplete.themes.min.css
|   |
|   \---js
|           jquery.easy-autocomplete.min.js
```

```
|   ajax_selects_singers_db
|   manage.py
|   run_local_dev_server.bat
|
+---projects
|   |   admin.py
|   |   apps.py
|   |   forms.py
|   |   models.py
```

```
|   |   tests.py
|   |   urls.py
|   |   views.py
|   |   __init__.py
|   |
|   +---migrations
|   |   |   0001_initial.py
|   |   |   __init__.py
|   |   |

|   +---templates
|   |   \---projects
|   |       \---project
|   |               update.html
|   |


|
+---projet_ajax
|   |   settings.py
|   |   urls.py
|   |   wsgi.py
|   |   __init__.py
|   |
|
+---singers
|   |   admin.py
|   |   apps.py
|   |   forms.py
|   |   lookups.py
|   |   models.py
|   |   tests.py
|   |   urls.py
|   |   views.py
|   |   __init__.py
|   |
|   +---migrations
|   |   |   0001_initial.py
|   |   |   0002_auto_20161017_1612.py
|   |   |   0003_auto_20161017_1632.py
|   |   |   __init__.py
|   |   |

|   |
|   +---templates
|   |   \---singers
|   |       +---author
|   |       |       update.html
|   |       |
|   |       +---book
|   |       |       update.html
|   |       |
|   |       \---song
|   |               update.html
|   |


|
+---static
|   +---css
```

```
|   |       easy-autocomplete.min.css
|   |       easy-autocomplete.themes.min.css
|   |
|   \---js
|           jquery.easy-autocomplete.min.js
|
\---templates
        base.html
        search_form.html
```

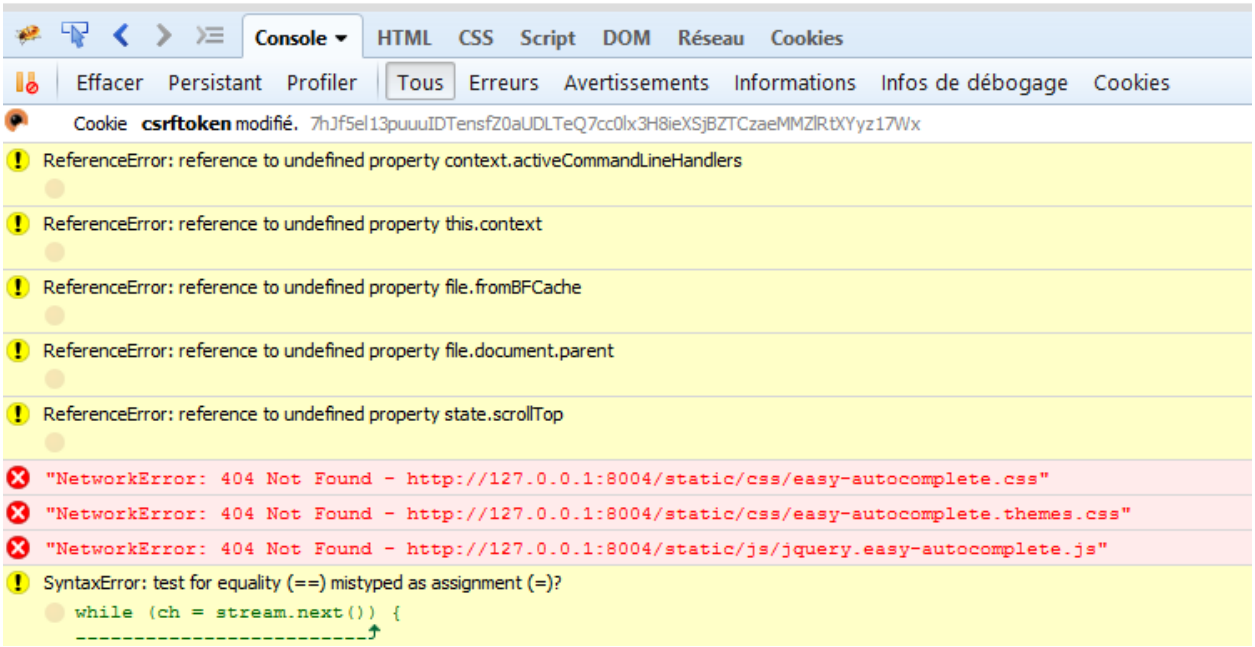
### Create a simple update_easy_simple.html

### Le fichier projects/templates/pojects/projet/update_easy_simple.html

```
{% load static %}
{% load staticfiles %}

<!DOCTYPE html>
<html lang="en">
    <head>
        <title>EasyAutocomplete simple example</title>

        <!-- STYLE -> CSS -->
            <link rel="stylesheet" href="{% static 'css/easy-autocomplete.css' %}"␣
↪type="text/css">
            <link rel="stylesheet" href="{% static 'css/easy-autocomplete.themes.css'
↪%}" type="text/css">

        <!-- end STYLE-->

    </head>
    <body>

        <!-- STRUCTURE -> HTML5 elements -->
            <h1>EasyAutocomplete - easy example</h1>
            <input id="simple" />
        <!-- end STRUCTURE-->

        <!--BEHAVIOR -> Javascript scripts-->
            <!-- Using jQuery with a CDN -->
            <script src="//code.jquery.com/jquery-1.11.2.js"></script>
            <script src="{% static 'js/jquery.easy-autocomplete.js' %}" type="text/
↪javascript"></script>

            <script>

                var options_easy_autocomplete = {
                    data: ["blue", "green", "pink", "red", "yellow"],

                };

                $("#simple").easyAutocomplete(options_easy_autocomplete);

            </script>
        <!-- end BEHAVIOR -->
```

```
    </body>

</html>
```

### First problem : the libraries are not found


```
from os.path import (join,
                     basename,
                     dirname)

# Build paths inside the project like this: os.path.join(BASE_DIR, ...)
```

```
DJANGO_ROOT = BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))
# Site name ('projet_ajax')
SITE_NAME = basename(DJANGO_ROOT)
# Absolute filesystem path to the top-level project folder:
PROJECT_ROOT=dirname(DJANGO_ROOT)

logging.info("DJANGO_ROOT={} SITE_NAME={} PROJECT_ROOT={}".format(
    DJANGO_ROOT,
    SITE_NAME,
    PROJECT_ROOT,
))


# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/dev/howto/static-files/
STATIC_ROOT = join(PROJECT_ROOT, 'staticfiles')
# STATIC_ROOT="C:/projects_id3/P5N001/XLOG5R372_id3_intranet/trunk/django-www/
↪staticfiles"
# Static files (CSS, JavaScript, Images)
# http://www.marinamele.com/taskbuster-django-tutorial/create-home-page-with-tdd-
↪staticfiles-templates-settings
# https://docs.djangoproject.com/en/dev/howto/static-files/
# http://whitenoise.evans.io/en/latest/django.html#runserver-nostatic
STATIC_URL = '/static/'
# https://docs.djangoproject.com/en/dev/howto/static-files/
# les répertoires ou sont sockés les fichiers statiques en developpement
STATICFILES_DIRS = [
    join(DJANGO_ROOT, 'static'),
]
```

### collectstatic

+:

```
python manage.py collectstatic
```

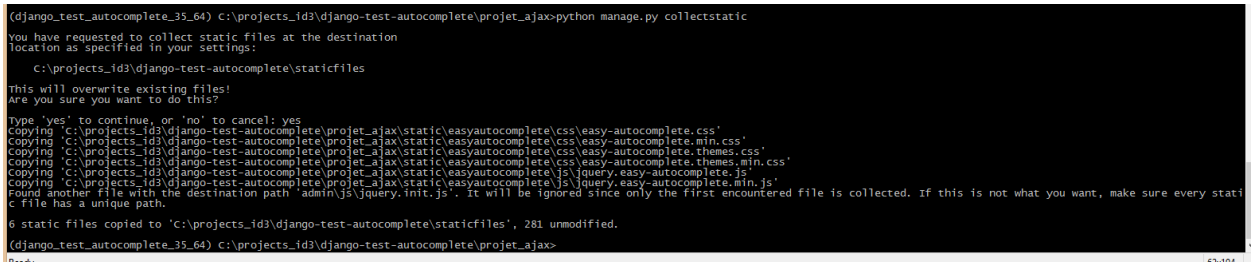
### New tree

```
+---doc_sphinx
|   +---actions
|   |   \---2016

+---projet_ajax
|   +---projects
|   |   +---migrations

\---staticfiles
    +---admin
    |   +---css
    |   +---fonts
    |   +---img
    |   |   \---gis
    |   \---js
    |       +---admin
```

```
    |       \---vendor
    |           +---jquery
    |           \---xregexp
    +---ajax_select
    |   +---css
    |   +---images
    |   \---js
    +---autocomplete_light
    |   \---vendor
    |       \---select2
    |           +---dist
    |           |   +---css
    |           |   \---js
    |           |       \---i18n
    |           +---src
    |           |   \---js
    |           |       \---select2
    |           |           +---compat
    |           |           +---data
    |           |           +---dropdown
    |           |           +---i18n
    |           |           \---selection
    |           +---tests
    |           |   +---a11y
    |           |   +---data
    |           |   +---dropdown
    |           |   +---integration
    |           |   +---options
    |           |   +---selection
    |           |   +---utils
    |           |   \---vendor
    |           \---vendor
    +---css
    +---debug_toolbar
    |   +---css
    |   +---img
    |   \---js
    +---django_extensions
    |   +---css
    |   +---img
    |   \---js
    \---js
```

### Create an easyautocomplete directory

```
python manage.py collectstatic
```



```
|
\---easyautocomplete
    +---css
    |       easy-autocomplete.css
    |       easy-autocomplete.min.css
    |       easy-autocomplete.themes.css
    |       easy-autocomplete.themes.min.css
    |
    \---js
            jquery.easy-autocomplete.js
            jquery.easy-autocomplete.min.js
```

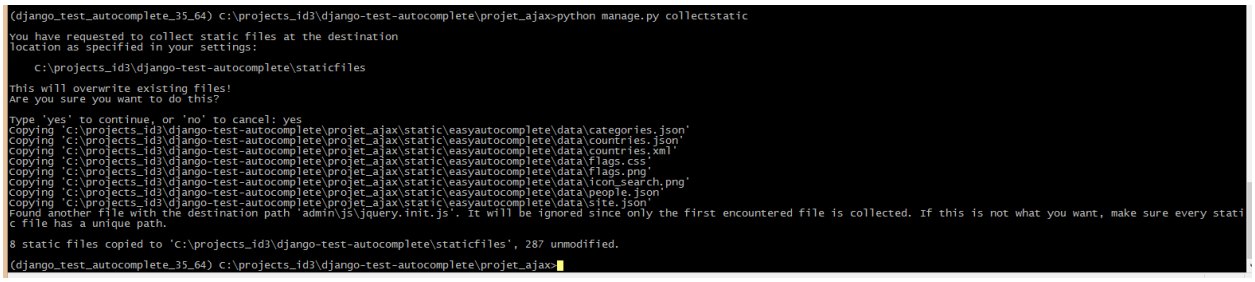
### Update the Django template file

```
{% load static %}
{% load staticfiles %}

<!DOCTYPE html>
<html lang="en">
    <head>
        <title>EasyAutocomplete simple example</title>

        <!-- STYLE -> CSS -->
            <link rel="stylesheet" href="{% static 'easyautocomplete/css/easy-
↪autocomplete.css' %}" type="text/css">
            <link rel="stylesheet" href="{% static 'easyautocomplete/css/easy-
↪autocomplete.themes.css' %}" type="text/css">

        <!-- end STYLE-->

    </head>
    <body>

        <!-- STRUCTURE -> HTML5 elements -->
            <h1>EasyAutocomplete - easy example</h1>
            <input id="simple" />
        <!-- end STRUCTURE-->

        <!--BEHAVIOR -> Javascript scripts-->
            <!-- Using jQuery with a CDN -->
            <script src="//code.jquery.com/jquery-1.11.2.js"></script>
            <script src="{% static 'easyautocomplete/js/jquery.easy-autocomplete.js'
↪%}" type="text/javascript"></script>

            <script>

                var options_easy_autocomplete = {
                    data: ["blue", "green", "pink", "red", "yellow"],

                };

                $("#simple").easyAutocomplete(options_easy_autocomplete);

            </script>
        <!-- end BEHAVIOR -->
```

```
    </body>

</html>
```

### OK it works !

### Add JSON files

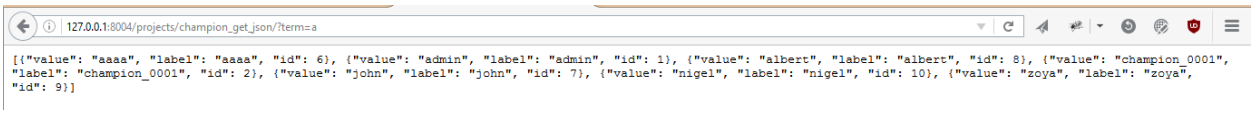



### Adding the AJAX call in the Django template

**See also:**

http://easyautocomplete.com/guide#sec-data-providers

### URL : http://127.0.0.1:8004/projects/champion_get_json/?term=a




### pip install httpie (clihttp)

**See also:**

- https://httpie.org/
- https://twitter.com/jkbrzt
- https://twitter.com/clihttp
- https://github.com/eliangcs/http-prompt

HTTPie—aitch-tee-tee-pie—is an open source CLI HTTP client that will make you smile: a user-friendly curl alternative that provides a simple http command designed for painless debugging and interaction with HTTP servers, RESTful APIs, and web services.

```
pip install httpie
```


```
(django_test_autocomplete_35_64) C:\projects_id3\django-test-autocomplete>pip install httpie
Collecting httpie
  Using cached httpie-0.9.6-py2.py3-none-any.whl
Requirement already satisfied (use --upgrade to upgrade): colorama>=0.2.4; sys_platform == "win32" in c:\project\python_envs\django_test_autocomplete_35_64\lib\
Requirement already satisfied (use --upgrade to upgrade): Pygments>=2.1.3 in c:\project\python_envs\django_test_autocomplete_35_64\lib\site-packages (from httpi
Collecting requests>=2.11.0 (from httpie)
  Using cached requests-2.11.1-py2.py3-none-any.whl
Installing collected packages: requests, httpie
Successfully installed httpie-0.9.6 requests-2.11.1

(django_test_autocomplete_35_64) C:\projects_id3\django-test-autocomplete>http http://127.0.0.1:8004/projects/champion_get_json/?term=a
HTTP/1.0 200 OK
Content-Type: application/json
Date: Fri, 21 Oct 2016 06:41:59 GMT
Server: WSGIServer/0.2 CPython/3.5.2
X-Frame-Options: SAMEORIGIN

[
    {
        "id": 6,
        "label": "aaaa",
        "value": "aaaa"
    },
    {
        "id": 1,
        "label": "admin",
        "value": "admin"
    },
    {
        "id": 8,
        "label": "albert",
        "value": "albert"
    },
    {
        "id": 2,
        "label": "champion_0001",
        "value": "champion_0001"
    },
    {
        "id": 7,
        "label": "john",
        "value": "john"
    },
    {
        "id": 10,
        "label": "nigel",
        "value": "nigel"
    },
    {
        "id": 9,
        "label": "zoya",
        "value": "zoya"
    }
]


(django_test_autocomplete_35_64) C:\projects_id3\django-test-autocomplete>
```

**http http://127.0.0.1:8004/projects/champion_get_json/?term=a**


```
[
    {
        "id": 6,
        "label": "aaaa",
        "value": "aaaa"
    },
    {
        "id": 1,
        "label": "admin",
        "value": "admin"
    },
    {
        "id": 8,
        "label": "albert",
        "value": "albert"
    },
    {
        "id": 2,
        "label": "champion_0001",
        "value": "champion_0001"
    },
    {
        "id": 7,
        "label": "john",
        "value": "john"
    },
    {
        "id": 10,
        "label": "nigel",
        "value": "nigel"
    },
    {
        "id": 9,
        "label": "zoya",
        "value": "zoya"
    }
]
```


**http http://easyautocomplete.com/api/countrySearch.php?phrase=co**


```
[
    {
        "name": "COCOS (KEELING) ISLANDS"
    },
    {
        "name": "COLOMBIA"
    },
    {
        "name": "COMOROS"
    },
    {
        "name": "CONGO"
    },
    {
```

```
(sphinx_35) C:\projects_id3\django-test-autocomplete\doc_sphinx>http http://easyautocomplete.com/api/countrySearch.php?phrase=co
HTTP/1.1 200 OK
Connection: close
Content-Encoding: gzip
Content-Type: application/json
Date: Fri, 21 Oct 2016 06:45:16 GMT
Server: Apache
Transfer-Encoding: chunked
Vary: Accept-Encoding,User-Agent

[
    {
        "name": "COCOS (KEELING) ISLANDS"
    },
    {
        "name": "COLOMBIA"
    },
    {
        "name": "COMOROS"
    },
    {
        "name": "CONGO"
    },
    {
        "name": "CONGO, THE DEMOCRATIC REPUBLIC OF THE"
    },
    {
        "name": "COOK ISLANDS"
    },
    {
        "name": "COSTA RICA"
    },
    {
        "name": "COTE D IVOIRE"
    },
    {
        "name": "MEXICO"
    },
    {
        "name": "MONACO"
    },
    {
        "name": "MOROCCO"
    },
    {
        "name": "PUERTO RICO"
    },
    {
        "name": "TURKS AND CAICOS ISLANDS"
    }
]
```


```
        "name": "CONGO, THE DEMOCRATIC REPUBLIC OF THE"
    },
    {
        "name": "COOK ISLANDS"
    },
    {
        "name": "COSTA RICA"
    },
    {
        "name": "COTE D IVOIRE"
    },
    {
        "name": "MEXICO"
    },
    {
        "name": "MONACO"
    },
    {
        "name": "MOROCCO"
    },
    {
        "name": "PUERTO RICO"
    },
    {
        "name": "TURKS AND CAICOS ISLANDS"
    }
]
```

### Avec countrySearch.php

```
<input id="provider-remote" />
```

```
var options = {
    url: function(phrase) {
        return "api/countrySearch.php?phrase=" + phrase + "&format=json";
    },

    getValue: "value"
};

$("#provider-remote").easyAutocomplete(options);
```

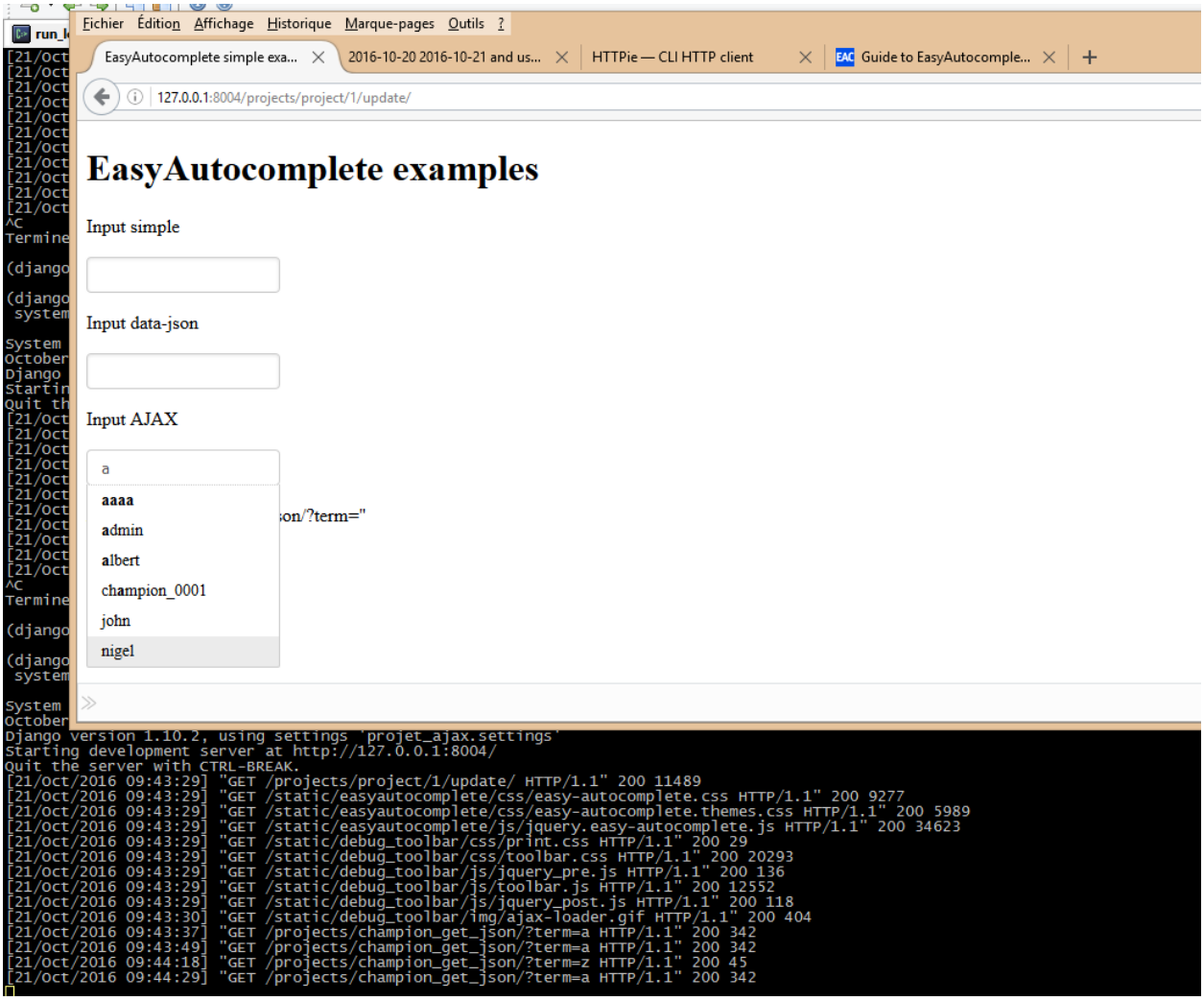
### Avec Python/Django OK the first step is DONE

**See also:**

- http://127.0.0.1:8004/projects/project/1/update/
- http://127.0.0.1:8004/projects/champion_get_json/?term=a

### The Template

```
{# ------------------------------------------------------------------------------------
↪---------#}
{# simple AJAX call #}
{# http://easyautocomplete.com/guide#sec-data-providers #}
{# ------------------------------------------------------------------------------------
↪---------#}
{% comment %}
(django_test_autocomplete_35_64) C:\projects_id3\django-test-autocomplete>http http://
↪127.0.0.1:8004/projects/champion_get_json/?term=z
HTTP/1.0 200 OK
Content-Type: application/json
Date: Fri, 21 Oct 2016 07:51:46 GMT
Server: WSGIServer/0.2 CPython/3.5.2
X-Frame-Options: SAMEORIGIN

[
    {
        "id": 9,
        "label": "zoya",
        "value": "zoya"
    }
]
{% endcomment %}
var options_easy_autocomplete_ajax = {
    {# we have to build this URL: http://127.0.0.1:8004/projects/champion_get_json/?
↪term=a #}
    url: function(term) {
        return "{% url 'projects:champion_get_json' %}" + "?term=" + term;
    },
    getValue: "value",
```


Fichier Édition Affichage Historique Marque-pages Outils ?
EasyAutocomplete simple exa...
2016-10-20 2016-10-21 and us...
HTTPie — CLI HTTP client
Guide to EasyAutocomple...
127.0.0.1:8004/projects/project/1/update/
EasyAutocomplete examples
Input simple
Input data-json
Input AJAX
a
aaaa
admin
albert
champion_0001
john
nigel
Django version 1.10.2, using settings 'projet_ajax.settings'
Starting development server at http://127.0.0.1:8004/
Quit the server with CTRL-BREAK.
[21/Oct/2016 09:43:29] "GET /projects/project/1/update/ HTTP/1.1" 200 11489
[21/Oct/2016 09:43:29] "GET /static/easyautocomplete/css/easy-autocomplete.css HTTP/1.1" 200 9277
[21/Oct/2016 09:43:29] "GET /static/easyautocomplete/css/easy-autocomplete.themes.css HTTP/1.1" 200 5989
[21/Oct/2016 09:43:29] "GET /static/easyautocomplete/js/jquery.easy-autocomplete.js HTTP/1.1" 200 34623
[21/Oct/2016 09:43:29] "GET /static/debug_toolbar/css/print.css HTTP/1.1" 200 29
[21/Oct/2016 09:43:29] "GET /static/debug_toolbar/css/toolbar.css HTTP/1.1" 200 20293
[21/Oct/2016 09:43:29] "GET /static/debug_toolbar/js/jquery_pre.js HTTP/1.1" 200 136
[21/Oct/2016 09:43:29] "GET /static/debug_toolbar/js/toolbar.js HTTP/1.1" 200 12552
[21/Oct/2016 09:43:29] "GET /static/debug_toolbar/js/jquery_post.js HTTP/1.1" 200 118
[21/Oct/2016 09:43:30] "GET /static/debug_toolbar/img/ajax-loader.gif HTTP/1.1" 200 404
[21/Oct/2016 09:43:37] "GET /projects/champion_get_json/?term=a HTTP/1.1" 200 342
[21/Oct/2016 09:43:49] "GET /projects/champion_get_json/?term=a HTTP/1.1" 200 342
[21/Oct/2016 09:44:18] "GET /projects/champion_get_json/?term=z HTTP/1.1" 200 45
[21/Oct/2016 09:44:29] "GET /projects/champion_get_json/?term=a HTTP/1.1" 200 342

```
};
$("#data-ajax").easyAutocomplete(options_easy_autocomplete_ajax);
```

### The View ChampionAutoCompleteView

```
class ChampionAutoCompleteView(FormView):
    """
    Documentation
    =============

    - https://ccbv.co.uk/projects/Django/1.9/django.views.generic.edit/FormView/

    """
    def get(self, request, *args, **kwargs):
        """term is sent by the jquery-ui autocomplete widget.

        The filter is on the username and the user email.

        """
        term = request.GET.get("term")
        if term:
            users = User.objects.filter(Q(username__icontains=term)
                                        | Q(email__icontains=term)).order_by('username
↪')
        else:
            users = User.objects.all()[:50]

        results = []
        for user in users:
            user_json = {}
            user_json['id'] = user.id
            user_json['label'] = user.username
            user_json['value'] = user.username
            results.append(user_json)

        data = json.dumps(results)
        mimetype = 'application/json'
        return HttpResponse(data, mimetype)
```

### The projects urls.py

```
# calls by jquery-ui autocomplete (AJAX calls)
# http://127.0.0.1:8004/projects/champion_get_json/?term=a
url(r'^champion_get_json/$',
    ChampionAutoCompleteView.as_view(),
    name='champion_get_json'),
```

### This is the happy end of the first step

## 2016-10-19 improve the Django templates + unobstrusive Javascript (step1)

**Contents**



### The Javascript code must be in the the body element

**See also:**

- https://www.amazon.fr/gp/product/1617292079/ref=oh_aui_detailpage_o00_s00?ie=UTF8&psc=1

#### 1.2.1 Separating behavior from structure

For all the same reasons that it's desirable to segregate style from structure within an HTML document, it's just as beneficial (if not more so) to separate the *behavior* from the structure. Ideally, an HTML page should be structured as shown in figure 1.2, with structure, style, and behavior each partitioned nicely in its own niche.

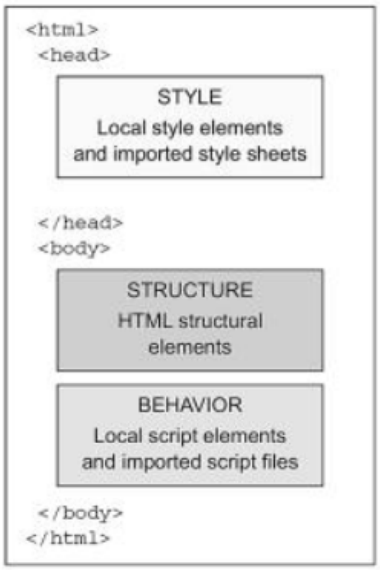



**Figure 1.2 With structure, style, and behavior each neatly tucked away within a page, readability and maintainability are maximized.**

This strategy, known as *unobtrusive JavaScript*, is now embraced by every major JavaScript library, helping page authors achieve this useful separation on their pages. As the library that popularized this movement, jQuery's core is well optimized for producing unobtrusive JavaScript easily. Unobtrusive JavaScript considers *any* JavaScript expressions or statements placed within or among HTML tags in the `<body>` of HTML pages, either as attributes of HTML elements (such as `onclick`) or in script blocks placed anywhere other than the very end of the body of the page, to be incorrect.

"But how can I instrument the button without the `onclick` attribute?" you might ask. Consider the following change to the button element:

```
<button id="test-button">Click Me</button>
```

Much simpler! But now, you'll note, the button doesn't do anything. You can click it all day long, and no behavior will result. Let's fix that.

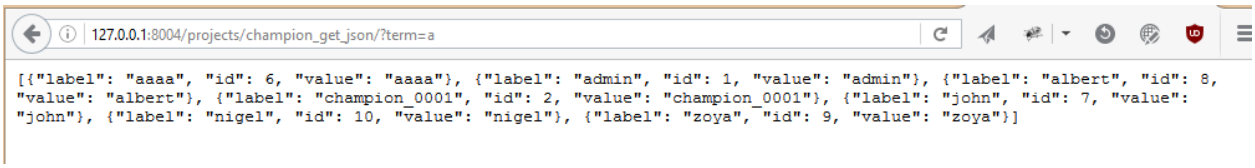
### Modify names

- champion_auto_complete -> champion_get_json

### Test champion_get_json : http://127.0.0.1:8004/projects/champion_get_json/?term=a

**See also:**

http://127.0.0.1:8004/projects/champion_get_json/?term=a




### Rendering fields manually

**See also:**

- https://docs.djangoproject.com/en/dev/topics/forms/#rendering-fields-manually
- https://docs.djangoproject.com/en/dev/topics/forms/#looping-over-hidden-and-visible-fields

### Adding django-debug-toolbar

**See also:**

- https://django-debug-toolbar.readthedocs.io/en/stable/index.html
- https://django-debug-toolbar.readthedocs.io/en/stable/installation.html

```
pip install django-debug-toolbar
```

```
    # https://django-extensions.readthedocs.org/en/latest
    'django_extensions',

    # http://django-ajax-selects.readthedocs.io/en/latest/index.html
    'ajax_select',
    # http://django-crispy-forms.readthedocs.io/en/latest/install.html#installing-
↪django-crispy-forms
    'crispy_forms',
    # https://github.com/jazzband/django-debug-toolbar
    'debug_toolbar',

    # # https://docs.djangoproject.com/en/dev/ref/applications/#django.apps.AppConfig
    'singers.apps.SingersConfig',
    # http://guiqinqian.blogspot.fr/2012/01/using-jquery-auto-complete-in-django.html
    'projects.apps.ProjectsConfig'
]
```

```
if settings.DEBUG:
    import debug_toolbar
    urlpatterns += [
```

```
        url(r'^__debug__/', include(debug_toolbar.urls)),
    ]
```

```
MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',

    'debug_toolbar.middleware.DebugToolbarMiddleware'
]
```

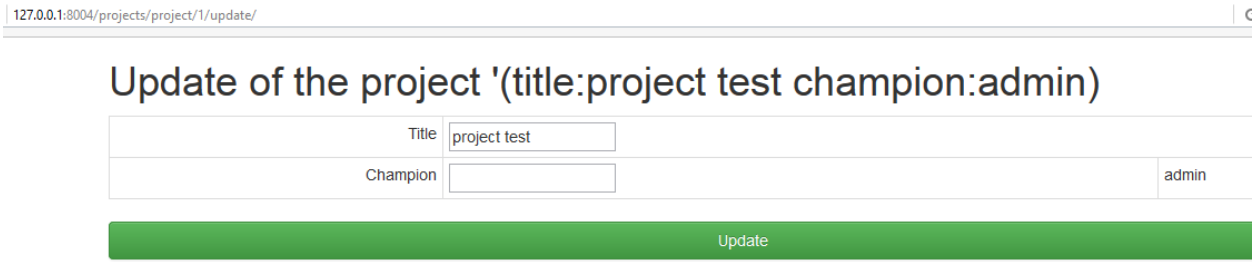
**Warning:** do not forget to set the new INTERNAL_IPS variable

INTERNAL_IPS = '127.0.0.1'

### First step for a better look

**See also:**

http://127.0.0.1:8004/projects/project/1/update/




## 2016-10-18 improve the jquery-ui autocomplete look and feel

**See also:**

- http://api.jqueryui.com/autocomplete/
- https://jqueryui.com/autocomplete/
- https://jqueryui.com/autocomplete/#combobox

**Contents**

        * *Remote option*
    - *Try https://jqueryui.com/autocomplete/#combobox*
    - *Try https://jqueryui.com/autocomplete/#maxheight*
    - *Try https://jqueryui.com/autocomplete/#categories*
    - *Try https://jqueryui.com/autocomplete/#remote-jsonp*

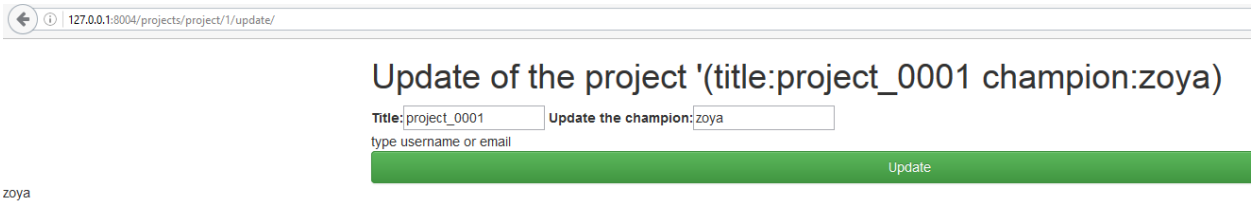
### Last look

**See also:**

- http://127.0.0.1:8004/projects/project/1/update/

```
(django_test_autocomplete_35_64) C:\projects_id3\django-test-autocomplete\projet_ajax>
↪python manage.py runserver 127.0.0.1:8004 Performing system checks...
```

```
System check identified no issues (0 silenced).
October 19, 2016 - 09:08:19
Django version 1.10.2, using settings 'projet_ajax.settings'
Starting development server at http://127.0.0.1:8004/
Quit the server with CTRL-BREAK.
```




```
{% extends "base.html" %}
{% load static %}
{% load staticfiles %}


{# STYLE CSS #}
{% block stylesheet %}
    {{ block.super }}
    {# https://www.bootstrapcdn.com/ #}
    <link rel="stylesheet" href="http://code.jquery.com/ui/1.12.1/themes/base/jquery-
↪ui.csss">

    <style>
    {# https://jqueryui.com/autocomplete/#maxheight #}
    .ui-autocomplete {
        max-height: 100px;
        overflow-y: auto;
        /* prevent horizontal scrollbar */
        overflow-x: hidden;
    }
    * html .ui-autocomplete {
        height: 100px;
```

```
    }
    </style>

{% endblock  %}


{% block content %}

    <h1>Update of the project '(title:{{ project.title }} champion:{{ project.champion }}) </h1>
    <p></p>
    <p></p>
    {# https://docs.djangoproject.com/en/dev/topics/forms/ #}
    <form id="id_form_project_update" action="{% url 'projects:project_update' project.id %}" method="post">
        {% csrf_token %}
        <div class="forms">
            {{ form.id }}
            {{ form.non_field_errors }}
            {# Include the hidden fields #}
            {% for hidden in form.hidden_fields %}
                {{ hidden }}
            {% endfor %}
            <table id="id_table" class="table table-hover table-bordered table-condensed">
                <tbody>
                        <tr>
                            <td class="text-right">Title:</td>
                            <td>{{ form.title }}</td>
                        </tr>
                        <tr>
                            <td class="text-right">Champion:</td>
                            <td> <span id="id_champion_display">{{ project.champion }}</span> {{ form.champion_term }}  </td>
                        </tr>
                </tbody>
            </table>
        </div>
        <input type="submit" name="btn_update"  value="Update" class="btn btn-success btn-block" />
    </form>


    {% if messages %}
        <ul class="messages">
            {% comment %} https://getbootstrap.com/components/#alerts {% endcomment %}
            {% for message in messages %}
            <!--li{% if message.tags %} class="{{ message.tags }}"{% endif %} -->
                {% if message.level == DEFAULT_MESSAGE_LEVELS.ERROR %}
                    <div class="alert alert-danger" role="alert">{{ message }}</div>
                {% elif message.level == DEFAULT_MESSAGE_LEVELS.SUCCESS %}
                    <div class="alert alert-success" role="alert">{{ message }}</div>
                {% else %}
                    <div class="alert alert-info" role="alert">{{ message }}</div>
                {% endif %}
            <!-- /li -->
            {% endfor %}
        </ul>
```

```
    {% endif %}


{% endblock content %}


{# BEHAVIOR Javascript #}
{% block javascript %}
    {{ block.super }}

    {# http://code.jquery.com/ui/ #}
    <script src="http://code.jquery.com/ui/1.12.1/jquery-ui.min.js"></script>

{% endblock javascript %}


{# DOM ready #}
{% block domready %}

    $( "#id_champion_term" ).autocomplete({
          {# http://api.jqueryui.com/autocomplete/#option-source #}
          source: "{% url 'projects:champion_get_json' %}",
          {# http://api.jqueryui.com/autocomplete/#option-autoFocus #}
          autoFocus:true,
          {# http://api.jqueryui.com/autocomplete/#option-minLength #}
          minLength:1,
          {# http://api.jqueryui.com/autocomplete/#event-select #}
          select:function(event,ui) {
              $("#id_champion").val(ui.item.id);
          }
    });

{% endblock %}
```

```
#!/usr/bin/python
# -*- coding: utf8 -*-
"""The project's forms.

"""

from django import forms

from .models import Project

class ProjectChampionForm(forms.ModelForm):
    """The champion project form"""
    champions_choice_list = forms.CharField(max_length=100,
                                            help_text='type username or email')

    class Meta:
        model = Project
        fields = ('title',
                  'champions_choice_list', 'champion',)

    def __init__(self, *args, **kwargs):
        super(ProjectChampionForm, self).__init__(*args, **kwargs)
        self.fields['champions_choice_list'].label = "Update the champion"
        self.fields['champion'].widget = forms.HiddenInput()
```

### jquery-ui autocomplete options already used

### Remote option

- https://jqueryui.com/autocomplete/#remote

The Autocomplete widgets provides suggestions while you type into the field. Here the suggestions are bird names, displayed when at least two characters are entered into the field.

The datasource is a server-side script which returns JSON data, specified via a simple URL for the source-option. In addition, the minLength-option is set to 2 to avoid queries that would return too many results and the select-event is used to display some feedback.

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>jQuery UI Autocomplete - Remote datasource</title>
  <link rel="stylesheet" href="//code.jquery.com/ui/1.12.1/themes/base/jquery-ui.css">
  <link rel="stylesheet" href="/resources/demos/style.css">
  <style>
  .ui-autocomplete-loading {
    background: white url("images/ui-anim_basic_16x16.gif") right center no-repeat;
  }
  </style>

  <script src="https://code.jquery.com/jquery-1.12.4.js"></script>
  <script src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"></script>
  <script>
  $( function() {
    function log( message ) {
      $( "<div>" ).text( message ).prependTo( "#log" );
      $( "#log" ).scrollTop( 0 );
    }

    $( "#birds" ).autocomplete({
      source: "search.php",
      minLength: 2,
      select: function( event, ui ) {
        log( "Selected: " + ui.item.value + " aka " + ui.item.id );
      }
    });
  } );
  </script>
</head>
<body>

<div class="ui-widget">
  <label for="birds">Birds: </label>
  <input id="birds">
</div>

<div class="ui-widget" style="margin-top:2em; font-family:Arial">
  Result:
```

```
  <div id="log" style="height: 200px; width: 300px; overflow: auto;" class="ui-widget-
↪content"></div>
</div>


</body>
</html>
```

### Try https://jqueryui.com/autocomplete/#combobox

**See also:**

- http://www.learningjquery.com/2010/06/a-jquery-ui-combobox-under-the-hood

### Try https://jqueryui.com/autocomplete/#maxheight

**See also:**

- https://jqueryui.com/autocomplete/#maxheight

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>jQuery UI Autocomplete - Scrollable results</title>
  <link rel="stylesheet" href="//code.jquery.com/ui/1.12.1/themes/base/jquery-ui.css">
  <link rel="stylesheet" href="/resources/demos/style.css">
  <style>
  .ui-autocomplete {
    max-height: 100px;
    overflow-y: auto;
    /* prevent horizontal scrollbar */
    overflow-x: hidden;
  }
  /* IE 6 doesn't support max-height
   * we use height instead, but this forces the menu to always be this tall
   */
  * html .ui-autocomplete {
    height: 100px;
  }
  </style>
  <script src="https://code.jquery.com/jquery-1.12.4.js"></script>
  <script src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"></script>
  <script>
  $( function() {
    var availableTags = [
      "ActionScript",
      "AppleScript",
      "Asp",
      "BASIC",
      "C",
      "C++",
      "Clojure",
      "COBOL",
      "ColdFusion",
```

```
      "Erlang",
      "Fortran",
      "Groovy",
      "Haskell",
      "Java",
      "JavaScript",
      "Lisp",
      "Perl",
      "PHP",
      "Python",
      "Ruby",
      "Scala",
      "Scheme"
    ];
    $( "#tags" ).autocomplete({
      source: availableTags
    });
  } );
  </script>
</head>
<body>

<div class="ui-widget">
  <label for="tags">Tags: </label>
  <input id="tags">
</div>


</body>
</html>
```

### Try https://jqueryui.com/autocomplete/#categories

**See also:**

- https://jqueryui.com/autocomplete/#categories

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>jQuery UI Autocomplete - Categories</title>
  <link rel="stylesheet" href="//code.jquery.com/ui/1.12.1/themes/base/jquery-ui.css">
  <link rel="stylesheet" href="/resources/demos/style.css">
  <style>
  .ui-autocomplete-category {
    font-weight: bold;
    padding: .2em .4em;
    margin: .8em 0 .2em;
    line-height: 1.5;
  }
  </style>
  <script src="https://code.jquery.com/jquery-1.12.4.js"></script>
  <script src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"></script>
  <script>
  $( function() {
```

```
    $.widget( "custom.catcomplete", $.ui.autocomplete, {
      _create: function() {
        this._super();
        this.widget().menu( "option", "items", "> :not(.ui-autocomplete-category)" );
      },
      _renderMenu: function( ul, items ) {
        var that = this,
          currentCategory = "";
        $.each( items, function( index, item ) {
          var li;
          if ( item.category != currentCategory ) {
            ul.append( "<li class='ui-autocomplete-category'>" + item.category + "</
↪li>" );
            currentCategory = item.category;
          }
          li = that._renderItemData( ul, item );
          if ( item.category ) {
            li.attr( "aria-label", item.category + " : " + item.label );
          }
        });
      }
    });
    var data = [
      { label: "anders", category: "" },
      { label: "andreas", category: "" },
      { label: "antal", category: "" },
      { label: "annhhx10", category: "Products" },
      { label: "annk K12", category: "Products" },
      { label: "annttop C13", category: "Products" },
      { label: "anders andersson", category: "People" },
      { label: "andreas andersson", category: "People" },
      { label: "andreas johnson", category: "People" }
    ];

    $( "#search" ).catcomplete({
      delay: 0,
      source: data
    });
  } );
  </script>
</head>
<body>

<label for="search">Search: </label>
<input id="search">


</body>
</html>
```

### Try https://jqueryui.com/autocomplete/#remote-jsonp

**See also:**

- https://jqueryui.com/autocomplete/#remote-jsonp

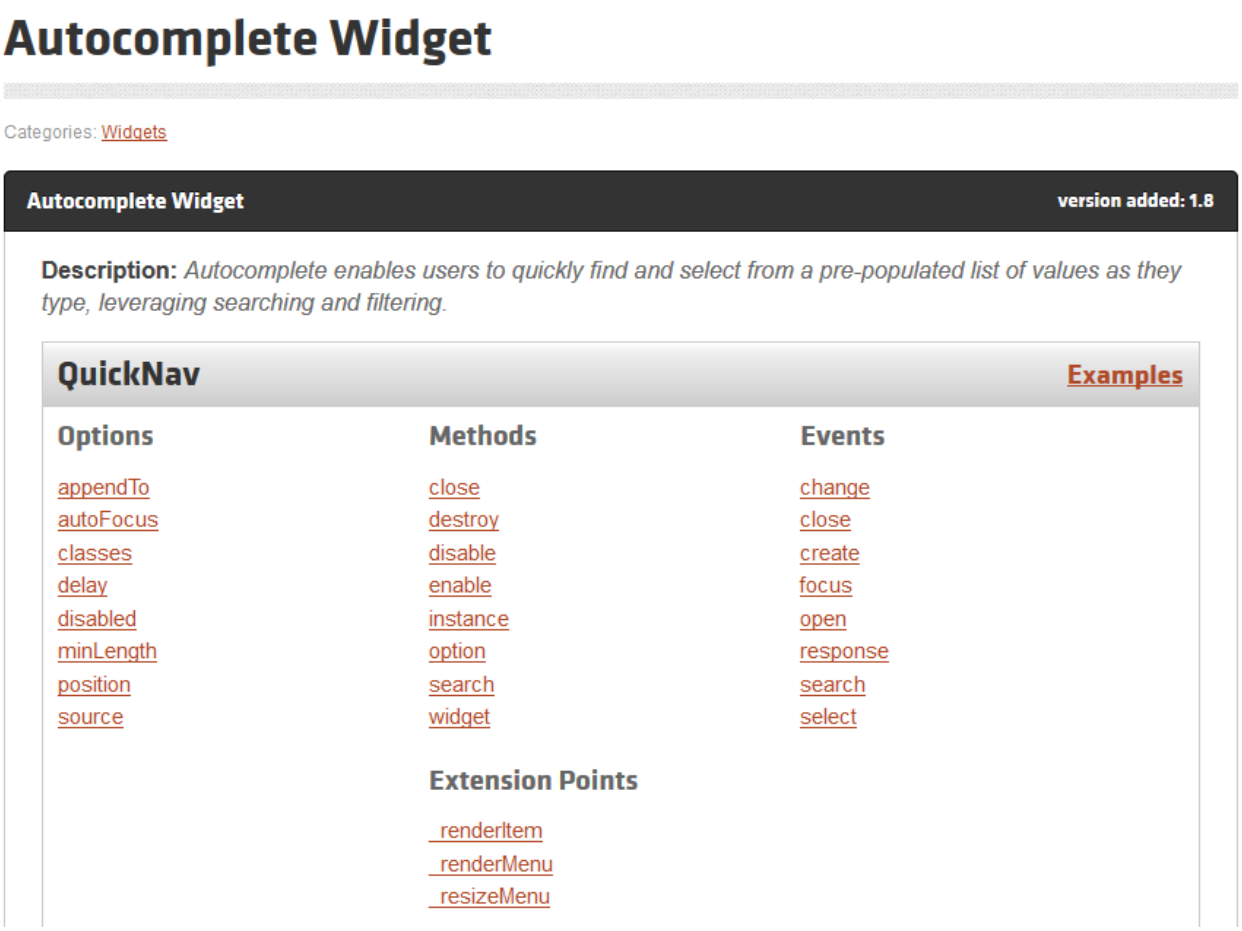
```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>jQuery UI Autocomplete - Remote JSONP datasource</title>
  <link rel="stylesheet" href="//code.jquery.com/ui/1.12.1/themes/base/jquery-ui.css">
  <link rel="stylesheet" href="/resources/demos/style.css">
  <style>
  .ui-autocomplete-loading {
    background: white url("images/ui-anim_basic_16x16.gif") right center no-repeat;
  }
  </style>
  <script src="https://code.jquery.com/jquery-1.12.4.js"></script>
  <script src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"></script>
  <script>
  $( function() {
    function log( message ) {
      $( "<div>" ).text( message ).prependTo( "#log" );
      $( "#log" ).scrollTop( 0 );
    }

    $( "#birds" ).autocomplete({
      source: function( request, response ) {
        $.ajax( {
          url: "search.php",
          dataType: "jsonp",
          data: {
            term: request.term
          },
          success: function( data ) {
            response( data );
          }
        } );
      },
      minLength: 2,
      select: function( event, ui ) {
        log( "Selected: " + ui.item.value + " aka " + ui.item.id );
      }
    } );
  } );
  </script>
</head>
<body>

<div class="ui-widget">
  <label for="birds">Birds: </label>
  <input id="birds">
</div>

<div class="ui-widget" style="margin-top:2em; font-family:Arial">
  Result:
  <div id="log" style="height: 200px; width: 300px; overflow: auto;" class="ui-widget-
↪content"></div>
</div>


</body>
```

```
</html>
```

## 2016-10-18 try a more simple approach with jquery-ui autocomplete widget

**See also:**

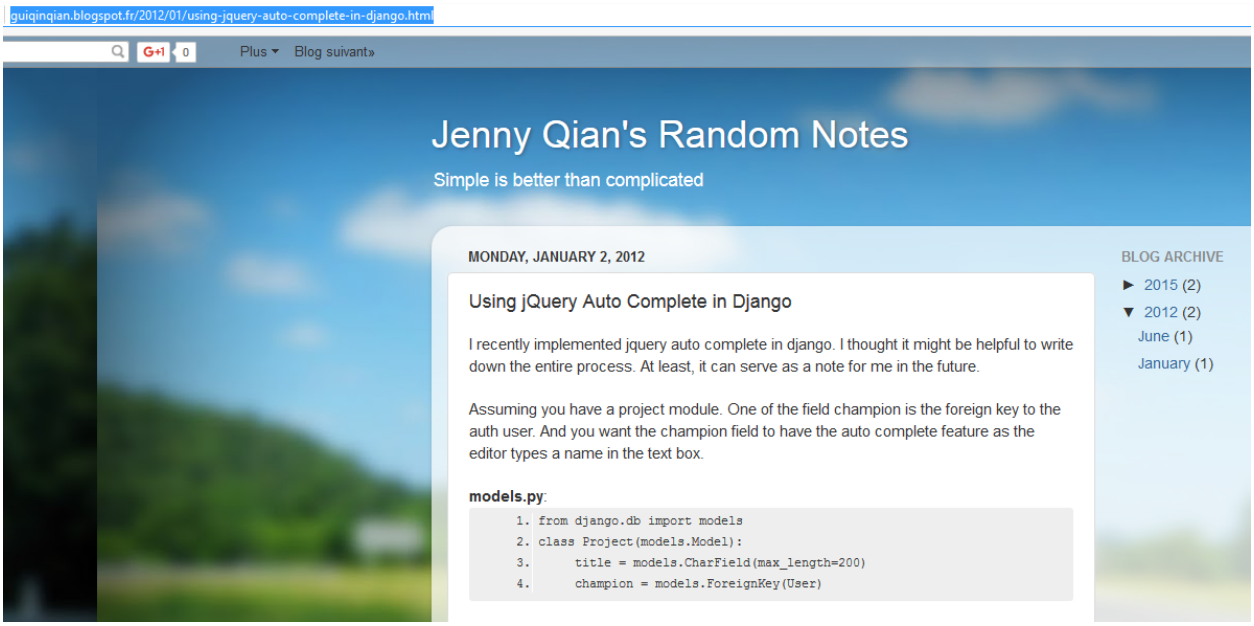
- https://jqueryui.com/autocomplete/
- http://guiqinqian.blogspot.fr/2012/01/using-jquery-auto-complete-in-django.html
- https://github.com/jquery/jquery-ui




**Contents**

- *makemigrations*
- *sqlmigrate*
- *migrate*
- *Create the projects/forms.py file and add 2 forms*
- *Create the projects/urls.py file and add the champion auto complete URL*
- *Add the projects.urls file to the root urls.py*
- *Add the ProjectUpdateView*
- *Create the templates/projects/project/update.html django template*
- *Add the jquery-ui javascript code*
- *Add the admin interface*
- *Create some projects with the admin interface*
- *Reading Project record from the commande line*
- *Add some users with the admin interface*
- *Add some users with the command line interface*
- *Try the URL : http://127.0.0.1:8000/projects/project/1/update*

### Installation




Other sources:

- http://blog.appliedinformaticsinc.com/autocomplete-input-field-in-django-template-with-jquery-ui/
- https://openclassrooms.com/courses/decouvrez-la-puissance-de-jquery-ui/l-autocompletion-1

### Create a `projects` application

```
python manage.py startapp projects
```

```
|   ajax_selects_singers_db
|   manage.py
|   tree.txt
|
+---projects
|   |   admin.py
|   |   apps.py
|   |   models.py
|   |   tests.py
|   |   views.py
|   |   __init__.py
|   |
|   \---migrations
|           __init__.py
```

### Add the Django model `Project`

```
#!/usr/bin/python
# -*- coding: utf8 -*-
"""The project's models.

"""
from __future__ import unicode_literals
from django.utils.encoding import python_2_unicode_compatible

from django.db import models
from django.core.urlresolvers import reverse

# https://docs.djangoproject.com/en/dev/ref/contrib/auth/#user-model
from django.contrib.auth.models import User



@python_2_unicode_compatible
class Project(models.Model):
    """A project with a title and a champion which is the foreign key to the
       auth user.

    Documentation
    =============

    - http://guiqinqian.blogspot.fr/2012/01/using-jquery-auto-complete-in-django.html

    """
    title = models.CharField(max_length=200)
    champion = models.ForeignKey(User)

    def __str__(self):
        return "{} {}".format(self.title, self.champion)

    def get_absolute_url(self):
        """
```

```
        https://docs.djangoproject.com/en/dev/ref/class-based-views/generic-editing/
        """
        return reverse('projects:project_update',
                       kwargs={'pk': self.pk})
```

### Add the `project` in the INSTALLED_APPS

```
    # # https://docs.djangoproject.com/en/dev/ref/applications/#django.apps.AppConfig
    'singers.apps.SingersConfig',
    # http://guiqinqian.blogspot.fr/2012/01/using-jquery-auto-complete-in-django.html
    'projects.apps.ProjectsConfig'
]
```

```
"""
Django settings for projet_ajax project.

Generated by 'django-admin startproject' using Django 1.10.2.

For more information on this file, see
https://docs.djangoproject.com/en/1.10/topics/settings/

For the full list of settings and their values, see
https://docs.djangoproject.com/en/1.10/ref/settings/
"""

import os
import logging

from os.path import (join,
                     basename,
                     dirname)

# Build paths inside the project like this: os.path.join(BASE_DIR, ...)
DJANGO_ROOT = BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))
# Site name ('projet_ajax')
SITE_NAME = basename(DJANGO_ROOT)
# Absolute filesystem path to the top-level project folder:
PROJECT_ROOT=dirname(DJANGO_ROOT)

logging.info("DJANGO_ROOT={} SITE_NAME={} PROJECT_ROOT={}".format(
    DJANGO_ROOT,
    SITE_NAME,
    PROJECT_ROOT,
))

# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/1.10/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = 'q#e(dtsmq9^!(%9*un6tuq5v)2!@00ftgk$!@h1*(e0iwo@=cc'

# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True

# for the debug toolbar
INTERNAL_IPS = '127.0.0.1'
```

```
44
45  DEBUG_TOOLBAR_PANELS = [
46      'debug_toolbar.panels.versions.VersionsPanel',
47      'debug_toolbar.panels.timer.TimerPanel',
48      'debug_toolbar.panels.settings.SettingsPanel',
49      'debug_toolbar.panels.headers.HeadersPanel',
50      'debug_toolbar.panels.request.RequestPanel',
51      'debug_toolbar.panels.sql.SQLPanel',
52      'debug_toolbar.panels.staticfiles.StaticFilesPanel',
53      'debug_toolbar.panels.templates.TemplatesPanel',
54      'debug_toolbar.panels.cache.CachePanel',
55      'debug_toolbar.panels.signals.SignalsPanel',
56      'debug_toolbar.panels.logging.LoggingPanel',
57      'debug_toolbar.panels.redirects.RedirectsPanel',
58  ]
59
60
61  ALLOWED_HOSTS = []
62
63
64  # Applications definition
65  INSTALLED_APPS = [
66      'dal',
67      'dal_select2',
68
69      'django.contrib.admin',
70      'django.contrib.auth',
71      'django.contrib.contenttypes',
72      'django.contrib.sessions',
73      'django.contrib.messages',
74      'django.contrib.staticfiles',
75
76      # https://django-extensions.readthedocs.org/en/latest
77      'django_extensions',
78
79      # http://django-ajax-selects.readthedocs.io/en/latest/index.html
80      'ajax_select',
81      # http://django-crispy-forms.readthedocs.io/en/latest/install.html#installing-
    ↪django-crispy-forms
82      'crispy_forms',
83      # https://github.com/jazzband/django-debug-toolbar
84      # 'debug_toolbar',
85
86      # # https://docs.djangoproject.com/en/dev/ref/applications/#django.apps.AppConfig
87      'singers.apps.SingersConfig',
88      # http://guiqinqian.blogspot.fr/2012/01/using-jquery-auto-complete-in-django.html
89      'projects.apps.ProjectsConfig'
90  ]
91
92  MIDDLEWARE = [
93      'django.middleware.security.SecurityMiddleware',
94      'django.contrib.sessions.middleware.SessionMiddleware',
95      'django.middleware.common.CommonMiddleware',
96      'django.middleware.csrf.CsrfViewMiddleware',
97      'django.contrib.auth.middleware.AuthenticationMiddleware',
98      'django.contrib.messages.middleware.MessageMiddleware',
99      'django.middleware.clickjacking.XFrameOptionsMiddleware',
100
```

```
    # 'debug_toolbar.middleware.DebugToolbarMiddleware'
]


ROOT_URLCONF = 'projet_ajax.urls'

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [join(BASE_DIR, 'templates')],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]

WSGI_APPLICATION = 'projet_ajax.wsgi.application'


# Database
# https://docs.djangoproject.com/en/1.10/ref/settings/#databases
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': os.path.join(BASE_DIR,'ajax_selects_singers_db')
    }
}

# Password validation
# https://docs.djangoproject.com/en/1.10/ref/settings/#auth-password-validators

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME': 'django.contrib.auth.password_validation.
↪UserAttributeSimilarityValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
    },
]


# Internationalization
# https://docs.djangoproject.com/en/1.10/topics/i18n/

LANGUAGE_CODE = 'en-us'
```

```

TIME_ZONE = 'UTC'

USE_I18N = True

USE_L10N = True

USE_TZ = True


# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/dev/howto/static-files/
STATIC_ROOT = join(PROJECT_ROOT, 'staticfiles')
# STATIC_ROOT="C:/projects_id3/P5N001/XLOG5R372_id3_intranet/trunk/django-www/staticfiles"
# Static files (CSS, JavaScript, Images)
# http://www.marinamele.com/taskbuster-django-tutorial/create-home-page-with-tdd-staticfiles-templates-settings
# https://docs.djangoproject.com/en/dev/howto/static-files/
# http://whitenoise.evans.io/en/latest/django.html#runserver-nostatic
STATIC_URL = '/static/'
# https://docs.djangoproject.com/en/dev/howto/static-files/
# les répertoires ou sont sockés les fichiers statiques en developpement
STATICFILES_DIRS = [
    join(DJANGO_ROOT, 'static'),
]

AJAX_LOOKUP_CHANNELS = {
    # simplest way, automatically construct a search channel by passing a dict
    'label': {'model': 'singers.label'
            , 'search_field': 'name'},

    # Custom channels are specified with a tuple
    # channel: ( module.where_lookup_is, ClassNameOfLookup )
    # 'person': ('singers.lookups', 'PersonLookup'),
    # 'group': ('singers.lookups', 'GroupLookup'),
    # 'song': ('singers.lookups', 'SongLookup'),
}

# voir https://github.com/jordij/bctt.nz/blob/master/ttwellington/settings/base.py
MEDIA_ROOT = join(BASE_DIR, 'media')
MEDIA_URL = '/media/'

```

### makemigrations

```
python manage.py makemigrations projects
```

```
Migrations for 'projects':
  projects\migrations\0001_initial.py:
    - Create model Project
```

```
# -*- coding: utf-8 -*-
# Generated by Django 1.10.2 on 2016-10-18 10:45
from __future__ import unicode_literals
```

```

from django.conf import settings
from django.db import migrations, models
import django.db.models.deletion


class Migration(migrations.Migration):

    initial = True

    dependencies = [
        migrations.swappable_dependency(settings.AUTH_USER_MODEL),
    ]

    operations = [
        migrations.CreateModel(
            name='Project',
            fields=[
                ('id', models.AutoField(auto_created=True, primary_key=True,␣
↪serialize=False, verbose_name='ID')),
                ('title', models.CharField(max_length=200)),
                ('champion', models.ForeignKey(on_delete=django.db.models.deletion.
↪CASCADE, to=settings.AUTH_USER_MODEL)),
            ],
        ),
    ]
```

### sqlmigrate

```
python manage.py sqlmigrate projects
```

```
(ajax_django_35) C:\projects_id3\django_ajax_select\projet_ajax>python manage.py␣
↪sqlmigrate projects 0001
```

```
BEGIN;
--
-- Create model Project
--
CREATE TABLE "projects_project" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT,
↪"title" varchar(200) NOT NULL, "champion_id" integer NOT NULL REFERENCES "auth_user
↪" ("id"));
CREATE INDEX "projects_project_68b9a77a" ON "projects_project" ("champion_id");
COMMIT;
```

### migrate

```
python manage.py migrate projects
```

```
Operations to perform:
  Apply all migrations: projects
Running migrations:
  Applying projects.0001_initial... OK
```

### Create the projects/forms.py file and add 2 forms

```
#!/usr/bin/python
# -*- coding: utf8 -*-
"""The project's forms.

"""

from django import forms

from .models import Project

class ProjectChampionForm(forms.ModelForm):
    """The champion project form"""
    champions_choice_list = forms.CharField(max_length=100,
                                            help_text='type username or email')

    class Meta:
        model = Project
        fields = ('title',
                  'champions_choice_list', 'champion',)

    def __init__(self, *args, **kwargs):
        super(ProjectChampionForm, self).__init__(*args, **kwargs)
        self.fields['champions_choice_list'].label = "Update the champion"
        self.fields['champion'].widget = forms.HiddenInput()
```

### Create the projects/urls.py file and add the champion auto complete URL

```
# -*- coding: utf-8 -*-

from django.conf.urls import url


from .views_json import (ApiEACGetProjectsJSONView,
                         ApiEACGetchampionsJSONView)

from .views_django_autocomplete_light import (ApiUserDjangoAutocompleteLight,
                                              ProjectDjangoAutoCompleteUpdateView,
                                              ProjectAutocompleteView)

from .views import (ProjectUpdateView,
                    ProjectUpdateViewEasyAutoComplete,
                    ProjectUpdateViewJQueryUIAutoComplete)


urlpatterns = [
     url(r'^project/(?P<pk>\d+)/update/$',
         ProjectUpdateView.as_view(),
         name='project_update'),

    url(r'^project/(?P<pk>\d+)/updateeasy/$',
        ProjectUpdateViewEasyAutoComplete.as_view(),
        name='project_update_easy'),

    url(r'^project/(?P<pk>\d+)/update_jquery_ui/$',
```

```
        ProjectUpdateViewJQueryUIAutoComplete.as_view(),
        name='project_update_jquery_ui'),

    url(r'^project/(?P<pk>\d+)/update_django_autocomplete/$',
        ProjectDjangoAutoCompleteUpdateView.as_view(),
        name='project_update_django_autocomplete'),

     # calls by jquery EasyAutocomplete
     # http://127.0.0.1:8004/projects/api_get_champions/?term=a
     url(r'^api_get_champions/$',
         ApiEACGetchampionsJSONView.as_view(),
         name='api_get_champions'),

    # calls by jquery EasyAutocomplete (EAC)
    # http://127.0.0.1:8004/projects/api_get_projects/?term=a
    url(r'^apis_get_projects/$',
        ApiEACGetProjectsJSONView.as_view(),
        name='api_get_projects'),

     # calling example:
     # http://127.0.0.1:8004/projects/project_autocomplete/?q=a
     url(
        r'^project_autocomplete/$',
        ProjectAutocompleteView.as_view(),
        name='project_autocomplete',
     ),

    url(r'^api/get_users/$',
        ApiUserDjangoAutocompleteLight.as_view(),
        name='api_get_users'),
]
```

### Add the projects.urls file to the root urls.py

```
url(r'^projects/', include('projects.urls', namespace='projects')),
```

```
#!/usr/bin/python
# -*- coding: utf8 -*-
"""projet_ajax URL Configuration

The `urlpatterns` list routes URLs to views. For more information please see:
    https://docs.djangoproject.com/en/1.10/topics/http/urls/
Examples:
Function views
    1. Add an import:  from my_app import views
    2. Add a URL to urlpatterns:  url(r'^$', views.home, name='home')
Class-based views
    1. Add an import:  from other_app.views import Home
    2. Add a URL to urlpatterns:  url(r'^$', Home.as_view(), name='home')
Including another URLconf
    1. Import the include() function: from django.conf.urls import url, include
    2. Add a URL to urlpatterns:  url(r'^blog/', include('blog.urls'))
"""

from django.conf import settings
from django.conf.urls import (include,
```

```
                                url)

from django.contrib import admin

from ajax_select import urls as ajax_select_urls
from singers import views


urlpatterns = [
    url(r'^admin/', admin.site.urls),

    url(r'^search',
        view=views.search_form,
        name='search_form'),
    url(r'^admin/lookups/', include(ajax_select_urls)),

    url(r'^projects/', include('projects.urls', namespace='projects')),
    url(r'^singers/', include('singers.urls', namespace='singers')),
]


if settings.DEBUG:
    from django.contrib.staticfiles.urls import staticfiles_urlpatterns
    from django.conf.urls.static import static
    urlpatterns += staticfiles_urlpatterns()
    # voir p.405 "Django By Example"
    # https://docs.djangoproject.com/en/dev/howto/static-files/#serving-files-
↪uploaded-by-a-user-during-development
    urlpatterns += static(settings.MEDIA_URL ,
                          document_root=settings.MEDIA_ROOT)

    import debug_toolbar
    urlpatterns += [
        url(r'^__debug__/', include(debug_toolbar.urls)),
    ]
```

### Add the ProjectUpdateView

```


import logging

from django.core.urlresolvers import reverse

from django.views.generic.edit import UpdateView
from django.http import HttpResponseRedirect


from .models import Project
from .forms import ProjectChampionForm


# Get an instance of a logger
logger = logging.getLogger(__name__)


```

```
class ProjectUpdateView(UpdateView):
    """
            url(r'^projects/project/(?P<pk>\d+)/update/$', ProjctUpdate.as_view(), name='project_update'),

    Documentation:

    - http://ccbv.co.uk/projects/Django/1.10/django.views.generic.edit/UpdateView/

    """
    model = Project
    form_class = ProjectChampionForm
    context_object_name = 'project'
    template_name = 'projects/project/update_easy_simple.html'

    def get_object(self, queryset=None):
        """Pour mémoriser self.demande_article"""
        self.object = super(ProjectUpdateView, self).get_object(queryset)
        return self.object

    def post(self, request, *args, **kwargs):
        logger.warning("Hello from ProjectUpdateView !")
        if "cancel" in request.POST:
            url = self.get_success_url()
            return HttpResponseRedirect(url)
        else:
            return super(ProjectUpdateView, self).post(request, *args, **kwargs)



class ProjectUpdateViewEasyAutoComplete(UpdateView):
    """Update thye view with the jQuery EasyAutocomplete plugin.

    Documentation:

    - http://ccbv.co.uk/projects/Django/1.10/django.views.generic.edit/UpdateView/

    """
    model = Project
    form_class = ProjectChampionForm
    context_object_name = 'project'
    template_name = 'projects/project/update_easyautocomplete.html'

    def get_object(self, queryset=None):
        """Pour mémoriser self.demande_article"""
        self.object = super(ProjectUpdateViewEasyAutoComplete, self).get_object(queryset)
        return self.object

    def post(self, request, *args, **kwargs):
        logger.warning("Hello from ProjectUpdateViewEasyAutoComplete !")
        if "cancel" in request.POST:
            url = reverse('projects:project_update_easy', kwargs={'pk': self.pk})
            return HttpResponseRedirect(url)
        else:
            return super(ProjectUpdateViewEasyAutoComplete, self).post(request, *args, **kwargs)
```

```
74
75
76  class ProjectUpdateViewJQueryUIAutoComplete(UpdateView):
77      """Update the view with the jQuery UI Autocomplete plugin.
78
79      Documentation:
80
81      - http://ccbv.co.uk/projects/Django/1.10/django.views.generic.edit/UpdateView/
82
83      """
84      model = Project
85      form_class = ProjectChampionForm
86      context_object_name = 'project'
87      template_name = 'projects/project/update_jquery_ui_autocomplete.html'
88
89      def get_object(self, queryset=None):
90          """Pour mémoriser self.demande_article"""
91          self.object = super(ProjectUpdateViewJQueryUIAutoComplete, self).get_
    ↪object(queryset)
92          return self.object
93
94      def post(self, request, *args, **kwargs):
95          logger.warning("Hello from ProjectUpdateViewJQueryUIAutoComplete !")
96          return super(ProjectUpdateViewJQueryUIAutoComplete, self).post(request, *args,
    ↪ **kwargs)
97
98
99
100
```

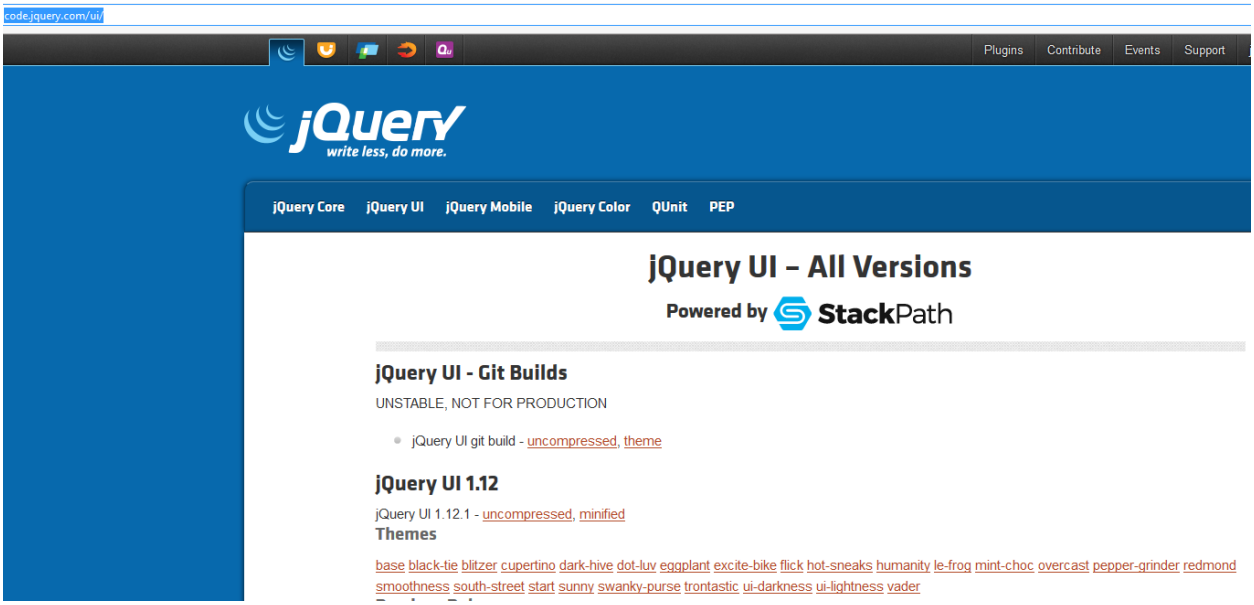
### Create the templates/projects/project/update.html django template

**See also:**

- http://code.jquery.com/ui/

This template file is used by the ProjectUpdateView.

```
1   {% extends "base.html" %}
2   {% load static %}
3   {% load staticfiles %}
4
5
6   {# STYLE CSS #}
7   {% block stylesheet %}
8       {{ block.super }}
9       {# https://www.bootstrapcdn.com/ #}
10      <link rel="stylesheet" href="http://code.jquery.com/ui/1.12.1/themes/base/jquery-
    ↪ui.csss">
11
12      <style>
13      {# https://jqueryui.com/autocomplete/#maxheight #}
14      .ui-autocomplete {
15          max-height: 100px;
16          overflow-y: auto;
17          /* prevent horizontal scrollbar */
18          overflow-x: hidden;
```




```
19      }
20      * html .ui-autocomplete {
21          height: 100px;
22      }
23      </style>
24
25  {% endblock  %}
26
27
28  {% block content %}
29
30      <h1>Update of the project '(title:{{ project.title }} champion:{{ project.
    ↪champion }}) </h1>
31      <p></p>
32      <p></p>
33      {# https://docs.djangoproject.com/en/dev/topics/forms/ #}
34      <form id="id_form_project_update" action="{% url 'projects:project_update'␣
    ↪project.id %}" method="post">
35          {% csrf_token %}
36          <div class="forms">
37              {{ form.id }}
38              {{ form.non_field_errors }}
39              {# Include the hidden fields #}
40              {% for hidden in form.hidden_fields %}
41                  {{ hidden }}
42              {% endfor %}
43              <table id="id_table" class="table table-hover table-bordered table-
    ↪condensed">
44                  <tbody>
45                          <tr>
46                              <td class="text-right">Title:</td>
47                              <td>{{ form.title }}</td>
48                          </tr>
49                          <tr>
50                              <td class="text-right">Champion:</td>
51                              <td> <span id="id_champion_display">{{ project.champion }}
    ↪</span> {{ form.champion_term }}  </td>
```

```
                        </tr>
                </tbody>
            </table>
        </div>
        <input type="submit" name="btn_update"  value="Update" class="btn btn-success btn-block" />
    </form>


    {% if messages %}
        <ul class="messages">
            {% comment %} https://getbootstrap.com/components/#alerts {% endcomment %}
            {% for message in messages %}
            <!--li{% if message.tags %} class="{{ message.tags }}"{% endif %} -->
                {% if message.level == DEFAULT_MESSAGE_LEVELS.ERROR %}
                    <div class="alert alert-danger" role="alert">{{ message }}</div>
                {% elif message.level == DEFAULT_MESSAGE_LEVELS.SUCCESS %}
                    <div class="alert alert-success" role="alert">{{ message }}</div>
                {% else %}
                    <div class="alert alert-info" role="alert">{{ message }}</div>
                {% endif %}
            <!-- /li -->
            {% endfor %}
        </ul>
    {% endif %}


{% endblock content %}


{# BEHAVIOR Javascript #}
{% block javascript %}
    {{ block.super }}

    {# http://code.jquery.com/ui/ #}
    <script src="http://code.jquery.com/ui/1.12.1/jquery-ui.min.js"></script>

{% endblock javascript %}


{# DOM ready #}
{% block domready %}

    $( "#id_champion_term" ).autocomplete({
          {# http://api.jqueryui.com/autocomplete/#option-source #}
          source: "{% url 'projects:champion_get_json' %}",
          {# http://api.jqueryui.com/autocomplete/#option-autoFocus #}
          autoFocus:true,
          {# http://api.jqueryui.com/autocomplete/#option-minLength #}
          minLength:1,
          {# http://api.jqueryui.com/autocomplete/#event-select #}
          select:function(event,ui) {
              $("#id_champion").val(ui.item.id);
          }
    });

{% endblock %}
```

### Add the jquery-ui javascript code

**See also:**

- http://api.jqueryui.com/autocomplete/
- http://api.jqueryui.com/autocomplete/#option-source
- http://api.jqueryui.com/autocomplete/#option-minLength
- http://api.jqueryui.com/autocomplete/#event-select

```
$( "#id_champion_display" ).autocomplete({
      source: "{% url champion_auto_complete %}",
      minLength:2,
      select:function(event,ui) {
        $("#id_champion").val(ui.item.id)
      }
});
```

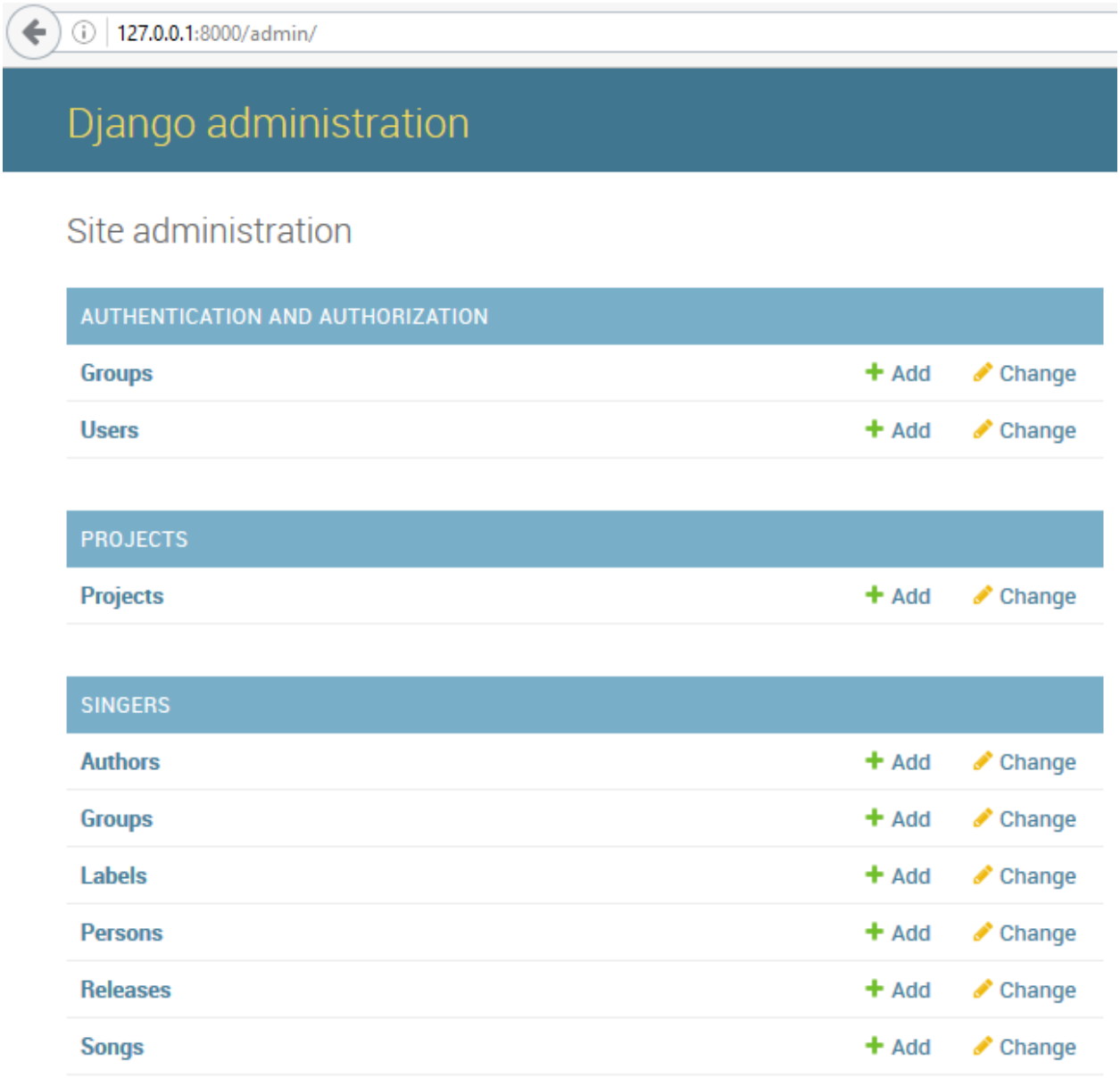
### Add the admin interface

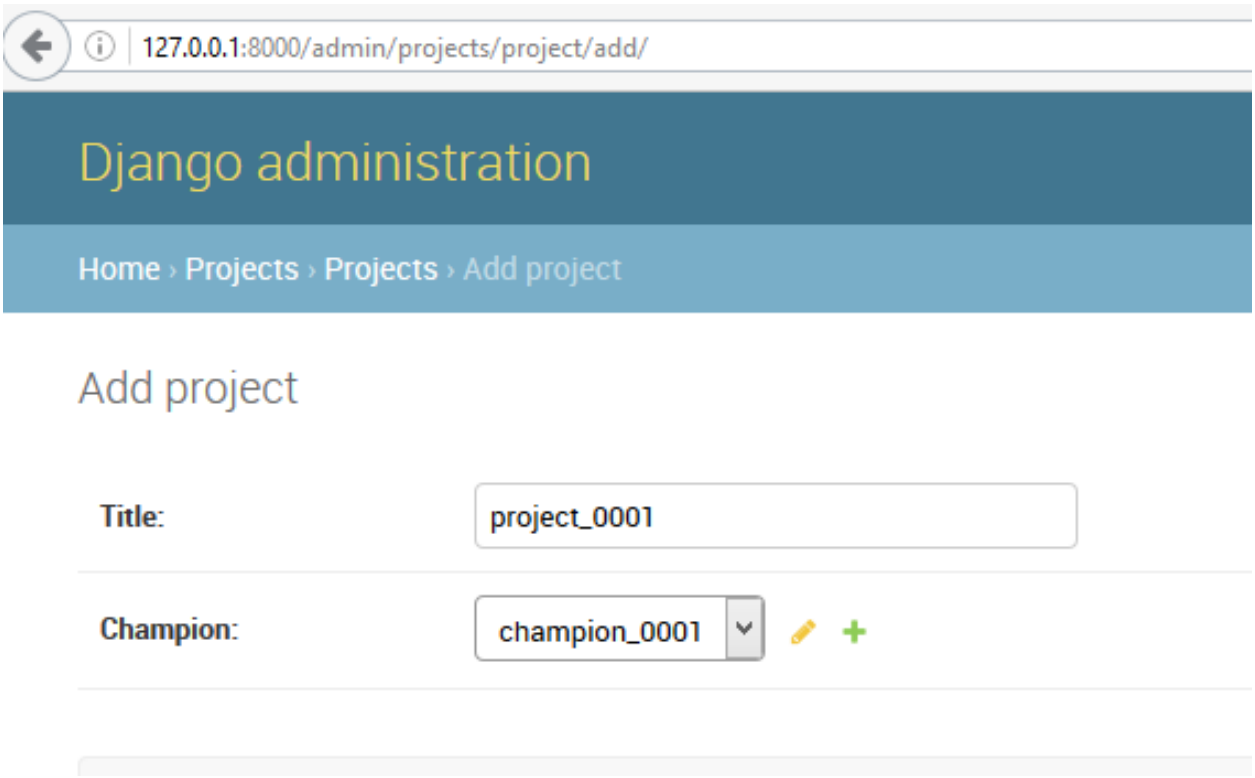
```
#!/usr/bin/python
# -*- coding: UTF-8 -*-
"""Project Administration.

"""

from django.contrib import admin

from .models import Project

from .forms_django_autocomplete_light import ProjectFormDjangoAutocomplete

@admin.register(Project)
class ProjectAdmin(admin.ModelAdmin):
    """Project administration

    Documentation
    =============

    - https://docs.djangoproject.com/en/dev/ref/contrib/admin/#modeladmin-objects

    """
    form = ProjectFormDjangoAutocomplete
    list_display = ('title', 'champion')
    search_fields = ('title', 'champion')
    list_filter = ('title', 'champion')
```

### Create some projects with the admin interface

### Reading Project record from the commande line

```
python manage.py shell_plus
```

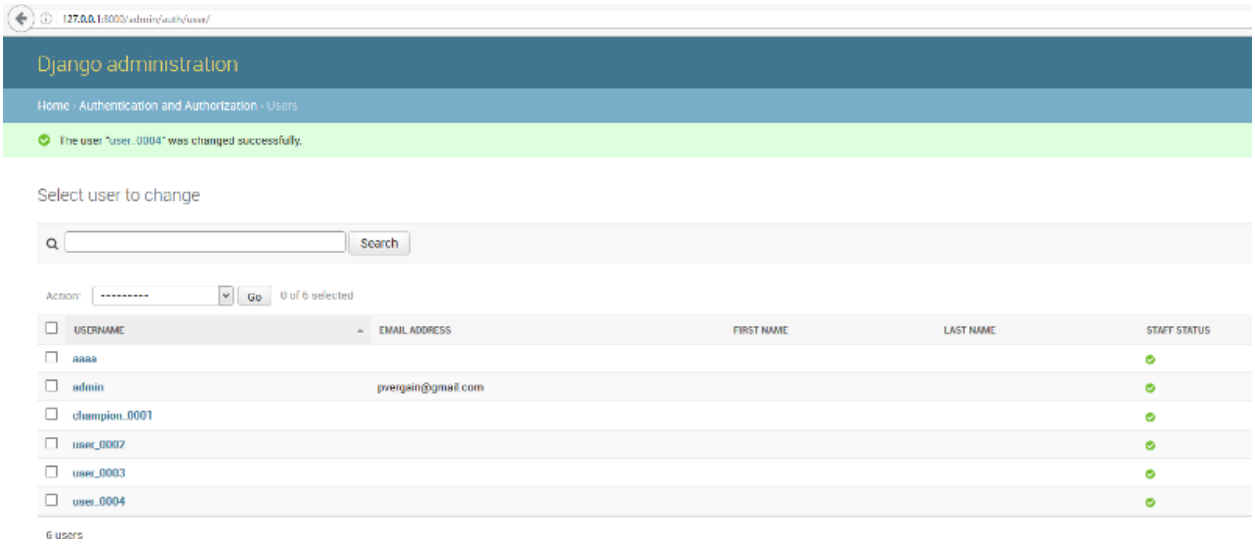

127.0.0.1:8000/admin/
Django administration
Site administration
AUTHENTICATION AND AUTHORIZATION
Groups
Add
Change
Users
Add
Change
PROJECTS
Projects
Add
Change
SINGERS
Authors
Add
Change
Groups
Add
Change
Labels
Add
Change
Persons
Add
Change
Releases
Add
Change
Songs
Add
Change




```
(ajax_django_35) C:\projects_id3\django_ajax_select\projet_ajax>python manage.py␣
↪shell_plus
# Shell Plus Model Imports
from django.contrib.admin.models import LogEntry
from django.contrib.auth.models import Group, Permission, User
from django.contrib.contenttypes.models import ContentType
from django.contrib.sessions.models import Session
from projects.models import Project
from singers.models import Author, Book, Group, Label, Person, Release, Song
# Shell Plus Django Imports
from django.db.models import Avg, Case, Count, F, Max, Min, Prefetch, Q, Sum, When
from django.db import transaction
from django.conf import settings
from django.utils import timezone
from django.urls import reverse
from django.core.cache import cache
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:18:55) [MSC v.1900 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 5.1.0 -- An enhanced Interactive Python.
?         -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help      -> Python's own help system.
object?   -> Details about 'object', use 'object??' for extra details.

In [1]: p1=Project.objects.first()

In [2]: p1
Out[2]: <Project: project_0001 champion_0001>
```

```
In [3]: p1.id
Out[3]: 1
```

### Add some users with the admin interface




### Add some users with the command line interface

**See also:**

- https://docs.djangoproject.com/en/dev/topics/auth/default/#user-objects

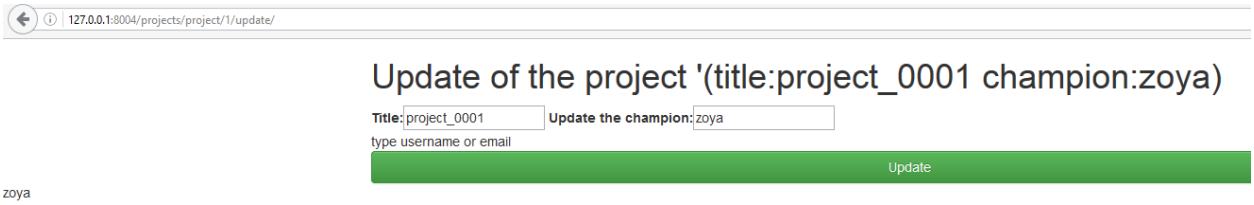
The most direct way to create users is to use the included create_user() helper function:

```
In [4]: user = User.objects.create_user('john', 'lennon@thebeatles.com', 'johnpassword
↪')

In [5]: user = User.objects.create_user('albert', 'albert@thebeatles.com',
↪'albertpassword')

In [6]: user = User.objects.create_user('zoya', 'zoya@thebeatles.com', 'zoyapassword')

In [7]: user = User.objects.create_user('nigel', 'nigel@thebeatles.com',
↪'nigelpassword')

In [8]: users=User.objects.all()

In [9]: users
Out[9]: <QuerySet [<User: admin>, <User: champion_0001>, <User: user_0002>, <User:␣
↪user_0003>, <User: user_0004>, <User: aaaa>, <User: john>, <User: albert>, <User:␣
↪zoya>, <User: nigel>]>
```

### Try the URL : http://127.0.0.1:8000/projects/project/1/update

**OK, this time this works fine !**

**Warning:** La saisie est agréable, il faut améliorer le look.




```
[18/Oct/2016 15:17:26] "GET /projects/project/1/update/ HTTP/1.1" 200 3423
[18/Oct/2016 15:17:44] "GET /projects/champion_auto_complete/?term=zo HTTP/1.1" 200 45
[18/Oct/2016 15:17:51] "POST /projects/project/1/update/ HTTP/1.1" 302 0
[18/Oct/2016 15:17:51] "GET /projects/project/1/update/ HTTP/1.1" 200 3418
[18/Oct/2016 15:18:47] "GET /projects/champion_auto_complete/?term=ni HTTP/1.1" 200 48
[18/Oct/2016 15:19:00] "POST /projects/project/1/update/ HTTP/1.1" 302 0
[18/Oct/2016 15:19:00] "GET /projects/project/1/update/ HTTP/1.1" 200 3420
```

## 2016-10-18 create some Django Templates : base.html

**See also:**

- https://docs.djangoproject.com/en/dev/ref/templates/language/#

**Contents**



### The Django Template templates/base.html

```
{# https://docs.djangoproject.com/en/dev/ref/templates/builtins/#std:templatetag-static #}
{% load static %}
{# https://docs.djangoproject.com/en/dev/ref/templates/builtins/#tz #}
{% load tz %}

<!DOCTYPE html>
```

```
{# https://www.w3.org/TR/2014/REC-html5-20141028/syntax.html#the-doctype #}
<!--[if lt IE 7]>      <html class="no-js lt-ie9 lt-ie8 lt-ie7"> <![endif]-->
<!--[if IE 7]>         <html class="no-js lt-ie9 lt-ie8"> <![endif]-->
<!--[if IE 8]>         <html class="no-js lt-ie9"> <![endif]-->
<!--[if gt IE 8]><!--> <html class="no-js"> <!--<![endif]-->
{# https://www.w3.org/TR/html5/semantics.html#the-html-element #}
    <head>
        {# https://www.w3.org/TR/html5/document-metadata.html#attr-meta-charset #}
        <meta charset="utf-8">
        {# https://n.survol.fr/n/x-ua-compatible-ieedge #}
        <!-- meta http-equiv="X-UA-Compatible" content="IE=edge" -->
        {# https://www.w3.org/TR/html5/dom.html#the-title-attribute #}
        {# https://www.w3.org/TR/html5/document-metadata.html#the-title-element #}
        <title>
            {% block head_title %}
            {% endblock %}
        </title>

        {# https://www.w3.org/TR/html5/document-metadata.html#attr-meta-name #}
↪

        {# https://developer.mozilla.org/fr/docs/Mozilla/Mobile/Balise_meta_viewport
↪#}
        <meta name="viewport" content="width=device-width, initial-scale=1" />


        {# CSS #}
        {% block stylesheet %}
            {# https://docs.djangoproject.com/en/dev/ref/templates/language/#template-
↪inheritance #}
            {# If you need to get the content of the block from the parent template,
↪the {{ block.super }} variable will do the trick #}
            {# {{ block.super }} #}

            {# https://blog.getbootstrap.com/2016/07/25/bootstrap-3-3-7-released
↪(2016-07-25) Added support for jQuery 3 #}
            {# https://www.bootstrapcdn.com/ #}
            <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.
↪3.7/css/bootstrap.min.css">
            <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.
↪3.7/css/bootstrap-theme.min.css">

        {% endblock  %}



    </head>
    <body class="{% block bodyclass %} {% endblock %}">


        <div id="conteneur" class="container">
            {% block content %}
            {% endblock %}

            {% block footer %}
            {% endblock %}

        </div> <!-- id conteneur -->
```

```
58
59          {# Javascript #}
60          {% block javascript %}
61
62              {# a employer dans les templates fils->{{ block.super }} #}
63              {# https://docs.djangoproject.com/en/dev/ref/templates/language/#template-
    ↪inheritance #}
64              {# If you need to get the content of the block from the parent template,␣
    ↪the {{ block.super }} variable will do the trick #}
65              <!-- Liaison aux scripts Javascript -->
66              <!-- CDN JQuery Placer jquery avant bootstrap, select2 -->
67              {# https://github.com/jquery/jquery-migrate/#README #}
68              {# https://blog.jquery.com/2016/09/22/jquery-3-1-1-released/ #}
69              {# https://code.jquery.com/ #}
70              <script src="https://code.jquery.com/jquery-3.1.1.min.js"></script>
71              <script src="https://code.jquery.com/jquery-migrate-3.0.0.min.js"></
    ↪script>
72
73              <!-- Code javascript pour bootstrap 3.3.7 (2016-07-25) -->
74              <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.
    ↪min.js"> </script>
75
76          {% endblock javascript %}
77
78          {# p.155 Adding AJAX actions with jQuery, Django By Example from Antonio Melé
    ↪#}
79          {# https://www.amazon.com/Django-Example-Antonio-Mele/dp/1784391913/ #}
80          <script>
81              $(document).ready(function(){
82                  {% block domready %}
83                  {% endblock %}
84              });
85          </script>
86
87      </body>
88  </html>
```

```
|   ajax_selects_singers_db
|   manage.py
|
+---projet_ajax
|   |   settings.py
|   |   urls.py
|   |   wsgi.py
|   |   __init__.py
|
+---singers
|   |   admin.py
|   |   apps.py
|   |   forms.py
|   |   lookups.py
|   |   models.py
|   |   tests.py
|   |   urls.py
|   |   views.py
|   |   __init__.py
|   |
|   +---migrations
```

```
|   |   |   0001_initial.py
|   |   |   0002_auto_20161017_1612.py
|   |   |   0003_auto_20161017_1632.py
|   |   |   __init__.py
|   |
|
\---templates
        base.html
        search_form.html
```

### The Django Template singers/templates/song/update.html

**See also:**

- https://docs.djangoproject.com/en/dev/ref/templates/builtins/#std:templatetag-extends

The template `templates/singers/song/update.html` extends the `templatesbase.html` **base template**.

```
{% extends "base.html" %}
{% load static %}
{% load staticfiles %}
{% load crispy_forms_tags %}

{% block content %}

<h1>Update of the song {{ song.title }} {{ song.group }} </h1>

{# https://docs.djangoproject.com/en/dev/topics/forms/ #}
<form id="song_update" method="post" action=".">
    {% csrf_token %}
    {{ form }}

    {# https://django-ajax-selects.readthedocs.io/en/latest/Outside-of-Admin.html #}
    {{ form.media }}
    <input type="submit" name="btn_update"  value="Update" class="btn btn-success btn-
↪block" />
</form>


{% if messages %}
    <ul class="messages">
        {% comment %} https://getbootstrap.com/components/#alerts {% endcomment %}
        {% for message in messages %}
        <!--li{% if message.tags %} class="{{ message.tags }}"{% endif %} -->
            {% if message.level == DEFAULT_MESSAGE_LEVELS.ERROR %}
                <div class="alert alert-danger" role="alert">{{ message }}</div>
            {% elif message.level == DEFAULT_MESSAGE_LEVELS.SUCCESS %}
                <div class="alert alert-success" role="alert">{{ message }}</div>
            {% else %}
                <div class="alert alert-info" role="alert">{{ message }}</div>
            {% endif %}
        <!-- /li -->
        {% endfor %}
    </ul>
{% endif %}

```

```
38
39 {% endblock content %}
```

### Add the django-extensions and ipython modules

```
pip install django-extensions ipython
```

In the projet_ajax/settings.py file:

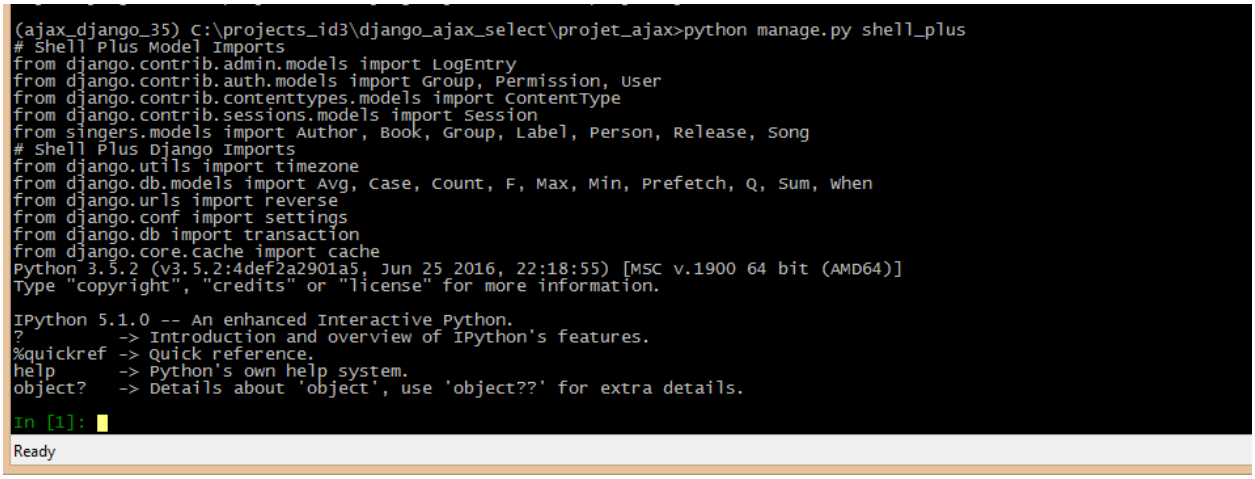
```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',

    # https://django-extensions.readthedocs.org/en/latest
    'django_extensions',
```

### New django commands

```
[django_extensions]
    admin_generator
    clean_pyc
    clear_cache
    compile_pyc
    create_app
    create_command
    create_jobs
    create_template_tags
    describe_form
    drop_test_database
    dumpscript
    export_emails
    find_template
    generate_secret_key
    graph_models
    mail_debug
    notes
    passwd
    pipchecker
    print_settings
    print_user_for_session
    reset_db
    runjob
    runjobs
    runprofileserver
    runscript
    runserver_plus
    set_default_site
    set_fake_emails
    set_fake_passwords
    shell_plus
```

```
    show_template_tags
    show_templatetags
    show_urls
    sqlcreate
    sqldiff
    sqldsn
    sync_s3
    syncdata
    unreferenced_files
    update_permissions
    validate_templates
```

The *shell_plus* command is very usefull.




### Get the first song

```
In [1]:  first_song = Song.objects.all().first()

In [2]: first_song.id
Out[2]: 1
```

### Try the URL : http://127.0.0.1:8000/singers/song/1/update

**See also:**

- http://127.0.0.1:8000/singers/song/1/update

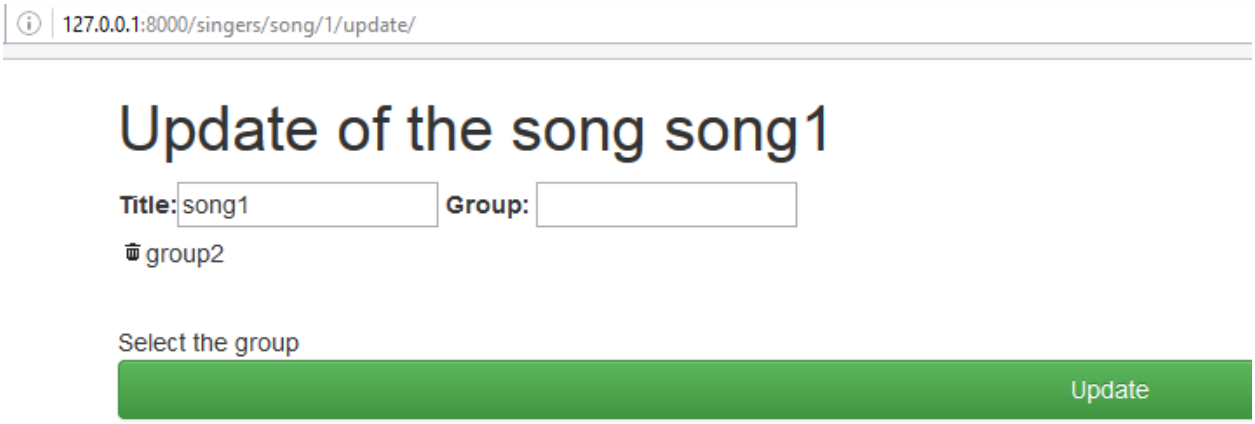
Not very nice and not very easy to use.

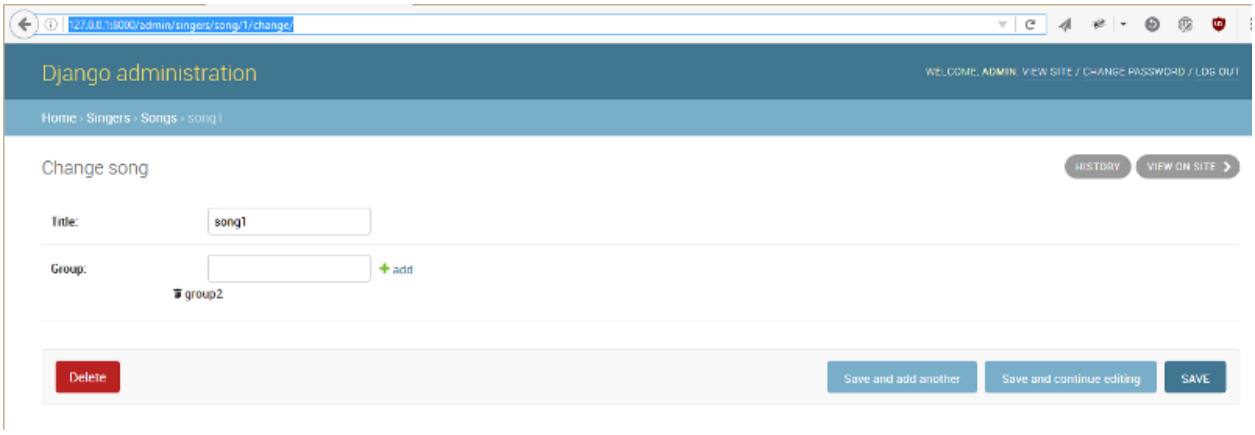
### Compared to the django admin interface

**See also:**

- http://127.0.0.1:8000/admin/singers/song/1/change/

More nice but not very easy to use.


127.0.0.1:8000/singers/song/1/update/
Update of the song song1
Title: song1
Group:
group2
Select the group
Update


Django administration
WELCOME, ADMIN VIEW SITE / CHANGE PASSWORD / LOG OUT
Home › Singers › Songs › song1
Change song
HISTORY
VIEW ON SITE
Title:
song1
Group:
add
group2
Delete
Save and add another
Save and continue editing
SAVE

### Conclusion

Try the django-autocomplete-light module.

## 2016-10-17 create a normal Django view

**Contents**



### Le fichier singers/models.py

```
#!/usr/bin/python
# -*- coding: utf8 -*-
"""The singers models.

"""
from __future__ import unicode_literals
from django.utils.encoding import python_2_unicode_compatible

from django.db import models
from django.core.urlresolvers import reverse


@python_2_unicode_compatible
class Person(models.Model):
    """ an actual singular human being """
    name = models.CharField(blank=True, max_length=100)
    email = models.EmailField()

    def __str__(self):
        return self.name


@python_2_unicode_compatible
class Group(models.Model):
    """ a music group """
    name = models.CharField(max_length=200,
                            unique=True,
                            help_text="Name of the group")
    members = models.ManyToManyField(Person,
        blank=True,
        help_text="Enter text to search for and add each member of the group.")
    url = models.URLField(blank=True)

    def __str__(self):
        return self.name
```

```
36
37
38  @python_2_unicode_compatible
39  class Label(models.Model):
40      """ a record label """
41      name = models.CharField(max_length=200,
42                              unique=True)
43
44      owner = models.ForeignKey(Person,
45                                blank=True,
46                                null=True)
47
48      url = models.URLField(blank=True)
49
50      def __str__(self):
51          return self.name
52
53  @python_2_unicode_compatible
54  class Song(models.Model):
55      """ a song """
56      title = models.CharField(blank=False, max_length=200)
57      group = models.ForeignKey(Group)
58
59      def __str__(self):
60          return self.title
61
62      def get_absolute_url(self):
63          """
64          https://docs.djangoproject.com/en/dev/ref/class-based-views/generic-editing/
65          """
66          return reverse('singers:song_update',
67                         kwargs={'pk': self.pk})
68
69
70  @python_2_unicode_compatible
71  class Release(models.Model):
72      """ a music release/product """
73      title = models.CharField(max_length=100)
74
75      catalog = models.CharField(blank=True,
76                                 max_length=100)
77
78      group = models.ForeignKey(Group, blank=True, null=True,
79                                verbose_name="  (group)")
80
81      label = models.ForeignKey(Label,
82                                blank=False,
83                                null=False)
84
85      songs = models.ManyToManyField(Song,
86                                     blank=True)
87
88      def __str__(self):
89          return self.title
90
91  @python_2_unicode_compatible
92  class Author(models.Model):
93      """ Author has multiple books,
```

```
        via foreign keys
    """
    name = models.CharField(max_length=100)

    def __str__(self):
        return self.name


@python_2_unicode_compatible
class Book(models.Model):
    """ Book has no admin, its an inline in the Author admin"""

    author = models.ForeignKey(Author)
    title = models.CharField(max_length=100)
    about_group = models.ForeignKey(Group)
    mentions_persons = models.ManyToManyField(Person,
                                              help_text="Person lookup renders html in menu")

    def __str__(self):
        return self.title

```

### Le fichier singers/forms.py

```
# -*- coding: utf-8 -*-

from __future__ import unicode_literals

from django.forms.models import ModelForm
from ajax_select import make_ajax_field
from dal import autocomplete

from .models import (Release,
                     Song,
                     Author,
                     Book)

class ReleaseForm(ModelForm):

    class Meta:
        model = Release
        exclude = []

    # args:  this model, fieldname on this model, lookup_channel_name
    group = make_ajax_field(Release, 'group', 'group', show_help_text=True)

    label = make_ajax_field(Release, 'label', 'label', help_text="Search for label by name")

    # any extra kwargs are passed onto the field, so you may pass a custom help_text here
    # songs = make_ajax_field(Release,'songs','song', help_text=u"Search for song by title")

    # testing bug with no help text supplied
```

```
    songs = make_ajax_field(Release, 'songs', 'song', help_text="", show_help_
↪text=True)

    # these are from a fixed array defined in lookups.py
    title = make_ajax_field(Release, 'title', 'cliche', help_text="Autocomplete will␣
↪suggest clichés about cats.")


class SongForm(ModelForm):
    """The Django form"""

    class Meta:
        model = Song
        fields = ['title', 'group']

    # args:  this model, fieldname on this model, lookup_channel_name
    group = make_ajax_field(Song, # this model
                            'group', # fieldname on this model
                            'group', # lookup_channel_name
                            help_text="Select the group")


class AuthorForm(ModelForm):
    class Meta:
        model = Author
        fields = ('name',)
        widgets = {
            'name': autocomplete.Select(url='singers:author_autocomplete')
        }

class BookForm(ModelForm):
    class Meta:
        model = Book
        fields = ('title', 'author',)
        widgets = {
            'author': autocomplete.ModelSelect2(url='singers:author_autocomplete')
        }


```

### Le fichier singers/views.py

```
# -*- coding: utf-8 -*-

from __future__ import unicode_literals
from django import forms

from django.shortcuts import render_to_response

# https://docs.djangoproject.com/en/dev/ref/templates/api/
from django.template import RequestContext
from django.http import HttpResponseRedirect

from django.views.generic.edit import UpdateView

from ajax_select.fields import AutoCompleteField
```

```

# https://stackoverflow.com/questions/37915224/django-autocomplete-light-widgets-not-showing-up?rq=1
from dal import autocomplete

from .models import (Song,
                     Author,
                     Book)

from .forms import (SongForm,
                    AuthorForm,
                    BookForm)

class SearchForm(forms.Form):

    q = AutoCompleteField('cliche',
        required=True,
        help_text="Autocomplete will suggest clichés about cats, but you can enter anything you like.",
        label="Favorite Cliché",
        attrs={'size': 100})


def search_form(request):

    dd = {}
    if 'q' in request.GET:
        dd['entered'] = request.GET.get('q')
    initial = {'q': "\"This is an initial value,\" said O'Leary."}
    form = SearchForm(initial=initial)
    dd['form'] = form
    return render_to_response('search_form.html',
                              dd,
                              context=RequestContext(request))



class SongUpdate(UpdateView):
    """
            url(r'^singers/song/(?P<pk>\d+)/update/$', SongUpdate.as_view(), name='song_update'),

    Documentation:

    - http://ccbv.co.uk/projects/Django/1.9/django.views.generic.edit/UpdateView/

    """
    model = Song
    form_class = SongForm
    context_object_name = 'song'
    template_name = 'singers/song/update.html'

    def get_object(self, queryset=None):
        """Pour mémoriser self.demande_article"""
        self.object = super(SongUpdate, self).get_object(queryset)
        return self.object

    def post(self, request, *args, **kwargs):
```

```
        if "cancel" in request.POST:
            url = self.get_success_url()
            return HttpResponseRedirect(url)
        else:
            return super(SongUpdate, self).post(request, *args, **kwargs)


class AuthorAutocomplete(autocomplete.Select2QuerySetView):
    def get_queryset(self):
        # Don't forget to filter out results depending on the visitor !

        qs = Author.objects.all()

        if self.q:
            qs = qs.filter(name__istartswith=self.q)

        return qs



class AuthorUpdate(UpdateView):
    """
            url(r'^singers/song/(?P<pk>\d+)/update/$', SongUpdate.as_view(), name=
↪'song_update'),

    Documentation:

    - http://ccbv.co.uk/projects/Django/1.9/django.views.generic.edit/UpdateView/

    """
    model = Author
    form_class = AuthorForm
    context_object_name = 'author'
    template_name = 'singers/author/update.html'

    def get_object(self, queryset=None):
        """Pour mémoriser self.demande_article"""
        self.object = super(AuthorUpdate, self).get_object(queryset)
        return self.object

    def post(self, request, *args, **kwargs):
        if "cancel" in request.POST:
            url = self.get_success_url()
            return HttpResponseRedirect(url)
        else:
            return super(AuthorUpdate, self).post(request, *args, **kwargs)

class BookUpdate(UpdateView):
    """
            url(r'^singers/song/(?P<pk>\d+)/update/$', SongUpdate.as_view(), name=
↪'song_update'),

    Documentation:

    - http://ccbv.co.uk/projects/Django/1.9/django.views.generic.edit/UpdateView/

    """
    model = Book
```

```
    form_class = BookForm
    context_object_name = 'author'
    template_name = 'singers/author/update.html'

    def get_object(self, queryset=None):
        """Pour mémoriser self.demande_article"""
        self.object = super(BookUpdate, self).get_object(queryset)
        return self.object

    def post(self, request, *args, **kwargs):
        if "cancel" in request.POST:
            url = self.get_success_url()
            return HttpResponseRedirect(url)
        else:
            return super(BookUpdate, self).post(request, *args, **kwargs)
```

### Le fichier singers/urls.py

```
#!/usr/bin/python
# -*- coding: utf8 -*-

from django.conf.urls import url
from  .views import (SongUpdate,
                     BookUpdate,
                     AuthorAutocomplete)


urlpatterns = [
    url(r'^song/(?P<pk>\d+)/update/$', SongUpdate.as_view(), name='song_update'),
    url(r'^book/(?P<pk>\d+)/update/$', BookUpdate.as_view(), name='book_update'),
    url(r'^author_autocomplete/$', AuthorAutocomplete.as_view(), name='author_
↪autocomplete'),
]
```

## 2016-10-17 create singers SQL tables

**Contents**



The

### makemigrations singers

```
(ajax_django_35) projet_ajax>python manage.py makemigrations singers
```

```
Migrations for 'singers':
  singers\migrations\0001_initial.py:
    - Create model Author
    - Create model Book
    - Create model Group
    - Create model Label
    - Create model Person
    - Create model Release
    - Create model Song
    - Add field songs to release
    - Add field owner to label
    - Add field members to group
    - Add field about_group to book
    - Add field author to book
    - Add field mentions_persons to book
```

### sqlmigrate singers 0001

```
(ajax_django_35) C:\projects_id3\django_ajax_select\projet_ajax>python manage.py␣
↪sqlmigrate singers 0001
```

```
BEGIN;
--
-- Create model Author
--
CREATE TABLE "singers_author" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "name
↪" varchar(100) NOT NULL);
--
-- Create model Book
--
CREATE TABLE "singers_book" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "title"␣
↪varchar(100) NOT NULL);
--
-- Create model Group
--
CREATE TABLE "singers_group" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "name"␣
↪varchar(200) NOT NULL UNIQUE, "url" varchar(200) NOT NULL);
--
-- Create model Label
--
CREATE TABLE "singers_label" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "name"␣
↪varchar(200) NOT NULL UNIQUE, "url" varchar(200) NOT NULL);
--
-- Create model Person
--
CREATE TABLE "singers_person" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "name
↪" varchar(100) NOT NULL, "email" varchar(254) NOT NULL);
--
-- Create model Release
--
CREATE TABLE "singers_release" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT,
↪"title" varchar(100) NOT NULL, "catalog" varchar(100) NOT NULL,
```

```
"group_id" integer NULL REFERENCES "singers_group" ("id"), "label_id" integer NOT␣
↪NULL REFERENCES "singers_label" ("id"));
--
-- Create model Song
--
CREATE TABLE "singers_song" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "title"␣
↪varchar(200) NOT NULL, "group_id" integer NOT NULL REFERENCES "singers_group" ("id
↪"));
--
-- Add field songs to release
--
CREATE TABLE "singers_release_songs" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT,
↪ "release_id" integer NOT NULL REFERENCES "singers_release" ("id"), "song_id"␣
↪integer NOT NULL REFERENCES "singers_song" ("id"));
--
-- Add field owner to label
--
ALTER TABLE "singers_label" RENAME TO "singers_label__old";
CREATE TABLE "singers_label" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "name"␣
↪varchar(200) NOT NULL UNIQUE, "url" varchar(200) NOT NULL,
"owner_id" integer NULL REFERENCES "singers_person" ("id"));
INSERT INTO "singers_label" ("url", "id", "owner_id", "name") SELECT "url", "id",␣
↪NULL, "name" FROM "singers_label__old";
DROP TABLE "singers_label__old";
CREATE INDEX "singers_release_0e939a4f" ON "singers_release" ("group_id");
CREATE INDEX "singers_release_abec2aca" ON "singers_release" ("label_id");
CREATE INDEX "singers_song_0e939a4f" ON "singers_song" ("group_id");
CREATE UNIQUE INDEX "singers_release_songs_release_id_be8581cc_uniq" ON "singers_
↪release_songs" ("release_id", "song_id");
CREATE INDEX "singers_release_songs_b07ccb57" ON "singers_release_songs" ("release_id
↪");
CREATE INDEX "singers_release_songs_e5c27d73" ON "singers_release_songs" ("song_id");
CREATE INDEX "singers_label_5e7b1936" ON "singers_label" ("owner_id");
--
-- Add field members to group
--
CREATE TABLE "singers_group_members" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT,
↪ "group_id" integer NOT NULL REFERENCES "singers_group" ("id"), "person_id" integer␣
↪NOT NULL REFERENCES "singers_person" ("id"));
--
-- Add field about_group to book
--
ALTER TABLE "singers_book" RENAME TO "singers_book__old";
CREATE TABLE "singers_book" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "title"␣
↪varchar(100) NOT NULL, "about_group_id" integer NOT NULL REFERENCES "singers_group"␣
↪("id"));
INSERT INTO "singers_book" ("title", "id", "about_group_id") SELECT "title", "id",␣
↪NULL FROM "singers_book__old";
DROP TABLE "singers_book__old";
CREATE UNIQUE INDEX "singers_group_members_group_id_6767fcf1_uniq" ON "singers_group_
↪members" ("group_id", "person_id");
CREATE INDEX "singers_group_members_0e939a4f" ON "singers_group_members" ("group_id");
CREATE INDEX "singers_group_members_a8452ca7" ON "singers_group_members" ("person_id
↪");
CREATE INDEX "singers_book_13ef9fae" ON "singers_book" ("about_group_id");
--
-- Add field author to book
--
```

```
ALTER TABLE "singers_book" RENAME TO "singers_book__old";
CREATE TABLE "singers_book" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "title"␣
↪varchar(100) NOT NULL, "about_group_id" integer NOT NULL REFERENCES "singers_group"␣
↪("id"), "author_id" integer NOT NULL REFERENCES "singers_author" ("id"));
INSERT INTO "singers_book" ("author_id", "title", "id", "about_group_id") SELECT NULL,
↪ "title", "id", "about_group_id" FROM "singers_book__old";
DROP TABLE "singers_book__old";
CREATE INDEX "singers_book_13ef9fae" ON "singers_book" ("about_group_id");
CREATE INDEX "singers_book_4f331e2f" ON "singers_book" ("author_id");
--
-- Add field mentions_persons to book
--
CREATE TABLE "singers_book_mentions_persons" ("id" integer NOT NULL PRIMARY KEY␣
↪AUTOINCREMENT, "book_id" integer NOT NULL REFERENCES "singers_book" ("id"), "person_
↪id" integer NOT NULL REFERENCES "singers_person" ("id"));
CREATE UNIQUE INDEX "singers_book_mentions_persons_book_id_873aadf4_uniq" ON "singers_
↪book_mentions_persons" ("book_id", "person_id");
CREATE INDEX "singers_book_mentions_persons_0a4572cc" ON "singers_book_mentions_
↪persons" ("book_id");
CREATE INDEX "singers_book_mentions_persons_a8452ca7" ON "singers_book_mentions_
↪persons" ("person_id");
COMMIT;
```

### migrate singers

**See also:**

- https://docs.djangoproject.com/en/dev/intro/tutorial02/
- https://docs.djangoproject.com/fr/1.10/intro/tutorial02/

```
(ajax_django_35) C:\projects_id3\django_ajax_select\projet_ajax>python manage.py␣
↪migrate singers
```

```
Operations to perform:
  Apply all migrations: singers
Running migrations:
  Applying singers.0001_initial... OK
```

### show-migrations

```
(ajax_django_35) C:\projects_id3\django_ajax_select\projet_ajax>python manage.py␣
↪showmigrations
```

```
admin
 [X] 0001_initial
 [X] 0002_logentry_remove_auto_add
auth
 [X] 0001_initial
 [X] 0002_alter_permission_name_max_length
 [X] 0003_alter_user_email_max_length
 [X] 0004_alter_user_username_opts
 [X] 0005_alter_user_last_login_null
 [X] 0006_require_contenttypes_0002
```

```
 [X] 0007_alter_validators_add_error_messages
 [X] 0008_alter_user_username_max_length
contenttypes
 [X] 0001_initial
 [X] 0002_remove_content_type_name
sessions
 [X] 0001_initial
singers
 [X] 0001_initial
```

### createsuperuser

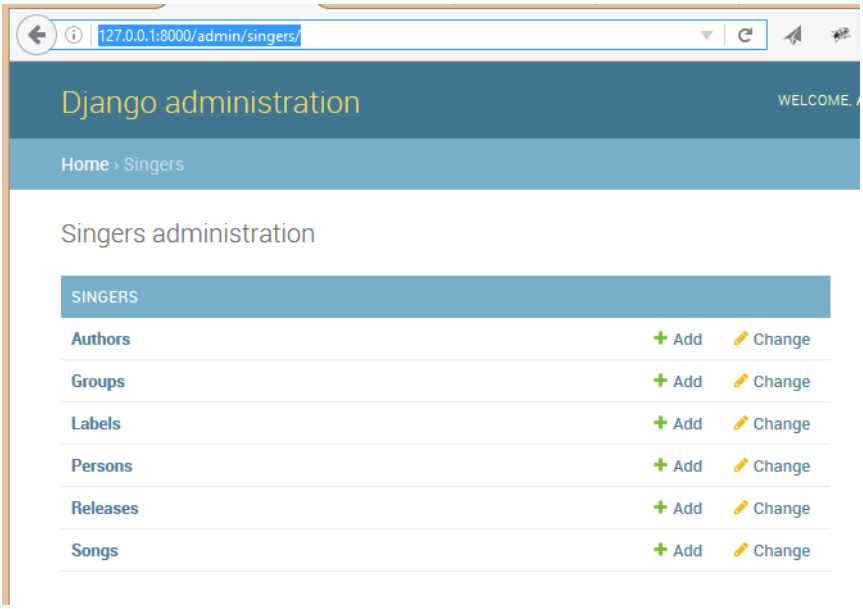
```
(ajax_django_35) C:\projects_id3\django_ajax_select\projet_ajax>python manage.py␣
↪createsuperuser
```

```
Username (leave blank to use 'pvergain'): admin
Email address: pvergain@gmail.com
Password:
Password (again):
This password is too short. It must contain at least 8 characters.
This password is too common.
Password:
Password (again):
The password is too similar to the email address.
Password:
Password (again):
Superuser created successfully.
```

### Connect to the admin interface

**See also:**

http://127.0.0.1:8000/admin/



CHAPTER 4

# Django test AJAX autocomplete development

## Github Singers's development

**See also:**

- https://swcarpentry.github.io/git-novice/
- https://github.com/github/gitignore

**Contents**



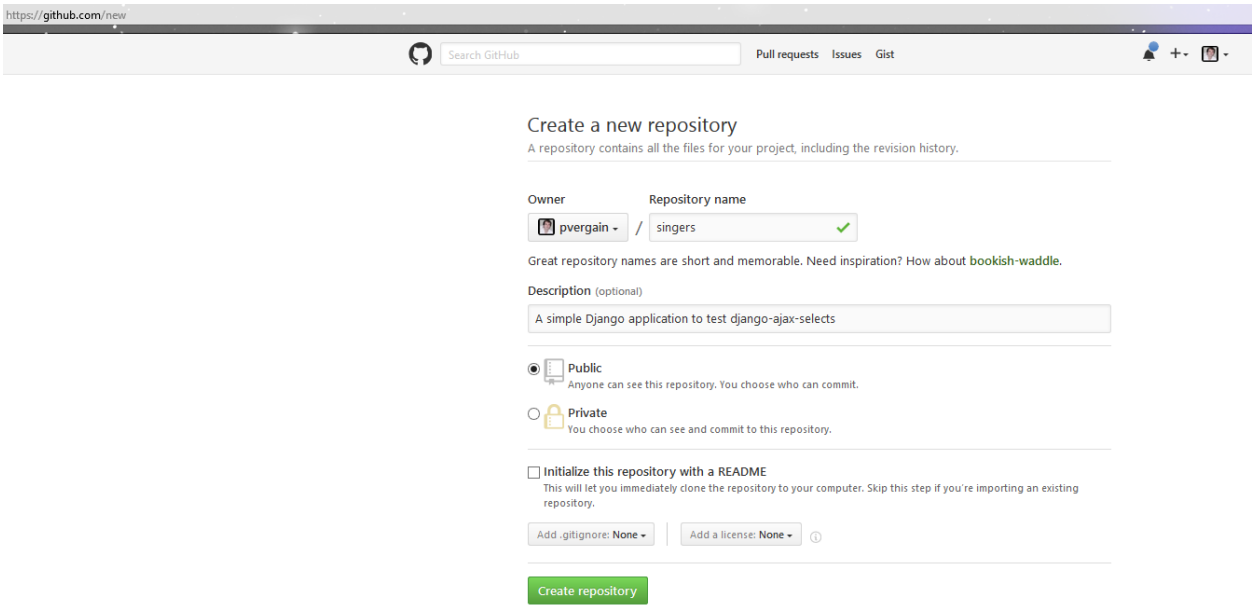
### Create new project

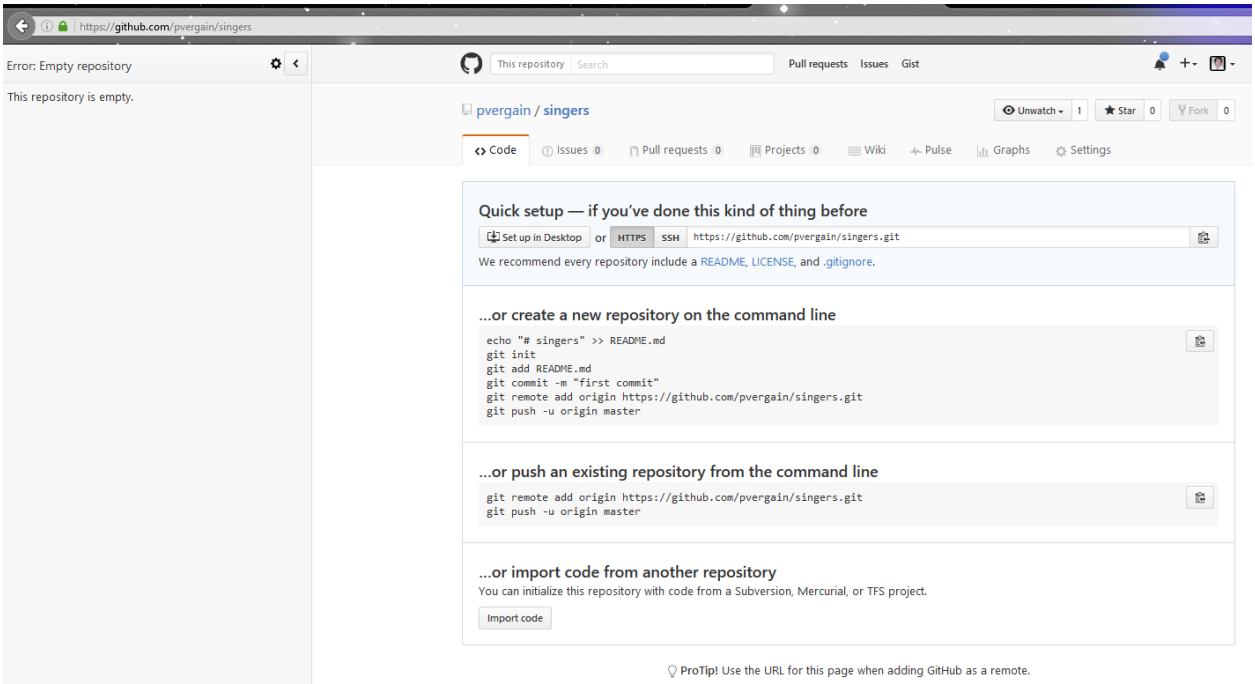
### Create singers project

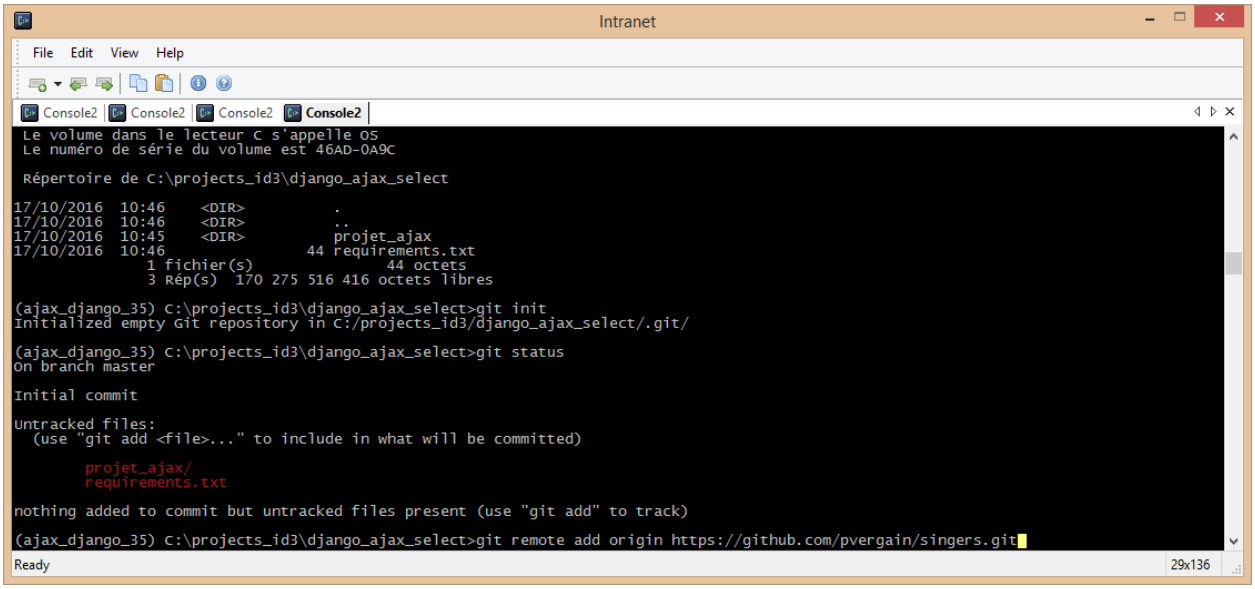
### Git remote

**See also:**

https://swcarpentry.github.io/git-novice/07-github/


https://github.com/new
Search GitHub
Pull requests
Issues
Gist
Create a new repository
A repository contains all the files for your project, including the revision history.
Owner
Repository name
pvergain
singers
Great repository names are short and memorable. Need inspiration? How about bookish-waddle.
Description (optional)
A simple Django application to test django-ajax-selects
Public
Anyone can see this repository. You choose who can commit.
Private
You choose who can see and commit to this repository.
Initialize this repository with a README
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.
Add .gitignore: None
Add a license: None
Create repository


https://github.com/pvergain/singers
Error: Empty repository
This repository is empty.
This repository
Search
Pull requests
Issues
Gist
pvergain / singers
Unwatch
1
Star
0
Fork
0
Code
Issues 0
Pull requests 0
Projects 0
Wiki
Pulse
Graphs
Settings
Quick setup — if you've done this kind of thing before
Set up in Desktop
or
HTTPS
SSH
https://github.com/pvergain/singers.git
We recommend every repository include a README, LICENSE, and .gitignore.
...or create a new repository on the command line
echo "# singers" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/pvergain/singers.git
git push -u origin master
...or push an existing repository from the command line
git remote add origin https://github.com/pvergain/singers.git
git push -u origin master
...or import code from another repository
You can initialize this repository with code from a Subversion, Mercurial, or TFS project.
Import code
ProTip! Use the URL for this page when adding GitHub as a remote.




# Singers's Python virtualenv

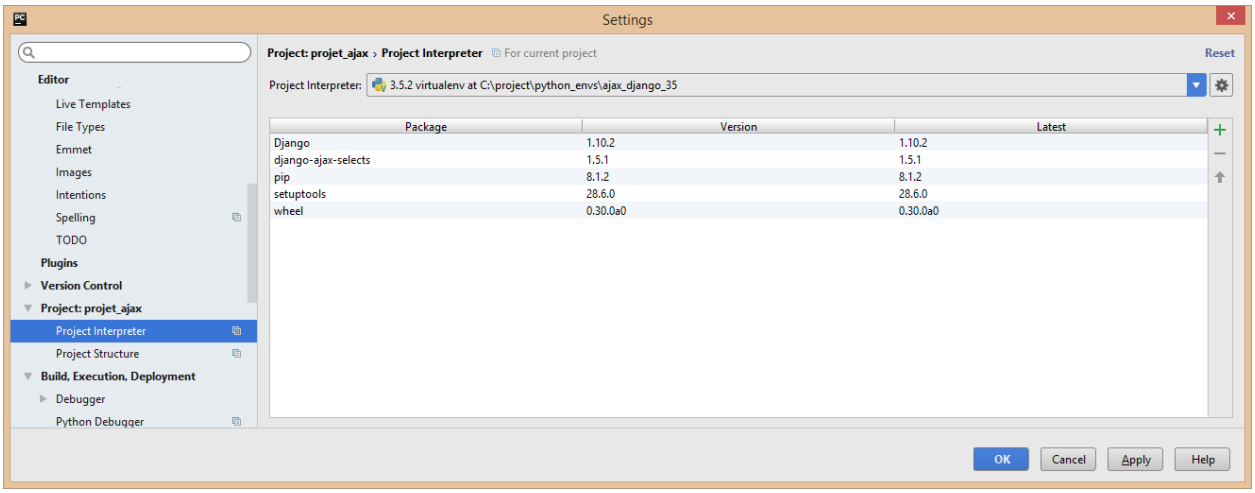
**See also:**

- https://swcarpentry.github.io/git-novice/

**Contents**



## ajax_django35 Python 3.5 virtualenv



# Django

## Django packages

## Django tips

**See also:**

- https://gist.github.com/kyle-eshares/0c30d81c4f8f201a41b916a2dd864b2c#file-values-py
- https://medium.com/@hansonkd/performance-problems-in-the-django-orm-1f62b3d04785#.qoy9oemyt

```
>>> Book.objects.values('title', 'author__name')
<QuerySet [{'author__name': u'Nikolai Gogol', 'title': u'The Overcoat'}, {'author__
↪name': u'Leo Tolstoy', 'title': u'War and Peace'}]>

#  Retrieve values as a tuple
>>> Book.objects.values_list('title', 'author__name')
<QuerySet [(u'The Overcoat', u'Nikolai Gogol'),
(u'War and Peace', u'Leo Tolstoy')]>
>>> Book.objects.values_list('title')
<QuerySet [(u'The Overcoat',), (u'War and Peace',)]>

#  With one value, it is easier to flatten the list
>>> Book.objects.values_list('title', flat=True)
<QuerySet [u'The Overcoat', u'War and Peace']>
```

Solving Performance Problems in the Django ORM

Django is a wonderful tool which has helped thousands boost their productivity when writing web applications. Like any framework, when you first start out and the data model is simple, things are speedy. When you start adding real-world constraints and the data-model grows in complexity you'll probably find that the same strategies you used in the beginning won't be as effective. As you learn more about your problem domain, you need to adapt your code accordingly.

All frameworks require upfront knowledge about how the internals work in order to write high performance code. Django is fast, but sometimes it allows you to unwittingly write slow code.

What to look for

With something as complex like a web application, it is hard to know where to start.

A bottom-up technique to working with the data, starting from where the data lives to how it is displayed, gives a pragmatic approach to debugging performance problems.

The vast majority of performance problems in a web application are related to accessing the database. Unless you are dealing with large amounts of data and know what you are doing, don't approach the problem thinking about what the Big-O notation is for your View. The overhead of a database call will dwarf the overhead of loops and template rendering. Without first addressing how you are using the database, you can't move on to fixing other problems.

Django has a detailed summary of how to optimize working with the database. The ORM is large and strategies are needed to build efficient code from the beginning.

When approaching optimization, code can often become unclear. If faced with a choice between a small performance gain or clear code, understandable code should always come first. It takes practice to know where to place the threshold.

The first step to fixing a problem is being able to identify it. When dealing with the ORM, there are a few things you can do. Understand django.db.connection, which records the queries made with the current connection.

This can be cumbersome and as you make more and more queries, it can be hard to digest the information.

In the shell, use django-exension's shell_plus command with the –print-sql flag turned on.

With the server, a middleware should run in the background of your DEBUG environment and log queries and point out duplicates.

Django-debug-toolbar provides this information in the page itself.

For our examples, we will use a classic author/book schema. Unexpected Queries

When checking the existence of an author for a book or grabbing the id of the author, it is tempting to want to use the author field directly.

It's tricky, but if the author object isn't needed, you potentially made an extra query for nothing. If you use the author value later, it doesn't have an impact. A good habit is to play it safe and always use the column name attribute. Size and Existence

Knowing when to use count and exists is tough. Django caches querysets, so when you are using the data of the queryset, use the built-in python operations. When not using the data, use the queryset methods.

The same holds true when you need the size of the queryset. If you are using the queryset, use len. If you only need the size use count.

Getting only what you need

By default, Django requests all the managed columns of the table and populates a Python object. When you only need a subset of columns from the table, consider using values and values_list. These methods skip the step of creating a complex python object and instead use dicts, tuples or even plain values. They can even handle getting columns through relationships straight forward.

Handling many rows

When you evaluate a queryset, Django caches the values. This makes sense if you will iterate over the queryset multiple times, but it doesn't make sense in an instance where you loop through once.

What Django actually does is load all books into memory and then iterate through each one. We want Django to hold the SQL connection open and read each row and call do_stuff before reading the next row. iterator to the rescue!

An added benefit of the iterator, is that it allows you to write linear data like table or a CSV as a stream. You can write a file or serve content to a user incrementally.

This is especially powerful when combined with values and values_list, because it keeps the minimum amount of information in memory as possible.

Using iterator also comes in handy during migrations where you need to mutate every row in a table. Just because the migration isn't client facing doesn't mean you should slack when it comes to efficiency. A long running migration could mean transaction locks or downtime.

Relationship Problems

Django's ORM allows you to interact with a relational database in a way that feels natural for the Object-Oriented Python programming language.

The code is precise code and semantically clear. Django uses lazy loading to only load the author if you need it. This is great, but can lead to an explosion of queries.

Django recognizes the problem and provides select_related and prefetch_related to solve it.

Using prefetch_related and select_related is critical when writing a Django app.

A caveat to prefetch_related is that if you plan on filtering the related queryset (author.books.filter(..)) the cache populated by prefetch_related won't be used and you will have to use a Prefetch object. Sometimes things can get complex and you might better off making 2 queries, one for the parents and one for the children, and then grouping

the children by the parent. If your prefetch plans get too complicated, evaluate how much you value the speed boost compared to making slightly less efficient but clearer code.

When select_related doesn't help

It is tempting to throw a select_related on everything, but there are certain circumstances that don't make sense. See the following result of the query. id in python gives you the unique id of an object instance. Objects with the same id value are the same instance.

You can be doing more work than needed. select_related creates a new instance for each row of the query, consuming memory. SQL also returns duplicate information for each row. If you are making a query where all the values of your select_related are the same, you'll want to use something else. Use related querysets or flip the query and use prefetch_related.

With the related queryset author.books.all(), Django caches the value of the author for each book using the same instance of the already queried value.

A hidden side affect is that if you use select_related and alter an author instance, that change won't propagate to the other authors in the queryset (even if they represent the same row) because they are different instances in python memory. With related querysets, the changes will propagate. Easier doesn't always mean better

Django makes following relationships too easy. This results in functions that cannot manage their own side-effects. When you pass in a model instance to a function and use a relationship, it is practically impossible to know if the relationship has already been fetched.

Will either author_name_length or process_author_books make a query? We can't tell. The relationship features of the ORM are so enticing that it's natural that we would want to write code this way. Using these functions without a select_related or prefetch_related in a loop can accidentally result in hundreds of queries. Django will happily make the queries without saying a peep. It is up to you to monitor your SQL logs and the callers of the function to figure out if it should pre-fetch or not.

We can rewrite our functions to be explicit by passing in a flattened data-structure that isn't a model (like a namedtuple), but we shouldn't have to think about it.

How do we fix it?

Knowing that we have this problem, how can we extend Django to be more explicit about resource consumption? Many database wrappers have solved this in different ways. In Ecto, the DB wrapper for Elixir, an unfetched relationship returns a Ecto.Association.NotLoaded struct instead of implicitly making the query.

Lets imagine a version of Django that implemented this behavior in a pythonic way.

An implementation can be relatively minimal.

There isn't a one size fits all answer to utilizing the ORM. Most of the time the performance gains for small apps won't make much of a difference. You should first seek to make your code clear and then work on optimizing it.

As your app grows, it is important to practice good hygiene when working with the ORM. Developing good habits now regarding consumption of resources will lead to big benefits later.

Optimization is a lot to handle, but a few simple rules can go a long way.

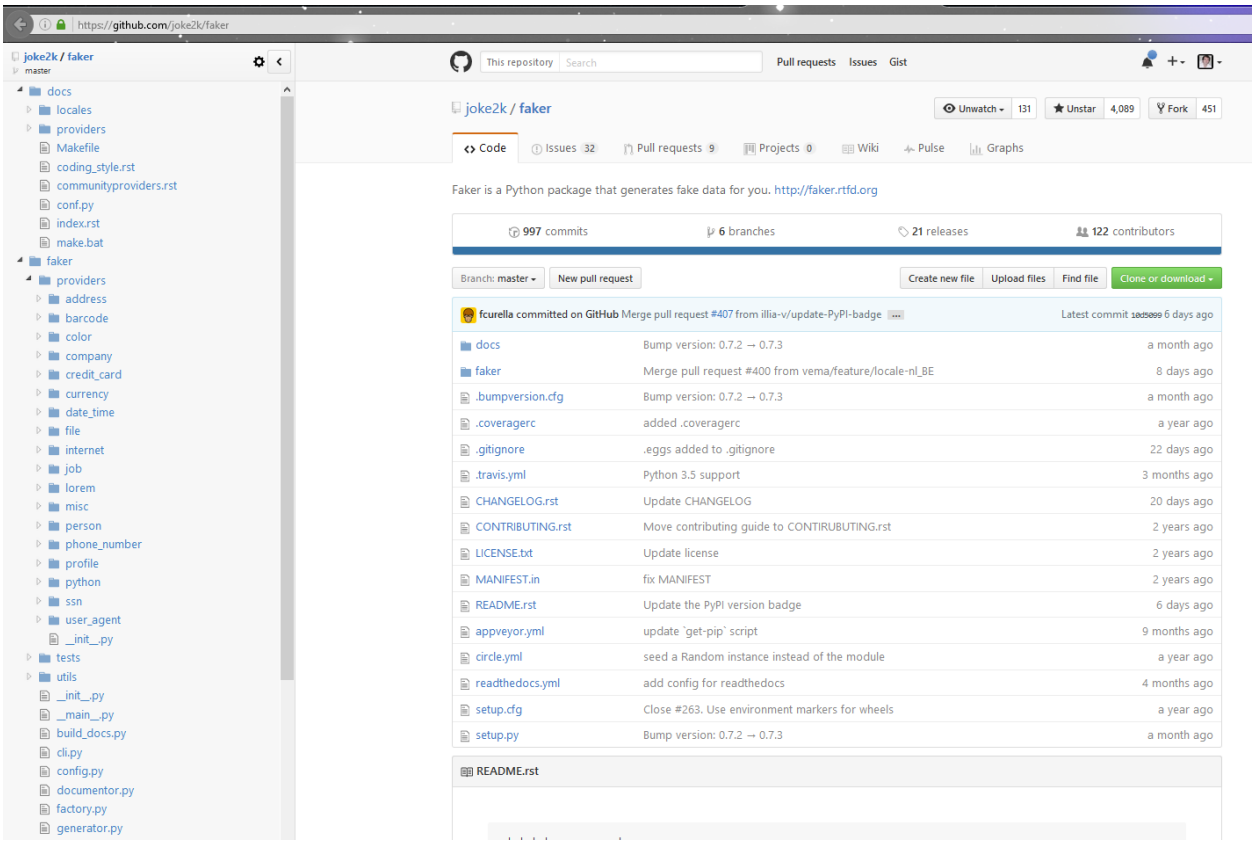
- Make a habit of isolating code and recording the queries it produces
- Queries should not be in loops
- Understand how the ORM caches data
- Know when Django will make a query
- Don't over-optimize at the expense of clarity

# Python modules

## Python faker module

**See also:**

- https://github.com/joke2k/faker
- https://faker.readthedocs.io/en/master/
- http://blog.districtdatalabs.com/a-practical-guide-to-anonymizing-datasets-with-python-faker




### Description

Faker is a Python package that generates fake data for you. Whether you need to bootstrap your database, create good-looking XML documents, fill-in your persistence to stress test it, or anonymize data taken from a production service, Faker is for you.

### install

```
(django_test_autocomplete_35_64) C:\projects_id3\django-test-autocomplete>pip install␣
↪faker
```

```
Collecting faker
  Downloading Faker-0.7.3-py2.py3-none-any.whl (529kB)
```

```
    100% |################################| 532kB 1.2MB/s
Requirement already satisfied (use --upgrade to upgrade): six in c:\project\python_
↪envs\django_test_autocomplete_35_64\lib\site-packages (from faker)
Requirement already satisfied (use --upgrade to upgrade): python-dateutil>=2.4 in␣
↪c:\project\python_envs\django_test_autocomplete_35_64\lib\site-packages (from faker)
Installing collected packages: faker
Successfully installed faker-0.7.3
```

### Test

**See also:**

- https://docs.djangoproject.com/eb/dev/ref/contrib/auth/
- https://faker.readthedocs.io/en/master/providers/faker.providers.internet.html
- https://faker.readthedocs.io/en/master/providers/faker.providers.person.html

Les attributs principaux de l'utilisateur par défaut sont :

- username
- password
- email
- first_name
- last_name

```
(django_test_autocomplete_35_64) C:\projects_id3\django-test-autocomplete>faker␣
↪address
```

```
042 Wilson Isle
Port Alan, MT 63548-2890
```

```
from faker import Factory

for lang in ['dk_DK', 'en_GB', 'fa_IR', 'it_IT', 'es_ES']:
    fake=Factory.create(lang)
    for _ in range(20):
        first_name = fake.first_name_female()
        last_name = fake.last_name()
        username = '{}_{}'.format(first_name, last_name)
        email =  '{}_{}@{}'.format(first_name,
                                   last_name,
                                   fake.free_email_domain())
        female=User.objects.get_or_create(username=username,
                                          first_name=first_name,
                                          last_name=last_name,
                                          email=email,
                                          password='1234')
```

```
In [18]: User.objects.count()
Out[18]: 11

In [19]:     fake=Factory.create('fr_FR')
```

```
    ...:     for _ in range(20):
    ...:         first_name = fake.first_name_female()
    ...:         last_name = fake.last_name()
    ...:         username = '{}_{}'.format(first_name, last_name)
    ...:         email =  '{}_{}@{}'.format(first_name,
    ...:                                    last_name,
    ...:                                    fake.free_email_domain())
    ...:         female=User.objects.get_or_create(username=username,
    ...:                                           first_name=first_name,
    ...:                                           last_name=last_name,
    ...:                                           email=email,
    ...:                                           password='1234')
    ...:

In [20]: User.objects.count()
Out[20]: 31


for lang in ['dk_DK', 'en_GB', 'fa_IR', 'it_IT', 'es_ES']:
    fake=Factory.create(lang)
    for _ in range(20):
        first_name = fake.first_name_male()
        last_name = fake.last_name()
        username = '{}_{}'.format(first_name, last_name)
        email =  '{}_{}@{}'.format(first_name,
                                   last_name,
                                   fake.free_email_domain())
        female=User.objects.get_or_create(username=username,
                                          first_name=first_name,
                                          last_name=last_name,
                                          email=email,
                                          password='1234')
```

```
In [24]:     for lang in ['dk_DK', 'en_GB', 'fa_IR', 'it_IT', 'es_ES']:
    ...:         fake=Factory.create(lang)
    ...:         for _ in range(20):
    ...:             first_name = fake.first_name_female()
    ...:             last_name = fake.last_name()
    ...:             username = '{}_{}'.format(first_name, last_name)
    ...:             email =  '{}_{}@{}'.format(first_name,
    ...:                                        last_name,
    ...:                                        fake.free_email_domain())
    ...:             female=User.objects.get_or_create(username=username,
    ...:                                               first_name=first_name,
    ...:                                               last_name=last_name,
    ...:                                               email=email,
    ...:                                               password='1234')
    ...:

In [25]: User.objects.count()
Out[25]: 151
```

```
In [26]:     for lang in ['dk_DK', 'en_GB', 'fa_IR', 'it_IT', 'es_ES']:
    ...:         fake=Factory.create(lang)
    ...:         for _ in range(20):
    ...:             first_name = fake.first_name_male()
    ...:             last_name = fake.last_name()
```

```
    ...:             username = '{}_{}'.format(first_name, last_name)
    ...:             email =  '{}_{}@{}'.format(first_name,
    ...:                                        last_name,
    ...:                                        fake.free_email_domain())
    ...:             female=User.objects.get_or_create(username=username,
    ...:                                               first_name=first_name,
    ...:                                               last_name=last_name,
    ...:                                               email=email,
    ...:                                               password='1234')
    ...:

In [27]: User.objects.count()
Out[27]: 251
```

```
from faker import Factory
for lang in ['nl_NL', 'cs_CZ', 'el_GR', 'fi_FI', 'sv_SE']:
    fake=Factory.create(lang)
    for _ in range(20):
        first_name = fake.first_name_female()
        last_name = fake.last_name()
        username = '{}_{}'.format(first_name, last_name)
        email =  '{}_{}@{}'.format(first_name,
                                   last_name,
                                   fake.free_email_domain())
        female=User.objects.get_or_create(username=username,
                                          first_name=first_name,
                                          last_name=last_name,
                                          email=email,
                                          password='1234')
```

## 2016-10-21 Receive 2 new excellent books about jQuery: `jQuery In Action` and `jquery UI in Action`

**See also:**

- https://twitter.com/tjvantoll
- https://twitter.com/AurelioDeRosa
- http://developer.telerik.com/featured/is-jquery-still-relevant/
- https://trends.builtwith.com/javascript/jQuery

**Contents**

    * *clone*
  - *jQuery UI in Action*
  - *jquery plugins*

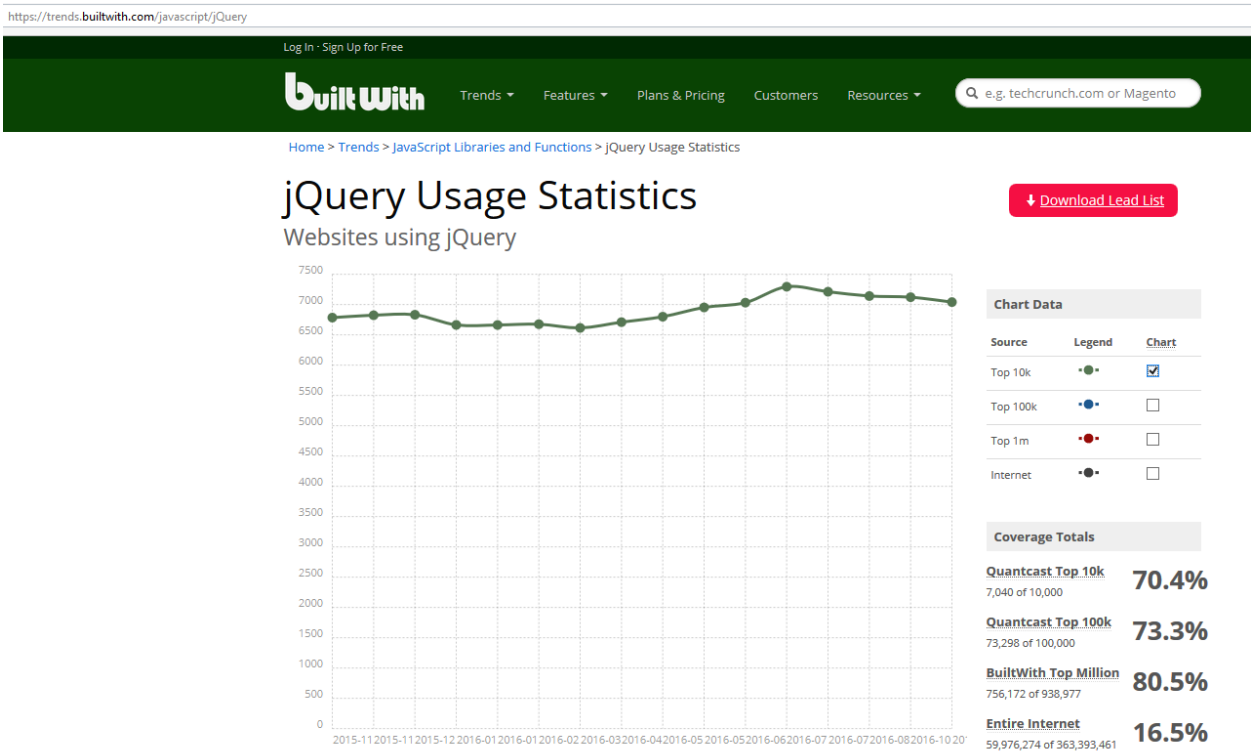



Fig. 4.1: https://trends.builtwith.com/javascript/jQuery

## If you were starting a new web app tomorrow, would you use jQuery ?

**See also:**

- https://twitter.com/tjvantoll
- https://twitter.com/AurelioDeRosa
- http://developer.telerik.com/featured/is-jquery-still-relevant/

### Aurelio De Rosa

```
I don't use it by default anymore, but it also depends on the project and the browsers
I have to support. These days I don't have to support IE8 anymore, but I'm
still maintaining some code bases developed a couple of years ago that needed
to support IE8. So, I still see jQuery around from time to time.

I think we can all agree that jQuery is not the tool for building SPAs, but
to be honest, it was never conceived for that.
```

#### TJ VanToll

```
Exactly. I still find jQuery to provide the exact mix of functionality I
need for little sites like that.
I would probably not, however, use jQuery if I were starting on a large complex
app nowadays. I think modern frameworks like Angular 2 just provide better
paradigms for building apps at that scale.
```

### jQuery in Action (third edition)

**See also:**

- https://github.com/AurelioDeRosa/jquery-in-action
- https://twitter.com/AurelioDeRosa
- https://www.manning.com/books/jquery-in-action-third-edition

#### clone

```
git clone https://github.com/AurelioDeRosa/jquery-in-action
```

### jQuery UI in Action

**See also:**

- https://github.com/tjvantoll/jquery-ui-in-action-demos
- https://twitter.com/tjvantoll
- https://twitter.com/jenlooper
- https://www.manning.com/books/jquery-ui-in-action

### jquery plugins

#### jquery plugins

#### jquery_ui_autocomplete_scroll plugin

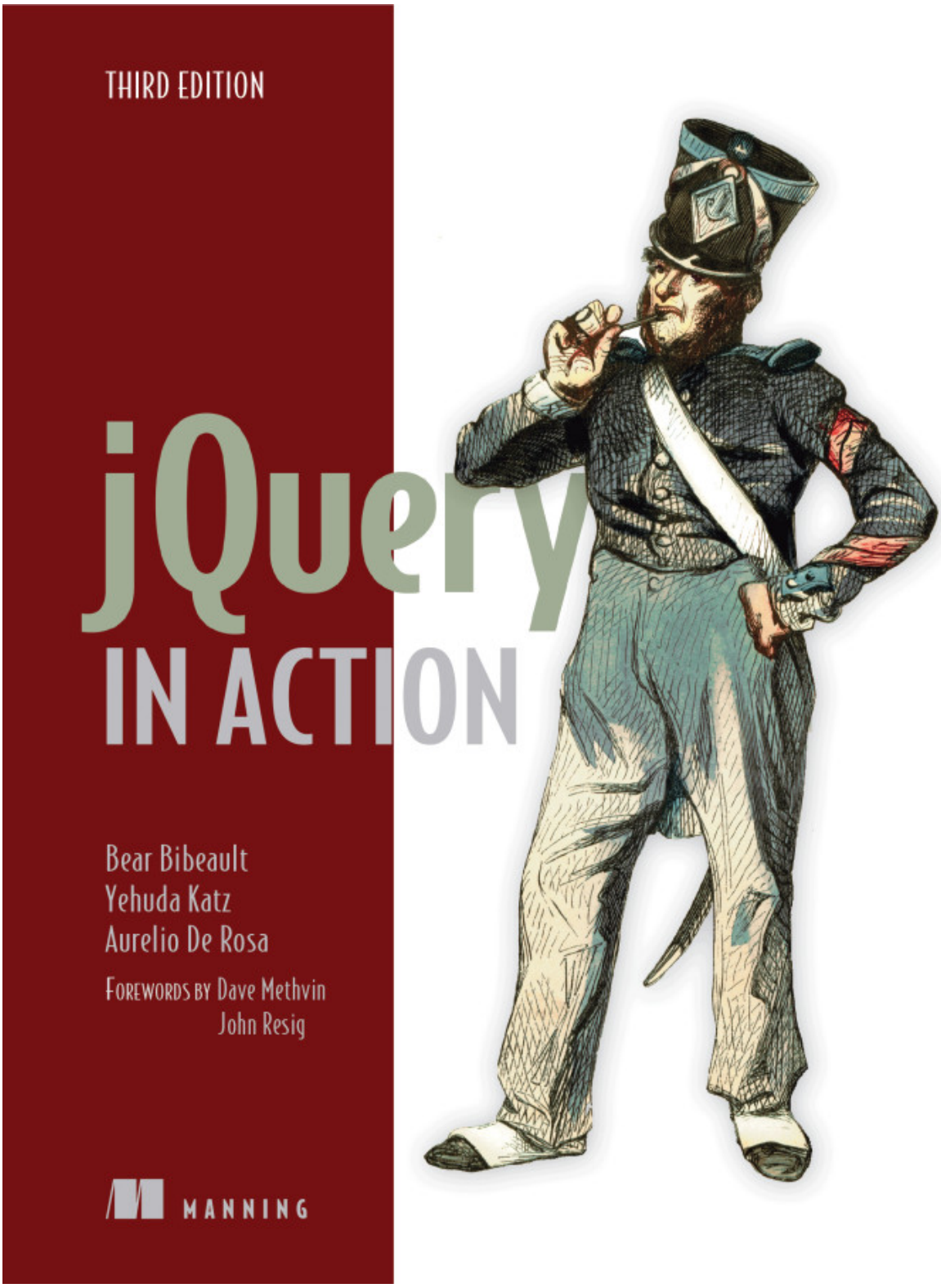
**See also:**

- https://github.com/anseki
- https://anseki.github.io/jquery-ui-autocomplete-scroll/
- https://github.com/anseki/jquery-ui-autocomplete-scroll

#### jquery select2 plugin

**See also:**

- https://github.com/select2/select2


THIRD EDITION
jQuery
IN ACTION
Bear Bibeault
Yehuda Katz
Aurelio De Rosa
FOREWORDS BY Dave Methvin
John Resig
MANNING

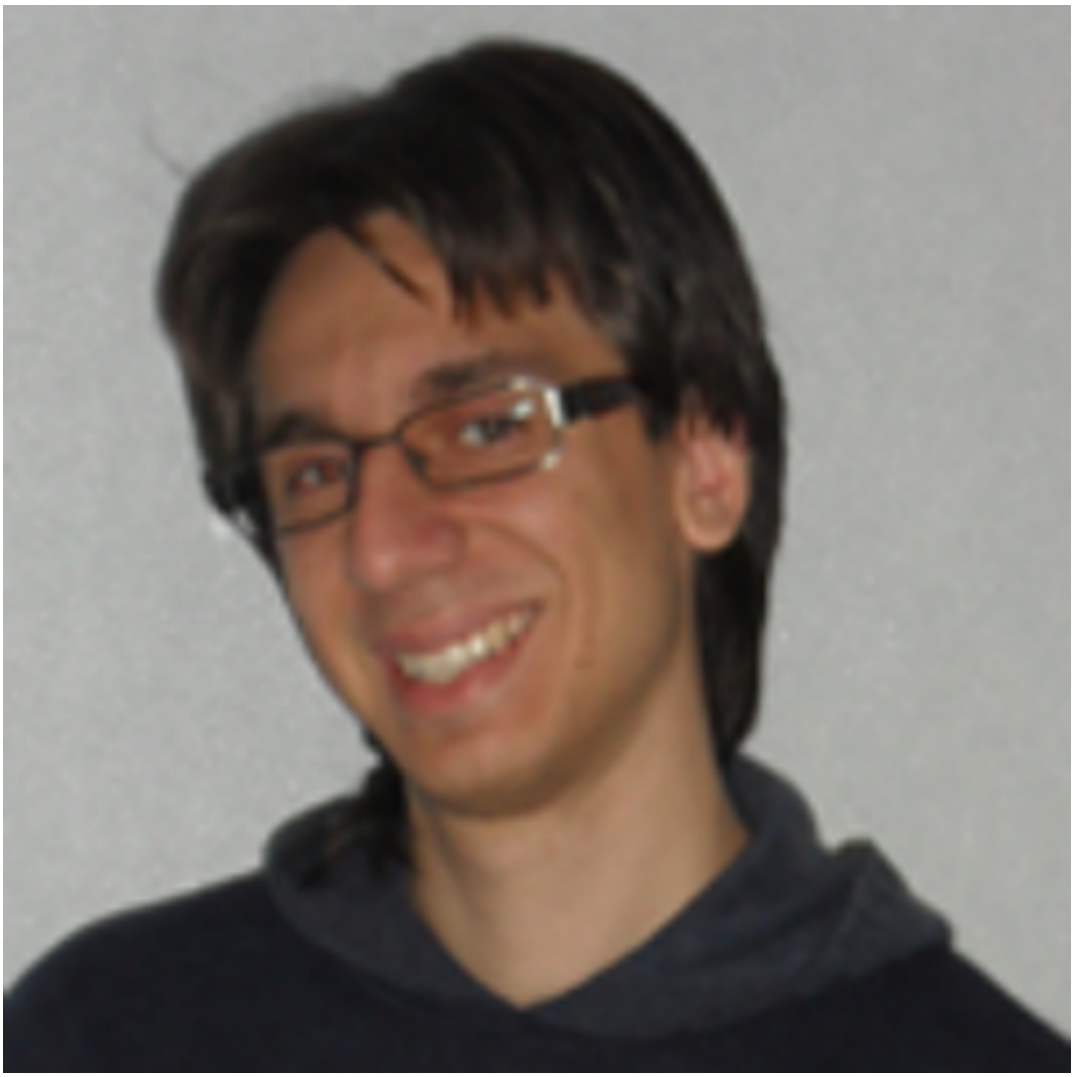
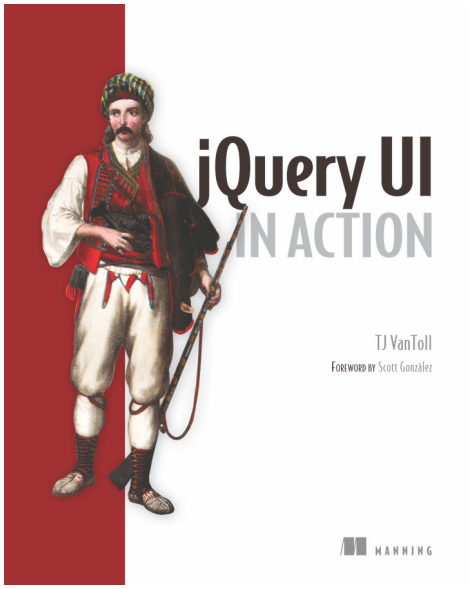


Fig. 4.2: Aurelio De Rosa (https://twitter.com/AurelioDeRosa)




- https://select2.github.io/announcements-4.0.html

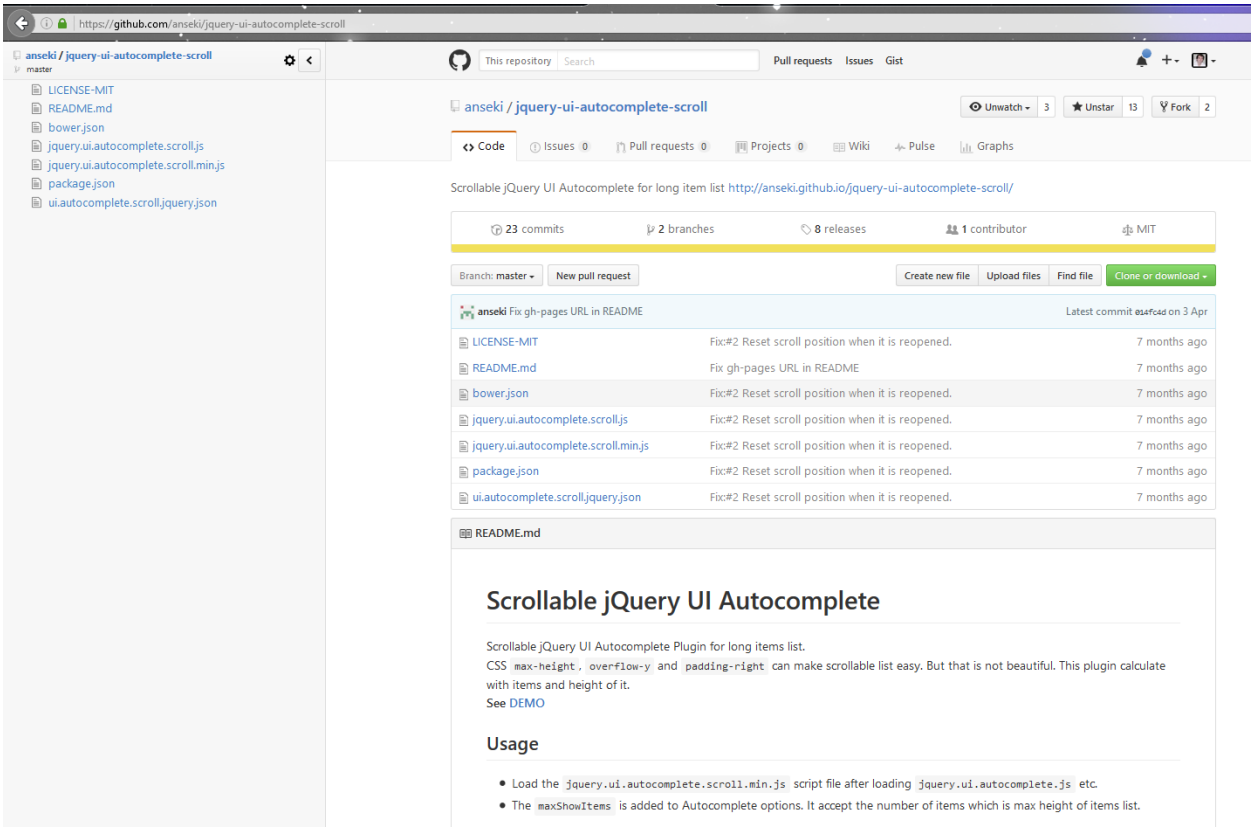
**Contents**

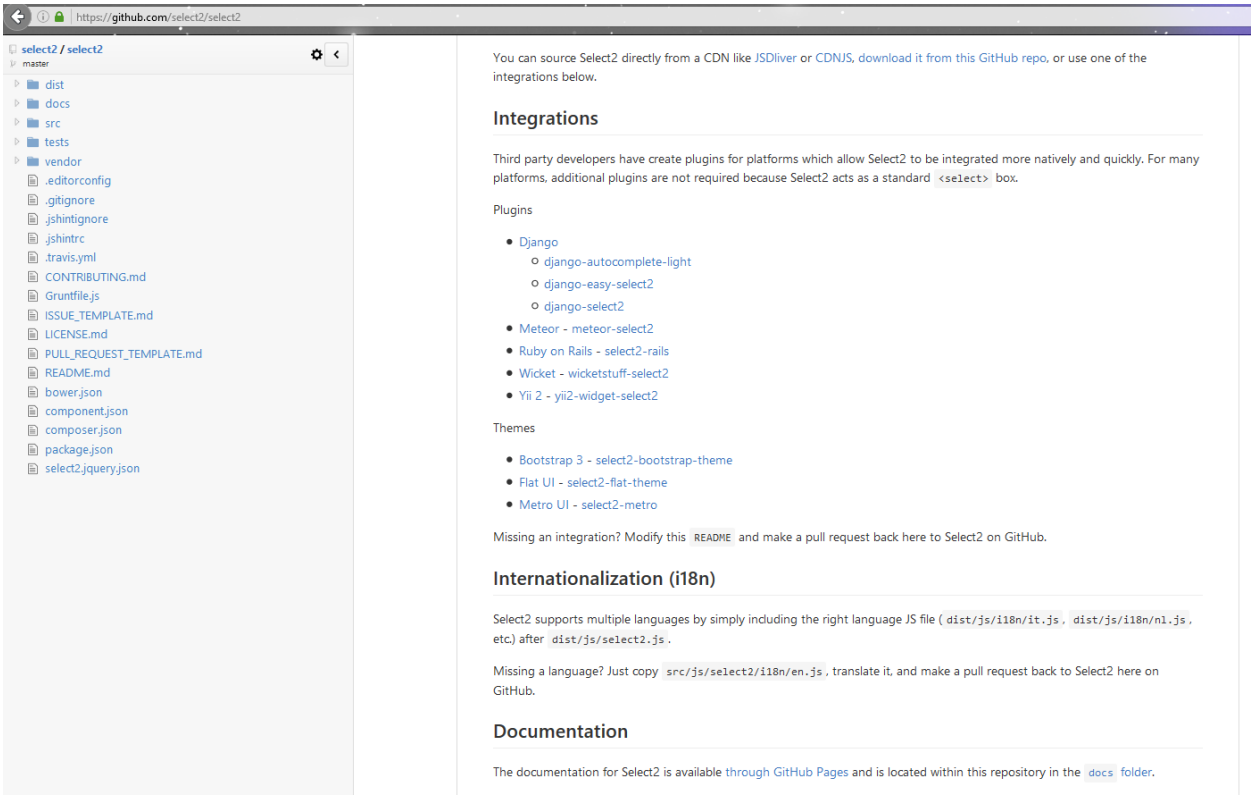


### Use cases

- Enhancing native selects with search.
- Enhancing native selects with a better multi-select interface.
- Loading data from JavaScript: easily load items via AJAX and have them searchable.
- Nesting optgroups: native selects only support one level of nesting. Select2 does not have this restriction.
- Tagging: ability to add new items on the fly.
- Working with large, remote datasets: ability to partially load a dataset based on the search term.
- Paging of large datasets: easy support for loading more pages when the results are scrolled to the end.
- Templating: support for custom rendering of results and selections.

Fig. 4.3: TJ Vantoll (https://twitter.com/tjvantoll)


https://github.com/anseki/jquery-ui-autocomplete-scroll
anseki / jquery-ui-autocomplete-scroll
master
LICENSE-MIT
README.md
bower.json
jquery.ui.autocomplete.scroll.js
jquery.ui.autocomplete.scroll.min.js
package.json
ui.autocomplete.scroll.jquery.json
This repository
Search
Pull requests
Issues
Gist
Unwatch 3
Unstar 13
Fork 2
Code
Issues 0
Pull requests 0
Projects 0
Wiki
Pulse
Graphs
Scrollable jQuery UI Autocomplete for long item list http://anseki.github.io/jquery-ui-autocomplete-scroll/
23 commits
2 branches
8 releases
1 contributor
MIT
Branch: master
New pull request
Create new file
Upload files
Find file
Clone or download
anseki Fix gh-pages URL in README
Latest commit e14fc4d on 3 Apr
LICENSE-MIT Fix:#2 Reset scroll position when it is reopened. 7 months ago
README.md Fix gh-pages URL in README 7 months ago
bower.json Fix:#2 Reset scroll position when it is reopened. 7 months ago
jquery.ui.autocomplete.scroll.js Fix:#2 Reset scroll position when it is reopened. 7 months ago
jquery.ui.autocomplete.scroll.min.js Fix:#2 Reset scroll position when it is reopened. 7 months ago
package.json Fix:#2 Reset scroll position when it is reopened. 7 months ago
ui.autocomplete.scroll.jquery.json Fix:#2 Reset scroll position when it is reopened. 7 months ago
README.md
Scrollable jQuery UI Autocomplete
Scrollable jQuery UI Autocomplete Plugin for long items list.
CSS max-height, overflow-y and padding-right can make scrollable list easy. But that is not beautiful. This plugin calculate with items and height of it.
See DEMO
Usage
Load the jquery.ui.autocomplete.scroll.min.js script file after loading jquery.ui.autocomplete.js etc.
The maxShowItems is added to Autocomplete options. It accept the number of items which is max height of items list.


https://github.com/select2/select2
select2 / select2
master
dist
docs
src
tests
vendor
.editorconfig
.gitignore
.jshintignore
.jshintrc
.travis.yml
CONTRIBUTING.md
Gruntfile.js
ISSUE_TEMPLATE.md
LICENSE.md
PULL_REQUEST_TEMPLATE.md
README.md
bower.json
component.json
composer.json
package.json
select2.jquery.json
You can source Select2 directly from a CDN like JSDliver or CDNJS, download it from this GitHub repo, or use one of the integrations below.
Integrations
Third party developers have create plugins for platforms which allow Select2 to be integrated more natively and quickly. For many platforms, additional plugins are not required because Select2 acts as a standard <select> box.
Plugins
Django
django-autocomplete-light
django-easy-select2
django-select2
Meteor - meteor-select2
Ruby on Rails - select2-rails
Wicket - wicketstuff-select2
Yii 2 - yii2-widget-select2
Themes
Bootstrap 3 - select2-bootstrap-theme
Flat UI - select2-flat-theme
Metro UI - select2-metro
Missing an integration? Modify this README and make a pull request back here to Select2 on GitHub.
Internationalization (i18n)
Select2 supports multiple languages by simply including the right language JS file (dist/js/i18n/it.js, dist/js/i18n/nl.js, etc.) after dist/js/select2.js.
Missing a language? Just copy src/js/select2/i18n/en.js, translate it, and make a pull request back to Select2 here on GitHub.
Documentation
The documentation for Select2 is available through GitHub Pages and is located within this repository in the docs folder.

CHAPTER 5

# Versions

**See also:**

- http://keepachangelog.com
- http://semver.org/
- https://github.com/pvergain/django-test-autocomplete.git

## Creating tags from the command line

To create a tag on your current branch, run this:

```
git tag <tagname>
```

This will create a local tag with the current state of the branch you are on. When pushing to your remote repo, tags are NOT included by default. You will need to explicitly say that you want to push your tags to your remote repo:

```
git push origin --tags
```

## Versions

### Version 0.2.0 (2016-10-27)

**See also:**

- http://keepachangelog.com
- https://github.com/pvergain/django-test-autocomplete

```
git tag 0.2.0
git push origin --tags
```

in doc_sphinx/conf.py:

```
# The short X.Y version.
version = '0.2.0'
# The full version, including alpha/beta/rc tags.
release = '0.2.0'
```

### Added

- the documentation is on read the docs (https://django-test-autocomplete.readthedocs.io/en/latest/)

### Changed

The django-autocomplete-light is adopted.

### Miscellaneous

### Fixed

### Test

## Version 0.1.0 (2016-10-21)

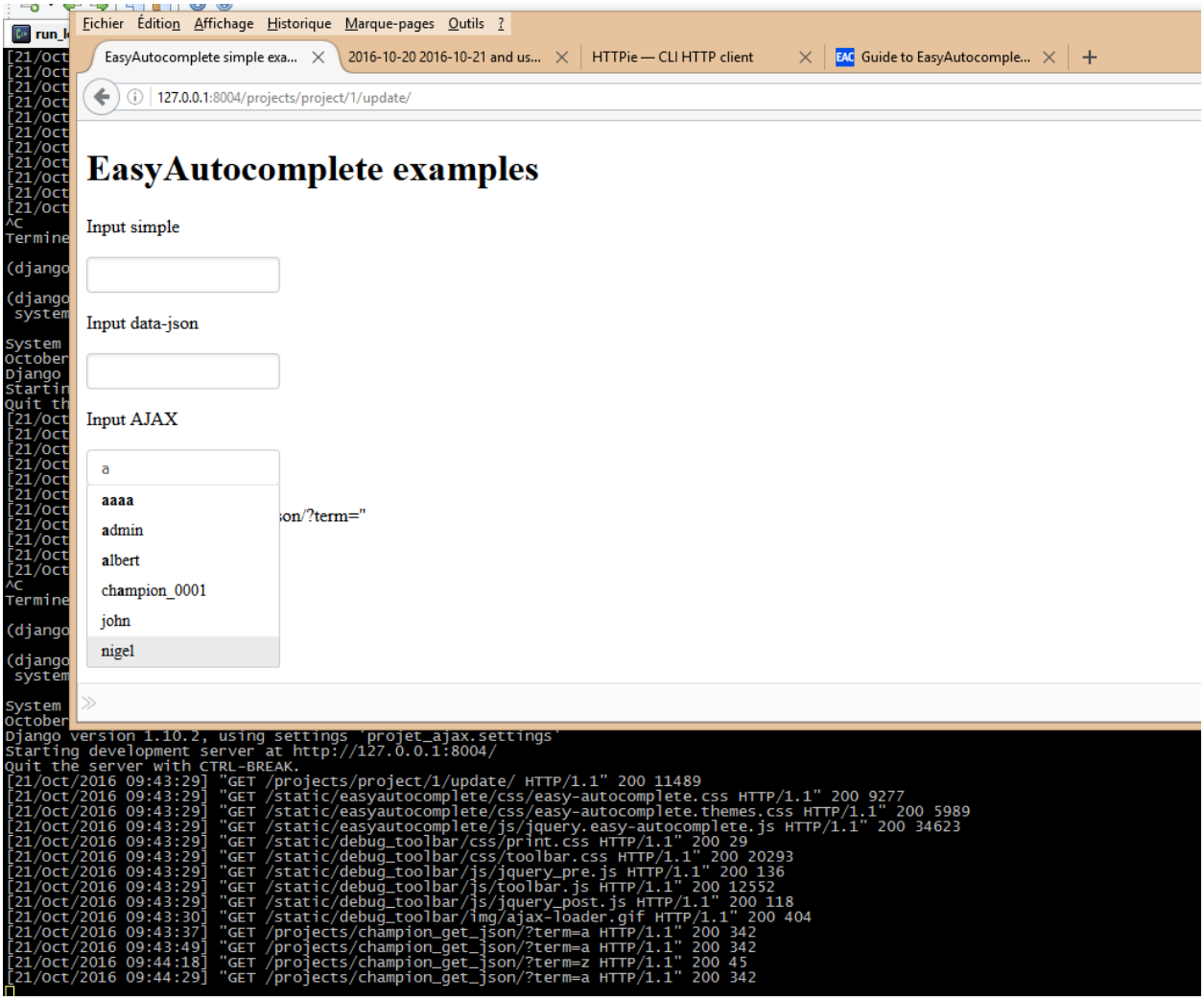
**See also:**

- http://keepachangelog.com
- https://github.com/pvergain/django-test-autocomplete/tree/0.1.0

```
git tag 0.1.0

(django_test_autocomplete_35_64) C:\projects_id3\django-test-autocomplete>git tag 0.1.
↪0

(django_test_autocomplete_35_64) C:\projects_id3\django-test-autocomplete>git push␣
↪origin --tags
Fatal: ArgumentException encountered.
"C:/Program Files (x86)/GitExtensions/GitCredentialWinStore/git-credential-winstore.
↪exe" get: C:/Program Files (x86)/GitExtensions/GitCredentialWinStore/git-credential-
↪winstore.exe: No such file or directory
Username for 'https://github.com': pvergain
Password for 'https://pvergain@github.com':
Fatal: ArgumentException encountered.
"C:/Program Files (x86)/GitExtensions/GitCredentialWinStore/git-credential-winstore.
↪exe" store: C:/Program Files (x86)/GitExtensions/GitCredentialWinStore/git-
↪credential-winstore.exe: No such file or directory
Total 0 (delta 0), reused 0 (delta 0)
To https://github.com/pvergain/django-test-autocomplete.git
 * [new tag]         0.1.0 -> 0.1.0
```

### Added

**See also:**


Fichier Édition Affichage Historique Marque-pages Outils ?
EasyAutocomplete simple exa...
2016-10-20 2016-10-21 and us...
HTTPie — CLI HTTP client
Guide to EasyAutocomple...
127.0.0.1:8004/projects/project/1/update/
EasyAutocomplete examples
Input simple
Input data-json
Input AJAX
a
aaaa
admin
albert
champion_0001
john
nigel
Django version 1.10.2, using settings 'projet_ajax.settings'
Starting development server at http://127.0.0.1:8004/
Quit the server with CTRL-BREAK.
[21/Oct/2016 09:43:29] "GET /projects/project/1/update/ HTTP/1.1" 200 11489
[21/Oct/2016 09:43:29] "GET /static/easyautocomplete/css/easy-autocomplete.css HTTP/1.1" 200 9277
[21/Oct/2016 09:43:29] "GET /static/easyautocomplete/css/easy-autocomplete.themes.css HTTP/1.1" 200 5989
[21/Oct/2016 09:43:29] "GET /static/easyautocomplete/js/jquery.easy-autocomplete.js HTTP/1.1" 200 34623
[21/Oct/2016 09:43:29] "GET /static/debug_toolbar/css/print.css HTTP/1.1" 200 29
[21/Oct/2016 09:43:29] "GET /static/debug_toolbar/css/toolbar.css HTTP/1.1" 200 20293
[21/Oct/2016 09:43:29] "GET /static/debug_toolbar/js/jquery_pre.js HTTP/1.1" 200 136
[21/Oct/2016 09:43:29] "GET /static/debug_toolbar/js/toolbar.js HTTP/1.1" 200 12552
[21/Oct/2016 09:43:29] "GET /static/debug_toolbar/js/jquery_post.js HTTP/1.1" 200 118
[21/Oct/2016 09:43:30] "GET /static/debug_toolbar/img/ajax-loader.gif HTTP/1.1" 200 404
[21/Oct/2016 09:43:37] "GET /projects/champion_get_json/?term=a HTTP/1.1" 200 342
[21/Oct/2016 09:43:49] "GET /projects/champion_get_json/?term=a HTTP/1.1" 200 342
[21/Oct/2016 09:44:18] "GET /projects/champion_get_json/?term=z HTTP/1.1" 200 45
[21/Oct/2016 09:44:29] "GET /projects/champion_get_json/?term=a HTTP/1.1" 200 342

- *Adding the AJAX call in the Django template*

This is the First version !

The best autocomplete jquery plugin for the non-admin Django is the jquery EasyAutocomplete plugin

### Changed

### Miscellaneous

### Fixed

### Test

CHAPTER 6

# Glossary

**AJAX** Ajax (also AJAX; /edæks/; short for asynchronous JavaScript and XML) is a set of web development techniques using many web technologies on the client-side to create asynchronous Web applications. With Ajax, web applications can send data to and retrieve from a server asynchronously (in the background) without interfering with the display and behavior of the existing page. By decoupling the data interchange layer from the presentation layer, Ajax allows for web pages, and by extension web applications, to change content dynamically without the need to reload the entire page.

In practice, modern implementations commonly substitute JSON for XML due to the advantages of being native to JavaScript.

Source: https://en.wikipedia.org/wiki/Ajax_(programming)

**JSON** JSON (canonically pronounced /desn/ JAY-sn;[1] sometimes JavaScript Object Notation) is an open-standard format that uses human-readable text to transmit data objects consisting of attribute–value pairs. It is the most common data format used for asynchronous browser/server communication, largely replacing XML which is used by AJAX.

JSON is a language-independent data format. It derives from JavaScript, but as of 2016, code to generate and parse JSON-format data is available in many programming languages. The official Internet media type for JSON is application/json. The JSON filename extension is .json.

Douglas Crockford originally specified the JSON format; two competing standards, RFC 7159 and ECMA-404, define it. The ECMA standard describes only the allowed syntax, whereas the RFC also provides some semantic and security considerations.

Source: https://en.wikipedia.org/wiki/JSON

# Index

## U



## V