
Django test AJAX autocomplete Documentation

Release 0.2.0

pvergain@gmail.com

Apr 05, 2017

1	Autocomplete introduction	5
1.1	Definitions	5
1.1.1	Definition 1	5
1.1.2	Definition 2	5
1.2	Arguments	6
2	Autocomplete jQuery plugins tests	7
2.1	Specification	7
2.2	Test the different jQuery plugings	8
2.2.1	django-autocomplete-light test	8
	install	10
	update the settings.py module	10
	tree	10
	projects/views_django_autocomplete_light.py	16
	projects/api/get_users/?q (projects/urls.py)	16
	test	17
	projects/forms_django_autocomplete_light.py	17
	Using autocompletes in the admin : nice !	17
	Using autocompletes outside the admin	17
	projects/views_django_autocomplete_light.py	17
	Django template	19
	Test URL http://127.0.0.1:8004/projects/project/1/update_django_autocomplete	22
	Passing options to select2	22
2.2.2	jQuery EasyAutocomplete test	24
	Running the local Django web server	24
	Testing the JSON API view	24
	Testing the form, step1	25
	projects/urls.py	25
	The forms.py part	25
	The HTML and JavaScript part	26
	Step2 : initialize the easyAutocomplete placeholder with the value of champion	27
	Step3 : hide the select champion form field	28
	Before the update	28
	After the update	28
2.2.3	jQuery UI autocomplete test	29
	Running the local Django web server	29
	The projects/forms.py module	30
	The HTML and JavaScript part (the Django Template)	31

	The projects/views.py module	32
	The projects/urls.py module	33
	The jQuery UI scrollbar module	33
3	Actions	35
3.1	Actions 2016	35
3.1.1	2016-10-25 installing gitsome	35
	install	36
3.1.2	2016-10-25 testing jquery-ui autocomplete and django-autocomplete-light	37
	Conclusion	37
3.1.3	2016-10-25 Building fake users with the Python faker module	37
	Motivation	37
3.1.4	2016-10-24 Learning jQuery 101	37
3.1.5	2016-10-21 Receive 2 new excellent books about jQuery: jQuery In Action and jquery UI in Action	37
3.1.6	2016-10-21 testing the jquery EasyAutocomplete (EAC) options for an AJAX call	37
	Renaming views	39
	Add a great number of projects into the 'project' table	39
	Write an API view/urls to select projects	39
	The view	39
	The URL	40
	Tests with httpie	40
	Update the Django template file in order to call this view	41
	6 elements is not enough	41
	Ajout placeholder	41
	list['onSelectedItemEvent']	41
	HTML5	44
	Javascript	44
	list['showAnimation']	44
	TODO : tester categories	44
3.1.7	2016-10-21 pushing documentation on readthedocs	45
	Connect to readthedocs	45
	Import a projet	45
	Advanced parameters	49
3.1.8	2016-10-20 2016-10-21 First step: using the jquery EasyAutocomplete plugin with Django	49
	Why ? EasyAutocomplete can easily become one of the best autocomplete plugins available for free	52
	EasyAutocomplete files tree (Javascript, CSS, JSON, PHP files)	52
	Javascript SCSS and CSS EasyAutocomplete files	55
	EasyAutocomplete-master/src/sass/easy-autocomplete.scss	55
	EasyAutocomplete-master/dist/easy-autocomplete.css	63
	EasyAutocomplete-master/dist/jquery.easy-autocomplete.js	69
	EasyAutocomplete simple example	100
	EasyAutocomplete-master/demo/example_simple.html	101
	EasyAutocomplete JSON example	102
	EasyAutocomplete-master/demo/example_json.html	102
	EasyAutocomplete-master/demo/resources/countries.json	103
	EasyAutocomplete flags example	105
	EasyAutocomplete-master/demo/example_flags.html	105
	EasyAutocomplete-master/demo/resources/flags.css	106
	EasyAutocomplete-master/demo/resources/flags.png	111
	EasyAutocomplete categories example	111
	EasyAutocomplete-master/demo/example_categories.html	111
	EasyAutocomplete-master/demo/resources/categories.json	113

EasyAutocomplete static_link example	113
EasyAutocomplete-master/demo/example_static_link.html	113
EasyAutocomplete-master/demo/resources/site.json	114
EasyAutocomplete email example	115
EasyAutocomplete-master/demo/example_email.html	115
EasyAutocomplete-master/demo/resources/people.json	116
EasyAutocomplete Django integration	120
Include JS and CSS files from the distribution (static files)	120
Exemples of static files in the Django World	121
The Django contrib module	121
Django_By_Example_Code/Chapter 1/mysite/blog	122
Django_By_Example_Code/Chapter 8	123
Django_By_Example_Code/Chapter 13	125
Django settings example	128
Integration in our Django test project	128
Create a simple update_easy_simple.html	129
Le fichier projects/templates/projects/projet/update_easy_simple.html	130
First problem : the libraries are not found	130
collectstatic	132
New tree	132
Create an easyautocomplete directory	133
Update the Django template file	134
OK it works !	135
Add JSON files	135
Adding the AJAX call in the Django template	135
URL : http://127.0.0.1:8004/projects/champion_get_json/?term=a	135
pip install httpie (clihttp)	135
http http://127.0.0.1:8004/projects/champion_get_json/?term=a	137
http http://easyautocomplete.com/api/countrySearch.php?phrase=co	137
Avec countrySearch.php	139
Avec Python/Django OK the first step is DONE	139
This is the happy end of the first step	141
3.1.9 2016-10-19 improve the Django templates + unobstrusive Javascript (step1)	141
The Javascript code must be in the the body element	142
Modify names	143
Test champion_get_json : http://127.0.0.1:8004/projects/champion_get_json/?term=a	143
Rendering fields manually	143
Adding django-debug-toolbar	143
First step for a better look	144
3.1.10 2016-10-18 improve the jquery-ui autocomplete look and feel	144
Last look	145
jquery-ui autocomplete options already used	148
Remote option	148
Try https://jqueryui.com/autocomplete/#combobox	149
Try https://jqueryui.com/autocomplete/#maxheight	149
Try https://jqueryui.com/autocomplete/#categories	150
Try https://jqueryui.com/autocomplete/#remote-jsonp	151
3.1.11 2016-10-18 try a more simple approach with jquery-ui autocomplete widget	153
Installation	154
Create a projects application	155
Add the Django model Project	155
Add the project in the INSTALLED_APPS	156
makemigrations	159
sqlmigrate	160

	migrate	160
	Create the projects/forms.py file and add 2 forms	161
	Create the projects/urls.py file and add the champion auto complete URL	161
	Add the projects.urls file to the root urls.py	162
	Add the ProjectUpdateView	163
	Create the templates/projects/project/update.html django template	165
	Add the jquery-ui javascript code	168
	Add the admin interface	168
	Create some projects with the admin interface	168
	Reading Project record from the commande line	168
	Add some users with the admin interface	171
	Add some users with the command line interface	171
	Try the URL : http://127.0.0.1:8000/projects/project/1/update	171
3.1.12	2016-10-18 create some Django Templates : base.html	172
	The Django Template templates/base.html	172
	The Django Template singers/templates/song/update.html	175
	Add the django-extensions and ipython modules	176
	New django commands	176
	Get the first song	177
	Try the URL : http://127.0.0.1:8000/singers/song/1/update	177
	Compared to the django admin interface	177
	Conclusion	179
3.1.13	2016-10-17 create a normal Django view	179
	Le fichier singers/models.py	179
	Le fichier singers/forms.py	181
	Le fichier singers/views.py	182
	Le fichier singers/urls.py	185
3.1.14	2016-10-17 create singers SQL tables	185
	makemigrations singers	186
	sqlmigrate singers 0001	186
	migrate singers	188
	show-migrations	188
	createsuperuser	189
	Connect to the admin interface	189
4	Django test AJAX autocomplete development	191
4.1	Github Singers's development	191
4.1.1	Create new project	191
4.1.2	Create singers project	191
4.1.3	Git remote	191
4.2	Singers's Python virtualenv	191
4.2.1	ajax_django35 Python 3.5 virtualenv	193
4.3	Django	194
4.3.1	Django packages	194
4.4	Python modules	194
4.4.1	Python faker module	194
	Description	194
	install	195
	Test	195
4.5	2016-10-21 Receive 2 new excellent books about jQuery: jQuery In Action and jquery UI in Action	197
4.5.1	If you were starting a new web app tomorrow, would you use jQuery ?	198
	Aurelio De Rosa	198
	TJ VanToll	199

4.5.2	jQuery in Action (third edition)	199
	clone	199
4.5.3	jQuery UI in Action	199
4.5.4	jquery plugins	199
	jquery plugins	199
	jquery_ui_autocomplete_scroll plugin	199
	jquery select2 plugin	202
5	Versions	205
5.1	Creating tags from the command line	205
5.2	Versions	205
5.2.1	Version 0.2.0 (2016-10-27)	205
	Added	206
	Changed	206
	Miscellaneous	206
	Fixed	206
	Test	206
5.2.2	Version 0.1.0 (2016-10-21)	206
	Added	206
	Changed	208
	Miscellaneous	208
	Fixed	208
	Test	208
6	Glossary	209

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/Ajax_(programming))
- <https://en.wikipedia.org/wiki/JSON>

Dango test AJAX autocomplete

Projet Dango test AJAX autocomplete

Release 0.2.0

Date Apr 05, 2017

Authors pvergain

- [genindex](#)
- [search](#)
- [Glossary](#)

Thanks <https://readthedocs.org/>

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

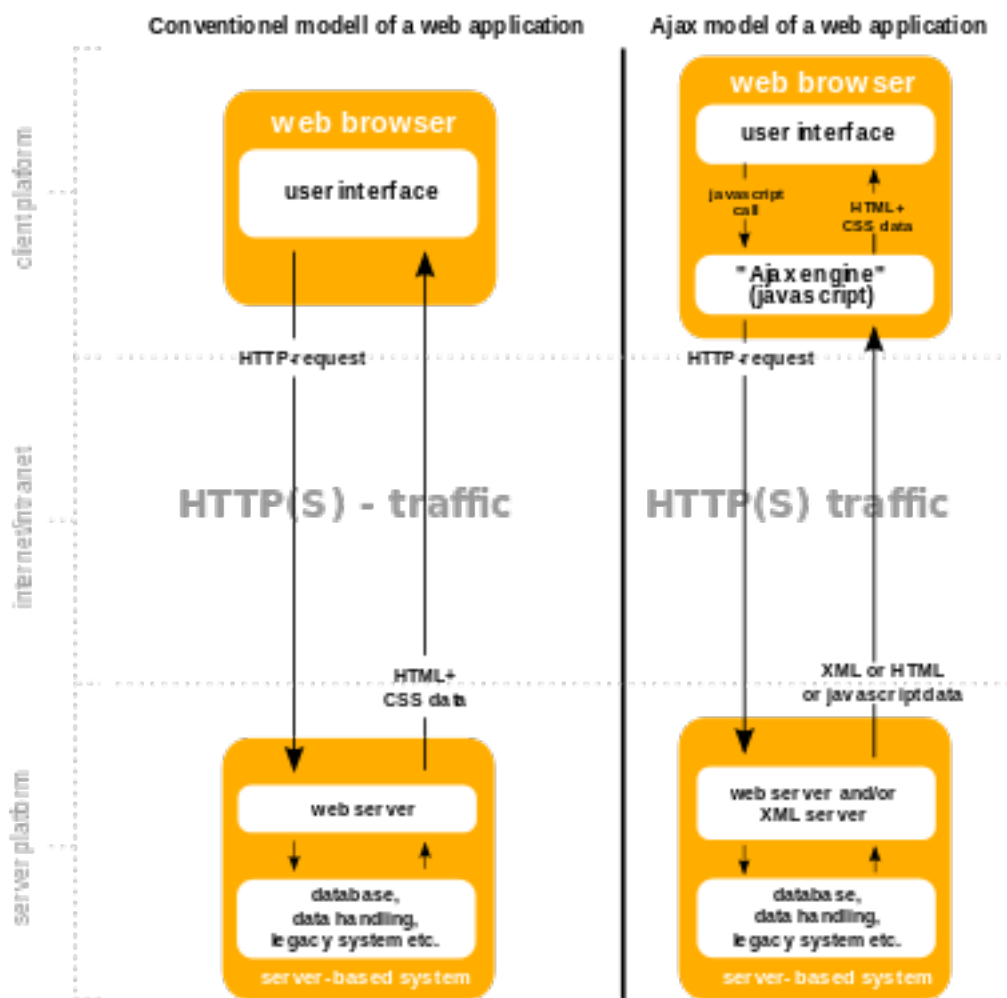


Fig. 1: *AJAX* source: [https://en.wikipedia.org/wiki/Ajax_\(programming\)](https://en.wikipedia.org/wiki/Ajax_(programming))



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



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

Autocomplete introduction

Contents

- *Autocomplete introduction*
 - *Definitions*
 - * *Definition 1*
 - * *Definition 2*
 - *Arguments*

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.

Autocomplete jQuery plugins tests

Contents

- *Autocomplete jQuery plugins tests*
 - *Specification*
 - *Test the different jQuery plugings*

Specification

Write a form for selecting ‘champion’ for a project.

```
1  #!/usr/bin/python
2  # -*- coding: utf8 -*-
3  """The project's models.
4
5  """
6  from __future__ import unicode_literals
7  from django.utils.encoding import python_2_unicode_compatible
8
9  from django.db import models
10 from django.core.urlresolvers import reverse
11
12 # https://docs.djangoproject.com/en/dev/ref/contrib/auth/#user-model
13 from django.contrib.auth.models import User
14
15
16
17 @python_2_unicode_compatible
18 class Project(models.Model):
19     """A project with a title and a champion which is the foreign key to the
20         auth user.
21
22     Documentation
23     =====
24
25     - http://guiqinqian.blogspot.fr/2012/01/using-jquery-auto-complete-in-django.html
26
27     """
```

```
28 title = models.CharField(max_length=200)
29 champion = models.ForeignKey(User)
30
31 def __str__(self):
32     return "{} {}".format(self.title, self.champion)
33
34 def get_absolute_url(self):
35     """
36     https://docs.djangoproject.com/en/dev/ref/class-based-views/generic-editing/
37     """
38     return reverse('projects:project_update',
39                   kwargs={'pk': self.pk})
```

Test the different jQuery plugings

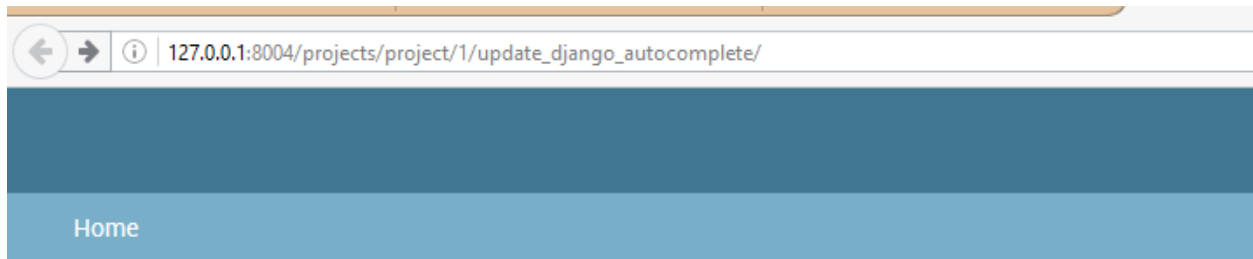
django-autocomplete-light test

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-autocomplete-light test*
 - *install*
 - * *update the settings.py module*
 - *tree*
 - *projects/views_django_autocomplete_light.py*
 - *projects/api/get_users/?q (projects/urls.py)*
 - * *test*
 - *projects/forms_django_autocomplete_light.py*
 - *Using autocompletes in the admin : nice !*
 - *Using autocompletes outside the admin*
 - * *projects/views_django_autocomplete_light.py*
 - * *Django template*
 - *Test URL http://127.0.0.1:8004/projects/project/1/update_django_autocomplete*
 - *Passing options to select2*



Test Django autocomplete

Update of the project '(title:project test 2 champion:Jalmari_Utriainen)

Title:

Champion:

- as
- Martin_Porras
- Mats_Asplund
- Osvaldo_Piras
- Santiago_Abascal
- Silas_Jeppesen
- Tazio_Basile
- Tomas_Holmkvist

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')
        }
```

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



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()
```

  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

```

        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" %}
{% 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/ #}

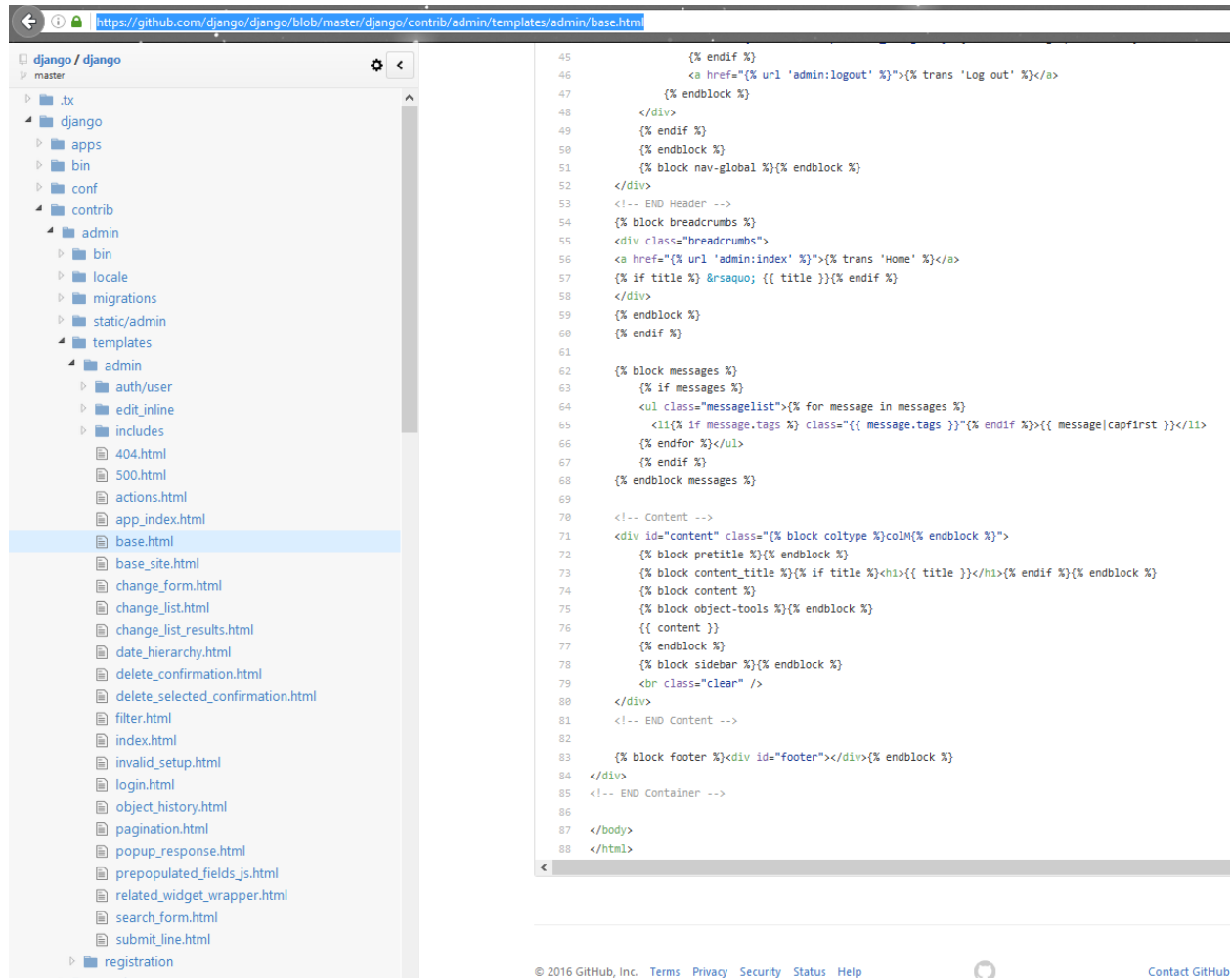
```

This screenshot shows the GitHub interface for the 'django-autocomplete-light' repository. The left sidebar displays the file tree, with 'test_project' expanded and 'templates' selected. The main content area shows the file 'select2_outside_admin.html' with 20 lines of code. The code is a Django template that extends 'base.html' and includes a form with a CSRF token and a submit button. The file is 399 bytes and was last modified by 'jpic'.

```
1 {% extends 'base.html' %}
2 {% # Don't forget that one ! #}
3 {% load static %}
4
5 {% block content %}
6 <div>
7     <form action="" method="post">
8         {% csrf_token %}
9         {{ form.as_p }}
10        <input type="submit" />
11    </form>
12 </div>
13 {% endblock %}
14
15 {% block footer %}
16 <script type="text/javascript" src="{% static 'admin/js/vendor/jquery/jquery.js' %}"></script>
17
18 {{ form.media }}
19 {% endblock %}
```

This screenshot shows the GitHub interface for the 'django-autocomplete-light' repository. The left sidebar displays the file tree, with 'test_project' expanded and 'templates' selected. The main content area shows the file 'base.html' with 55 lines of code. The code is a Django template that extends 'admin/base_site.html' and includes a form with a CSRF token and a submit button. The file is 1.87 KB and was last modified by 'jpic'.

```
1 {% extends "admin/base_site.html" %}
2 {% block branding %}
3 <h1>django-autocomplete-light demo: welcome</h1>
4 {% endblock %}
5
6 {% block breadcrumbs %}
7 <div class="breadcrumbs">
8 <a href="/">Home</a>
9 {% block title %}{% endblock %}
10 </div>
11 {% endblock %}
12
13 {% block content %}
14 <div style="display: inline-block; width: 49%; vertical-align: top;">
15 <h2>Welcome to django-autocomplete-light's demo project !</h2>
16 <p>
17     Here are a number of example apps which you can test manually, or let
18     robots do the work and run ./manage.py test.
19 </p>
20 <ul>
21 <li>
```

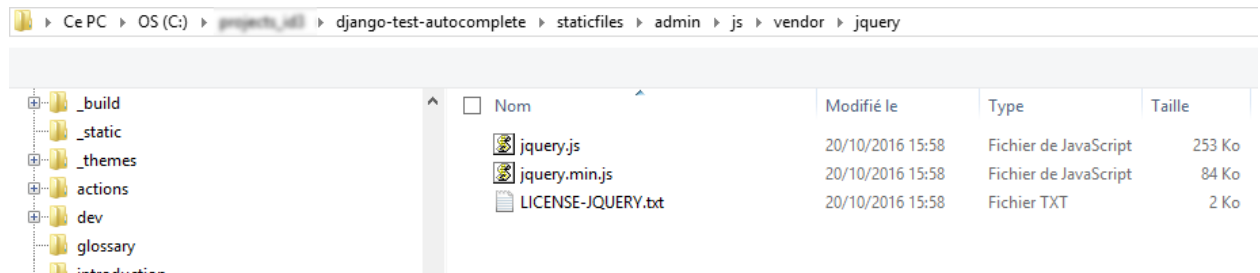


The screenshot shows the Django admin template file `base.html` from the Django project. The file is located at `https://github.com/django/django/blob/master/django/contrib/admin/templates/admin/base.html`. The left sidebar shows the file tree structure, and the main area displays the HTML code of the template. The code includes Django template tags for rendering blocks, messages, and navigation elements.

```

45         {% endif %}
46         <a href="{% url 'admin:logout' %}">{% trans 'Log out' %}</a>
47     {% endblock %}
48 </div>
49     {% endif %}
50 {% endblock %}
51 {% block nav-global %}{% endblock %}
52 </div>
53 <!-- END Header -->
54 {% block breadcrumbs %}
55 <div class="breadcrumbs">
56 <a href="{% url 'admin:index' %}">{% trans 'Home' %}</a>
57 {% if title %} &rsaquo; {{ title }}{% endif %}
58 </div>
59 {% endblock %}
60 {% endif %}
61
62 {% block messages %}
63     {% if messages %}
64     <ul class="messagelist">{% for message in messages %}
65     <li{% if message.tags %} class="{{ message.tags }}"{% endif %}>{{ message|capfirst }}</li>
66     {% endfor %}</ul>
67     {% endif %}
68 {% endblock messages %}
69
70 <!-- Content -->
71 <div id="content" class="{% block coltype %}colM{% endblock %}">
72     {% block pretitle %}{% endblock %}
73     {% block content_title %}{% if title %}<h1>{{ title }}</h1>{% endif %}{% endblock %}
74     {% block content %}
75     {% block object-tools %}{% endblock %}
76     {{ content }}
77     {% endblock %}
78     {% block sidebar %}{% endblock %}
79     <br class="clear" />
80 </div>
81 <!-- END Content -->
82
83 {% block footer %}<div id="footer"></div>{% endblock %}
84 </div>
85 <!-- END Container -->
86
87 </body>
88 </html>

```



The screenshot shows a Windows File Explorer window with the path `C:\projects\django-test-autocomplete\staticfiles\admin\js\vendor\jquery`. The directory contains three files:

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

```

<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" />
        </div>
    </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

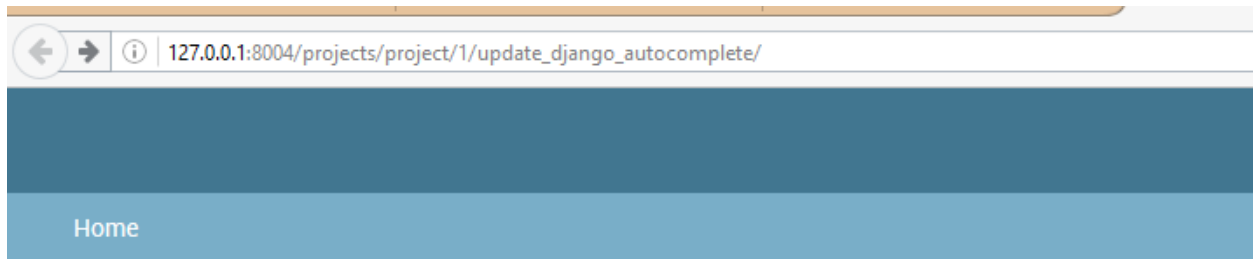
Select2 supports a bunch of options. These options may be set in data-* attributes.

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,
    }
)

```



Test Django autocomplete

Update of the project '(title:project test 2 champion:Jalmari_Utriainen)

Title:

Champion: x ▲

- Martin_Porras ▲
- Mats_Asplund
- Osvaldo_Piras
- Santiago_Abascal
- Silas_Jeppesen
- Tazio_Basile
- Tomas_Holmkvist ▼

```
    },  
  )
```

jQuery EasyAutocomplete test

Contents

- *jQuery EasyAutocomplete test*
 - *Running the local Django web server*
 - *Testing the JSON API view*
 - *Testing the form, step1*
 - * *projects/urls.py*
 - * *The forms.py part*
 - * *The HTML and JavaScript part*
 - *Step2 : initialize the easyAutocomplete placeholder with the value of champion*
 - *Step3 : hide the select champion form field*
 - * *Before the update*
 - * *After the update*

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.
```

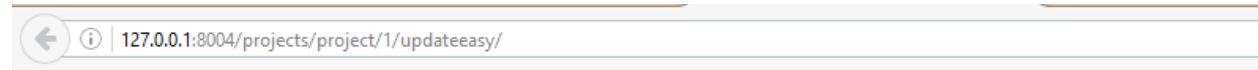
Testing the JSON API view

```
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



Test jquery EasyAutocomplete

Update of the project '(title:project test champion:john)

Title:

Champion:

- 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>

```

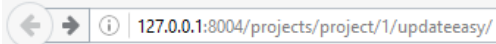
Step2 : initialize the easyAutocomplete placeholder with the value of champion

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);

```



Test jquery EasyAutocomplete

Update of the project '(title:project test 2 champion:albert)

Title:

Champion:

Champion value:

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

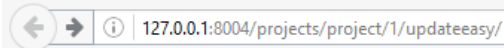
in the projects/forms.py file:

```
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);
```



Test jquery EasyAutocomplete

Update of the project '(title:project test 2 champion:zoya)

Title:

Champion:

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

- *jQuery UI autocomplete test*
 - *Running the local Django web server*
 - *The projects/forms.py module*
 - *The HTML and JavaScript part (the Django Template)*
 - *The projects/views.py module*
 - *The projects/urls.py module*
 - *The jQuery UI scrollbar module*

Running the local Django web server



Test jquery EasyAutocomplete

Update of the project '(title:project test 2 champion:Tazio_Basile)'

Title:

Champion:

- Abdul_Parker
- Adolf_Wahlberg
- Alexander_Dahlberg
- Alexandria_Weiss
- Amber_Taylor
- Anaïs_Bonneau
- André_Albert
- Anette_Robertsson
- Anežka_Bláhová
- Asser_Sirviö

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

```
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

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

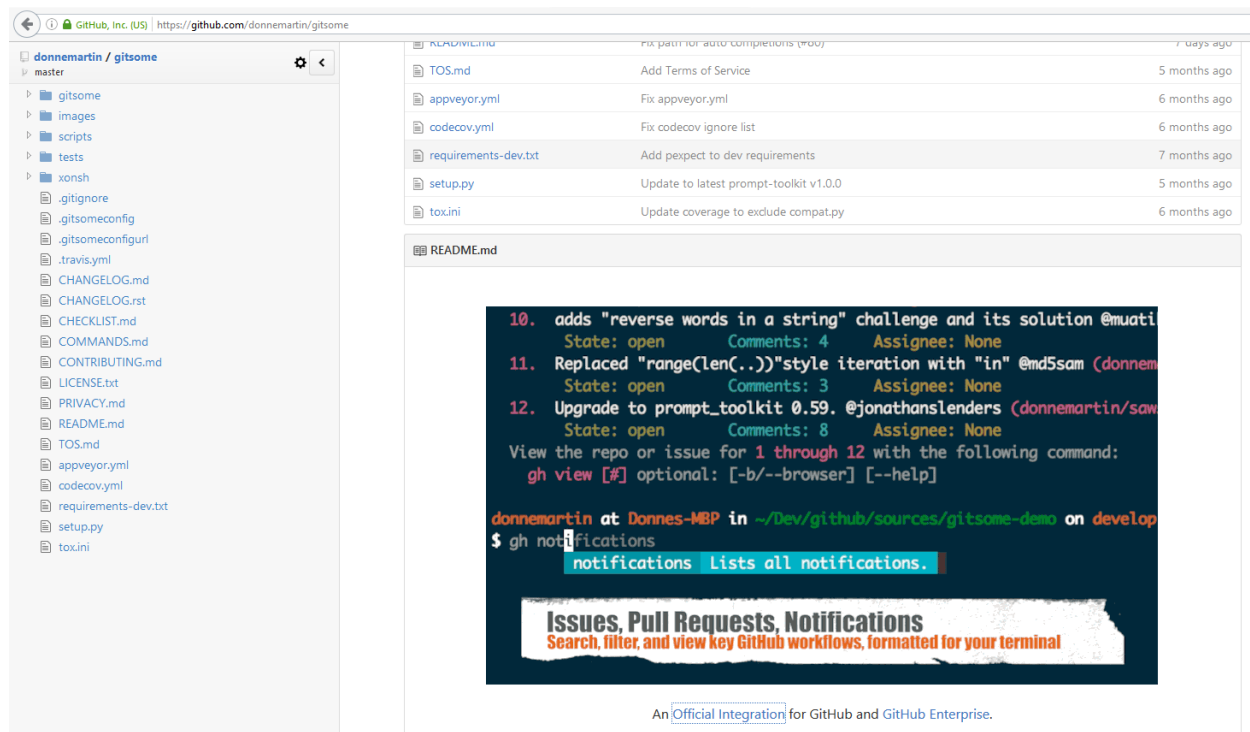


Fig. 3.1: An Official Integration for GitHub and GitHub Enterprise.

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)
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\128527ed75a5c07780bcef517c1be9a031b0f4672c
  Running setup.py bdist_wheel for numpydoc ... done
  Stored in directory: C:
↪\Users\pvergain\AppData\Local\pip\Cache\wheels\9d\ca\8c\9be286b195791bbbfebf407899de13eb12725a2c2
  Running setup.py bdist_wheel for ply ... done
  Stored in directory: C:
↪\Users\pvergain\AppData\Local\pip\Cache\wheels\c1\0c\bd\306a63396decbe8353a4a056fcb97a092be0e0355
  Running setup.py bdist_wheel for feedparser ... done
  Stored in directory: C:
↪\Users\pvergain\AppData\Local\pip\Cache\wheels\15\ce\10\b500f745822ea6db6ea8ed225c06b15c000d71016b
  Running setup.py bdist_wheel for uritemplate.py ... done
  Stored in directory: C:
↪\Users\pvergain\AppData\Local\pip\Cache\wheels\f0\28\64\52c28cc4428d1c79eb7e923cade677f5e63ae1d2fd
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 jquery-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*

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

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

Test jquery EasyAutocomplete

Update of the project '(title:project test 2 champion:zoya)

Title:

Champion:

- Alicia_Pareja
- Alighieri_Coppola
- Ana_Orozco
- Andrés_Vizcaino
- Arthur_Chauvet
- Belen_Cárdenas
- Bente_Christoffersen
- Blanca_Anguita
- Blanca_Aparicio
- Carl_Carlsen
- Carmen_Piquer
- Catrine_Petersen
- Cecco_De Angelis
- Colette_Guyot
- Corinne_Imbert
- Céline_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 testing the jquery EasyAutocomplete (EAC) options for an AJAX call*
 - *Renaming views*
 - *Add a great number of projects into the ‘project’ table*
 - *Write an API view/urls to select projects*
 - * *The view*
 - * *The URL*
 - * *Tests with httpie*
 - *Update the Django template file in order to call this view*
 - *6 elements is not enough*
 - *Ajout placeholder*
 - *list[‘onSelectItemEvent’]*
 - * *HTML5*
 - * *Javascript*
 - *list[‘showAnimation’]*
 - *TODO : tester categories*

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

Ajout placeholder

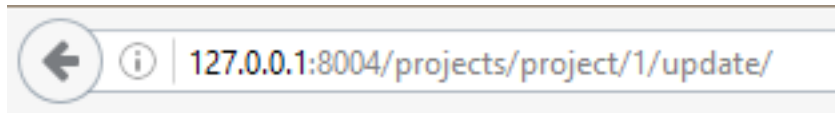
```
<!-- STRUCTURE -> HTML5 elements -->
  <h1>EasyAutocomplete</h1>

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

list['onSelectItemEvent']

See also:

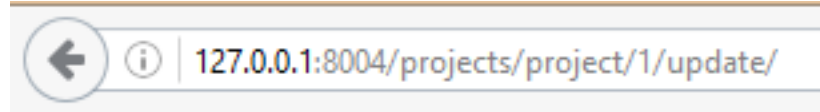
<http://easyautocomplete.com/guide#sec-function-selected-data>



EasyAutocomplete

Choose project

- python project:0
- python project:1
- python project:2
- python project:3
- python project:4
- python project:5



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
projet:19

EasyAutocomplete

Choose the project

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

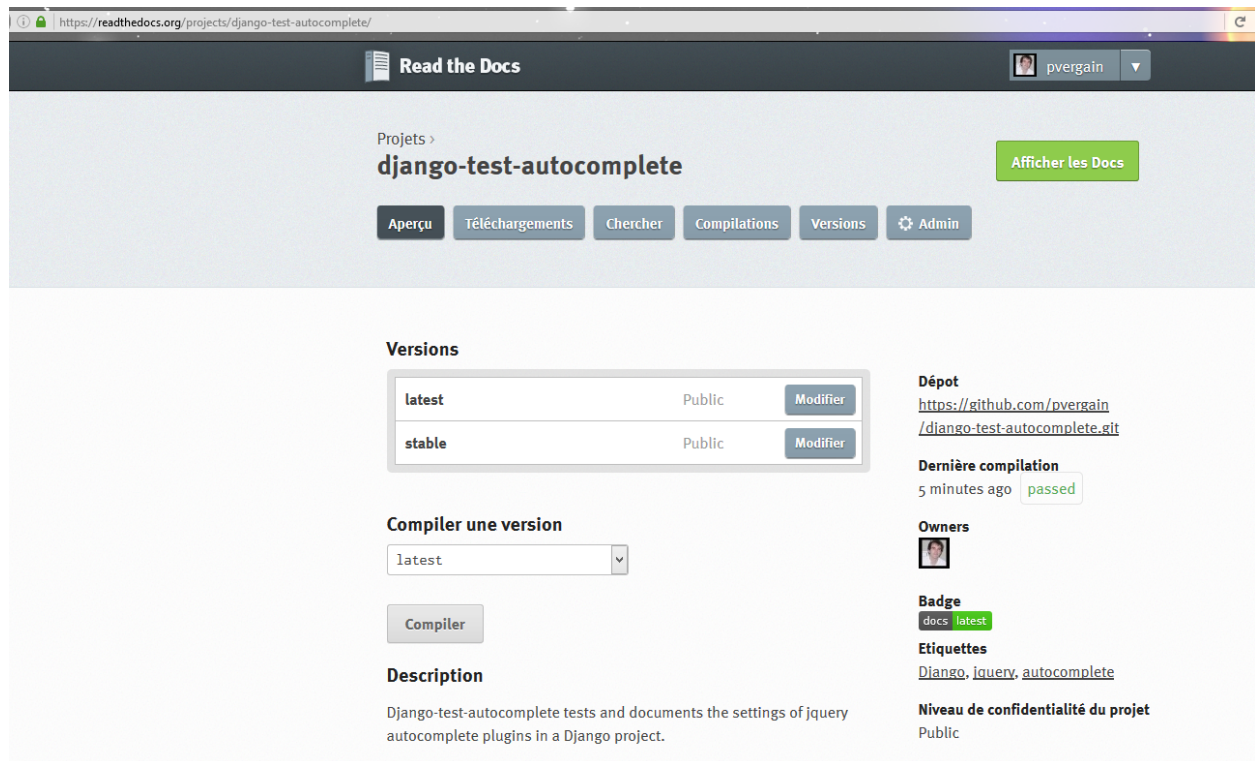
See also:

- <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/>



The screenshot shows the Read the Docs project page for **django-test-autocomplete**. The page has a dark header with the Read the Docs logo and a user profile for 'pvergain'. Below the header, the project name is displayed with a green 'Afficher les Docs' button. A navigation bar contains buttons for 'Aperçu', 'Téléchargements', 'Chercher', 'Compilations', 'Versions', and 'Admin'. The 'Versions' section shows a table with two versions: 'latest' and 'stable', both marked as 'Public' with 'Modifier' buttons. Below this is a 'Compiler une version' section with a dropdown menu set to 'latest' and a 'Compiler' button. The 'Description' section states: 'Django-test-autocomplete tests and documents the settings of jquery autocomplete plugins in a Django project.' On the right side, there is a 'Dépot' section with a GitHub link, a 'Dernière compilation' section showing '5 minutes ago' and 'passed', an 'Owners' section with a profile picture, a 'Badge' section with a 'docs latest' badge, an 'Etiquettes' section with 'Django, jquery, autocomplete', and a 'Niveau de confidentialité du projet' section set to 'Public'.

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

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

 **Read the Docs**

Connexion

Si vous n'avez pas encore créé de compte, merci de vous [sign up](#) au préalable.

Identifiant :

Mot de passe :


Se souvenir de moi : ☒

Mot de passe oublié ? [Réinitialisation.](#)

[GitHub](#) | [Docs](#). Made by [humans](#). Funded by [readers like you](#).



français [French]

https://readthedocs.org/dashboard/

 **Read the Docs**

Connexion avec pvergain réussie.

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 :


URL du dépôt :

Hosted documentation repository URL

Type de dépôt :

Edit advanced project options :
☒








https://readthedocs.org/dashboard/import/manual/

 Read the Docs

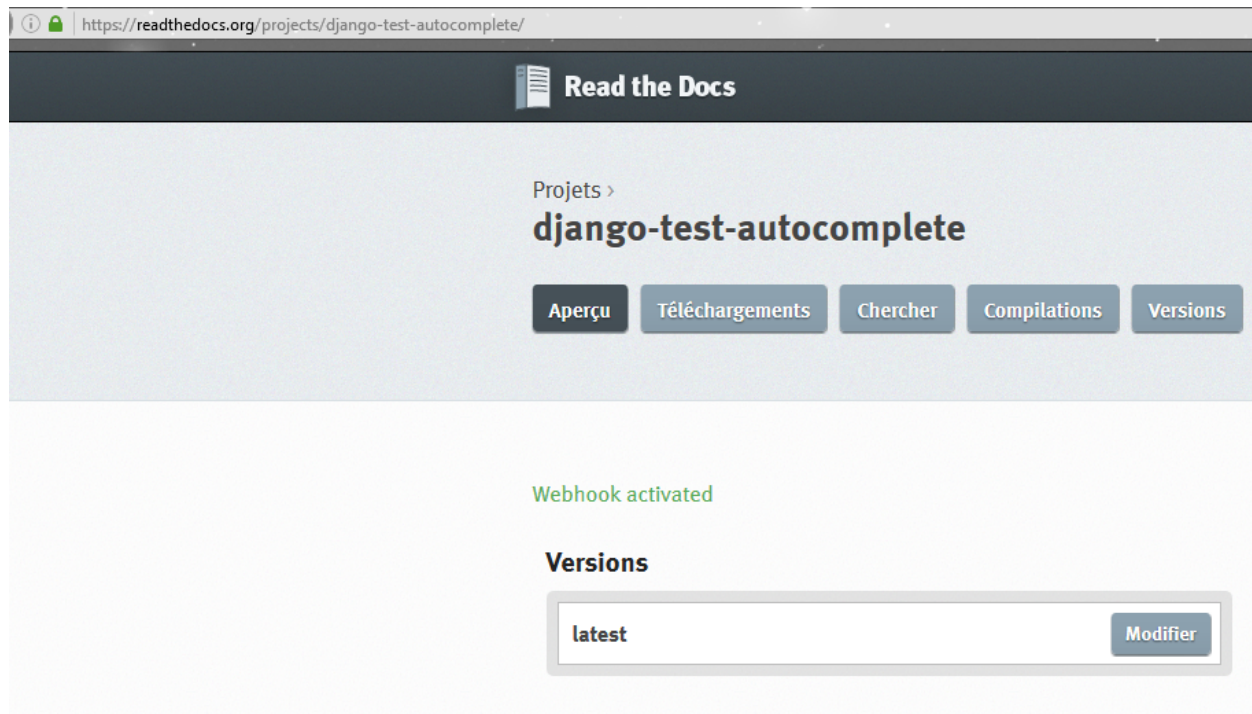
Project Extra Details

Here are a few more project options that you may need to configure.

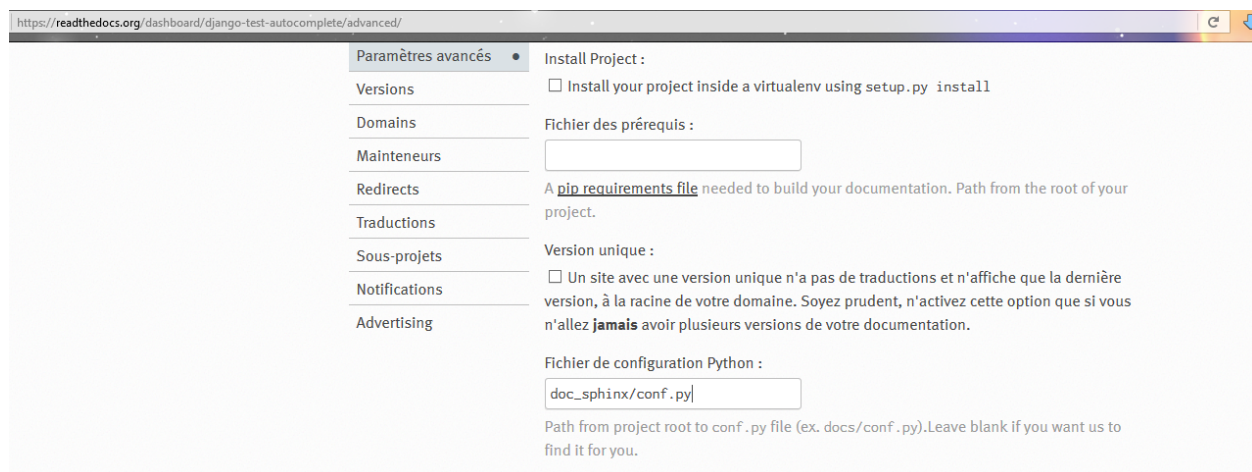
Description :

H1 H2 H3 | B I |   |   |   | 

Django-test-autocomplete tests and documents the settings of jquery
autocomplete plugins in a Django project.



The screenshot shows the Read the Docs website for the project `django-test-autocomplete`. The URL in the browser is `https://readthedocs.org/projects/django-test-autocomplete/`. The page features a dark header with the Read the Docs logo and the project name. Below the header, there's a navigation bar with buttons for `Aperçu`, `Téléchargements`, `Chercher`, `Compilations`, and `Versions`. The `Versions` button is highlighted. Below the navigation bar, there's a section titled `Webhook activated`. Underneath, there's a section titled `Versions` with a dropdown menu showing `latest` and a `Modifier` button.



The screenshot shows the Read the Docs advanced settings page for the project `django-test-autocomplete`. The URL in the browser is `https://readthedocs.org/dashboard/django-test-autocomplete/advanced/`. The page has a sidebar with a list of settings: `Paramètres avancés`, `Versions`, `Domains`, `Mainteneurs`, `Redirects`, `Traductions`, `Sous-projets`, `Notifications`, and `Advertising`. The `Paramètres avancés` option is selected. The main content area contains the following settings:

- 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 :**
 -
- 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

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

Contents

- 2016-10-20 2016-10-21 First step: using the jquery EasyAutocomplete plugin with Django
 - Why ? EasyAutocomplete can easily become one of the best autocomplete plugins available for free
 - EasyAutocomplete files tree (Javascript, CSS, JSON, PHP files)
 - Javascript SCSS and CSS EasyAutocomplete files
 - * EasyAutocomplete-master/src/sass/easy-autocomplete.scss
 - * EasyAutocomplete-master/dist/easy-autocomplete.css
 - * EasyAutocomplete-master/dist/jquery.easy-autocomplete.js
 - EasyAutocomplete simple example
 - * EasyAutocomplete-master/demo/example_simple.html

- *EasyAutocomplete JSON example*
 - * *EasyAutocomplete-master/demo/example_json.html*
 - * *EasyAutocomplete-master/demo/resources/countries.json*
- *EasyAutocomplete flags example*
 - * *EasyAutocomplete-master/demo/example_flags.html*
 - * *EasyAutocomplete-master/demo/resources/flags.css*
 - * *EasyAutocomplete-master/demo/resources/flags.png*
- *EasyAutocomplete categories example*
 - * *EasyAutocomplete-master/demo/example_categories.html*
 - * *EasyAutocomplete-master/demo/resources/categories.json*
- *EasyAutocomplete static_link example*
 - * *EasyAutocomplete-master/demo/example_static_link.html*
 - * *EasyAutocomplete-master/demo/resources/site.json*
- *EasyAutocomplete email example*
 - * *EasyAutocomplete-master/demo/example_email.html*
 - * *EasyAutocomplete-master/demo/resources/people.json*
- *EasyAutocomplete Django integration*
 - * *Include JS and CSS files from the distribution (static files)*
- *Exemples of static files in the Django World*
 - * *The Django contrib module*
 - * *Django_By_Example_Code/Chapter 1/mysite/blog*
 - * *Django_By_Example_Code/Chapter 8*
 - * *Django_By_Example_Code/Chapter 13*
- *Django settings example*
- *Integration in our Django test project*
- *Create a simple update_easy_simple.html*
 - * *Le fichier projects/templates/projects/projet/update_easy_simple.html*
 - * *First problem : the libraries are not found*
 - * *collectstatic*
 - * *New tree*
 - * *Create an easyautocomplete directory*
 - * *Update the Django template file*
 - * *OK it works !*
 - * *Add JSON files*
- *Adding the AJAX call in the Django template*

- * URL : `http://127.0.0.1:8004/projects/champion_get_json/?term=a`
- * `pip install httpie (clihttp)`
- * `http http://127.0.0.1:8004/projects/champion_get_json/?term=a`
- * `http http://easyautocomplete.com/api/countrySearch.php?phrase=co`
- * Avec `countrySearch.php`
- * Avec Python/Django OK the first step is DONE
 - The Template
 - The View `ChampionAutoCompleteView`
 - The projects `urls.py`
- This is the happy end of the first step



easyautocomplete.com

Home Guide Themes Examp

EasyAutocomplete

JQuery autocomplete plugin.

Try one of these examples: [Countries](#) [Countries with flags](#) [Heroes](#) [See more themes](#)

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": "Åland Islands", "code": "AX"},
  {"name": "Albania", "code": "AL"},
  {"name": "Algeria", "code": "DZ"},
  {"name": "American Samoa", "code": "AS"},
  ...
]
```

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..

126kr.com/article/69n5fxx65il

126Kr126kr.com

Google Search

Q

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

```
1  [
2    {
3      "name": "Afghanistan", "code": "AF"},
4      {
5        "name": "Aland Islands", "code": "AX"},
6        {
7          "name": "Albania", "code": "AL"},
8          {
9            "name": "Algeria", "code": "DZ"},
10           {
11             "name": "American Samoa", "code": "AS"},
12           ...
13         ]
```

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

```
C:.\n|\n|.gitattributes\n|.gitignore\n|.jscsrc\n|.jshinttrc\n|authors.txt\n|bower.json\n|Gruntfile.js\n|LICENSE.txt\n|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     &::-ms-input-placeholder      {@content}
22 }
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         }
35         box-shadow: 0 1px 2px rgba(0, 0, 0, 0.1) inset;
36         color: #555;
37         float: none;
38         padding: 6px 12px;
39
40         &:hover, &:focus {
41             box-shadow: none;
42         }
43     }
44
45     a {
46         display: block;

```

```
47     }
48
49
50     &.eac-blue-light {
51
52         input {
53
54             &:hover, &:focus {
55                 border-color: $blue-light;
56                 box-shadow: 0 1px 1px rgba(0, 0, 0, 0.075) inset, 0 0 8px $blue-
↪lighter;
57             }
58         }
59
60         ul {
61             border-color: $blue-light;
62             box-shadow: 0 1px 1px rgba(0, 0, 0, 0.075) inset, 0 0 8px $blue-lighter;
63
64             li, .eac-category {
65                 border-color: $blue-light;
66
67                 &.selected {
68                     background-color: lighten($blue-light, 30%);
69                 }
70             }
71         }
72     }
73
74     &.eac-green-light {
75
76         input {
77             &:hover, &:focus {
78                 border-color: $green-light;
79                 box-shadow: 0 1px 1px rgba(0, 0, 0, 0.075) inset, 0 0 8px $green-
↪lighter;
80             }
81         }
82
83         ul {
84             border-color: $green-light;
85             box-shadow: 0 1px 1px rgba(0, 0, 0, 0.075) inset, 0 0 8px $green-lighter;
86
87             li, .eac-category {
88                 border-color: $green-light;
89
90                 &.selected {
91                     background-color: lighten($green-light, 30%);
92                 }
93             }
94         }
95     }
96
97     &.eac-red-light {
98
99         input {
100             &:hover, &:focus {
```



```

103         border-color: $red-light;
104         box-shadow: 0 1px 1px rgba(0, 0, 0, 0.075) inset, 0 0 8px $red-
↪lighter;
105     }
106 }
107
108 ul {
109     border-color: $red-light;
110     box-shadow: 0 1px 1px rgba(0, 0, 0, 0.075) inset, 0 0 8px $red-lighter;
111
112     li, .eac-category {
113         border-color: $red-light;
114
115         &.selected {
116             background-color: lighten($red-light, 10%);
117         }
118     }
119 }
120 }
121 }
122
123 &.eac-yellow-light {
124
125     input {
126         &:hover, &:focus {
127             border-color: $yellow-light;
128             box-shadow: 0 1px 1px rgba(0, 0, 0, 0.075) inset, 0 0 8px $yellow-
↪lighter;
129         }
130     }
131
132     ul {
133         border-color: $yellow-light;
134         box-shadow: 0 1px 1px rgba(0, 0, 0, 0.075) inset, 0 0 8px $yellow-lighter;
135
136         li, .eac-category {
137             border-color: $yellow-light;
138
139             &.selected {
140                 background-color: lighten($yellow-light, 10%);
141             }
142         }
143     }
144 }
145 }
146
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         }
262     }
263
264     ul {
265         border-color: $blue;
266
267         li, .eac-category {
268             background-color: lighten($blue, 5%);
269             border-color: $blue;
270
271             &.selected {
272                 background-color: lighten($blue, 15%);
273                 color: #f6f6f6;

```

```
274         }
275     }
276 }
277
278
279
280 &.eac-yellow {
281     color: #333;
282
283     input {
284         background-color: lighten($yellow, 5%);
285         border: {
286             color: #333;
287             radius: 4px;
288         }
289         box-shadow: 0;
290         color: #333;
291
292         &:hover, &:focus {
293             border-color: #333;
294             box-shadow: 0;
295         }
296     }
297
298     ul {
299         border-color: #333;
300
301         li, .eac-category {
302             background-color: lighten($yellow, 5%);
303             border-color: #333;
304
305             &.selected {
306                 background-color: lighten($yellow, 15%);
307                 color: #333;
308             }
309         }
310     }
311 }
312
313
314 &.eac-purple {
315     color: #333;
316
317     input {
318         background-color: lighten($purple, 5%);
319         border: {
320             color: darken($purple, 5%);
321         }
322         box-shadow: 0;
323         color: #333;
324
325         &:hover, &:focus {
326             border-color: #333;
327             box-shadow: 0;
328         }
329     }
330
331     ul {
```

```

332         border-color: #333;
333
334     li, .eac-category {
335         background-color: lighten($purple, 5%);
336         border-color: #333;
337
338         &.selected {
339             background-color: lighten($purple, 12%);
340             color: #333;
341         }
342     }
343 }
344 }
345 }
346
347 &.eac-bootstrap {
348
349     input {
350         border-color: #ccc;
351         border-radius: 4px;
352         border-style: solid;
353         border-width: 1px;
354         color: #555;
355         padding: 6px 12px;
356     }
357 }
358
359 }
360
361
362
363 .easy-autocomplete-container {
364     left: 0;
365     position: absolute;
366     width: 100%;
367     z-index: 2;
368
369     ul {
370         background: none repeat scroll 0 0 #ffffff;
371         border-top: 1px dotted #ccc;
372         display: none;
373         margin-top: 0;
374         padding-bottom: 0;
375         padding-left: 0;
376         position: relative;
377         top: -1px;
378
379         li, .eac-category {
380             background: inherit;
381             border: {
382                 color: #ccc;
383                 image: none;
384                 style: solid;
385                 width: 0 1px;
386             }
387             display: block;
388             font: {
389                 size: 14px;

```

```
390         weight: normal;
391     }
392     padding: 4px 12px;
393 }
394
395 li {
396
397     &:last-child {
398         border: {
399             radius: 0 0 2px 2px;
400             width: 0 1px 1px;
401         }
402     }
403
404     &.selected {
405         background: none repeat scroll 0 0 #ebebeb;
406         cursor: pointer;
407
408         div {
409             font-weight: normal;
410         }
411     }
412
413     div {
414         display: block;
415         font-weight: normal;
416         word-break: break-all;
417     }
418
419     b {
420         font: {
421             weight: bold;
422         }
423     }
424 }
425
426 .eac-category {
427     font: {
428         color: #aaa;
429         style: italic;
430     }
431 }
432
433 }
434
435
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 .eac-icon-left {
449
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;
12 }
13 .easy-autocomplete input {
14     border-color: #ccc;
15     border-radius: 4px;
16     border-style: solid;
17     border-width: 1px;
18     box-shadow: 0 1px 2px rgba(0, 0, 0, 0.1) inset;
19     color: #555;
20     float: none;
21     padding: 6px 12px;
22 }
23 .easy-autocomplete input:hover, .easy-autocomplete input:focus {

```

```

24     box-shadow: none;
25 }
26 .easy-autocomplete a {
27     display: block;
28 }
29 .easy-autocomplete.eac-blue-light input:hover, .easy-autocomplete.eac-blue-light
↪input:focus {
30     border-color: #66afe9;
31     box-shadow: 0 1px 1px rgba(0, 0, 0, 0.075) inset, 0 0 8px rgba(102, 175, 233, 0.6);
32 }
33 .easy-autocomplete.eac-blue-light ul {
34     border-color: #66afe9;
35     box-shadow: 0 1px 1px rgba(0, 0, 0, 0.075) inset, 0 0 8px rgba(102, 175, 233, 0.6);
36 }
37 .easy-autocomplete.eac-blue-light ul li, .easy-autocomplete.eac-blue-light ul .eac-
↪category {
38     border-color: #66afe9;
39 }
40 .easy-autocomplete.eac-blue-light ul li.selected, .easy-autocomplete.eac-blue-light
↪ul .eac-category.selected {
41     background-color: #ecf5fc;
42 }
43 .easy-autocomplete.eac-green-light input:hover, .easy-autocomplete.eac-green-light
↪input:focus {
44     border-color: #41DB00;
45     box-shadow: 0 1px 1px rgba(0, 0, 0, 0.075) inset, 0 0 8px rgba(146, 237, 107, 0.6);
46 }
47 .easy-autocomplete.eac-green-light ul {
48     border-color: #41DB00;
49     box-shadow: 0 1px 1px rgba(0, 0, 0, 0.075) inset, 0 0 8px rgba(146, 237, 107, 0.6);
50 }
51 .easy-autocomplete.eac-green-light ul li, .easy-autocomplete.eac-green-light ul .eac-
↪category {
52     border-color: #41DB00;
53 }
54 .easy-autocomplete.eac-green-light ul li.selected, .easy-autocomplete.eac-green-light
↪ul .eac-category.selected {
55     background-color: #9eff75;
56 }
57 .easy-autocomplete.eac-red-light input:hover, .easy-autocomplete.eac-red-light input:
↪focus {
58     border-color: #ff5b5b;
59     box-shadow: 0 1px 1px rgba(0, 0, 0, 0.075) inset, 0 0 8px rgba(255, 90, 90, 0.6);
60 }
61 .easy-autocomplete.eac-red-light ul {
62     border-color: #ff5b5b;
63     box-shadow: 0 1px 1px rgba(0, 0, 0, 0.075) inset, 0 0 8px rgba(255, 90, 90, 0.6);
64 }
65 .easy-autocomplete.eac-red-light ul li, .easy-autocomplete.eac-red-light ul .eac-
↪category {
66     border-color: #ff5b5b;
67 }
68 .easy-autocomplete.eac-red-light ul li.selected, .easy-autocomplete.eac-red-light ul .
↪eac-category.selected {
69     background-color: #ff8e8e;
70 }
71 .easy-autocomplete.eac-yellow-light input:hover, .easy-autocomplete.eac-yellow-light
↪input:focus {

```



```

72     border-color: #ffdb00;
73     box-shadow: 0 1px 1px rgba(0, 0, 0, 0.075) inset, 0 0 8px rgba(255, 231, 84, 0.6);
74 }
75 .easy-autocomplete.eac-yellow-light ul {
76     border-color: #ffdb00;
77     box-shadow: 0 1px 1px rgba(0, 0, 0, 0.075) inset, 0 0 8px rgba(255, 231, 84, 0.6);
78 }
79 .easy-autocomplete.eac-yellow-light ul li, .easy-autocomplete.eac-yellow-light ul .
↪eac-category {
80     border-color: #ffdb00;
81 }
82 .easy-autocomplete.eac-yellow-light ul li.selected, .easy-autocomplete.eac-yellow-
↪light ul .eac-category.selected {
83     background-color: #ffe233;
84 }
85 .easy-autocomplete.eac-dark-light input:hover, .easy-autocomplete.eac-dark-light
↪input:focus {
86     border-color: #333;
87     box-shadow: 0 1px 1px rgba(0, 0, 0, 0.075) inset, 0 0 8px rgba(55, 55, 55, 0.6);
88 }
89 .easy-autocomplete.eac-dark-light ul {
90     border-color: #333;
91     box-shadow: 0 1px 1px rgba(0, 0, 0, 0.075) inset, 0 0 8px rgba(55, 55, 55, 0.6);
92 }
93 .easy-autocomplete.eac-dark-light ul li, .easy-autocomplete.eac-dark-light ul .eac-
↪category {
94     border-color: #333;
95 }
96 .easy-autocomplete.eac-dark-light ul li.selected, .easy-autocomplete.eac-dark-light
↪ul .eac-category.selected {
97     background-color: #4d4d4d;
98     color: #fff;
99 }
100 .easy-autocomplete.eac-dark {
101     color: #fff;
102 }
103 .easy-autocomplete.eac-dark input {
104     background-color: #404040;
105     border-radius: 4px;
106     box-shadow: 0;
107     color: #f6f6f6;
108 }
109 .easy-autocomplete.eac-dark input:hover, .easy-autocomplete.eac-dark input:focus {
110     border-color: #333;
111     box-shadow: 0;
112 }
113 .easy-autocomplete.eac-dark ul {
114     border-color: #333;
115 }
116 .easy-autocomplete.eac-dark ul li, .easy-autocomplete.eac-dark ul .eac-category {
117     background-color: #404040;
118     border-color: #333;
119 }
120 .easy-autocomplete.eac-dark ul li.selected, .easy-autocomplete.eac-dark ul .eac-
↪category.selected {
121     background-color: #737373;
122     color: #f6f6f6;
123 }

```

```
124 .easy-autocomplete.eac-dark-glass {
125     color: #fff;
126 }
127 .easy-autocomplete.eac-dark-glass input {
128     background-color: rgba(0, 0, 0, 0.8);
129     border-radius: 4px;
130     box-shadow: 0;
131     color: #f6f6f6;
132 }
133 .easy-autocomplete.eac-dark-glass input:hover, .easy-autocomplete.eac-dark-glass_
↪input:focus {
134     border-color: rgba(0, 0, 0, 0.8);
135     box-shadow: 0;
136 }
137 .easy-autocomplete.eac-dark-glass ul {
138     border-color: rgba(0, 0, 0, 0.8);
139 }
140 .easy-autocomplete.eac-dark-glass ul li, .easy-autocomplete.eac-dark-glass ul .eac-
↪category {
141     background-color: rgba(0, 0, 0, 0.8);
142     border-color: rgba(0, 0, 0, 0.8);
143 }
144 .easy-autocomplete.eac-dark-glass ul li.selected, .easy-autocomplete.eac-dark-glass_
↪ul .eac-category.selected {
145     background-color: rgba(64, 64, 64, 0.8);
146     color: #f6f6f6;
147 }
148 .easy-autocomplete.eac-dark-glass ul li:last-child, .easy-autocomplete.eac-dark-glass_
↪ul .eac-category:last-child {
149     border-radius: 0 0 4px 4px;
150 }
151 .easy-autocomplete.eac-blue {
152     color: #fff;
153 }
154 .easy-autocomplete.eac-blue input {
155     background-color: #6d9ed1;
156     border-radius: 4px;
157     box-shadow: 0;
158     color: #f6f6f6;
159 }
160 .easy-autocomplete.eac-blue input::-webkit-input-placeholder {
161     color: #f6f6f6;
162 }
163 .easy-autocomplete.eac-blue input:-moz-placeholder {
164     color: #f6f6f6;
165 }
166 .easy-autocomplete.eac-blue input::-moz-placeholder {
167     color: #f6f6f6;
168 }
169 .easy-autocomplete.eac-blue input:-ms-input-placeholder {
170     color: #f6f6f6;
171 }
172 .easy-autocomplete.eac-blue input:hover, .easy-autocomplete.eac-blue input:focus {
173     border-color: #5A91CB;
174     box-shadow: 0;
175 }
176 .easy-autocomplete.eac-blue ul {
177     border-color: #5A91CB;
```

```

178 }
179 .easy-autocomplete.eac-blue ul li, .easy-autocomplete.eac-blue ul .eac-category {
180     background-color: #6d9ed1;
181     border-color: #5A91CB;
182 }
183 .easy-autocomplete.eac-blue ul li.selected, .easy-autocomplete.eac-blue ul .eac-
184 ↪category.selected {
185     background-color: #94b8dd;
186     color: #f6f6f6;
187 }
188 .easy-autocomplete.eac-yellow {
189     color: #333;
190 }
191 .easy-autocomplete.eac-yellow input {
192     background-color: #ffdb7e;
193     border-color: #333;
194     border-radius: 4px;
195     box-shadow: 0;
196     color: #333;
197 }
198 .easy-autocomplete.eac-yellow input:hover, .easy-autocomplete.eac-yellow input:focus {
199     border-color: #333;
200     box-shadow: 0;
201 }
202 .easy-autocomplete.eac-yellow ul {
203     border-color: #333;
204 }
205 .easy-autocomplete.eac-yellow ul li, .easy-autocomplete.eac-yellow ul .eac-category {
206     background-color: #ffdb7e;
207     border-color: #333;
208 }
209 .easy-autocomplete.eac-yellow ul li.selected, .easy-autocomplete.eac-yellow ul .eac-
210 ↪category.selected {
211     background-color: #ffe9b1;
212     color: #333;
213 }
214 .easy-autocomplete.eac-purple {
215     color: #333;
216 }
217 .easy-autocomplete.eac-purple input {
218     background-color: #d6d1e7;
219     border-color: #b8afd5;
220     box-shadow: 0;
221     color: #333;
222 }
223 .easy-autocomplete.eac-purple input:hover, .easy-autocomplete.eac-purple input:focus {
224     border-color: #333;
225     box-shadow: 0;
226 }
227 .easy-autocomplete.eac-purple ul {
228     border-color: #333;
229 }
230 .easy-autocomplete.eac-purple ul li, .easy-autocomplete.eac-purple ul .eac-category {
231     background-color: #d6d1e7;
232     border-color: #333;
233 }
234 .easy-autocomplete.eac-purple ul li.selected, .easy-autocomplete.eac-purple ul .eac-
235 ↪category.selected {

```

```
233     background-color: #ebe8f3;
234     color: #333;
235 }
236 .easy-autocomplete.eac-bootstrap input {
237     border-color: #ccc;
238     border-radius: 4px;
239     border-style: solid;
240     border-width: 1px;
241     color: #555;
242     padding: 6px 12px;
243 }
244
245 .easy-autocomplete-container {
246     left: 0;
247     position: absolute;
248     width: 100%;
249     z-index: 2;
250 }
251 .easy-autocomplete-container ul {
252     background: none repeat scroll 0 0 #ffffff;
253     border-top: 1px dotted #ccc;
254     display: none;
255     margin-top: 0;
256     padding-bottom: 0;
257     padding-left: 0;
258     position: relative;
259     top: -1px;
260 }
261 .easy-autocomplete-container ul li, .easy-autocomplete-container ul .eac-category {
262     background: inherit;
263     border-color: #ccc;
264     border-image: none;
265     border-style: solid;
266     border-width: 0 1px;
267     display: block;
268     font-size: 14px;
269     font-weight: normal;
270     padding: 4px 12px;
271 }
272 .easy-autocomplete-container ul li:last-child {
273     border-radius: 0 0 2px 2px;
274     border-width: 0 1px 1px;
275 }
276 .easy-autocomplete-container ul li.selected {
277     background: none repeat scroll 0 0 #ebebeb;
278     cursor: pointer;
279 }
280 .easy-autocomplete-container ul li.selected div {
281     font-weight: normal;
282 }
283 .easy-autocomplete-container ul li div {
284     display: block;
285     font-weight: normal;
286     word-break: break-all;
287 }
288 .easy-autocomplete-container ul li b {
289     font-weight: bold;
290 }
```

```

291 .easy-autocomplete-container ul .eac-category {
292     font-color: #aaa;
293     font-style: italic;
294 }
295
296 .eac-description .eac-item span {
297     color: #aaa;
298     font-style: italic;
299     font-size: 0.9em;
300 }
301
302 .eac-icon-left .eac-item img {
303     margin-right: 4px;
304     max-height: 30px;
305 }
306
307 .eac-icon-right .eac-item {
308     margin-top: 8px;
309     min-height: 24px;
310     position: relative;
311 }
312 .eac-icon-right .eac-item img {
313     margin-left: 4px;
314     max-height: 30px;
315     position: absolute;
316     right: -4px;
317     top: -8px;
318 }
319
320 /*# sourceMappingURL=easy-autocomplete.css.map */

```

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

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

```

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 /*
11  * EasyAutocomplete - Configuration
12  */
13 var EasyAutocomplete = (function(scope) {
14
15     scope.Configuration = function Configuration(options) {
16         var defaults = {
17             data: "list-required",
18             url: "list-required",
19             dataType: "json",
20
21             listLocation: function(data) {

```

```
22         return data;
23     },
24     xmlElementName: "",
25     getValue: function(element) {
26         return element;
27     },
28     autocompleteOff: true,
29     placeholder: false,
30     ajaxCallback: function() {},
31     matchResponseProperty: false,
32     list: {
33         sort: {
34             enabled: false,
35             method: function(a, b) {
36                 a = defaults.getValue(a);
37                 b = defaults.getValue(b);
38                 if (a < b) {
39                     return -1;
40                 }
41                 if (a > b) {
42                     return 1;
43                 }
44                 return 0;
45             }
46         },
47         maxNumberOfElements: 6,
48         hideOnEmptyPhrase: true,
49         match: {
50             enabled: false,
51             caseSensitive: false,
52             method: function(element, phrase) {
53                 if (element.search(phrase) > -1) {
54                     return true;
55                 } else {
56                     return false;
57                 }
58             }
59         },
60         showAnimation: {
61             type: "normal", //normal/slide/fade
62             time: 400,
63             callback: function() {}
64         },
65         hideAnimation: {
66             type: "normal",
```

```

80         time: 400,
81         callback: function() {}
82     },
83
84     /* Events */
85     onClickEvent: function() {},
86     onSelectItemEvent: function() {},
87     onLoadEvent: function() {},
88     onChooseEvent: function() {},
89     onKeyEnterEvent: function() {},
90     onMouseOverEvent: function() {},
91     onMouseOutEvent: function() {},
92     onShowListEvent: function() {},
93     onHideListEvent: function() {}
94 },
95
96 highlightPhrase: true,
97
98 theme: "",
99
100 cssClasses: "",
101
102 minCharNumber: 0,
103
104 requestDelay: 0,
105
106 adjustWidth: true,
107
108 ajaxSettings: {},
109
110 preparePostData: function(data, inputPhrase) {return data;},
111
112 loggerEnabled: true,
113
114 template: "",
115
116 categoriesAssigned: false,
117
118 categories: [{
119     maxNumberOfElements: 4
120 }]
121
122 };
123
124 var externalObjects = ["ajaxSettings", "template"];
125
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         prepareDefaults();
161
162         mergeOptions();
163
164         if (defaults.loggerEnabled === true) {
165             printPropertiesThatDoesntExist(console, options);
166         }
167
168         addAjaxSettings();
169
170         processAfterMerge();
171         function prepareDefaults() {
172
173             if (options.dataType === "xml") {
174
175                 if (!options.getValue) {
176                     options.getValue = function(element) {
177                         return $(element).text();
178                     };
179                 }
180             }
181
182             if (!options.list) {
183                 options.list = {};
184             }
185
186             if (!options.list.sort) {
187                 options.list.sort = {};
188             }
189
190             if (!options.list.sort) {
191                 options.list.sort = {};
192             }
193         }
194     }
195 }
```



```

193
194         options.list.sort.method = function(a, b) {
195             a = options.getValue(a);
196             b = options.getValue(b);
197             if (a < b) {
198                 return -1;
199             }
200             if (a > b) {
201                 return 1;
202             }
203             return 0;
204         };
205
206         if (!options.list.match) {
207             options.list.match = {};
208         }
209
210         options.list.match.method = function(element, phrase)
↪ {
211
212             if (element.search(phrase) > -1) {
213                 return true;
214             } else {
215                 return false;
216             }
217         };
218
219     }
220     if (options.categories !== undefined && options.categories_
↪ instanceof Array) {
221
222         var categories = [];
223
224         for (var i = 0, length = options.categories.length; i
↪ < length; i += 1) {
225
226             var category = options.categories[i];
227
228             for (var property in defaults.categories[0]) {
229
230                 if (category[property] === undefined)
↪ {
231                     category[property] = defaults.
↪ categories[0][property];
232                 }
233             }
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
254                         target[propertyName]
↪instanceof Array) {
255                         mergedObject[propertyName] =
↪target[propertyName];
256                     } else {
257                         mergeObjects(source[propertyName],
↪target[propertyName]);
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         function processAfterMerge() {
271
272             if (defaults.url !== "list-required" && typeof defaults.url !
↪=== "function") {
273
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;
284                 };
285             }
286
287             if (typeof defaults.listLocation === "string") {
288                 var defaultlistLocation = defaults.listLocation;
289
290                 if (defaults.dataType.toUpperCase() === "XML") {
291                     defaults.listLocation = function(data) {
292                         return $(data).
↪find(defaultlistLocation);
293                     };
294                 } else {
```

```

295         defaults.listLocation = function(data) {
296             return data[defaultlistLocation];
297         };
298     }
299 }
300
301 if (typeof defaults.getValue === "string") {
302     var defaultsGetValue = defaults.getValue;
303     defaults.getValue = function(element) {
304         return element[defaultsGetValue];
305     };
306 }
307
308 if (options.categories !== undefined) {
309     defaults.categoriesAssigned = true;
310 }
311
312 }
313
314 function addAjaxSettings() {
315
316     if (options.ajaxSettings !== undefined && typeof options.
↪ ajaxSettings === "object") {
317         defaults.ajaxSettings = options.ajaxSettings;
318     } else {
319         defaults.ajaxSettings = {};
320     }
321 }
322
323
324 function isAssigned(name) {
325     if (defaults[name] !== undefined && defaults[name] !== null) {
326         return true;
327     } else {
328         return false;
329     }
330 }
331
332 function printPropertiesThatDoesntExist(console, optionsToCheck) {
333
334     checkPropertiesIfExist(defaults, optionsToCheck);
335
336     function checkPropertiesIfExist(source, target) {
337         for(var property in target) {
338             if (source[property] === undefined) {
339                 console.log("Property '" + property + "
↪ ' does not exist in EasyAutocomplete options API.");
340             }
341
342             if (typeof source[property] === "object" && $.
↪ isArray(property, externalObjects) === -1) {
343                 checkPropertiesIfExist(source[property],
↪ target[property]);
344             }
345         }
346     }
347 }
348

```

```
349         return scope;
350
351     })(EasyAutocomplete || {});
352
353
354     /*
355     * EasyAutocomplete - Logger
356     */
357     var EasyAutocomplete = (function(scope) {
358
359         scope.Logger = function Logger() {
360
361             this.error = function(message) {
362                 console.log("ERROR: " + message);
363             };
364
365             this.warning = function(message) {
366                 console.log("WARNING: " + message);
367             };
368         };
369
370         return scope;
371     })(EasyAutocomplete || {});
372
373
374
375     /*
376     * EasyAutocomplete - Constants
377     */
378     var EasyAutocomplete = (function(scope) {
379
380         scope.Constants = function Constants() {
381             var constants = {
382                 CONTAINER_CLASS: "easy-autocomplete-container",
383                 CONTAINER_ID: "eac-container-",
384
385                 WRAPPER_CSS_CLASS: "easy-autocomplete"
386             };
387
388             this.getValue = function(propertyName) {
389                 return constants[propertyName];
390             };
391         };
392
393         return scope;
394     })(EasyAutocomplete || {});
395
396
397
398     /*
399     * EasyAutocomplete - ListBuilderService
400     *
401     * @author Łukasz Pawełczak
402     */
403
404     var EasyAutocomplete = (function(scope) {
405
406         scope.ListBuilderService = function ListBuilderService(configuration,   

407         processResponseData) {
```

```

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
450         return listBuilder;
451     };
452
453     this.processData = function(listBuilder, inputPhrase) {
454
455         for(var i = 0, length = listBuilder.length; i < length; i+=1)
↪{
456             listBuilder[i].data = ↵
↪processResponseData(configuration, listBuilder[i], inputPhrase);
457         }
458
459         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         function convertToListBuilder(category, data) {
476
477             var builder = {};
478
479             if(configuration.get("dataType").toUpperCase() === "XML") {
480
481                 builder = convertXmlToListBuilder();
482             } else {
483
484                 builder = convertDataToListBuilder();
485             }
486
487             if (category.header !== undefined) {
488                 builder.header = category.header;
489             }
490
491             if (category.maxNumberOfElements !== undefined) {
492                 builder.maxNumberOfElements = category.
↪maxNumberOfElements;
493             }
494
495             if (configuration.get("list").maxNumberOfElements !==
↪undefined) {
496
497                 builder.maxListSize = configuration.get("list").
↪maxNumberOfElements;
498             }
499
500             if (category.getValue !== undefined) {
501
502                 if (typeof category.getValue === "string") {
503                     var defaultsGetValue = category.getValue;
504                     builder.getValue = function(element) {
505                         return element[defaultsGetValue];
506                     };
507                 } else if (typeof category.getValue === "function") {
508                     builder.getValue = category.getValue;
509                 }
510             }
511         }
```

```

512         } else {
513             builder.getValue = configuration.get("getValue");
514         }
515     }
516
517     return builder;
518
519
520     function convertXmlToListBuilder() {
521
522         var builder = {},
523             listLocation;
524
525         if (category.xmlElementName !== undefined) {
526             builder.xmlElementName = category.
527 ↪ xmlElementName;
528         }
529
530         if (category.listLocation !== undefined) {
531             listLocation = category.listLocation;
532         } else if (configuration.get("listLocation") !==
533 ↪ undefined) {
534             listLocation = configuration.get("listLocation
535 ↪ ");
536         }
537
538         if (listLocation !== undefined) {
539             if (typeof listLocation === "string") {
540                 builder.data = $(data).
541 ↪ find(listLocation);
542             } else if (typeof listLocation === "function
543 ↪ ") {
544                 builder.data = listLocation(data);
545             } else {
546                 builder.data = data;
547             }
548         }
549         return builder;
550     }
551
552     function convertDataToListBuilder() {
553
554         var builder = {};
555
556         if (category.listLocation !== undefined) {
557             if (typeof category.listLocation === "string
558 ↪ ") {
559                 builder.data = data[category.
560 ↪ listLocation];
561             } else if (typeof category.listLocation ===
562 ↪ "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 process 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 processData(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
765                 return elementValue + " - <span>" + _
↪ element[_fields.description] + "</span>";
766             };
767         } else if (typeof _fields.description === "function")
↪ {
768             buildMethod = function(elementValue, element)
↪ {
769                 return elementValue + " - <span>" + _
↪ fields.description(element) + "</span>";
770             };
771         }
772
773         return buildMethod;
774     }
775
776     if (template.type === "iconRight") {
777
778         if (typeof _fields.iconSrc === "string") {
779             buildMethod = function(elementValue, element)
↪ {
780                 return elementValue + "<img class=
↪ 'eac-icon' src='" + element[_fields.iconSrc] + "' />";
781             };
782         } else if (typeof _fields.iconSrc === "function") {
783             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
796             return "<img class='eac-icon' src='"
↪ + element[_fields.iconSrc] + "' />" + elementValue;
797         };
798     } else if (typeof _fields.iconSrc === "function") {
799         buildMethod = function(elementValue, element)
↪ {
800
801             return "<img class='eac-icon' src='"
↪ + _fields.iconSrc(element) + "' />" + elementValue;
802         };
803     }
804
805     return buildMethod;
806 }
807
808 if(template.type === "links") {
809
810     if (typeof _fields.link === "string") {
811         buildMethod = function(elementValue, element)
↪ {
812
813             return "<a href='" + element[_fields.
↪ link] + "' >" + elementValue + "</a>";
814         };
815     } else if (typeof _fields.link === "function") {
816         buildMethod = function(elementValue, element)
↪ {
817
818             return "<a href='" + _fields.
↪ link(element) + "' >" + elementValue + "</a>";
819         };
820     }
821
822     return buildMethod;
823 }
824
825 if (template.type === "custom") {
826
827     return template.method;
828 }
829
830 return genericTemplates.basic.method;
831
832 },

```

```

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     templateClass = function(options) {
848         var emptyStringFunction = function() {return ""};
849
850         if (!options || !options.type) {
851
852             return emptyStringFunction;
853         }
854
855         if (options.type && genericTemplates[options.type]) {
856             return (function () {
857                 var _cssClass = genericTemplates[options.
858 ↪type].cssClass;
859
860                 return function() { return _cssClass;};
861             })();
862         } else {
863             return emptyStringFunction;
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     scope.main = function Core($input, options) {
886
887         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,
898     ↪scope.proccess),
899
900         checkParam = config.equals,
901
902         $field = $input,
903         $container = "",
904         elementsList = [],
905         selectedElement = -1,
906         requestDelayTimeoutId;
907
908     scope.consts = consts;
909
910     this.getConstants = function() {
911         return consts;
912     };
913
914     this.getConfiguration = function() {
915         return config;
916     };
917
918     this.getContainer = function() {
919         return $container;
920     };
921
922     this.getSelectedItemId = function() {
923         return selectedElement;
924     };
925
926     this.getItems = function () {
927         return elementsList;
928     };
929
930     this.getItemData = function(index) {
931
932         if (elementsList.length < index || elementsList[index] ===
933     ↪undefined) {
934
935             return -1;
936         } else {
937             return elementsList[index];
938         }
939     };
940
941     this.getSelectedItemData = function() {
942         return this.getItemData(selectedElement);
943     };
944
945     this.build = function() {
946         prepareField();
947     };

```

```

945     this.init = function() {
946         init();
947     };
948     function init() {
949
950         if ($field.length === 0) {
951             logger.error("Input field doesn't exist.");
952             return;
953         }
954
955         if (!config.checkDataUrlProperties()) {
956             logger.error("One of options variables 'data' or 'url
957 ↪ ' must be defined.");
958             return;
959         }
960
961         if (!config.checkRequiredProperties()) {
962             logger.error("Will not work without mentioned_
963 ↪ properties.");
964             return;
965         }
966
967         prepareField();
968         bindEvents();
969     }
970     function prepareField() {
971
972
973         if ($field.parent().hasClass(consts.getValue("WRAPPER_CSS_
974 ↪ CLASS")) {
975             removeContainer();
976             removeWrapper();
977         }
978
979         createWrapper();
980         createContainer();
981
982         $container = $("#" + getContainerId());
983         if (config.get("placeholder")) {
984             $field.attr("placeholder", config.get("placeholder"));
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 ↪ ") {
994                 classes += " eac-" + config.get("theme");
995             }
996
997             if (config.get("cssClasses") && config.get("cssClasses
998 ↪ ") !== "") {

```

```

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 = $("

").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() {
1040
1041                     switch(config.get("list").
↪showAnimation.type) {
1042
1043                         case "slide":
1044                             var
↪animationTime = config.get("list").showAnimation.time,
1045                             callback
↪= config.get("list").showAnimation.callback;
1046
1047                             $elements_
↪container.find("ul").slideDown(animationTime, callback);
1048                             break;
1049


```



```

1050                                     case "fade":
1051                                         var
1052         ↪animationTime = config.get("list").showAnimation.time,
1053                                     callback
1054         ↪= config.get("list").showAnimation.callback;
1055                                     $selements_
1056         ↪container.find("ul").fadeIn(animationTime), callback;
1057                                     break;
1058                                     default:
1059                                         $selements_
1060         ↪container.find("ul").show();
1061                                     break;
1062                                     }
1063         ↪onShowListEvent();
1064                                     config.get("list").
1065                                     })
1066                                     /* List hide animation */
1067                                     .on("hide.eac", function() {
1068         ↪hideAnimation.type) {
1069                                     switch(config.get("list").
1070                                     case "slide":
1071                                         var
1072         ↪animationTime = config.get("list").hideAnimation.time,
1073                                     callback
1074         ↪= config.get("list").hideAnimation.callback;
1075                                     $selements_
1076         ↪container.find("ul").slideUp(animationTime, callback);
1077                                     break;
1078                                     case "fade":
1079                                         var
1080         ↪animationTime = config.get("list").hideAnimation.time,
1081                                     callback
1082         ↪= config.get("list").hideAnimation.callback;
1083                                     $selements_
1084         ↪container.find("ul").fadeOut(animationTime, callback);
1085                                     break;
1086                                     default:
1087                                         $selements_
1088         ↪container.find("ul").hide();
1089                                     break;
1090                                     }
1091         ↪onHideListEvent();
1092                                     config.get("list").
1093                                     })
1094                                     .on("selectElement.eac", function() {
1095         ↪li").removeClass("selected");
1096                                     $selements_container.find("ul

```

```

1094                                     $elements_container.find("ul_
↪li").eq(selectedElement).addClass("selected");
1095
1096                                     config.get("list").
↪onSelectItemEvent();
1097                                     })
1098                                     .on("loadElements.eac",_
↪function(event, listBuilders, phrase) {
1099
1100                                     var $item = "",
1101                                     $listContainer =
↪$elements_container.find("ul");
1102
1103                                     $listContainer
1104                                     .empty()
1105                                     .detach();
1106
1107                                     elementsList = [];
1108                                     var counter = 0;
1109                                     for(var builderIndex = 0,_
1110                                     ↪listBuildersLength = listBuilders.length; builderIndex < listBuildersLength; ↪
↪builderIndex += 1) {
1111
1112                                     var listData =_
↪listBuilders[builderIndex].data;
1113
1114                                     if (listData.length_
↪=== 0) {
1115                                     continue;
1116                                     }
1117
1118                                     if_
↪(listBuilders[builderIndex].header !== undefined && listBuilders[builderIndex].
↪header.length > 0) {
1119
↪$listContainer.append("<div class='eac-category' >" + listBuilders[builderIndex].
↪header + "</div>");
1120                                     }
1121
1122                                     for(var i = 0,_
↪listDataLength = listData.length; i < listDataLength && counter <_
↪listBuilders[builderIndex].maxListSize; i += 1) {
1123                                     $item = $("
↪<li><div class='eac-item'></div></li>");
1124
1125                                     (function() {
1126                                     var j_
↪= i,
1127                                     ↪= counter,
1128                                     ↪= listBuilders[builderIndex].getValue(listData[j]);
1129
1130                                     $item.
↪find("> div")
1131
↪on("click", function() {

```

itemCounter_

elementsValue

```

1133 ↪ $field.val (elementsValue) .trigger ("change");
1135
1136 ↪ = itemCounter;
1137
1138 ↪
1139
1140 ↪ get ("list") .onClickEvent ();
1141
1142 ↪ get ("list") .onChooseEvent ();
1143
1144 ↪ )
1145
1146 ↪ mouseover (function () {
1147
1148 ↪ = itemCounter;
1149
1150 ↪
1151
1152 ↪ get ("list") .onMouseOverEvent ();
1153
1154 ↪ )
1155
1156 ↪ mouseout (function () {
1157
1158 ↪ get ("list") .onMouseOutEvent ();
1159
1160 ↪ )
1161
1162 ↪ html (template.build (highlight (elementsValue, phrase), listData [j]));
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
```

```

1172                                     $field.next("." + consts.getValue("CONTAINER_CLASS")).
↪remove();
1173                                     }
1174
1175     function highlight(string, phrase) {
1176
1177         if(config.get("highlightPhrase") && phrase !== "") {
1178             return highlightPhrase(string, phrase);
1179         } else {
1180             return string;
1181         }
1182     }
1183
1184     function escapeRegExp(str) {
1185         return str.replace(/[\-\[\]\|\{\}\|\(\)\*\+\?\.\\\^\
↪$\|]/g, "\\$&");
1186     }
1187
1188     function highlightPhrase(string, phrase) {
1189         var escapedPhrase = escapeRegExp(phrase);
1190         return (string + "").replace(new RegExp("(" + ↪
↪escapedPhrase + ")", "gi"), "<b>$1</b>");
1191     }
1192
1193
1194
1195
1196     }
1197     function getContainerId() {
1198
1199         var elementId = $field.attr("id");
1200
1201         elementId = consts.getValue("CONTAINER_ID") + elementId;
1202
1203         return elementId;
1204     }
1205     function bindEvents() {
1206
1207         bindAllEvents();
1208
1209
1210         function bindAllEvents() {
1211             if (checkParam("autocompleteOff", true)) {
1212                 removeAutocomplete();
1213             }
1214
1215             bindFocusOut();
1216             bindKeyup();
1217             bindKeydown();
1218             bindKeyPress();
1219             bindFocus();
1220             bindBlur();
1221         }
1222
1223         function bindFocusOut() {
1224             $field.focusout(function () {
1225
1226                 var fieldValue = $field.val(),

```

```

1227         phrase;
1228
1229         if (!config.get("list").match.caseSensitive) {
1230             fieldValue = fieldValue.toLowerCase();
1231         }
1232
1233         for (var i = 0, length = elementsList.length;
1234 ↪ i < length; i += 1) {
1235
1236             phrase = config.get("getValue
1237 ↪") (elementsList[i]);
1238
1239             if (!config.get("list").match.
1240 ↪caseSensitive) {
1241
1242                 phrase = phrase.toLowerCase();
1243             }
1244
1245             if (phrase === fieldValue) {
1246                 selectedElement = i;
1247                 selectElement(selectedElement);
1248
1249                 return;
1250             }
1251         }
1252     });
1253
1254     function bindKeyup() {
1255         $field
1256         .off("keyup")
1257         .keyup(function(event) {
1258
1259             switch(event.keyCode) {
1260
1261                 case 27:
1262
1263                     hideContainer();
1264                     loseFieldFocus();
1265                     break;
1266
1267                 case 38:
1268
1269                     event.preventDefault();
1270
1271                     if(elementsList.length > 0 &&
1272 ↪selectedElement > 0) {
1273
1274                         selectedElement -= 1;
1275
1276                         $field.val(config.get(
1277 ↪"getValue") (elementsList[selectedElement]));
1278
1279                         selectElement(selectedElement);
1280
1281                     }
1282
1283                     break;
1284
1285                 case 40:

```

```

1278                                     event.preventDefault();
1279
1280                                     if(elementsList.length > 0 &&
1281 ↪selectedElement < elementsList.length - 1) {
1282
1283                                         selectedElement += 1;
1284
1285                                         $field.val(config.get(
1286 ↪"getValue")(elementsList[selectedElement]));
1287
1288                                         selectElement(selectedElement);
1289 ↪
1290                                     }
1291
1292                                     break;
1293
1294                                     default:
1295
1296                                         if (event.keyCode > 40 ||
1297 ↪event.keyCode === 8) {
1298
1299                                             var inputPhrase =
1300 ↪$field.val();
1301
1302                                             if (!(config.get("list
1303 ↪").hideOnEmptyPhrase === true && event.keyCode === 8 && inputPhrase === "")) {
1304
1305                                                 if (config.
1306 ↪get("requestDelay") > 0) {
1307
1308                                                     if
1309 ↪(requestDelayTimeoutId !== undefined) {
1310
1311                                                         clearTimeout
1312 ↪
1313                                                         }
1314
1315                                                         requestDelayTimeoutId
1316 ↪= setTimeout(function () { loadData(inputPhrase);}, config.get("requestDelay"));
1317                                                         } else {
1318                                                         loadData(inputPhrase);
1319 ↪
1320                                                         }
1321
1322                                                         } else {
1323 ↪
1324                                                         hideContainer();
1325 ↪
1326                                                         }
1327
1328                                                         }
1329
1330                                     }
1331
1332                                     break;
1333
1334                                     }
1335
1336                                     function loadData(inputPhrase) {

```

```

1324                                     if (inputPhrase.length < config.get(
1325 ↪ "minCharNumber")) {
1326                                     return;
1327                                     }
1328
1329
1330                                     if (config.get("data") !== "list-
1331 ↪ required") {
1332                                     var data = config.get("data");
1333
1334                                     var listBuilders = _
1335 ↪ listBuilderService.init(data);
1336                                     listBuilders = _
1337 ↪ listBuilderService.updateCategories(listBuilders, data);
1338                                     listBuilders = _
1339 ↪ listBuilderService.processData(listBuilders, inputPhrase);
1340                                     loadElements(listBuilders, _
1341 ↪ inputPhrase);
1342                                     if ($field.parent().find("li
1343 ↪ ").length > 0) {
1344                                     showContainer();
1345                                     } else {
1346                                     hideContainer();
1347                                     }
1348                                     }
1349
1350                                     var settings = createAjaxSettings();
1351
1352                                     if (settings.url === undefined || _
1353 ↪ settings.url === "") {
1354                                     settings.url = config.get("url
1355 ↪ ");
1356                                     }
1357
1358                                     if (settings.dataType === undefined _
1359 ↪ || settings.dataType === "") {
1360                                     settings.dataType = config.
1361 ↪ get("dataType");
1362                                     }
1363
1364                                     if (settings.url !== undefined && _
1365 ↪ settings.url !== "list-required") {
1366                                     settings.url = settings.
1367 ↪ url(inputPhrase);
1368                                     settings.data = config.get(
1369 ↪ "preparePostData")(settings.data, inputPhrase);
1370                                     $.ajax(settings)

```

```

1368                                     .done(function(data) {
1369
1370                                     var
1371
1372                                     listBuilders
1373
1374                                     listBuilders
1375                                     if
1376                                     (checkInputPhraseMatchResponse(inputPhrase, data)) {
1377                                     listBuilders
1378                                     listBuildersService.processData(listBuilders, inputPhrase);
1379                                     loadElements(listBuilders);
1380                                     inputPhrase);
1381                                     }
1382                                     if
1383                                     (listBuilderService.checkIfDataExists(listBuilders) && $field.parent().find("li").
1384                                     length > 0) {
1385                                     showContainer();
1386                                     } else {
1387                                     hideContainer();
1388                                     }
1389                                     config.get(
1390                                     "ajaxCallback") ();
1391                                     })
1392                                     .fail(function() {
1393                                     logger.
1394                                     warning("Fail to load response data");
1395                                     })
1396                                     .always(function() {
1397                                     });
1398                                     }
1399
1400
1401                                     function createAjaxSettings() {
1402                                     var settings = {},
1403                                     ajaxSettings = config.
1404                                     get("ajaxSettings") || {};
1405                                     for (var set in ajaxSettings)
1406                                     {
1407                                     settings[set] =
1408                                     ajaxSettings[set];
1409                                     }
1410

```



```

1411         return settings;
1412     }
1413
1414     function
1415     ↪checkInputPhraseMatchResponse(inputPhrase, data) {
1416         if (config.get(
1417         ↪"matchResponseProperty") !== false) {
1418             if (typeof config.get(
1419         ↪"matchResponseProperty") === "string") {
1420                 return
1421         ↪(data[config.get("matchResponseProperty")] === inputPhrase);
1422             }
1423             if (typeof config.get(
1424         ↪"matchResponseProperty") === "function") {
1425                 return
1426         ↪(config.get("matchResponseProperty")(data) === inputPhrase);
1427             }
1428             return true;
1429         } else {
1430             return true;
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             })
1448             .keydown(function(event) {
1449                 if (event.keyCode === 13 &&
1450         ↪selectedElement > -1) {
1451
1452                     $field.val(config.get(
1453         ↪"getValue")(elementsList[selectedElement]));
1454
1455                     config.get("list").
1456         ↪onKeyEnterEvent();
1457                     config.get("list").
1458         ↪onChooseEvent();
1459
1460                     selectedElement = -1;
1461                     hideContainer();

```

```

1459                                     event.preventDefault();
1460                                     }
1461                                 });
1462                             }
1463                         }
1464
1465                     function bindKeypress() {
1466                         $field
1467                             .off("keypress");
1468                     }
1469
1470                     function bindFocus() {
1471                         $field.focus(function() {
1472
1473                             if ($field.val() !== "" && elementsList.
1474     ↪length > 0) {
1475
1476                                 selectedElement = -1;
1477                                 showContainer();
1478                             }
1479                         });
1480                     }
1481
1482                     function bindBlur() {
1483                         $field.blur(function() {
1484                             setTimeout(function() {
1485
1486                                 selectedElement = -1;
1487                                 hideContainer();
1488                                 }, 250);
1489                             });
1490                     }
1491
1492                     function removeAutocomplete() {
1493                         $field.attr("autocomplete", "off");
1494                     }
1495                 }
1496
1497                 function showContainer() {
1498                     $container.trigger("show.eac");
1499                 }
1500
1501                 function hideContainer() {
1502                     $container.trigger("hide.eac");
1503                 }
1504
1505                 function selectElement(index) {
1506
1507                     $container.trigger("selectElement.eac", index);
1508                 }
1509
1510                 function loadElements(list, phrase) {
1511                     $container.trigger("loadElements.eac", [list, phrase]);
1512                 }
1513
1514                 function loseFieldFocus() {
1515

```

```

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 && $(input).attr("id").length >
↪ 0) {
1530             return true;
1531         } else {
1532             return false;
1533         }
1534     };
1535
1536     scope.assignRandomId = function(input) {
1537
1538         var fieldId = "";
1539
1540         do {
1541             fieldId = "eac-" + Math.floor(Math.random() * 10000);
1542         } while ($("#" + fieldId).length !== 0);
1543
1544         elementId = scope.consts.getValue("CONTAINER_ID") + fieldId;
1545
1546         $(input).attr("id", fieldId);
1547
1548     };
1549
1550     scope.setHandle = function(handle, id) {
1551         scope.eacHandles[id] = handle;
1552     };
1553
1554
1555     return scope;
1556
1557 }) (EasyAutocomplete || {});
1558
1559 (function($) {
1560
1561     $.fn.easyAutocomplete = function(options) {
1562
1563         return this.each(function() {
1564             var $this = $(this),
1565                 eacHandle = new EasyAutocomplete.main($this, options);
1566
1567             if (!EasyAutocomplete.inputHasId($this)) {
1568                 EasyAutocomplete.assignRandomId($this);
1569             }
1570
1571             eacHandle.init();
1572

```

```
1573         EasyAutocomplete.setHandle(eacHandle, $this.attr("id"));
1574     });
1575
1576     };
1577
1578     $.fn.getSelectedItemId = function() {
1579         var inputId = $(this).attr("id");
1580
1581         if (inputId !== undefined) {
1582             return EasyAutocomplete.getHandle(inputId).
↪getSelectedItemId();
1583         }
1584
1585         return -1;
1586     };
1587
1588     $.fn.getItems = function () {
1589         var inputId = $(this).attr("id");
1590
1591         if (inputId !== undefined) {
1592             return EasyAutocomplete.getHandle(inputId).getItems();
1593         }
1594
1595         return -1;
1596     };
1597
1598     $.fn.getItemData = function(index) {
1599         var inputId = $(this).attr("id");
1600
1601         if (inputId !== undefined && index > -1) {
1602             return EasyAutocomplete.getHandle(inputId).getItemData(index);
1603         }
1604
1605         return -1;
1606     };
1607
1608     $.fn.getSelectedItemData = function() {
1609         var inputId = $(this).attr("id");
1610
1611         if (inputId !== undefined) {
1612             return EasyAutocomplete.getHandle(inputId).
↪getSelectedItemData();
1613         }
1614
1615         return -1;
1616     };
1617
1618     })(jQuery);
```

EasyAutocomplete simple example

file:///C:/projects_id3/django-test-autocomplete/doc_sphinx/annexes/EasyAutocomplete-master/demo/example_simple.html

EasyAutocomplete - easy example

- blue
- green
- pink
- red
- yellow

EasyAutocomplete-master/demo/example_simple.html

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

```

1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <title>EasyAutocomplete simple example</title>
5
6     <!-- STYLE -> CSS -->
7     <link href="../dist/easy-autocomplete.min.css" rel="stylesheet" type=
↪ "text/css">
8     <!-- end STYLE-->
9
10    </head>
11    <body>
12
13      <!-- STRUCTURE -> HTML5 elements -->
14      <h1>EasyAutocomplete - easy example</h1>
15      <input id="simple" />
16      <!-- end STRUCTURE-->
17
18      <!--BEHAVIOR -> Javascript scripts-->
19      <script src="../lib/jquery-1.11.2.min.js"></script>
20      <script src="../dist/jquery.easy-autocomplete.min.js" type="text/
↪ javascript" ></script>
21      <script>
22
23        var options_easy_autocomplete = {
24          data: ["blue", "green", "pink", "red", "yellow"],
25
26        };
27
28        $("#simple").easyAutocomplete(options_easy_autocomplete);
29
30      </script>
31      <!-- end BEHAVIOR -->
32
33    </body>

```

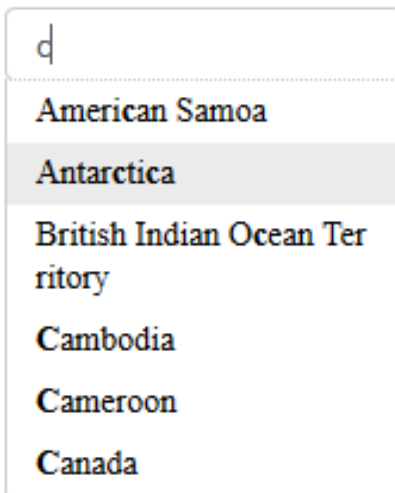
```
</html>
```

EasyAutocomplete JSON example

See also:

JSON

EasyAutocomplete - json example



EasyAutocomplete-master/demo/example_json.html

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

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <title>EasyAutocomplete json example</title>
5
6     <!-- STYLE -> CSS -->
7     <link href="../dist/easy-autocomplete.min.css" rel="stylesheet" type=
8     ↪ "text/css">
9     <!-- end STYLE -->
10  </head>
11  <body>
12
13    <!-- STRUCTURE -> HTML5 elements -->
14    <h1>EasyAutocomplete - json example</h1>
15    <input id="data-json" />
```

```

15     <!-- end STRUCTURE-->
16
17     <!--BEHAVIOR -> Javascript scripts-->
18     <script src="../../lib/jquery-1.11.2.min.js"></script>
19     <script src="../../dist/jquery.easy-autocomplete.min.js" type="text/
↪ javascript" ></script>
20     <script>
21         let options_easy_autocomplete = {
22             url: "resources/countries.json",
23
24             getValue: "name",
25
26             list: {
27                 match: {
28                     enabled: true
29                 }
30             }
31         };
32         $("#data-json").easyAutocomplete(options_easy_autocomplete);
33     </script>
34     <!-- end BEHAVIOR -->
35 </body>
36
37 </html>

```

EasyAutocomplete-master/demo/resources/countries.json

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

```

1  [
2    {"name": "Afghanistan", "code": "AF"},
3    {"name": "Albania", "code": "AL"},
4    {"name": "Algeria", "code": "DZ"},
5    {"name": "American Samoa", "code": "AS"},
6    {"name": "Andorra", "code": "AD"},
7    {"name": "Angola", "code": "AO"},
8    {"name": "Anguilla", "code": "AI"},
9    {"name": "Antarctica", "code": "AQ"},
10   {"name": "Antigua and Barbuda", "code": "AG"},
11   {"name": "Argentina", "code": "AR"},
12   {"name": "Armenia", "code": "AM"},
13   {"name": "Aruba", "code": "AW"},
14   {"name": "Australia", "code": "AU"},
15   {"name": "Austria", "code": "AT"},
16   {"name": "Azerbaijan", "code": "AZ"},
17   {"name": "Bahamas", "code": "BS"},
18   {"name": "Bahrain", "code": "BH"},
19   {"name": "Bangladesh", "code": "BD"},
20   {"name": "Barbados", "code": "BB"},
21   {"name": "Belarus", "code": "BY"},
22   {"name": "Belgium", "code": "BE"},
23   {"name": "Belize", "code": "BZ"},
24   {"name": "Benin", "code": "BJ"},
25   {"name": "Bermuda", "code": "BM"},
26   {"name": "Bhutan", "code": "BT"},
27   {"name": "Bolivia", "code": "BO"},

```

```
28 { "name": "Bosnia and Herzegovina", "code": "BA"},
29 { "name": "Botswana", "code": "BW"},
30 { "name": "Bouvet Island", "code": "BV"},
31 { "name": "Brazil", "code": "BR"},
32 { "name": "British Indian Ocean Territory", "code": "IO"},
33 { "name": "Brunei Darussalam", "code": "BN"},
34 { "name": "Bulgaria", "code": "BG"},
35 { "name": "Burkina Faso", "code": "BF"},
36 { "name": "Burundi", "code": "BI"},
37 { "name": "Cambodia", "code": "KH"},
38 { "name": "Cameroon", "code": "CM"},
39 { "name": "Canada", "code": "CA"},
40 { "name": "Cape Verde", "code": "CV"},
41 { "name": "Cayman Islands", "code": "KY"},
42 { "name": "Central African Republic", "code": "CF"},
43 { "name": "Chad", "code": "TD"},
44 { "name": "Chile", "code": "CL"},
45 { "name": "China", "code": "CN"},
46 { "name": "Christmas Island", "code": "CX"},
47 { "name": "Cocos (Keeling) Islands", "code": "CC"},
48 { "name": "Colombia", "code": "CO"},
49 { "name": "Comoros", "code": "KM"},
50 { "name": "Congo", "code": "CG"},
51 { "name": "Congo, The Democratic Republic of the", "code": "CD"},
52 { "name": "Cook Islands", "code": "CK"},
53 { "name": "Costa Rica", "code": "CR"},
54 { "name": "Cote D'Ivoire", "code": "CI"},
55 { "name": "Croatia", "code": "HR"},
56 { "name": "Cuba", "code": "CU"},
57 { "name": "Cyprus", "code": "CY"},
58 { "name": "Czech Republic", "code": "CZ"},
59 { "name": "Denmark", "code": "DK"},
60 { "name": "Djibouti", "code": "DJ"},
61 { "name": "Dominica", "code": "DM"},
62 { "name": "Dominican Republic", "code": "DO"},
63 { "name": "Ecuador", "code": "EC"},
64 { "name": "Egypt", "code": "EG"},
65 { "name": "El Salvador", "code": "SV"},
66 { "name": "Equatorial Guinea", "code": "GQ"},
67 { "name": "Eritrea", "code": "ER"},
68 { "name": "Estonia", "code": "EE"},
69 { "name": "Ethiopia", "code": "ET"},
70 { "name": "Falkland Islands (Malvinas)", "code": "FK"},
71 { "name": "Faroe Islands", "code": "FO"},
72 { "name": "Fiji", "code": "FJ"},
73 { "name": "Finland", "code": "FI"},
74 { "name": "France", "code": "FR"},
75 { "name": "French Guiana", "code": "GF"},
76 { "name": "French Polynesia", "code": "PF"},
77 { "name": "French Southern Territories", "code": "TF"},
78 { "name": "Gabon", "code": "GA"},
79 { "name": "Gambia", "code": "GM"},
80 { "name": "Georgia", "code": "GE"},
81 { "name": "Germany", "code": "DE"},
82 { "name": "Ghana", "code": "GH"},
83 { "name": "Gibraltar", "code": "GI"},
84 { "name": "Greece", "code": "GR"},
85 { "name": "Greenland", "code": "GL"},
```



```

86 { "name": "Grenada", "code": "GD" },
87 { "name": "Guadeloupe", "code": "GP" },
88 { "name": "Guam", "code": "GU" },
89 { "name": "Guatemala", "code": "GT" },
90 { "name": "Guernsey", "code": "GG" },
91 { "name": "Guinea", "code": "GN" },
92 { "name": "Guinea-Bissau", "code": "GW" },
93 { "name": "Guyana", "code": "GY" },
94 { "name": "Haiti", "code": "HT" },
95 { "name": "Heard Island and Mcdonald Islands", "code": "HM" },
96 { "name": "Holy See (Vatican City State)", "code": "VA" },
97 { "name": "Honduras", "code": "HN" },
98 { "name": "Hong Kong", "code": "HK" },
99 { "name": "Hungary", "code": "HU" },
100 { "name": "Iceland", "code": "IS" },

```

EasyAutocomplete flags example

EasyAutocomplete - country flags example



EasyAutocomplete-master/demo/example_flags.html

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

```

1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <title>EasyAutocomplete flags example</title>
5
6     <!-- STYLE -> CSS -->
7     <link href="../dist/easy-autocomplete.min.css" rel="stylesheet" type=
  ↪ "text/css">

```

```

8      <link href="resources/flags.css" rel="stylesheet" type="text/css" >
9      <!-- end STYLE-->
10
11     </head>
12     <body>
13         <!-- STRUCTURE -> HTML5 elements -->
14         <h1>EasyAutocomplete - country flags example</h1>
15
16         <input id="flags" />
17         <!-- end STRUCTURE-->
18
19         <!--BEHAVIOR -> Javascript scripts-->
20         <script src="../../lib/jquery-1.11.2.min.js"></script>
21         <script src="../../dist/jquery.easy-autocomplete.min.js" type="text/
→javascript" ></script>
22         <script>
23             var options_easy_autocomplete = {
24                 url: "resources/countries.json",
25
26                 getValue: "name",
27
28                 list: {
29                     match: {
30                         enabled: true
31                     },
32                     maxNumberOfElements: 10
33                 },
34
35                 template: {
36                     type: "custom",
37                     method: function(value, item) {
38                         return "<span class='flag flag-" + (item.code).
→toLowerCase() + "' ></span>" + value;
39                     }
40                 }
41             };
42
43             $("#flags").easyAutocomplete(options_easy_autocomplete);
44         </script>
45         <!-- end BEHAVIOR -->
46     </body>
47
48 </html>

```

EasyAutocomplete-master/demo/resources/flags.css

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

```

1 .flag {
2     width: 16px;
3     height: 11px;
4     background:url(flags.png) no-repeat;
5     display: inline-block;
6     margin-right: 5px;
7 }
8

```

```

9  .flag.flag-ad {background-position: -16px 0}
10 .flag.flag-ae {background-position: -32px 0}
11 .flag.flag-af {background-position: -48px 0}
12 .flag.flag-ag {background-position: -64px 0}
13 .flag.flag-ai {background-position: -80px 0}
14 .flag.flag-al {background-position: -96px 0}
15 .flag.flag-am {background-position: -112px 0}
16 .flag.flag-an {background-position: -128px 0}
17 .flag.flag-ao {background-position: -144px 0}
18 .flag.flag-ar {background-position: -160px 0}
19 .flag.flag-as {background-position: -176px 0}
20 .flag.flag-at {background-position: -192px 0}
21 .flag.flag-au {background-position: -208px 0}
22 .flag.flag-aw {background-position: -224px 0}
23 .flag.flag-az {background-position: -240px 0}
24 .flag.flag-ba {background-position: 0 -11px}
25 .flag.flag-bb {background-position: -16px -11px}
26 .flag.flag-bd {background-position: -32px -11px}
27 .flag.flag-be {background-position: -48px -11px}
28 .flag.flag-bf {background-position: -64px -11px}
29 .flag.flag-bg {background-position: -80px -11px}
30 .flag.flag-bh {background-position: -96px -11px}
31 .flag.flag-bi {background-position: -112px -11px}
32 .flag.flag-bj {background-position: -128px -11px}
33 .flag.flag-bm {background-position: -144px -11px}
34 .flag.flag-bn {background-position: -160px -11px}
35 .flag.flag-bo {background-position: -176px -11px}
36 .flag.flag-br {background-position: -192px -11px}
37 .flag.flag-bs {background-position: -208px -11px}
38 .flag.flag-bt {background-position: -224px -11px}
39 .flag.flag-bv {background-position: -240px -11px}
40 .flag.flag-bw {background-position: 0 -22px}
41 .flag.flag-by {background-position: -16px -22px}
42 .flag.flag-bz {background-position: -32px -22px}
43 .flag.flag-ca {background-position: -48px -22px}
44 .flag.flag-catalonia {background-position: -64px -22px}
45 .flag.flag-cd {background-position: -80px -22px}
46 .flag.flag-cf {background-position: -96px -22px}
47 .flag.flag-cg {background-position: -112px -22px}
48 .flag.flag-ch {background-position: -128px -22px}
49 .flag.flag-ci {background-position: -144px -22px}
50 .flag.flag-ck {background-position: -160px -22px}
51 .flag.flag-cl {background-position: -176px -22px}
52 .flag.flag-cm {background-position: -192px -22px}
53 .flag.flag-cn {background-position: -208px -22px}
54 .flag.flag-co {background-position: -224px -22px}
55 .flag.flag-cr {background-position: -240px -22px}
56 .flag.flag-cu {background-position: 0 -33px}
57 .flag.flag-cv {background-position: -16px -33px}
58 .flag.flag-cw {background-position: -32px -33px}
59 .flag.flag-cy {background-position: -48px -33px}
60 .flag.flag-cz {background-position: -64px -33px}
61 .flag.flag-de {background-position: -80px -33px}
62 .flag.flag-dj {background-position: -96px -33px}
63 .flag.flag-dk {background-position: -112px -33px}
64 .flag.flag-dm {background-position: -128px -33px}
65 .flag.flag-do {background-position: -144px -33px}
66 .flag.flag-dz {background-position: -160px -33px}

```

```
67 .flag.flag-ec {background-position: -176px -33px}
68 .flag.flag-ee {background-position: -192px -33px}
69 .flag.flag-eg {background-position: -208px -33px}
70 .flag.flag-eh {background-position: -224px -33px}
71 .flag.flag-england {background-position: -240px -33px}
72 .flag.flag-er {background-position: 0 -44px}
73 .flag.flag-es {background-position: -16px -44px}
74 .flag.flag-et {background-position: -32px -44px}
75 .flag.flag-eu {background-position: -48px -44px}
76 .flag.flag-fi {background-position: -64px -44px}
77 .flag.flag-fj {background-position: -80px -44px}
78 .flag.flag-fk {background-position: -96px -44px}
79 .flag.flag-fm {background-position: -112px -44px}
80 .flag.flag-fo {background-position: -128px -44px}
81 .flag.flag-fr {background-position: -144px -44px}
82 .flag.flag-ga {background-position: -160px -44px}
83 .flag.flag-gb {background-position: -176px -44px}
84 .flag.flag-gd {background-position: -192px -44px}
85 .flag.flag-ge {background-position: -208px -44px}
86 .flag.flag-gf {background-position: -224px -44px}
87 .flag.flag-gg {background-position: -240px -44px}
88 .flag.flag-gh {background-position: 0 -55px}
89 .flag.flag-gi {background-position: -16px -55px}
90 .flag.flag-gl {background-position: -32px -55px}
91 .flag.flag-gm {background-position: -48px -55px}
92 .flag.flag-gn {background-position: -64px -55px}
93 .flag.flag-gp {background-position: -80px -55px}
94 .flag.flag-gq {background-position: -96px -55px}
95 .flag.flag-gr {background-position: -112px -55px}
96 .flag.flag-gs {background-position: -128px -55px}
97 .flag.flag-gt {background-position: -144px -55px}
98 .flag.flag-gu {background-position: -160px -55px}
99 .flag.flag-gw {background-position: -176px -55px}
100 .flag.flag-gy {background-position: -192px -55px}
101 .flag.flag-hk {background-position: -208px -55px}
102 .flag.flag-hm {background-position: -224px -55px}
103 .flag.flag-hn {background-position: -240px -55px}
104 .flag.flag-hr {background-position: 0 -66px}
105 .flag.flag-ht {background-position: -16px -66px}
106 .flag.flag-hu {background-position: -32px -66px}
107 .flag.flag-ic {background-position: -48px -66px}
108 .flag.flag-id {background-position: -64px -66px}
109 .flag.flag-ie {background-position: -80px -66px}
110 .flag.flag-il {background-position: -96px -66px}
111 .flag.flag-im {background-position: -112px -66px}
112 .flag.flag-in {background-position: -128px -66px}
113 .flag.flag-io {background-position: -144px -66px}
114 .flag.flag-iq {background-position: -160px -66px}
115 .flag.flag-ir {background-position: -176px -66px}
116 .flag.flag-is {background-position: -192px -66px}
117 .flag.flag-it {background-position: -208px -66px}
118 .flag.flag-je {background-position: -224px -66px}
119 .flag.flag-jm {background-position: -240px -66px}
120 .flag.flag-jo {background-position: 0 -77px}
121 .flag.flag-jp {background-position: -16px -77px}
122 .flag.flag-ke {background-position: -32px -77px}
123 .flag.flag-kg {background-position: -48px -77px}
124 .flag.flag-kh {background-position: -64px -77px}
```

```

125 .flag.flag-ki {background-position: -80px -77px}
126 .flag.flag-km {background-position: -96px -77px}
127 .flag.flag-kn {background-position: -112px -77px}
128 .flag.flag-kp {background-position: -128px -77px}
129 .flag.flag-kr {background-position: -144px -77px}
130 .flag.flag-kurdistan {background-position: -160px -77px}
131 .flag.flag-kw {background-position: -176px -77px}
132 .flag.flag-ky {background-position: -192px -77px}
133 .flag.flag-kz {background-position: -208px -77px}
134 .flag.flag-la {background-position: -224px -77px}
135 .flag.flag-lb {background-position: -240px -77px}
136 .flag.flag-lc {background-position: 0 -88px}
137 .flag.flag-li {background-position: -16px -88px}
138 .flag.flag-lk {background-position: -32px -88px}
139 .flag.flag-lr {background-position: -48px -88px}
140 .flag.flag-ls {background-position: -64px -88px}
141 .flag.flag-lt {background-position: -80px -88px}
142 .flag.flag-lu {background-position: -96px -88px}
143 .flag.flag-lv {background-position: -112px -88px}
144 .flag.flag-ly {background-position: -128px -88px}
145 .flag.flag-ma {background-position: -144px -88px}
146 .flag.flag-mc {background-position: -160px -88px}
147 .flag.flag-md {background-position: -176px -88px}
148 .flag.flag-me {background-position: -192px -88px}
149 .flag.flag-mg {background-position: -208px -88px}
150 .flag.flag-mh {background-position: -224px -88px}
151 .flag.flag-mk {background-position: -240px -88px}
152 .flag.flag-ml {background-position: 0 -99px}
153 .flag.flag-mm {background-position: -16px -99px}
154 .flag.flag-mn {background-position: -32px -99px}
155 .flag.flag-mo {background-position: -48px -99px}
156 .flag.flag-mp {background-position: -64px -99px}
157 .flag.flag-mq {background-position: -80px -99px}
158 .flag.flag-mr {background-position: -96px -99px}
159 .flag.flag-ms {background-position: -112px -99px}
160 .flag.flag-mt {background-position: -128px -99px}
161 .flag.flag-mu {background-position: -144px -99px}
162 .flag.flag-mv {background-position: -160px -99px}
163 .flag.flag-mw {background-position: -176px -99px}
164 .flag.flag-mx {background-position: -192px -99px}
165 .flag.flag-my {background-position: -208px -99px}
166 .flag.flag-mz {background-position: -224px -99px}
167 .flag.flag-na {background-position: -240px -99px}
168 .flag.flag-nc {background-position: 0 -110px}
169 .flag.flag-ne {background-position: -16px -110px}
170 .flag.flag-nf {background-position: -32px -110px}
171 .flag.flag-ng {background-position: -48px -110px}
172 .flag.flag-ni {background-position: -64px -110px}
173 .flag.flag-nl {background-position: -80px -110px}
174 .flag.flag-no {background-position: -96px -110px}
175 .flag.flag-np {background-position: -112px -110px}
176 .flag.flag-nr {background-position: -128px -110px}
177 .flag.flag-nu {background-position: -144px -110px}
178 .flag.flag-nz {background-position: -160px -110px}
179 .flag.flag-om {background-position: -176px -110px}
180 .flag.flag-pa {background-position: -192px -110px}
181 .flag.flag-pe {background-position: -208px -110px}
182 .flag.flag-pf {background-position: -224px -110px}

```

```
183 .flag.flag-pg {background-position: -240px -110px}
184 .flag.flag-ph {background-position: 0 -121px}
185 .flag.flag-pk {background-position: -16px -121px}
186 .flag.flag-pl {background-position: -32px -121px}
187 .flag.flag-pm {background-position: -48px -121px}
188 .flag.flag-pn {background-position: -64px -121px}
189 .flag.flag-pr {background-position: -80px -121px}
190 .flag.flag-ps {background-position: -96px -121px}
191 .flag.flag-pt {background-position: -112px -121px}
192 .flag.flag-pw {background-position: -128px -121px}
193 .flag.flag-py {background-position: -144px -121px}
194 .flag.flag-qa {background-position: -160px -121px}
195 .flag.flag-re {background-position: -176px -121px}
196 .flag.flag-ro {background-position: -192px -121px}
197 .flag.flag-cs {background-position: -208px -121px}
198 .flag.flag-ru {background-position: -224px -121px}
199 .flag.flag-rw {background-position: -240px -121px}
200 .flag.flag-sa {background-position: 0 -132px}
201 .flag.flag-sb {background-position: -16px -132px}
202 .flag.flag-sc {background-position: -32px -132px}
203 .flag.flag-scotland {background-position: -48px -132px}
204 .flag.flag-sd {background-position: -64px -132px}
205 .flag.flag-se {background-position: -80px -132px}
206 .flag.flag-sg {background-position: -96px -132px}
207 .flag.flag-sh {background-position: -112px -132px}
208 .flag.flag-si {background-position: -128px -132px}
209 .flag.flag-sk {background-position: -144px -132px}
210 .flag.flag-sl {background-position: -160px -132px}
211 .flag.flag-sm {background-position: -176px -132px}
212 .flag.flag-sn {background-position: -192px -132px}
213 .flag.flag-so {background-position: -208px -132px}
214 .flag.flag-somaliland {background-position: -224px -132px}
215 .flag.flag-sr {background-position: -240px -132px}
216 .flag.flag-ss {background-position: 0 -143px}
217 .flag.flag-st {background-position: -16px -143px}
218 .flag.flag-sv {background-position: -32px -143px}
219 .flag.flag-sx {background-position: -48px -143px}
220 .flag.flag-sy {background-position: -64px -143px}
221 .flag.flag-sz {background-position: -80px -143px}
222 .flag.flag-tc {background-position: -96px -143px}
223 .flag.flag-td {background-position: -112px -143px}
224 .flag.flag-tf {background-position: -128px -143px}
225 .flag.flag-tg {background-position: -144px -143px}
226 .flag.flag-th {background-position: -160px -143px}
227 .flag.flag-tj {background-position: -176px -143px}
228 .flag.flag-tk {background-position: -192px -143px}
229 .flag.flag-tl {background-position: -208px -143px}
230 .flag.flag-tm {background-position: -224px -143px}
231 .flag.flag-tn {background-position: -240px -143px}
232 .flag.flag-to {background-position: 0 -154px}
233 .flag.flag-tr {background-position: -16px -154px}
234 .flag.flag-tt {background-position: -32px -154px}
235 .flag.flag-tv {background-position: -48px -154px}
236 .flag.flag-tw {background-position: -64px -154px}
237 .flag.flag-tz {background-position: -80px -154px}
238 .flag.flag-ua {background-position: -96px -154px}
239 .flag.flag-ug {background-position: -112px -154px}
240 .flag.flag-um {background-position: -128px -154px}
```



```

241 .flag.flag-us {background-position: -144px -154px}
242 .flag.flag-uy {background-position: -160px -154px}
243 .flag.flag-uz {background-position: -176px -154px}
244 .flag.flag-va {background-position: -192px -154px}
245 .flag.flag-vc {background-position: -208px -154px}
246 .flag.flag-ve {background-position: -224px -154px}
247 .flag.flag-vg {background-position: -240px -154px}
248 .flag.flag-vi {background-position: 0 -165px}
249 .flag.flag-vn {background-position: -16px -165px}
250 .flag.flag-vu {background-position: -32px -165px}
251 .flag.flag-wales {background-position: -48px -165px}
252 .flag.flag-wf {background-position: -64px -165px}
253 .flag.flag-ws {background-position: -80px -165px}
254 .flag.flag-xk {background-position: -96px -165px}
255 .flag.flag-ye {background-position: -112px -165px}
256 .flag.flag-yt {background-position: -128px -165px}
257 .flag.flag-za {background-position: -144px -165px}
258 .flag.flag-zanzibar {background-position: -160px -165px}
259 .flag.flag-zm {background-position: -176px -165px}
260 .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

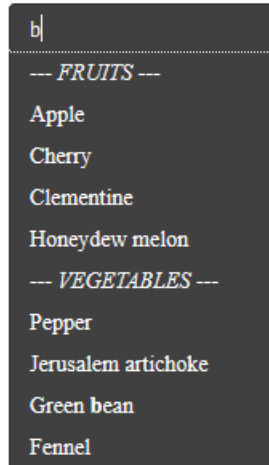
Listing 3.9: EasyAutocomplete-master/demo/example_categories.html

```

1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <title>EasyAutocomplete categories</title>
5     <!-- STYLE -> CSS -->
6     <link href="../dist/easy-autocomplete.min.css" rel="stylesheet" type=
↪ "text/css">
7     <link href="../dist/easy-autocomplete.themes.min.css" rel="stylesheet" ↪
↪ type="text/css">

```

EasyAutocomplete - categories example



```

8         <link href="//fonts.googleapis.com/css?family=Roboto:400,300,300italic,
↳400italic,600,600italic&subset=latin,cyrillic-ext,greek-ext,latin-ext,cyrillic'
↳rel='stylesheet' type='text/css'>
9         <!-- end STYLE-->
10    </head>
11    <body>
12        <!-- STRUCTURE -> HTML5 elements -->
13        <h1>EasyAutocomplete - categories example</h1>
14        <input id="data-categories" placeholder="Fruits and vegetables" />
15        <!-- end STRUCTURE-->
16
17        <!--BEHAVIOR -> Javascript scripts-->
18        <script src="../../lib/jquery-1.11.2.min.js"></script>
19        <script src="../../dist/jquery.easy-autocomplete.min.js" type="text/
↳javascript"></script>
20        <script>
21            var options_easy_autocomplete = {
22                url: "resources/categories.json",
23
24                categories: [{
25                    listLocation: "fruits",
26                    header: "--- FRUITS ---"
27                }, {
28                    listLocation: "vegetables",
29                    header: "--- VEGETABLES ---"
30                }],
31
32                list: {
33                    match: {
34                        enabled: false
35                    },
36                    maxNumberOfElements: 10

```



```

37         },
38
39         theme: "dark"
40     };
41
42     $("#data-categories").easyAutocomplete(options_easy_autocomplete);
43
44     </script>
45     <!-- end BEHAVIOR -->
46 </body>
47
48 </html>

```

EasyAutocomplete-master/demo/resources/categories.json

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

```

1 {
2     "fruits": ["Apple", "Cherry", "Clementine", "Honeydew melon", "Watermelon",
3     ↪ "Satsuma"],
4     "vegetables": ["Pepper", "Jerusalem artichoke", "Green bean", "Fennel",
5     ↪ "Courgette", "Yam"]
6 }

```

EasyAutocomplete static_link example



EasyAutocomplete-master/demo/example_static_link.html

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

```

1 <!DOCTYPE html>
2 <html lang="en">
3     <head>
4         <title>EasyAutocomplete static links</title>
5         <!-- STYLE -> CSS -->

```

```

6         <link href="../../dist/easy-autocomplete.min.css" rel="stylesheet" type=
↪ "text/css">
7         <!-- end STYLE-->
8     </head>
9     <body>
10
11         <!-- STRUCTURE -> HTML5 elements -->
12         <h1>EasyAutocomplete - static links</h1>
13         <input id="data-links" />
14         <!-- end STRUCTURE-->
15
16
17         <!--BEHAVIOR -> Javascript scripts-->
18         <script src="../../lib/jquery-1.11.2.min.js"></script>
19         <script src="../../dist/jquery.easy-autocomplete.min.js" type="text/
↪ javascript" ></script>
20         <script>
21             var options_easy_autocomplete = {
22                 url: "resources/site.json",
23
24                 getValue: "text",
25
26                 template: {
27                     type: "links",
28                     fields: {
29                         link: "site"
30                     }
31                 },
32
33                 list: {
34                     match: {
35                         enabled: false
36                     }
37                 },
38
39                 theme: "plate-dark"
40             };
41
42             $("#data-links").easyAutocomplete(options_easy_autocomplete);
43
44         </script>
45         <!-- end BEHAVIOR -->
46     </body>
47
48 </html>

```

EasyAutocomplete-master/demo/resources/site.json

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

```

1  [ {
2      "text": "Home",
3      "site": "http://easyautocomplete.com"
4  },
5  {
6      "text": "Guide",

```

```

7  "site": "http://easyautocomplete.com/guide"
8  },
9  {
10 "text": "Examples",
11 "site": "http://easyautocomplete.com/examples"
12 },
13 {
14 "text": "Themes",
15 "site": "http://easyautocomplete.com/themes"
16 },
17 {
18 "text": "Download",
19 "site": "http://easyautocomplete.com/download"
20 }
21 ]

```

EasyAutocomplete email example

file:///C:/projects_id3/django-test-autocomplete/doc_sphinx/annexes/EasyAutocomplete-master/demo/example_email.html

EasyAutocomplete - email example

enim.Curabitur.massa@
enim.Curabitur.massa@
convalliserat.edu - enim.
Curabitur.massa@convallis
erat.edu
Mauris.nulla@Curabitur
.net - Mauris.nulla@Cura
bitur.net

EasyAutocomplete-master/demo/example_email.html

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

```

1  <!DOCTYPE html>
2  <html lang="en">
3    <head>
4      <title>EasyAutocomplete email example</title>
5      <!-- STYLE -> CSS -->
6      <link href="../dist/easy-autocomplete.min.css" rel="stylesheet" type=
7      ↪ "text/css">
8      <!-- end STYLE-->
9    </head>
10   <body>
11
12     <!-- STRUCTURE -> HTML5 elements -->
13     <h1>EasyAutocomplete - email example</h1>

```

```
13     <input id="data-email" />
14     <!-- end STRUCTURE-->
15
16     <!--BEHAVIOR -> Javascript scripts-->
17     <script src="../../lib/jquery-1.11.2.min.js"></script>
18     <script src="../../dist/jquery.easy-autocomplete.min.js" type="text/
19     ↪javascript" ></script>
20     <script>
21         var options_easy_autocomplete = {
22             url: "resources/people.json",
23
24             getValue: "email",
25
26             template: {
27                 type: "description",
28                 fields: {
29                     description: "email"
30                 }
31             },
32
33             list: {
34                 match: {
35                     enabled: true
36                 }
37             },
38
39             theme: "plate-dark"
40         };
41
42         $("#data-email").easyAutocomplete(options_easy_autocomplete);
43     </script>
44     <!-- end BEHAVIOR -->
45
46 </body>
47 </html>
```

EasyAutocomplete-master/demo/resources/people.json

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

```
1  [
2    {
3      "name": "Declan Haley2 s",
4      "email": "Cr12as.lorem.lorem@nonquam.ca"
5    },
6    {
7      "name": "Francis Marsh",
8      "email": "ne2que@arcu.edu"
9    },
10   {
11     "name": "Gage Figueroa",
12     "email": "Sed.auctor.odio@magnis.ca"
13   },
14   {
15     "name": "Asher Gay",
16     "email": "Phasellus@nonsapien.ca"
```

```
16     },
17     {
18         "name": "Erasmus Roach",
19         "email": "amet@aptent.net"
20     },
21     {
22         "name": "Francis Johnston",
23         "email": "Ut.sagittis.lobortis@Donecsollicitudin.com"
24     },
25     {
26         "name": "Dustin Mckinney",
27         "email": "velit.Quisque@liberodui.co.uk"
28     },
29     {
30         "name": "Zane Camacho",
31         "email": "et.netus@Phasellusvitae.co.uk"
32     },
33     {
34         "name": "Murphy Larsen",
35         "email": "sit.amet@felisadipiscing.com"
36     },
37     {
38         "name": "Shad Kemp",
39         "email": "justo@laoreetlectus.co.uk"
40     },
41     {
42         "name": "Octavius Wolfe",
43         "email": "sociis@aptenttacitisociosqu.org"
44     },
45     {
46         "name": "Cooper Bell",
47         "email": "Quisque@apurus.ca"
48     },
49     {
50         "name": "Lev Mccarthy",
51         "email": "diam@amet.co.uk"
52     },
53     {
54         "name": "Arthur Mejia",
55         "email": "Nulla.dignissim@urna.ca"
56     },
57     {
58         "name": "Ferris Cameron",
59         "email": "vulputate.velit@In.net"
60     },
61     {
62         "name": "Ivor Greer",
63         "email": "nisi@Nuncmauriselit.ca"
64     },
65     {
66         "name": "Alan Church",
67         "email": "et.ipsu@dolor.co.uk"
68     },
69     {
70         "name": "Walter Berry",
71         "email": "neque@lacus.edu"
72     },
73     {
```

```
74     "name": "Hyatt Bowman",
75     "email": "lobortis.ultrices@mollis.org"
76 },
77 {
78     "name": "Acton Bradley",
79     "email": "commodo.tincidunt@ut.com"
80 },
81 {
82     "name": "Duncan Hancock",
83     "email": "Donec@sem.net"
84 },
85 {
86     "name": "Macaulay Trujillo",
87     "email": "in.sodales.elit@Donec.ca"
88 },
89 {
90     "name": "Jack Estrada",
91     "email": "nibh@Quisquetinciduntpede.net"
92 },
93 {
94     "name": "Edward Wilkerson",
95     "email": "Quisque.fringilla.euismod@idliberoDonec.co.uk"
96 },
97 {
98     "name": "Clarke Mullen",
99     "email": "nunc.Quisque.ornare@leo.com"
100 },
101 {
102     "name": "Steel Rodriquez",
103     "email": "a@diam.ca"
104 },
105 {
106     "name": "Jerome Edwards",
107     "email": "tincidunt@consectetuer.net"
108 },
109 {
110     "name": "Quentin Blevins",
111     "email": "In.sclerisque@idrisus.co.uk"
112 },
113 {
114     "name": "Maxwell Baxter",
115     "email": "lacinia.mattis@erateget.ca"
116 },
117 {
118     "name": "Beau McClain",
119     "email": "dis.parturient@feugiatnon.co.uk"
120 },
121 {
122     "name": "Jarrod Valenzuela",
123     "email": "adipiscing@FuscefeugiatLorem.net"
124 },
125 {
126     "name": "Grant Rosario",
127     "email": "feugiat@rutrumFuscedolor.co.uk"
128 },
129 {
130     "name": "Wade Atkinson",
131     "email": "et@euerosNam.com"
```

```
132     },
133     {
134         "name": "Ryder Lindsey",
135         "email": "Quisque.fringilla.euismod@Suspendissealiquetmolestie.edu"
136     },
137     {
138         "name": "Upton Schmidt",
139         "email": "arcu.Nunc.mauris@quis.ca"
140     },
141     {
142         "name": "Seth Tate",
143         "email": "elit.a.feugiat@turpisnon.edu"
144     },
145     {
146         "name": "Emery Shields",
147         "email": "mi.Aliquam.gravida@vestibulummassarutrum.net"
148     },
149     {
150         "name": "Slade Bolton",
151         "email": "venenatis.a@fringillami.co.uk"
152     },
153     {
154         "name": "Lucas Yates",
155         "email": "purus.in@idnuncinterdum.ca"
156     },
157     {
158         "name": "Dean Rosa",
159         "email": "vulputate.ullamcorper@commodoipsum.ca"
160     },
161     {
162         "name": "Ahmed Daniel",
163         "email": "ligula@idsapien.ca"
164     },
165     {
166         "name": "Hector Hardin",
167         "email": "egestas@Nullam.edu"
168     },
169     {
170         "name": "Byron Dickerson",
171         "email": "sociis@Proinultrices.co.uk"
172     },
173     {
174         "name": "Myles Workman",
175         "email": "magnis.dis@odioPhasellusat.co.uk"
176     },
177     {
178         "name": "Jerry Howard",
179         "email": "tortor.nibh@sitamet.edu"
180     },
181     {
182         "name": "Igor Chase",
183         "email": "sagittis@gravida.edu"
184     },
185     {
186         "name": "Aquila McClure",
187         "email": "dictum@Phasellusinfelis.co.uk"
188     },
189     {
```

```
190     "name": "John Hansen",
191     "email": "ut@molestietellus.com"
192 },
193 {
194     "name": "Forrest Goodwin",
195     "email": "enim.nec.tempus@massalobortis.com"
196 },
197 {
198     "name": "Griffin Lott",
199     "email": "enim.Curabitur.massa@convalliserat.edu"
200 },
```

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/>

```
|
| +---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
|
```

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.

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
  |   |   tagadd.svg
  |   |   tagarrowright.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
|       | tagadd.png
|       | tagarrowright.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
| | |         glyphsicons-halflings-regular.eot
| | |         glyphsicons-halflings-regular.svg
| | |         glyphsicons-halflings-regular.ttf
| | |         glyphsicons-halflings-regular.woff
| | |
| | |     +---img
| | |         glyphsicons-halflings-white.png
| | |         glyphsicons-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/projects/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__)))
```

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

EasyAutocomplete - easy example

The screenshot shows a web browser's developer console with the following content:

- Cookie **csrftoken** modifié. 7hJf5el13puuuIDTensfZ0aUDLTcQ7cc0lx3H8ieXSjBZTCzaeMMZlRtXYyz17Wx
- ReferenceError: reference to undefined property context.activeCommandLineHandlers
- ReferenceError: reference to undefined property this.context
- ReferenceError: reference to undefined property file.fromBFCache
- ReferenceError: reference to undefined property file.document.parent
- ReferenceError: reference to undefined property state.scrollTop
- NetworkError: 404 Not Found - http://127.0.0.1:8004/static/css/easy-autocomplete.css"
- NetworkError: 404 Not Found - http://127.0.0.1:8004/static/css/easy-autocomplete.themes.css"
- NetworkError: 404 Not Found - http://127.0.0.1:8004/static/js/jquery.easy-autocomplete.js"
- SyntaxError: test for equality (==) mistyped as assignment (=)?

```
while (ch = stream.next()) {  
    -----↑
```

```
# 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
|                   |   +---ally
|                   |   +---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
```

```

(django_test_autocomplete_35_64) C:\projects_id3\django-test-autocomplete\projet_ajax>python manage.py collectstatic
You have requested to collect static files at the destination
location as specified in your settings:
    c:\projects_id3\django-test-autocomplete\staticfiles
This will overwrite existing files!
Are you sure you want to do this?
Type 'yes' to continue, or 'no' to cancel: yes
Copying 'c:\projects_id3\django-test-autocomplete\projet_ajax\static\easyautocomplete\css\easy-autocomplete.css'
Copying 'c:\projects_id3\django-test-autocomplete\projet_ajax\static\easyautocomplete\css\easy-autocomplete.min.css'
Copying 'c:\projects_id3\django-test-autocomplete\projet_ajax\static\easyautocomplete\css\easy-autocomplete.themes.css'
Copying 'c:\projects_id3\django-test-autocomplete\projet_ajax\static\easyautocomplete\css\easy-autocomplete.themes.min.css'
Copying 'c:\projects_id3\django-test-autocomplete\projet_ajax\static\easyautocomplete\js\jquery.easy-autocomplete.js'
Copying 'c:\projects_id3\django-test-autocomplete\projet_ajax\static\easyautocomplete\js\jquery.easy-autocomplete.min.js'
Found another file with the destination path 'admin\js\jquery.init.js'. It will be ignored since only the first encountered file is collected. If this is not what you want, make sure every static file has a unique path.
6 static files copied to 'c:\projects_id3\django-test-autocomplete\staticfiles', 281 unmodified.
(django_test_autocomplete_35_64) C:\projects_id3\django-test-autocomplete\projet_ajax>

```

```
|
\---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

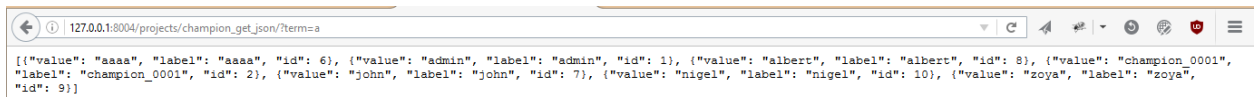
```
(django_test_autocomplete_35_64) C:\projects_id3\django-test-autocomplete\proj_ajax>python manage.py collectstatic
You have requested to collect static files at the destination
location as specified in your settings:
    C:\projects_id3\django-test-autocomplete\staticfiles
This will overwrite existing files!
Are you sure you want to do this?
Type 'yes' to continue, or 'no' to cancel: yes
Copying 'C:\projects_id3\django-test-autocomplete\proj_ajax\static\easyautocomplete\data/categories.json'
Copying 'C:\projects_id3\django-test-autocomplete\proj_ajax\static\easyautocomplete\data/countries.json'
Copying 'C:\projects_id3\django-test-autocomplete\proj_ajax\static\easyautocomplete\data/countries.xml'
Copying 'C:\projects_id3\django-test-autocomplete\proj_ajax\static\easyautocomplete\data/flags.css'
Copying 'C:\projects_id3\django-test-autocomplete\proj_ajax\static\easyautocomplete\data/flags.png'
Copying 'C:\projects_id3\django-test-autocomplete\proj_ajax\static\easyautocomplete\data/icon_search.png'
Copying 'C:\projects_id3\django-test-autocomplete\proj_ajax\static\easyautocomplete\data/people.json'
Copying 'C:\projects_id3\django-test-autocomplete\proj_ajax\static\easyautocomplete\data/site.json'
Found another file with the destination path 'admin/js/jquery.init.js'. It will be ignored since only the first encountered file is collected. If this is not what you want, make sure every static file has a unique path.
8 static files copied to 'C:\projects_id3\django-test-autocomplete\staticfiles', 287 unmodified.
(django_test_autocomplete_35_64) C:\projects_id3\django-test-autocomplete\proj_ajax>
```

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



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

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 httpie)
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",
```

The screenshot displays a web browser window with the address bar showing `127.0.0.1:8004/projects/project/1/update/`. The page title is "EasyAutocomplete examples". It contains three sections:

- Input simple:** A single text input field.
- Input data-json:** A single text input field.
- Input AJAX:** A text input field with a dropdown menu showing suggestions: "aaaa", "admin", "albert", "champion_0001", "john", and "nigel". The input field contains the letter "a".

Below the browser window, a terminal window shows the Django development server logs. The logs indicate the server is running on `http://127.0.0.1:8004/` and show various HTTP requests, including GET requests for static files and AJAX calls to `/projects/champion_get_json/?term=a`.

```

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

- 2016-10-19 improve the Django templates + unobstrusive Javascript (step1)
 - The Javascript code must be in the the body element
 - Modify names
 - Test champion_get_json : http://127.0.0.1:8004/projects/champion_get_json/?term=a
 - Rendering fields manually
 - Adding django-debug-toolbar
 - First step for a better look

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.

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.

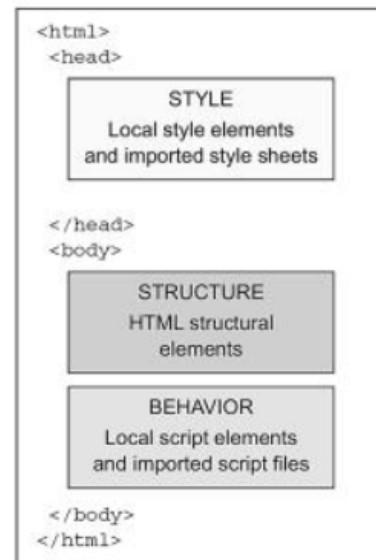


Figure 1.2 With structure, style, and behavior each neatly tucked away within a page, readability and maintainability are maximized.

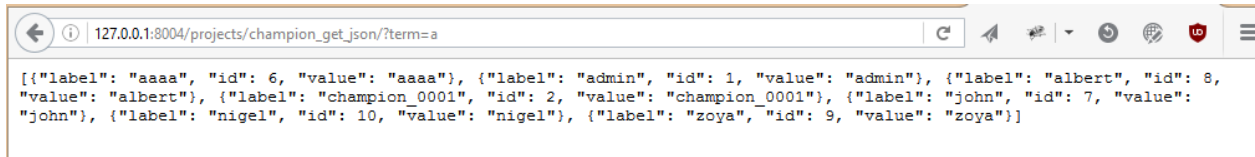
Modify names

- `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
'django_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/>

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

Update of the project '(title:project test champion:admin)

Title	<input type="text" value="project test"/>	
Champion	<input type="text"/>	admin

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

- *2016-10-18 improve the jquery-ui autocomplete look and feel*
 - *Last look*
 - *jquery-ui autocomplete options already used*

* *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.
```

```

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-
11     ui.css">
12
13     <style>
14         {% https://jqueryui.com/autocomplete/#maxheight #}
15         .ui-autocomplete {
16             max-height: 100px;
17             overflow-y: auto;
18             /* prevent horizontal scrollbar */
19             overflow-x: hidden;
20         }
21         * html .ui-autocomplete {
22             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>
52                     </tr>
53                 </tbody>
54             </table>
55             <div>
56                 <input type="submit" name="btn_update" value="Update" class="btn btn-success _
↪btn-block" />
57             </div>
58
59
60         {% if messages %}
61         <ul class="messages">
62             {% comment %} https://getbootstrap.com/components/#alerts {% endcomment %}
63             {% for message in messages %}
64             <!--li{% if message.tags %} class="{{ message.tags }}" {% endif %} -->
65                 {% if message.level == DEFAULT_MESSAGE_LEVELS.ERROR %}
66                     <div class="alert alert-danger" role="alert">{{ message }}</div>
67                 {% elif message.level == DEFAULT_MESSAGE_LEVELS.SUCCESS %}
68                     <div class="alert alert-success" role="alert">{{ message }}</div>
69                 {% else %}
70                     <div class="alert alert-info" role="alert">{{ message }}</div>
71                 {% endif %}
72             <!-- /li -->
73             {% endfor %}
74         </ul>

```

```

75     {% endif %}
76
77
78 {% endblock content %}
79
80
81 {% BEHAVIOR Javascript %}
82 {% block javascript %}
83     {{ block.super }}
84
85     {% http://code.jquery.com/ui/ %}
86     <script src="http://code.jquery.com/ui/1.12.1/jquery-ui.min.js"></script>
87
88 {% endblock javascript %}
89
90
91 {% DOM ready %}
92 {% block domready %}
93
94     $( "#id_champion_term" ).autocomplete({
95         {% http://api.jqueryui.com/autocomplete/#option-source %}
96         source: "{% url 'projects:champion_get_json' %}",
97         {% http://api.jqueryui.com/autocomplete/#option-autoFocus %}
98         autoFocus:true,
99         {% http://api.jqueryui.com/autocomplete/#option-minLength %}
100        minLength:1,
101        {% http://api.jqueryui.com/autocomplete/#event-select %}
102        select:function(event,ui) {
103            $("#id_champion").val(ui.item.id);
104        }
105    });
106
107 {% endblock %}

```

```

1  #!/usr/bin/python
2  # -*- coding: utf8 -*-
3  """The project's forms.
4
5  """
6
7  from django import forms
8
9  from .models import Project
10
11 class ProjectChampionForm(forms.ModelForm):
12     """The champion project form"""
13     champions_choice_list = forms.CharField(max_length=100,
14                                             help_text='type username or email')
15
16     class Meta:
17         model = Project
18         fields = ('title',
19                 'champions_choice_list', 'champion',)
20
21     def __init__(self, *args, **kwargs):
22         super(ProjectChampionForm, self).__init__(*args, **kwargs)
23         self.fields['champions_choice_list'].label = "Update the champion"
24         self.fields['champion'].widget = forms.HiddenInput()

```

jquery-ui autocomplete options already used

Remote option

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

The Autocomplete widget 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: "annhxx10", 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>

Autocomplete Widget

Categories: [Widgets](#)

Autocomplete Widget
version added: 1.8

Description: *Autocomplete enables users to quickly find and select from a pre-populated list of values as they type, leveraging searching and filtering.*

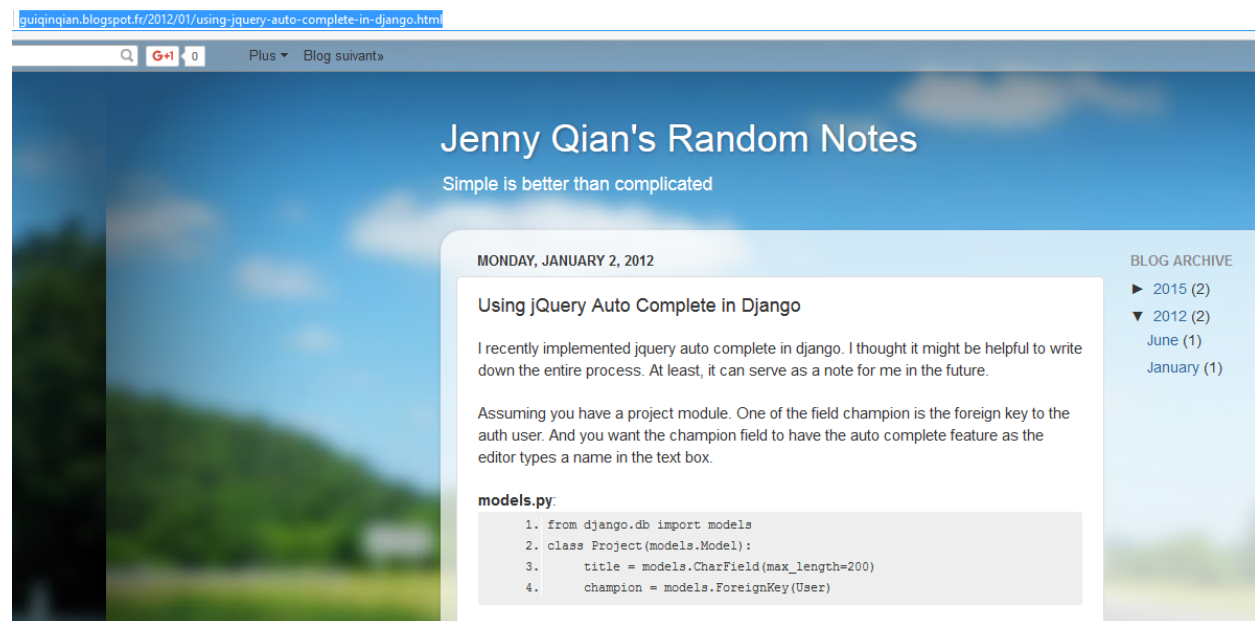
QuickNav		Examples
Options appendTo autoFocus classes delay disabled minLength position source	Methods close destroy disable enable instance option search widget	Events change close create focus open response search select
Extension Points renderItem renderMenu resizeMenu		

Contents

- 2016-10-18 try a more simple approach with jquery-ui autocomplete widget
 - Installation
 - Create a projects application
 - Add the Django model Project
 - Add the project in the INSTALLED_APPS

- *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

```
1  #!/usr/bin/python
2  # -*- coding: utf8 -*-
3  """The project's models.
4
5  """
6  from __future__ import unicode_literals
7  from django.utils.encoding import python_2_unicode_compatible
8
9  from django.db import models
10 from django.core.urlresolvers import reverse
11
12 # https://docs.djangoproject.com/en/dev/ref/contrib/auth/#user-model
13 from django.contrib.auth.models import User
14
15
16
17 @python_2_unicode_compatible
18 class Project(models.Model):
19     """A project with a title and a champion which is the foreign key to the
20     auth user.
21
22     Documentation
23     =====
24
25     - http://guiqinqian.blogspot.fr/2012/01/using-jquery-auto-complete-in-django.html
26
27     """
28     title = models.CharField(max_length=200)
29     champion = models.ForeignKey(User)
30
31     def __str__(self):
32         return "{} {}".format(self.title, self.champion)
33
34     def get_absolute_url(self):
35         """
```

```
36     https://docs.djangoproject.com/en/dev/ref/class-based-views/generic-editing/
37     """
38     return reverse('projects:project_update',
39                    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'
]
```

```
1  """
2  Django settings for projet_ajax project.
3
4  Generated by 'django-admin startproject' using Django 1.10.2.
5
6  For more information on this file, see
7  https://docs.djangoproject.com/en/1.10/topics/settings/
8
9  For the full list of settings and their values, see
10 https://docs.djangoproject.com/en/1.10/ref/settings/
11 """
12
13 import os
14 import logging
15
16 from os.path import (join,
17                      basename,
18                      dirname)
19
20 # Build paths inside the project like this: os.path.join(BASE_DIR, ...)
21 DJANGO_ROOT = BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))
22 # Site name ('projet_ajax')
23 SITE_NAME = basename(DJANGO_ROOT)
24 # Absolute filesystem path to the top-level project folder:
25 PROJECT_ROOT=dirname(DJANGO_ROOT)
26
27 logging.info("DJANGO_ROOT={} SITE_NAME={} PROJECT_ROOT={}".format(
28     DJANGO_ROOT,
29     SITE_NAME,
30     PROJECT_ROOT,
31 ))
32
33 # Quick-start development settings - unsuitable for production
34 # See https://docs.djangoproject.com/en/1.10/howto/deployment/checklist/
35
36 # SECURITY WARNING: keep the secret key used in production secret!
37 SECRET_KEY = 'q#e(dtsmq9^!(%9*un6tuq5v)2!@00ftgk$!@h1*(e0iwo@=cc'
38
39 # SECURITY WARNING: don't run with debug turned on in production!
40 DEBUG = True
41
42 # for the debug toolbar
43 INTERNAL_IPS = '127.0.0.1'
```

```

44 DEBUG_TOOLBAR_PANELS = [
45     'debug_toolbar.panels.versions.VersionsPanel',
46     'debug_toolbar.panels.timer.TimerPanel',
47     'debug_toolbar.panels.settings.SettingsPanel',
48     'debug_toolbar.panels.headers.HeadersPanel',
49     'debug_toolbar.panels.request.RequestPanel',
50     'debug_toolbar.panels.sql.SQLPanel',
51     'debug_toolbar.panels.staticfiles.StaticFilesPanel',
52     'debug_toolbar.panels.templates.TemplatesPanel',
53     'debug_toolbar.panels.cache.CachePanel',
54     'debug_toolbar.panels.signals.SignalsPanel',
55     'debug_toolbar.panels.logging.LoggingPanel',
56     'debug_toolbar.panels.redirects.RedirectsPanel',
57 ]
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-
82     ↪ 'django-crispy-forms',
83     'crispy_forms',
84     # https://github.com/jazzband/django-debug-toolbar
85     # 'debug_toolbar',
86
87     # # https://docs.djangoproject.com/en/dev/ref/applications/#django.apps.AppConfig
88     'singers.apps.SingersConfig',
89     # http://guiqinqian.blogspot.fr/2012/01/using-jquery-auto-complete-in-django.html
90     'projects.apps.ProjectsConfig'
91 ]
92
93 MIDDLEWARE = [
94     'django.middleware.security.SecurityMiddleware',
95     'django.contrib.sessions.middleware.SessionMiddleware',
96     'django.middleware.common.CommonMiddleware',
97     'django.middleware.csrf.CsrfViewMiddleware',
98     'django.contrib.auth.middleware.AuthenticationMiddleware',
99     'django.contrib.messages.middleware.MessageMiddleware',
100     'django.middleware.clickjacking.XFrameOptionsMiddleware',

```

```
101     # 'debug_toolbar.middleware.DebugToolbarMiddleware'
102 ]
103
104
105 ROOT_URLCONF = 'projet_ajax.urls'
106
107 TEMPLATES = [
108     {
109         'BACKEND': 'django.template.backends.django.DjangoTemplates',
110         'DIRS': [join(BASE_DIR, 'templates')],
111         'APP_DIRS': True,
112         'OPTIONS': {
113             'context_processors': [
114                 'django.template.context_processors.debug',
115                 'django.template.context_processors.request',
116                 'django.contrib.auth.context_processors.auth',
117                 'django.contrib.messages.context_processors.messages',
118             ],
119         },
120     ],
121 ]
122
123 WSGI_APPLICATION = 'projet_ajax.wsgi.application'
124
125
126 # Database
127 # https://docs.djangoproject.com/en/1.10/ref/settings/#databases
128 DATABASES = {
129     'default': {
130         'ENGINE': 'django.db.backends.sqlite3',
131         'NAME': os.path.join(BASE_DIR, 'ajax_selects_singers_db')
132     }
133 }
134
135 # Password validation
136 # https://docs.djangoproject.com/en/1.10/ref/settings/#auth-password-validators
137
138 AUTH_PASSWORD_VALIDATORS = [
139     {
140         'NAME': 'django.contrib.auth.password_validation.
141 ↪UserAttributeSimilarityValidator',
142     },
143     {
144         'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
145     },
146     {
147         'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
148     },
149     {
150         'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
151     },
152 ]
153
154 # Internationalization
155 # https://docs.djangoproject.com/en/1.10/topics/i18n/
156
157 LANGUAGE_CODE = 'en-us'
```

```

158 TIME_ZONE = 'UTC'
159
160
161 USE_I18N = True
162
163 USE_L10N = True
164
165 USE_TZ = True
166
167
168 # Static files (CSS, JavaScript, Images)
169 # https://docs.djangoproject.com/en/dev/howto/static-files/
170 STATIC_ROOT = join(PROJECT_ROOT, 'staticfiles')
171 # STATIC_ROOT="C:/projects_id3/P5N001/XLOG5R372_id3_intranet/trunk/django-www/
    ↳staticfiles"
172 # Static files (CSS, JavaScript, Images)
173 # http://www.marinamele.com/taskbuster-django-tutorial/create-home-page-with-tdd-
    ↳staticfiles-templates-settings
174 # https://docs.djangoproject.com/en/dev/howto/static-files/
175 # http://whitenoise.evans.io/en/latest/django.html#runserver-nostatic
176 STATIC_URL = '/static/'
177 # https://docs.djangoproject.com/en/dev/howto/static-files/
178 # les répertoires où sont stockés les fichiers statiques en développement
179 STATICFILES_DIRS = [
180     join(DJANGO_ROOT, 'static'),
181 ]
182
183 AJAX_LOOKUP_CHANNELS = {
184     # simplest way, automatically construct a search channel by passing a dict
185     'label': {'model': 'singers.label'
186              , 'search_field': 'name'},
187
188     # Custom channels are specified with a tuple
189     # channel: ( module.where_lookup_is, ClassNameOfLookup )
190     # 'person': ('singers.lookups', 'PersonLookup'),
191     # 'group': ('singers.lookups', 'GroupLookup'),
192     # 'song': ('singers.lookups', 'SongLookup'),
193 }
194
195 # voir https://github.com/jordij/bctt.nz/blob/master/ttwellington/settings/base.py
196 MEDIA_ROOT = join(BASE_DIR, 'media')
197 MEDIA_URL = '/media/'
198

```

makemigrations

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

```

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

```

```

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

```

```
4
5 from django.conf import settings
6 from django.db import migrations, models
7 import django.db.models.deletion
8
9
10 class Migration(migrations.Migration):
11
12     initial = True
13
14     dependencies = [
15         migrations.swappable_dependency(settings.AUTH_USER_MODEL),
16     ]
17
18     operations = [
19         migrations.CreateModel(
20             name='Project',
21             fields=[
22                 ('id', models.AutoField(auto_created=True, primary_key=True,
23 ↪ serialize=False, verbose_name='ID')),
24                 ('title', models.CharField(max_length=200)),
25                 ('champion', models.ForeignKey(on_delete=django.db.models.deletion.
26 ↪ CASCADE, to=settings.AUTH_USER_MODEL)),
27     ],
28     ),
29 ]
```

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

```

1  #!/usr/bin/python
2  # -*- coding: utf8 -*-
3  """The project's forms.
4
5  """
6
7  from django import forms
8
9  from .models import Project
10
11 class ProjectChampionForm(forms.ModelForm):
12     """The champion project form"""
13     champions_choice_list = forms.CharField(max_length=100,
14                                             help_text='type username or email')
15
16     class Meta:
17         model = Project
18         fields = ('title',
19                 'champions_choice_list', 'champion',)
20
21     def __init__(self, *args, **kwargs):
22         super(ProjectChampionForm, self).__init__(*args, **kwargs)
23         self.fields['champions_choice_list'].label = "Update the champion"
24         self.fields['champion'].widget = forms.HiddenInput()

```

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

```

1  # -*- coding: utf-8 -*-
2
3  from django.conf.urls import url
4
5
6  from .views_json import (ApiEACGetProjectsJSONView,
7                          ApiEACGetChampionsJSONView)
8
9  from .views_django_autocomplete_light import (ApiUserDjangoAutocompleteLight,
10                                              ProjectDjangoAutoCompleteUpdateView,
11                                              ProjectAutocompleteView)
12
13  from .views import (ProjectUpdateView,
14                    ProjectUpdateViewEasyAutoComplete,
15                    ProjectUpdateViewjQueryUIAutoComplete)
16
17
18  urlpatterns = [
19     url(r'^project/(?P<pk>\d+)/update/$',
20         ProjectUpdateView.as_view(),
21         name='project_update'),
22
23     url(r'^project/(?P<pk>\d+)/updateeasy/$',
24         ProjectUpdateViewEasyAutoComplete.as_view(),
25         name='project_update_easy'),
26
27     url(r'^project/(?P<pk>\d+)/update_jquery_ui/$',

```

```
28     ProjectUpdateViewJQueryUIAutoComplete.as_view(),
29     name='project_update_jquery_ui'),
30
31     url(r'^project/(?P<pk>\d+)/update_django_autocomplete/$',
32         ProjectDjangoAutoCompleteUpdateView.as_view(),
33         name='project_update_django_autocomplete'),
34
35     # calls by jquery EasyAutocomplete
36     # http://127.0.0.1:8004/projects/api_get_champions/?term=a
37     url(r'^api_get_champions/$',
38         ApiEACGetchampionsJSONView.as_view(),
39         name='api_get_champions'),
40
41     # calls by jquery EasyAutocomplete (EAC)
42     # http://127.0.0.1:8004/projects/api_get_projects/?term=a
43     url(r'^apis_get_projects/$',
44         ApiEACGetProjectsJSONView.as_view(),
45         name='api_get_projects'),
46
47     # calling example:
48     # http://127.0.0.1:8004/projects/project_autocomplete/?q=a
49     url(
50         r'^project_autocomplete/$',
51         ProjectAutocompleteView.as_view(),
52         name='project_autocomplete',
53     ),
54
55     url(r'^api/get_users/$',
56         ApiUserDjangoAutocompleteLight.as_view(),
57         name='api_get_users'),
58 ]
```

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

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

```
1  #!/usr/bin/python
2  # -*- coding: utf8 -*-
3  """projet_ajax URL Configuration
4
5  The `urlpatterns` list routes URLs to views. For more information please see:
6      https://docs.djangoproject.com/en/1.10/topics/http/urls/
7  Examples:
8  Function views
9      1. Add an import:  from my_app import views
10     2. Add a URL to urlpatterns:  url(r'^$', views.home, name='home')
11  Class-based views
12     1. Add an import:  from other_app.views import Home
13     2. Add a URL to urlpatterns:  url(r'^$', Home.as_view(), name='home')
14  Including another URLconf
15     1. Import the include() function: from django.conf.urls import url, include
16     2. Add a URL to urlpatterns:  url(r'^blog/', include('blog.urls'))
17  """
18
19  from django.conf import settings
20  from django.conf.urls import (include,
```

```

21         url)
22
23 from django.contrib import admin
24
25 from ajax_select import urls as ajax_select_urls
26 from singers import views
27
28
29 urlpatterns = [
30     url(r'^admin/', admin.site.urls),
31
32     url(r'^search',
33         view=views.search_form,
34         name='search_form'),
35     url(r'^admin/lookups/', include(ajax_select_urls)),
36
37     url(r'^projects/', include('projects.urls', namespace='projects')),
38     url(r'^singers/', include('singers.urls', namespace='singers')),
39 ]
40
41
42 if settings.DEBUG:
43     from django.contrib.staticfiles.urls import staticfiles_urlpatterns
44     from django.conf.urls.static import static
45     urlpatterns += staticfiles_urlpatterns()
46     # voir p.405 "Django By Example"
47     # https://docs.djangoproject.com/en/dev/howto/static-files/#serving-files-
48     ↳uploaded-by-a-user-during-development
49     urlpatterns += static(settings.MEDIA_URL ,
50                           document_root=settings.MEDIA_ROOT)
51
52     import debug_toolbar
53     urlpatterns += [
54         url(r'^__debug__/', include(debug_toolbar.urls)),
55     ]

```

Add the ProjectUpdateView

```

1
2
3 import logging
4
5 from django.core.urlresolvers import reverse
6
7 from django.views.generic.edit import UpdateView
8 from django.http import HttpResponseRedirect
9
10
11 from .models import Project
12 from .forms import ProjectChampionForm
13
14
15 # Get an instance of a logger
16 logger = logging.getLogger(__name__)
17
18

```

```

19 class ProjectUpdateView(UpdateView):
20     """
21         url(r'^projects/project/(?P<pk>\d+)/update/$', ProjctUpdate.as_view(),
↪ name='project_update'),
22
23     Documentation:
24
25     - http://ccbv.co.uk/projects/Django/1.10/django.views.generic.edit/UpdateView/
26
27     """
28     model = Project
29     form_class = ProjectChampionForm
30     context_object_name = 'project'
31     template_name = 'projects/project/update_easy_simple.html'
32
33     def get_object(self, queryset=None):
34         """Pour mémoriser self.demande_article"""
35         self.object = super(ProjectUpdateView, self).get_object(queryset)
36         return self.object
37
38     def post(self, request, *args, **kwargs):
39         logger.warning("Hello from ProjectUpdateView !")
40         if "cancel" in request.POST:
41             url = self.get_success_url()
42             return HttpResponseRedirect(url)
43         else:
44             return super(ProjectUpdateView, self).post(request, *args, **kwargs)
45
46
47
48
49 class ProjectUpdateViewEasyAutoComplete(UpdateView):
50     """Update thye view with the jQuery EasyAutocomplete plugin.
51
52     Documentation:
53
54     - http://ccbv.co.uk/projects/Django/1.10/django.views.generic.edit/UpdateView/
55
56     """
57     model = Project
58     form_class = ProjectChampionForm
59     context_object_name = 'project'
60     template_name = 'projects/project/update_easyautocomplete.html'
61
62     def get_object(self, queryset=None):
63         """Pour mémoriser self.demande_article"""
64         self.object = super(ProjectUpdateViewEasyAutoComplete, self).get_
↪ object(queryset)
65         return self.object
66
67     def post(self, request, *args, **kwargs):
68         logger.warning("Hello from ProjectUpdateViewEasyAutoComplete !")
69         if "cancel" in request.POST:
70             url = reverse('projects:project_update_easy', kwargs={'pk': self.pk})
71             return HttpResponseRedirect(url)
72         else:
73             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.css">
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.
31 ↪champion }})' </h1>
32     <p></p>
33     <p></p>
34     {# https://docs.djangoproject.com/en/dev/topics/forms/ #}
35     <form id="id_form_project_update" action="{% url 'projects:project_update' _
36 ↪project.id %}" method="post">
37         {% csrf_token %}
38         <div class="forms">
39             {{ form.id }}
40             {{ form.non_field_errors }}
41             {# Include the hidden fields #}
42             {% for hidden in form.hidden_fields %}
43                 {{ hidden }}
44             {% endfor %}
45             <table id="id_table" class="table table-hover table-bordered table-
46 ↪condensed">
47                 <tbody>
48                     <tr>
49                         <td class="text-right">Title:</td>
50                         <td>{{ form.title }}</td>
51                     </tr>
52                     <tr>
53                         <td class="text-right">Champion:</td>
54                         <td> <span id="id_champion_display">{{ project.champion }}
55 ↪</span> {{ form.champion_term }} </td>

```

```

52         </tr>
53     </tbody>
54 </table>
55 </div>
56 <input type="submit" name="btn_update" value="Update" class="btn btn-success_
↪ btn-block" />
57 </form>
58
59
60 {% if messages %}
61 <ul class="messages">
62     {% comment %} https://getbootstrap.com/components/#alerts {% endcomment %}
63     {% for message in messages %}
64 <!--li{% if message.tags %} class="{% message.tags %}"{% endif %} -->
65         {% if message.level == DEFAULT_MESSAGE_LEVELS.ERROR %}
66             <div class="alert alert-danger" role="alert">{{ message }}</div>
67         {% elif message.level == DEFAULT_MESSAGE_LEVELS.SUCCESS %}
68             <div class="alert alert-success" role="alert">{{ message }}</div>
69         {% else %}
70             <div class="alert alert-info" role="alert">{{ message }}</div>
71         {% endif %}
72     <!-- /li -->
73     {% endfor %}
74 </ul>
75 {% endif %}
76
77
78 {% endblock content %}
79
80
81 {% # BEHAVIOR Javascript #}
82 {% block javascript %}
83     {{ block.super }}
84
85     {% # http://code.jquery.com/ui/ #}
86     <script src="http://code.jquery.com/ui/1.12.1/jquery-ui.min.js"></script>
87
88 {% endblock javascript %}
89
90
91 {% # DOM ready #}
92 {% block domready %}
93
94     $( "#id_champion_term" ).autocomplete({
95         {% # http://api.jqueryui.com/autocomplete/#option-source #}
96         source: "{% url 'projects:champion_get_json' %}",
97         {% # http://api.jqueryui.com/autocomplete/#option-autoFocus #}
98         autoFocus:true,
99         {% # http://api.jqueryui.com/autocomplete/#option-minLength #}
100        minLength:1,
101        {% # http://api.jqueryui.com/autocomplete/#event-select #}
102        select:function(event,ui) {
103            $("#id_champion").val(ui.item.id);
104        }
105    });
106
107 {% 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

```
1  #!/usr/bin/python
2  # -*- coding: UTF-8 -*-
3  """Project Administration.
4
5  """
6
7  from django.contrib import admin
8
9  from .models import Project
10
11 from .forms_django_autocomplete_light import ProjectFormDjangoAutocomplete
12
13 @admin.register(Project)
14 class ProjectAdmin(admin.ModelAdmin):
15     """Project administration
16
17     Documentation
18     =====
19
20     - https://docs.djangoproject.com/en/dev/ref/contrib/admin/#modeladmin-objects
21
22     """
23     form = ProjectFormDjangoAutocomplete
24     list_display = ('title', 'champion')
25     search_fields = ('title', 'champion')
26     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/

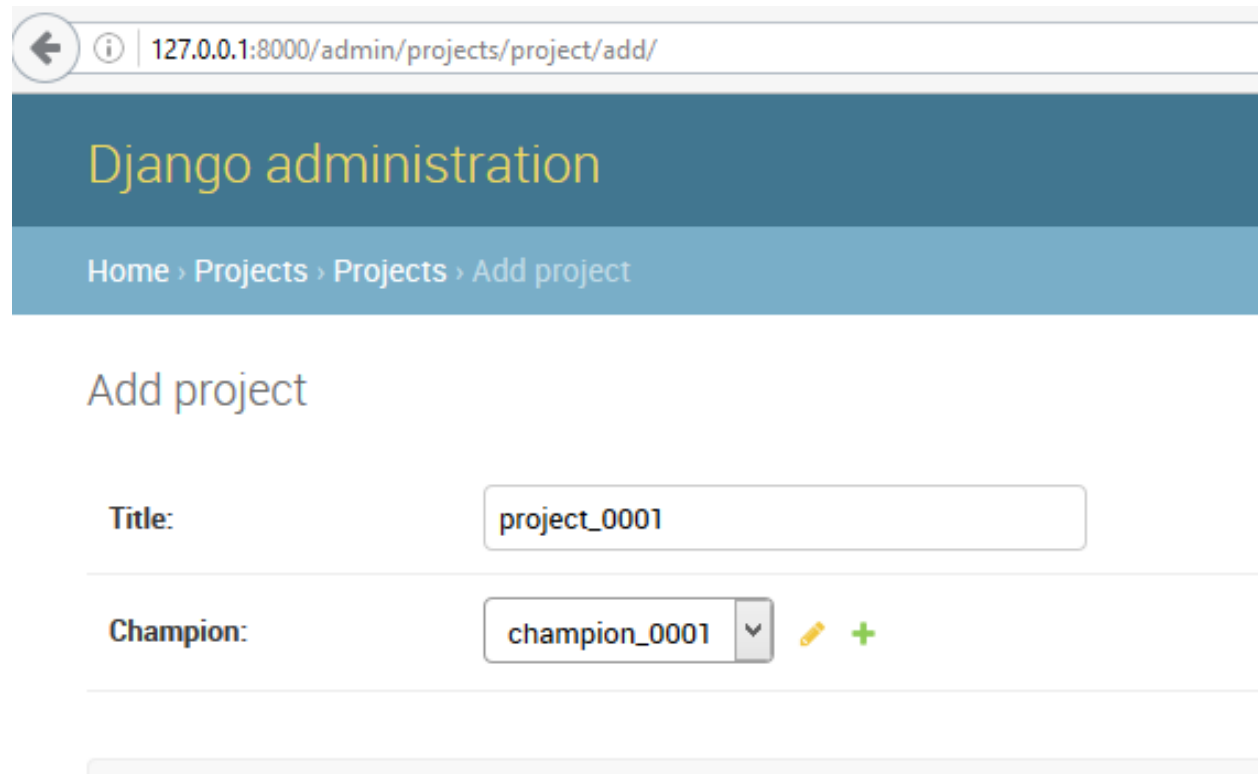
Django administration

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






127.0.0.1:8000/admin/projects/project/add/

Django administration

Home > Projects > Projects > Add project

Add project

Title:

Champion:   

```
(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.

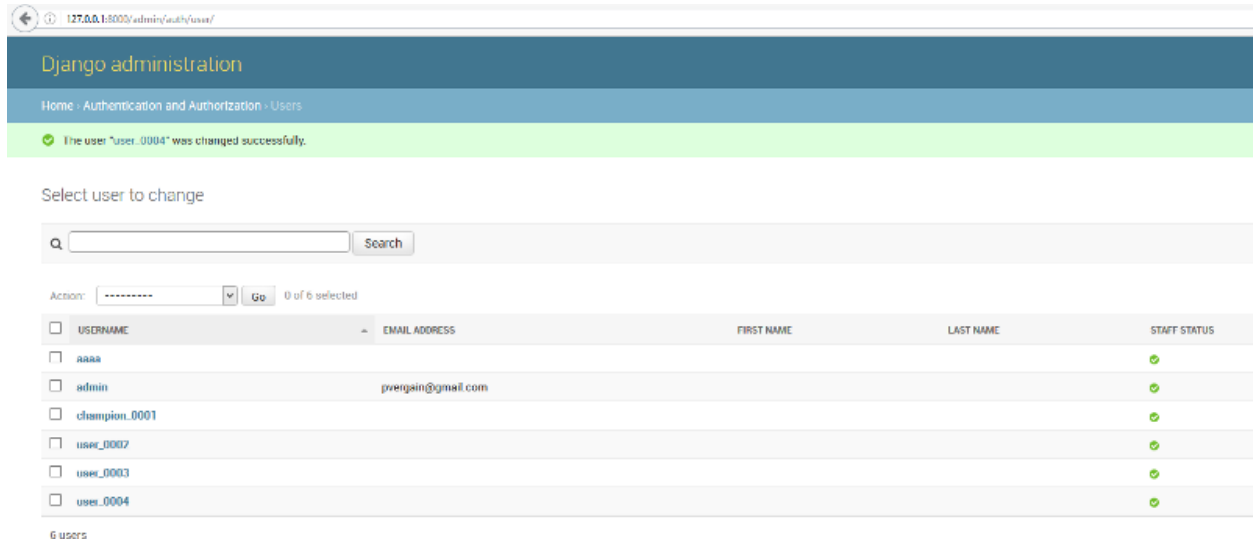
IPython 5.1.0 -- An enhanced Interactive Python.
? -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help -> Python's own help system.
object? -> Details about 'object', use 'object??' for extra details.

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.

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

Update of the project '(title:project_0001 champion:zoya)

Title: Update the champion:

type username or email

zoya

Update

```
[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

- 2016-10-18 create some Django Templates : base.html
 - The Django Template templates/base.html
 - The Django Template singers/templates/song/update.html
 - Add the django-extensions and ipython modules
 - * New django commands
 - Get the first song
 - Try the URL : `http://127.0.0.1:8000/singers/song/1/update`
 - Compared to the django admin interface
 - Conclusion

The Django Template templates/base.html

```
1 {% https://docs.djangoproject.com/en/dev/ref/templates/builtins/#std:templatetag-
   ↳ static %}
2 {% load static %}
3 {% https://docs.djangoproject.com/en/dev/ref/templates/builtins/#tz %}
4 {% load tz %}
5
6 <!DOCTYPE html>
```

```

7  {# https://www.w3.org/TR/2014/REC-html5-20141028/syntax.html#the-doctype #}
8  <!--[if lt IE 7]>      <html class="no-js lt-ie9 lt-ie8 lt-ie7"> <![endif]-->
9  <!--[if IE 7]>         <html class="no-js lt-ie9 lt-ie8"> <![endif]-->
10 <!--[if IE 8]>         <html class="no-js lt-ie9"> <![endif]-->
11 <!--[if gt IE 8]><!--> <html class="no-js"> <!--<![endif]-->
12 {# https://www.w3.org/TR/html5/semantics.html#the-html-element #}
13 <head>
14     {# https://www.w3.org/TR/html5/document-metadata.html#attr-meta-charset #}
15     <meta charset="utf-8">
16     {# https://n.survol.fr/n/x-ua-compatible-ieedge #}
17     <!-- meta http-equiv="X-UA-Compatible" content="IE=edge" -->
18     {# https://www.w3.org/TR/html5/dom.html#the-title-attribute #}
19     {# https://www.w3.org/TR/html5/document-metadata.html#the-title-element #}
20     <title>
21         {% block head_title %}
22         {% endblock %}
23     </title>
24
25     {# https://www.w3.org/TR/html5/document-metadata.html#attr-meta-name #}
26
27     {# https://developer.mozilla.org/fr/docs/Mozilla/Mobile/Balise_meta_viewport
28     {#}
29     <meta name="viewport" content="width=device-width, initial-scale=1" />
30
31     {# CSS #}
32     {% block stylesheet %}
33         {# https://docs.djangoproject.com/en/dev/ref/templates/language/#template-
34         {# If you need to get the content of the block from the parent template,
35         {# the {{ block.super }} variable will do the trick #}
36         {# {{ block.super }} #}
37
38         {# https://blog.getbootstrap.com/2016/07/25/bootstrap-3-3-7-released
39         {# (2016-07-25) Added support for jQuery 3 #}
40         {# https://www.bootstrapcdn.com/ #}
41         <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.
42         {# 3.7/css/bootstrap.min.css">
43         <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.
44         {# 3.7/css/bootstrap-theme.min.css">
45
46         {% endblock %}
47
48
49
50 </head>
51 <body class="{% block bodyclass %} {% endblock %}">
52
53     <div id="conteneur" class="container">
54         {% block content %}
55         {% endblock %}
56
57         {% block footer %}
58         {% endblock %}
59
60     </div> <!-- id conteneur -->

```

```

58     {% Javascript %}
59     {% block javascript %}
60
61         {# a employer dans les templates fils->{{ block.super }} #}
62         {# https://docs.djangoproject.com/en/dev/ref/templates/language/#template-
63         ↳inheritance #}
64         {# If you need to get the content of the block from the parent template,
65         ↳the {{ block.super }} variable will do the trick #}
66         <!-- Liaison aux scripts Javascript -->
67         <!-- CDN JQuery Placer jquery avant bootstrap, select2 -->
68         {# https://github.com/jquery/jquery-migrate/#README #}
69         {# https://blog.jquery.com/2016/09/22/jquery-3-1-1-released/ #}
70         {# https://code.jquery.com/ #}
71         <script src="https://code.jquery.com/jquery-3.1.1.min.js"></script>
72         <script src="https://code.jquery.com/jquery-migrate-3.0.0.min.js"></
73         ↳script>
74
75         <!-- Code javascript pour bootstrap 3.3.7 (2016-07-25) -->
76         <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.
77         ↳min.js"> </script>
78
79         {% endblock javascript %}
80
81         {# p.155 Adding AJAX actions with jQuery, Django By Example from Antonio Melé
82         ↳#}
83         {# https://www.amazon.com/Django-Example-Antonio-Mele/dp/1784391913/ #}
84         <script>
85             $(document).ready(function() {
86                 {% block domready %}
87                 {% endblock %}
88             });
89         </script>
90
91     </body>
92 </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**.

```

1  {% extends "base.html" %}
2  {% load static %}
3  {% load staticfiles %}
4  {% load crispy_forms_tags %}
5
6  {% block content %}
7
8  <h1>Update of the song {{ song.title }} {{ song.group }} </h1>
9
10  {# https://docs.djangoproject.com/en/dev/topics/forms/ #}
11  <form id="song_update" method="post" action=".">
12      {% csrf_token %}
13      {{ form }}
14
15      {# https://django-ajax-selects.readthedocs.io/en/latest/Outside-of-Admin.html #}
16      {{ form.media }}
17      <input type="submit" name="btn_update" value="Update" class="btn btn-success btn-
18  ↪block" />
19  </form>
20
21  {% if messages %}
22      <ul class="messages">
23          {% comment %} https://getbootstrap.com/components/#alerts {% endcomment %}
24          {% for message in messages %}
25              <!--li{% if message.tags %} class="{{ message.tags }}" {% endif %} -->
26              {% if message.level == DEFAULT_MESSAGE_LEVELS.ERROR %}
27                  <div class="alert alert-danger" role="alert">{{ message }}</div>
28              {% elif message.level == DEFAULT_MESSAGE_LEVELS.SUCCESS %}
29                  <div class="alert alert-success" role="alert">{{ message }}</div>
30              {% else %}
31                  <div class="alert alert-info" role="alert">{{ message }}</div>
32              {% endif %}
33              <!-- /li -->
34              {% endfor %}
35      </ul>
36  {% endif %}
37

```

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

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
dumpscrip
export_emails
find_template
generate_secret_key
graph_models
mail_debug
notes
passwd
pipchecker
print_settings
print_user_for_session
reset_db
runjob
runjobs
runprofiles
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.

```

(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 singers.models import Author, Book, Group, Label, Person, Release, Song
# Shell Plus Django Imports
from django.utils import timezone
from django.db.models import Avg, Case, Count, F, Max, Min, Prefetch, Q, Sum, When
from django.urls import reverse
from django.conf import settings
from django.db import transaction
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]:
Ready

```

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: Group:

 group2

Select the group

127.0.0.1:8000/admin/singers/song/1/change/

Django administration

WELCOME, ADMIN. [VIEW SITE](#) / [CHANGE PASSWORD](#) / [LOG OUT](#)


Home / Singers / Songs / song1

Change song

[HISTORY](#) [VIEW ON SITE](#)

Title:

Group: [+ add](#)

 group2

Conclusion

Try the django-autocomplete-light module.

2016-10-17 create a normal Django view

Contents

- *2016-10-17 create a normal Django view*
 - *Le fichier singers/models.py*
 - *Le fichier singers/forms.py*
 - *Le fichier singers/views.py*
 - *Le fichier singers/urls.py*

Le fichier singers/models.py

```

1  #!/usr/bin/python
2  # -*- coding: utf8 -*-
3  """The singers models.
4
5  """
6  from __future__ import unicode_literals
7  from django.utils.encoding import python_2_unicode_compatible
8
9  from django.db import models
10 from django.core.urlresolvers import reverse
11
12
13 @python_2_unicode_compatible
14 class Person(models.Model):
15     """ an actual singular human being """
16     name = models.CharField(blank=True, max_length=100)
17     email = models.EmailField()
18
19     def __str__(self):
20         return self.name
21
22
23 @python_2_unicode_compatible
24 class Group(models.Model):
25     """ a music group """
26     name = models.CharField(max_length=200,
27                             unique=True,
28                             help_text="Name of the group")
29     members = models.ManyToManyField(Person,
30                                     blank=True,
31                                     help_text="Enter text to search for and add each member of the group.")
32     url = models.URLField(blank=True)
33
34     def __str__(self):
35         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,
```

```

94         via foreign keys
95         """
96         name = models.CharField(max_length=100)
97
98         def __str__(self):
99             return self.name
100
101
102 @python_2_unicode_compatible
103 class Book(models.Model):
104     """ Book has no admin, its an inline in the Author admin """
105
106     author = models.ForeignKey(Author)
107     title = models.CharField(max_length=100)
108     about_group = models.ForeignKey(Group)
109     mentions_persons = models.ManyToManyField(Person,
110                                                help_text="Person lookup renders html_
111 ↪in menu")
112
113     def __str__(self):
114         return self.title

```

Le fichier singers/forms.py

```

1  # -*- coding: utf-8 -*-
2
3  from __future__ import unicode_literals
4
5  from django.forms.models import ModelForm
6  from ajax_select import make_ajax_field
7  from dal import autocomplete
8
9  from .models import (Release,
10                      Song,
11                      Author,
12                      Book)
13
14  class ReleaseForm(ModelForm):
15
16      class Meta:
17          model = Release
18          exclude = []
19
20          # args: this model, fieldname on this model, lookup_channel_name
21          group = make_ajax_field(Release, 'group', 'group', show_help_text=True)
22
23          label = make_ajax_field(Release, 'label', 'label', help_text="Search for label by_
24 ↪name")
25
26          # any extra kwargs are passed onto the field, so you may pass a custom help_text_
27 ↪here
28          # songs = make_ajax_field(Release, 'songs', 'song', help_text=u"Search for song by_
29 ↪title")
30
31          # testing bug with no help text supplied

```

```
29     songs = make_ajax_field(Release, 'songs', 'song', help_text="", show_help_
↪text=True)
30
31     # these are from a fixed array defined in lookups.py
32     title = make_ajax_field(Release, 'title', 'cliche', help_text="Autocomplete will_
↪suggest clichés about cats.")
33
34
35 class SongForm(ModelForm):
36     """The Django form"""
37
38     class Meta:
39         model = Song
40         fields = ['title', 'group']
41
42     # args:  this model, fieldname on this model, lookup_channel_name
43     group = make_ajax_field(Song, # this model
44                             'group', # fieldname on this model
45                             'group', # lookup_channel_name
46                             help_text="Select the group")
47
48
49 class AuthorForm(ModelForm):
50     class Meta:
51         model = Author
52         fields = ('name',)
53         widgets = {
54             'name': autocomplete.Select(url='singers:author_autocomplete')
55         }
56
57 class BookForm(ModelForm):
58     class Meta:
59         model = Book
60         fields = ('title', 'author',)
61         widgets = {
62             'author': autocomplete.ModelSelect2(url='singers:author_autocomplete')
63         }
64
65
```

Le fichier singers/views.py

```
1  # -*- coding: utf-8 -*-
2
3  from __future__ import unicode_literals
4  from django import forms
5
6  from django.shortcuts import render_to_response
7
8  # https://docs.djangoproject.com/en/dev/ref/templates/api/
9  from django.template import RequestContext
10 from django.http import HttpResponseRedirect
11
12 from django.views.generic.edit import UpdateView
13
14 from ajax_select.fields import AutoCompleteField
```

```

15
16 # https://stackoverflow.com/questions/37915224/django-autocomplete-light-widgets-not-
    ↳ showing-up?rq=1
17 from dal import autocomplete
18
19 from .models import (Song,
20                     Author,
21                     Book)
22
23 from .forms import (SongForm,
24                   AuthorForm,
25                   BookForm)
26
27 class SearchForm(forms.Form):
28
29     q = AutoCompleteField('cliche',
30                          required=True,
31                          help_text="Autocomplete will suggest clichés about cats, but you can enter_
    ↳ anything you like.",
32                          label="Favorite Cliché",
33                          attrs={'size': 100})
34
35
36 def search_form(request):
37
38     dd = {}
39     if 'q' in request.GET:
40         dd['entered'] = request.GET.get('q')
41     initial = {'q': "\"This is an initial value,\" said O'Leary.\""}
42     form = SearchForm(initial=initial)
43     dd['form'] = form
44     return render_to_response('search_form.html',
45                             dd,
46                             context=RequestContext(request))
47
48
49
50 class SongUpdate(UpdateView):
51     """
52     url(r'^singers/song/(?P<pk>\d+)/update/$', SongUpdate.as_view(), name=
    ↳ 'song_update'),
53
54     Documentation:
55
56     - http://ccbv.co.uk/projects/Django/1.9/django.views.generic.edit/UpdateView/
57
58     """
59     model = Song
60     form_class = SongForm
61     context_object_name = 'song'
62     template_name = 'singers/song/update.html'
63
64     def get_object(self, queryset=None):
65         """Pour mémoriser self.demande_article"""
66         self.object = super(SongUpdate, self).get_object(queryset)
67         return self.object
68
69     def post(self, request, *args, **kwargs):

```

```
70         if "cancel" in request.POST:
71             url = self.get_success_url()
72             return HttpResponseRedirect(url)
73         else:
74             return super(SongUpdate, self).post(request, *args, **kwargs)
75
76
77 class AuthorAutocomplete(autocomplete.Select2QuerySetView):
78     def get_queryset(self):
79         # Don't forget to filter out results depending on the visitor !
80
81         qs = Author.objects.all()
82
83         if self.q:
84             qs = qs.filter(name__istartswith=self.q)
85
86         return qs
87
88
89
90 class AuthorUpdate(UpdateView):
91     """
92     url(r'^singers/song/(?P<pk>\d+)/update/$', SongUpdate.as_view(), name=
93     ↪ 'song_update'),
94
95     Documentation:
96
97     - http://ccbv.co.uk/projects/Django/1.9/django.views.generic.edit/UpdateView/
98
99     """
100     model = Author
101     form_class = AuthorForm
102     context_object_name = 'author'
103     template_name = 'singers/author/update.html'
104
105     def get_object(self, queryset=None):
106         """Pour mémoriser self.demande_article"""
107         self.object = super(AuthorUpdate, self).get_object(queryset)
108         return self.object
109
110     def post(self, request, *args, **kwargs):
111         if "cancel" in request.POST:
112             url = self.get_success_url()
113             return HttpResponseRedirect(url)
114         else:
115             return super(AuthorUpdate, self).post(request, *args, **kwargs)
116
117 class BookUpdate(UpdateView):
118     """
119     url(r'^singers/song/(?P<pk>\d+)/update/$', SongUpdate.as_view(), name=
120     ↪ 'song_update'),
121
122     Documentation:
123
124     - http://ccbv.co.uk/projects/Django/1.9/django.views.generic.edit/UpdateView/
125
126     """
127     model = Book
```



```

126 form_class = BookForm
127 context_object_name = 'author'
128 template_name = 'singers/author/update.html'
129
130 def get_object(self, queryset=None):
131     """Pour mémoriser self.demande_article"""
132     self.object = super(BookUpdate, self).get_object(queryset)
133     return self.object
134
135 def post(self, request, *args, **kwargs):
136     if "cancel" in request.POST:
137         url = self.get_success_url()
138         return HttpResponseRedirect(url)
139     else:
140         return super(BookUpdate, self).post(request, *args, **kwargs)

```

Le fichier singers/urls.py

```

1  #!/usr/bin/python
2  # -*- coding: utf8 -*-
3
4  from django.conf.urls import url
5  from .views import (SongUpdate,
6                     BookUpdate,
7                     AuthorAutocomplete)
8
9
10 urlpatterns = [
11     url(r'^song/(?P<pk>\d+)/update/$', SongUpdate.as_view(), name='song_update'),
12     url(r'^book/(?P<pk>\d+)/update/$', BookUpdate.as_view(), name='book_update'),
13     url(r'^author_autocomplete/$', AuthorAutocomplete.as_view(), name='author_
14     ↪autocomplete'),
15 ]

```

2016-10-17 create singers SQL tables

Contents

- 2016-10-17 create singers SQL tables
 - makemigrations singers
 - sqlmigrate singers 0001
 - migrate singers
 - show-migrations
 - createsuperuser
 - Connect to the admin interface

The

makemigrations singers

```
(ajax_django_35) C:\projects_id3\django_ajax_select\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 NOT 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 NOT 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

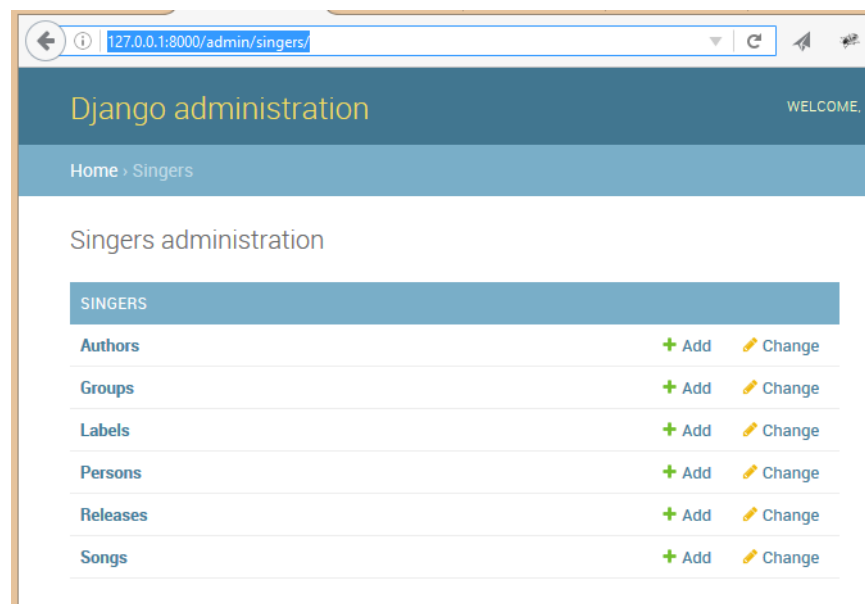
```
(ajax_django_35) C:\projects_id3\django_ajax_select\projet_ajax>python manage.py_
↪createsuperuser
```

```
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/>



Django test AJAX autocomplete development

Github Singers's development

See also:

- <https://swcarpentry.github.io/git-novice/>
- <https://github.com/github/gitignore>

Contents

- *Github Singers's development*
 - *Create new project*
 - *Create singers project*
 - *Git remote*

Create new project

Create singers project

Git remote

See also:

<https://swcarpentry.github.io/git-novice/07-github/>

Singers's Python virtualenv

See also:

- <https://swcarpentry.github.io/git-novice/>

Contents

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: pvergain / Repository name: 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.

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.


```

Le volume dans le lecteur C s'appelle os
Le numéro de série du volume est 46AD-0A9C

Répertoire de C:\projects_id3\django_ajax_select
17/10/2016  10:46    <DIR>        .
17/10/2016  10:46    <DIR>        ..
17/10/2016  10:45    <DIR>        projet_ajax
17/10/2016  10:46                44 requirements.txt
               1 fichier(s)          44 octets
               3 Rép(s)  170 275 516 416 octets libres

(ajax_django_35) C:\projects_id3\django_ajax_select>git init
Initialized empty Git repository in C:/projects_id3/django_ajax_select/.git/

(ajax_django_35) C:\projects_id3\django_ajax_select>git status
On branch master

Initial commit

Untracked files:
  (use "git add <file>..." to include in what will be committed)

    projet_ajax/
    requirements.txt

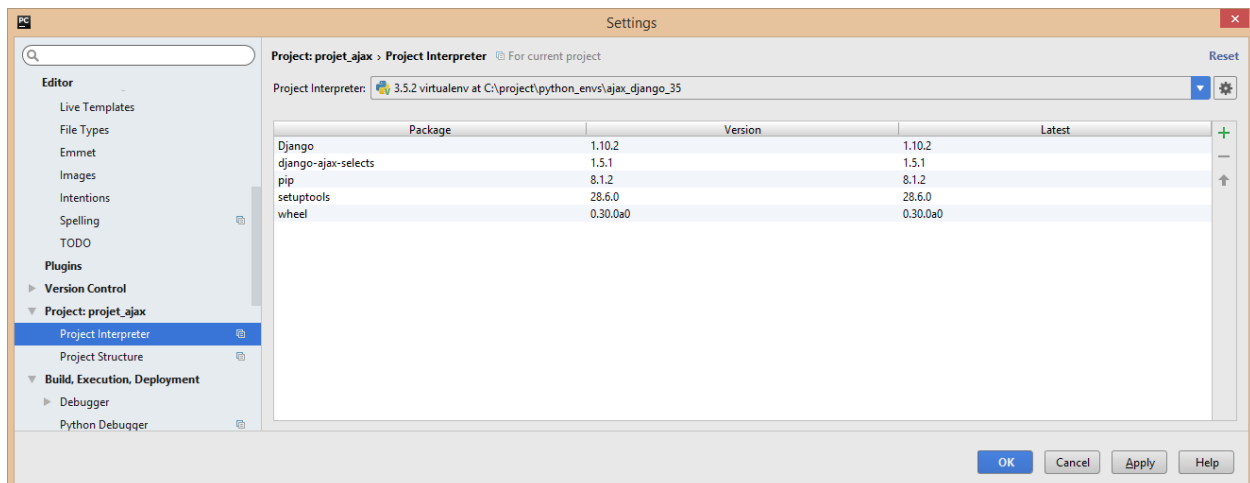
nothing added to commit but untracked files present (use "git add" to track)

(ajax_django_35) C:\projects_id3\django_ajax_select>git remote add origin https://github.com/pvergain/singers.git

```

- *Singers's Python virtualenv*
 - *ajax_django35 Python 3.5 virtualenv*

ajax_django35 Python 3.5 virtualenv



Django

Django packages

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>

The screenshot shows the GitHub repository for `joke2k/faker`. The repository is a Python package that generates fake data. The left sidebar shows the repository structure, including folders like `docs`, `locales`, `providers`, `Makefile`, `coding_style.rst`, `communityproviders.rst`, `conf.py`, `index.rst`, `make.bat`, `faker`, `providers`, `address`, `barcode`, `color`, `company`, `credit_card`, `currency`, `date_time`, `file`, `internet`, `job`, `lorem`, `misc`, `person`, `phone_number`, `profile`, `python`, `ssn`, `user_agent`, `__init__.py`, `tests`, `utils`, `__init__.py`, `__main__.py`, `build_docs.py`, `cli.py`, `config.py`, `documentor.py`, `factory.py`, and `generator.py`.

The main content area shows the repository name `joke2k / faker` and statistics: 997 commits, 6 branches, 21 releases, and 122 contributors. Below this, there is a table of recent commits:

Commit	Description	Time
docs	Bump version: 0.7.2 → 0.7.3	a month ago
faker	Merge pull request #400 from vema/feature/locale-nl_BE	8 days ago
.bumpversion.cfg	Bump version: 0.7.2 → 0.7.3	a month ago
.coveragerc	added .coveragerc	a year ago
.gitignore	.eggs added to .gitignore	22 days ago
.travis.yml	Python 3.5 support	3 months ago
CHANGELOG.rst	Update CHANGELOG	20 days ago
CONTRIBUTING.rst	Move contributing guide to CONTRIBUTING.rst	2 years ago
LICENSE.txt	Update license	2 years ago
MANIFEST.in	fix MANIFEST	2 years ago
README.rst	Update the PyPI version badge	6 days ago
appveyor.yml	update 'get-pip' script	9 months ago
circle.yml	seed a Random instance instead of the module	a year ago
readthedocs.yml	add config for readthedocs	4 months ago
setup.cfg	Close #263. Use environment markers for wheels	a year ago
setup.py	Bump version: 0.7.2 → 0.7.3	a month ago

The README.rst file is also visible at the bottom of the page.

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/en/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

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

- *If you were starting a new web app tomorrow, would you use jQuery ?*
 - * Aurelio De Rosa
 - * TJ VanToll
- *jQuery in Action (third edition)*
 - * clone
- *jQuery UI in Action*
- *jquery plugins*

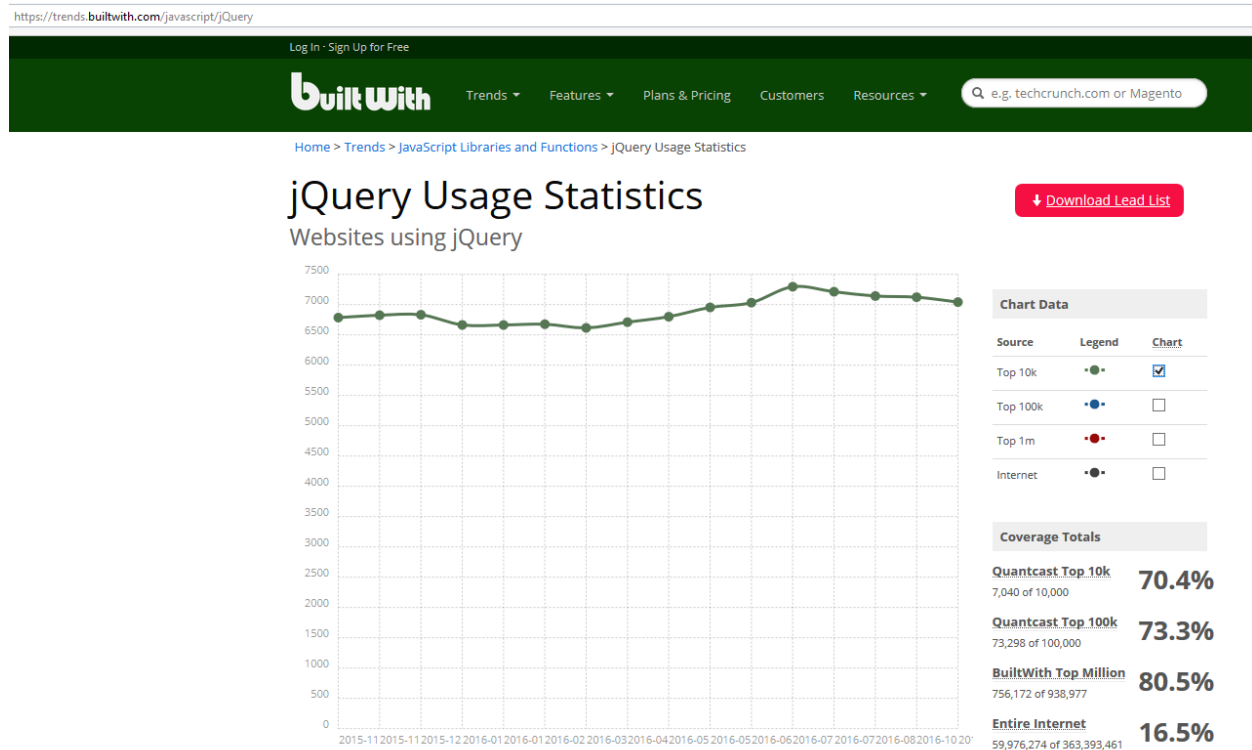


Fig. 4.1: <https://trends.builtwith.com/javascript/jquery>

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>

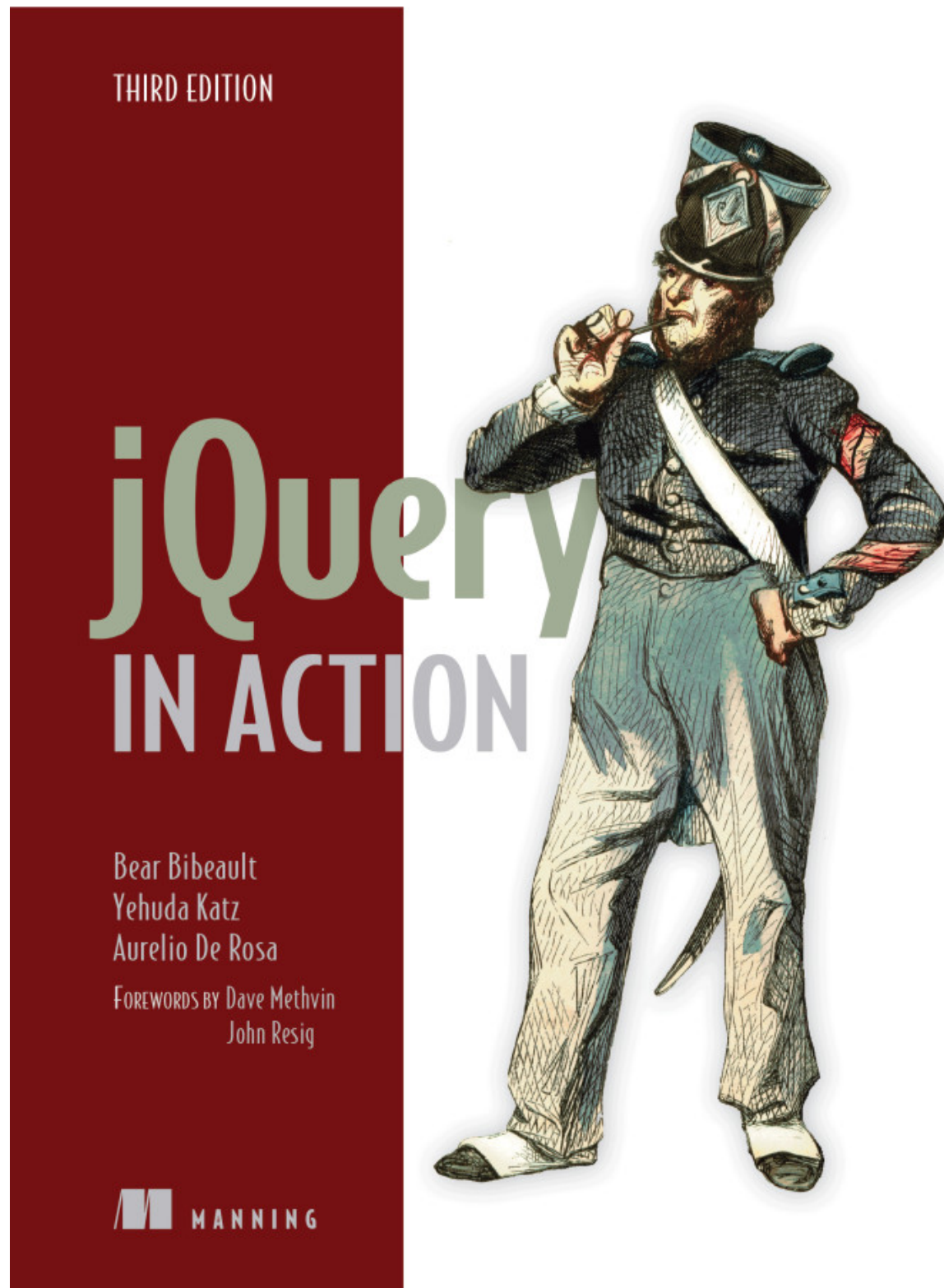
jquery plugins

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>



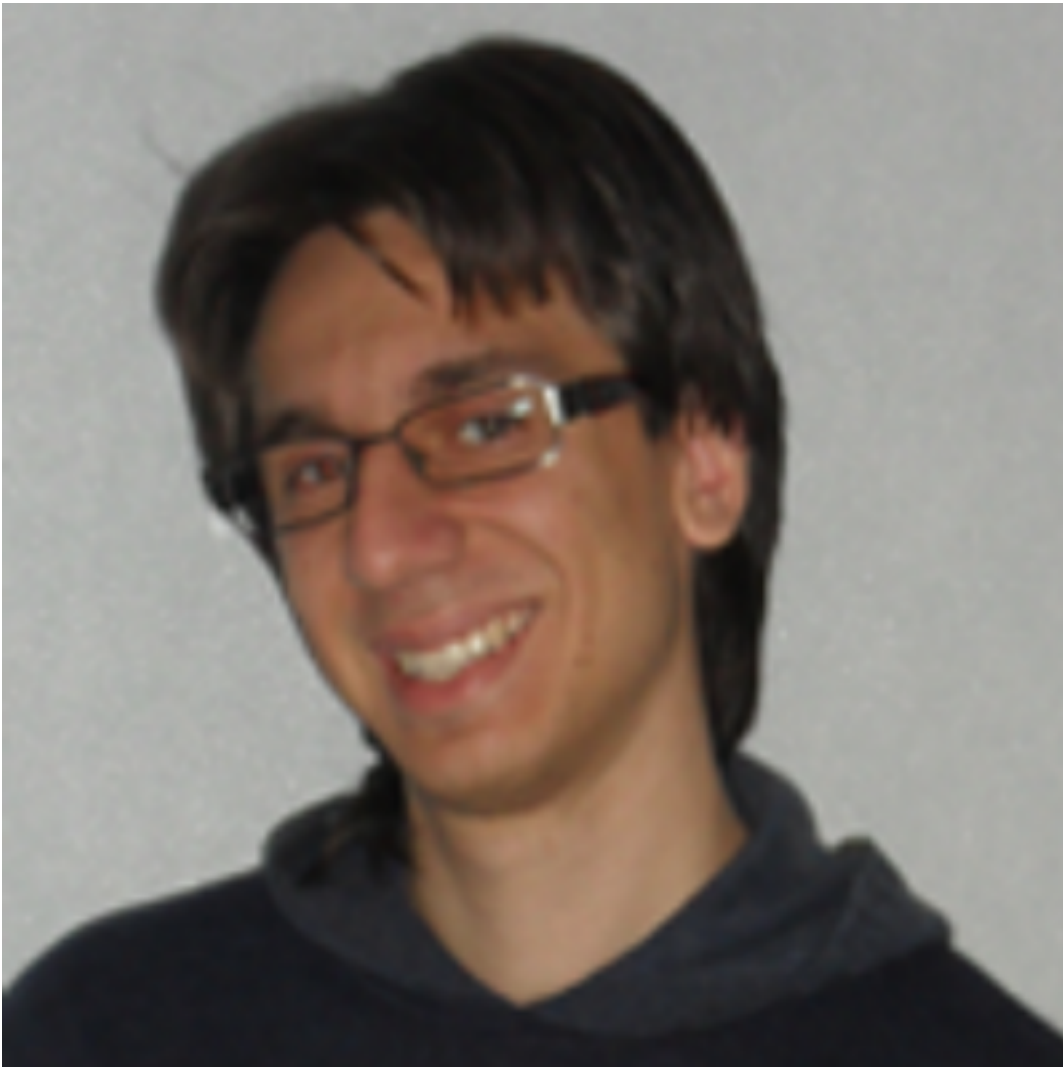
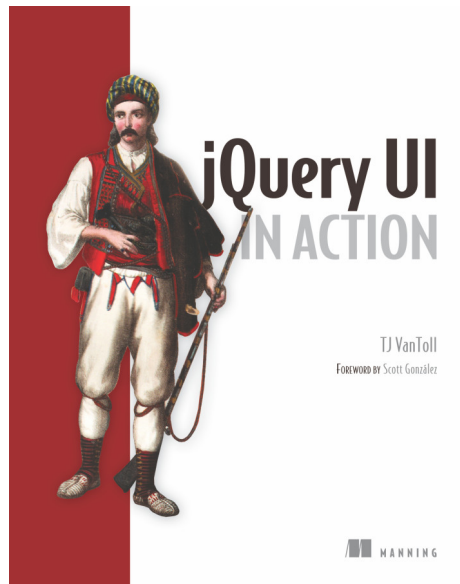


Fig. 4.2: Aurelio De Rosa (<https://twitter.com/AurelioDeRosa>)



jquery select2 plugin

See also:

- <https://github.com/select2/select2>
- <https://select2.github.io/announcements-4.0.html>

Contents

- *jquery select2 plugin*
 - *Use cases*

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.

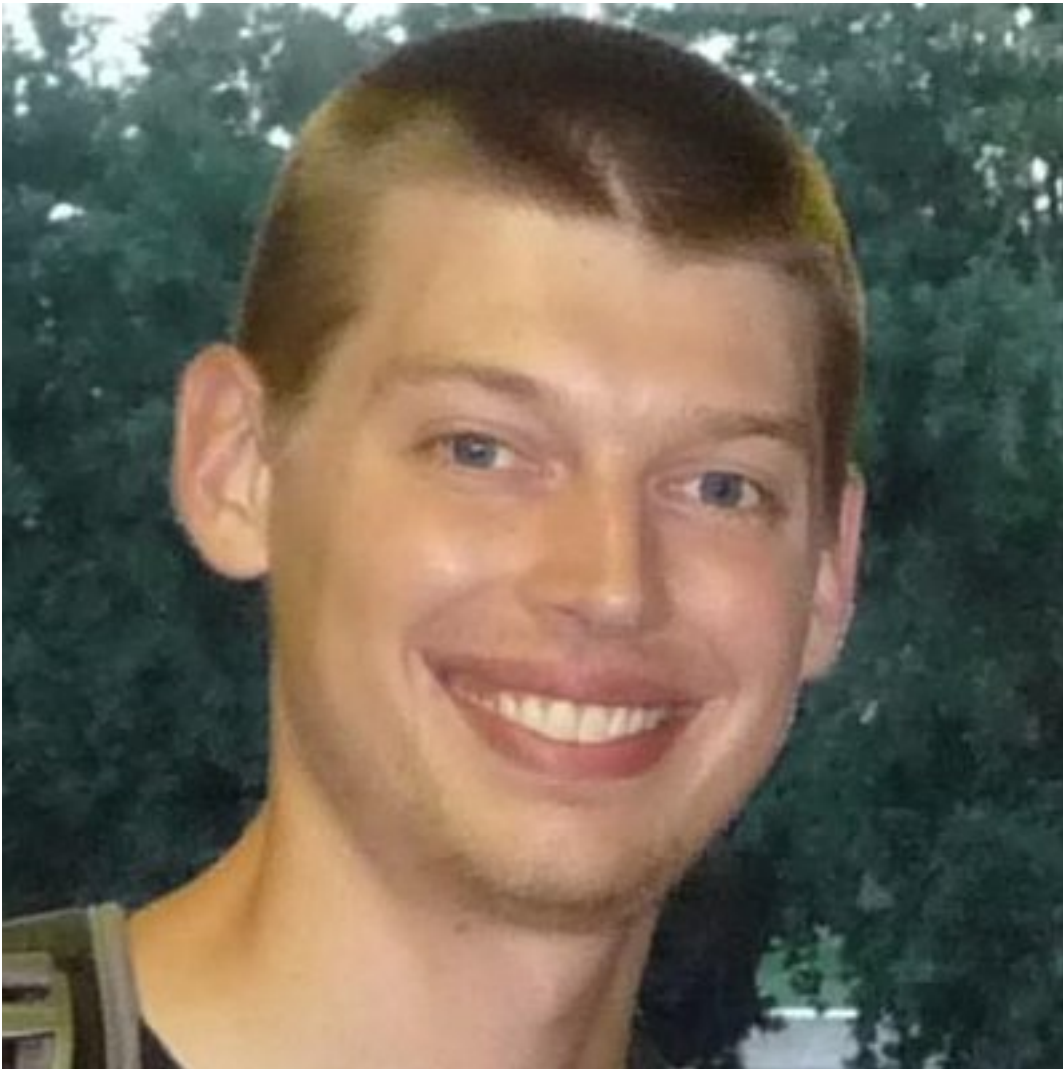


Fig. 4.3: TJ Vantoll (<https://twitter.com/tjvantoll>)

The screenshot shows the GitHub repository page for 'ansek / jquery-ui-autocomplete-scroll'. The left sidebar displays the repository's file structure, including 'LICENSE-MIT', 'README.md', 'bower.json', 'jquery.ui.autocomplete.scroll.js', 'jquery.ui.autocomplete.scroll.min.js', 'package.json', and 'ui.autocomplete.scroll.jquery.json'. The main content area shows the repository's overview, including 23 commits, 2 branches, 8 releases, and 1 contributor. A table lists recent commits, all of which are 'Fix #2 Reset scroll position when it is reopened.' and were made 7 months ago. Below the commit list is the 'README.md' file, which contains the following text:

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.

The screenshot shows the GitHub repository page for 'select2/select2'. The left sidebar displays the repository's file structure, including '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', and 'select2.jquery.json'. The main content area shows the repository's overview, including a description of the library and its integrations. The 'Integrations' section lists various platforms and themes that support Select2, including Django, Meteor, Ruby on Rails, Wicket, and Yii 2. The 'Internationalization (i18n)' section explains how to use the library with multiple languages. The 'Documentation' section mentions that the documentation is available through GitHub Pages and is located within the 'docs' folder.

You can source Select2 directly from a CDN like [JSDeliver](#) 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.

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:

- *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

EasyAutocomplete examples

Input simple

Input data-json

Input AJAX

aaaa
admin
albert
champion_0001
john
nigel

on/?term="

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
```

Changed

Miscellaneous

Fixed

Test

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_\(programming\)](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>

Symbols

- 0.1.0 (2016-10-21)
 - Version, 206
- 0.2.0 (2016-10-27)
 - Version, 205
- 2016-10-17
 - Actions, 179, 185
- 2016-10-26
 - Actions, 37

Numbers

- 101
 - jQuery, 37
- 2016
 - Actions, 35

A

- Actions
 - 2016-10-17, 179, 185
 - 2016-10-26, 37
 - 2016, 35
 - Project, 33
- AJAX, 209
- Aurelio De Rosa
 - jQuery, 197, 199
- Autocomplete
 - definitions, 1
- autocomplete
 - jQuery-ui, 153
 - Tests, 6

B

- base.html
 - Django, 144, 153, 172

C

- categories
 - EasyAutocomplete, 49
- CSS
 - EasyAutocomplete, 49

D

- definitions
 - Autocomplete, 1
- Development
 - Django, 193
 - Django test AJAX autocomplete, 189
- Django
 - base.html, 144, 153, 172
 - Development, 193
 - django-autocomplete-light, 8
 - django-extensions, 172
 - makemigrations, 185
 - normal Django Views, 179
 - Packages, 194
 - Template, 141
- Django AJAX autocomplete
 - Project, 1
- Django test AJAX autocomplete
 - Development, 189
- Django test Autocomplete
 - Versions, 202
- django-autocomplete-light
 - Django, 8
- django-extensions
 - Django, 172
- doc
 - read the docs, 45
- Documentation
 - Read the docs, 1

E

- EasyAutocomplete
 - categories, 49
 - CSS, 49
 - flags, 49
 - Javascript, 49
 - jQuery, 49
 - JSON, 49
 - SCSS, 49
- easyautocomplete

Tests, 24

F

Faker

Python, 37, 194

flags

EasyAutocomplete, 49

G

git

gitsome, 35

Github

singers, 191

swcarpentry, 191

github

<https://github.com/anseki>, 35, 37

gitsome

git, 35

Glossary

Project, 208

H

HTML5

placeholder, 37

httpie

Python, 49

<https://github.com/anseki>

github, 35, 37

I

inheritance

Template, 172

J

Javascript

EasyAutocomplete, 49

Unobstrusive, 141

jQuery, 197, 199

101, 37

Aurelio De Rosa, 197, 199

EasyAutocomplete, 49

jquery_ui_autocomplete_scroll, 199

select2, 199

TJ VanToll, 197, 199

UI, 153

jQuery EasyAutocomplete

plugin, 49

jQuery UI autocomplete

scrollbar, 29

Tests, 29

jQuery-ui

autocomplete, 153

jquery_ui_autocomplete_scroll

jQuery, 199

JSON, 209

EasyAutocomplete, 49

M

makemigrations

Django, 185

N

normal Django Views

Django, 179

P

Packages

Django, 194

placeholder

HTML5, 37

plugin

jQuery EasyAutocomplete, 49

Project

Actions, 33

Django AJAX autocomplete, 1

Glossary, 208

Python

Faker, 37, 194

httpie, 49

R

Read the docs

Documentation, 1

read the docs

doc, 45

S

scrollbar

jQuery UI autocomplete, 29

SCSS

EasyAutocomplete, 49

select2

jQuery, 199

singers

Github, 191

virtualenv, 191

swcarpentry

Github, 191

T

Template

Django, 141

inheritance, 172

Tests

autocomplete, 6

easyautocomplete, 24

- jQuery UI autocomplete, [29](#)
- TJ VanToll
 - jQuery, [197](#), [199](#)

U

- UI
 - jQuery, [153](#)
- Unobstrusive
 - Javascript, [141](#)

V

- Version
 - 0.1.0 (2016-10-21), [206](#)
 - 0.2.0 (2016-10-27), [205](#)
- Versions, [202](#)
 - Django test Autocomplete, [202](#)
- virtualenv
 - singers, [191](#)