
Django-ajax-search Documentation

Release 1.1.5

Prasenjit Singh

June 06, 2015

1	Getting started	3
2	Configuration	5
3	Search tools	7
4	Customize	9

Django-ajax-search is an AJAX-powered search-app for Django based websites.

In addition to displaying instant dropdown search results as one types, it's fully customizable and easy to use.

The **source code** for this app is hosted at <https://github.com/side-winder/django-ajax-search>.

Getting started

- Python 2.6 or Python 2.7
- Django >= 1.3
- jQuery >= 1.4.4

To install:

```
easy_install django-ajax-search
```

Next add *ajax_search* to your *INSTALLED_APPS* to include the related css/js:

```
INSTALLED_APPS = (  
    'django.contrib.staticfiles',  
    # Other apps here  
    'ajax_search',  
)
```

Once installed you should add the urls to your root url patterns:

```
urlpatterns = patterns('',  
    # Other patterns go here  
    url(r'^ajax_search/', include('ajax_search.urls')),  
)
```

Now that you're done with the setup, configure your installation.

Configuration

The jQuery library is not included in the distribution but must be included in your templates. Import it:

```
<script src="//ajax.googleapis.com/ajax/libs/jquery/1.7.2/jquery.min.js"></script>
```

Include the django-ajax-search media/static files to your project media/static folder. Then include the files in your templates. Make sure to use the correct path:

```
<link rel="stylesheet" type="text/css" href="/media/ajax_search/css/ajax_search.css" />
<script src="/media/ajax_search/js/ajax_search.js" type="text/javascript"></script>
```

Include global settings in your **settings.py** file.

Specify the maximum number of instant-search results you'd like to see in the search drop-down menu:

```
AJAX_SEARCH_LIMIT = 8
```

Specify the search helper function that we shall configure in a moment (use the full path):

```
AJAX_SEARCH_HELPER = 'app.views.search_helper'
```

Specify the search-results template where you'd like to see the results displayed. We shall configure it shortly:

```
SEARCH_RESULT_TEMPLATE = 'search_results.html'
```

Import the search form in your views and pass it to templates:

```
from ajax_search.forms import SearchForm

return render_to_response(template_name, {'searchform': SearchForm()})
```

In your templates where you want to have the search form, include the code (make sure to keep **autocomplete off** and don't change the class and id of any tag if you wish to include formatting):

```
<form class="searchfield" method="get" action="/ajax_search/search/" autocomplete="off">
  {{ searchentryform.q }}
  <input id="searchbuttonmain" type="submit" class="subbtn" value="Search">
</form>
```

Now that you've configured the basics, move on to the [Search Tools](#) section.

Search tools

The configuration is almost done. All that remains is to define a search function.

Define a new function on the path that you provided for the `AJAX_SEARCH_HELPER` setting in the previous section. Assuming that the model you want to search is called `Article` and you wish to search the **title** field of every article:

```
def search_helper(count, query):
    import itertools
    model_list = Article.objects.filter(title__icontains=query, status=1)
    for L in range(1, count+1):
        for subset in itertools.permutations(words, L):
            count1=1
            query1=subset[0]
            while count1!=len(subset):
                query1=query1+" "+subset[count1]
                count1+=1
            model_list = model_list | Article.objects.filter(title__icontains=query1, status=1)
    return model_list.distinct()
```

You may stop at the third line to return a very simple list that matches exactly the query provided with the title. The 'for' loops break the search query down and search the parts and various combinations thereof. We suggest you use this method.

Now all you need to do is specify the position of the drop-down menu. Read about it [here](#).

Customize

You will need to define the position of the dropdown list for instant-search.

In the template where you need the dropdown displayed, include this code in the body:

```
<div id='searchdropdown'></div>
```

In the `media/css` folder, open the `ajax_search.css` file. Therein you'll find specifications for the very first element called `searchdropdown`. Uncomment the lines and define its position according to your needs.

You may also make other changes to the CSS file to change the look and feel of the components.

Your search is now set-up. Try it by typing in the search-box.

In case of any issues, include the issue on Github after searching existing ones. You may also join and mail to the django-ajax-search Google Group at <https://groups.google.com/forum/?fromgroups#!forum/django-ajax-search> .

You may also mail to me directly at prasenjit0625@gmail.com.