download file via ajax django

Django - File Uploading.

It is generally useful for a web app to be able to upload files (profile picture, songs, pdf, words.). Let's discuss how to upload files in this chapter.

Uploading an Image.

Before starting to play with an image, make sure you have the Python Image Library (PIL) installed. Now to illustrate uploading an image, let's create a profile form, in our myapp/forms.py −

As you can see, the main difference here is just the forms.ImageField . ImageField will make sure the uploaded file is an image. If not, the form validation will fail.

Now let's create a "Profile" model to save our uploaded profile. This is done in myapp/models.py −

As you can see for the model, the ImageField takes a compulsory argument: upload_to . This represents the place on the hard drive where your images will be saved. Note that the parameter will be added to the MEDIA_ROOT option defined in your settings.py file.

Now that we have the Form and the Model, let's create the view, in myapp/views.py −

The part not to miss is, there is a change when creating a ProfileForm, we added a second parameters: request.FILES . If not passed the form validation will fail, giving a message that says the picture is empty.

Now, we just need the saved.html template and the profile.html template, for the form and the redirection page −

myapp/templates/saved.html −

myapp/templates/profile.html −

Next, we need our pair of URLs to get started: myapp/urls.py.

When accessing "/myapp/profile", we will get the following profile.html template rendered −

And on form post, the saved template will be rendered −

We have a sample for image, but if you want to upload another type of file, not just image, just replace the ImageField in both Model and Form with FileField .

Django - Ajax.

Ajax essentially is a combination of technologies that are integrated together to reduce the number of page loads. We generally use Ajax to ease end-user experience. Using Ajax in Django can be done by directly using an Ajax library like JQuery or others. Let's say you want to use JQuery, then you need to download and serve the library on your server through Apache or others. Then use it in your template, just like you might do while developing any Ajax-based application.

Another way of using Ajax in Django is to use the Django Ajax framework. The most commonly used is django-dajax which is a powerful tool to easily and super-quickly develop asynchronous presentation logic in web applications, using Python and almost no JavaScript source code. It supports four of the most popular Ajax frameworks: Prototype, jQuery, Dojo and MooTools.

Using Django-dajax.

First thing to do is to install django-dajax. This can be done using easy_install or pip −

This will automatically install django-dajaxice, required by django-dajax. We then need to configure both dajax and dajaxice.

Add dajax and dajaxice in your project settings.py in INSTALLED_APPS option −

Make sure in the same settings.py file, you have the following −

Now go to the myapp/url.py file and make sure you have the following to set dajax URLs and to load dajax statics js files −

Let us create a simple form based on our Dreamreal model to store it, using Ajax (means no refresh).

At first, we need our Dreamreal form in myapp/form.py.

Then we need an ajax.py file in our application: myapp/ajax.py. That's where is our logic, that's where we put the function that will be saving our form then return the popup −

Now let's create the dreamreal.html template, which has our form −

Add the view that goes with the template in myapp/views.py −

Add the corresponding URL in myapp/urls.py −

Now let's add the necessary in our template to make the Ajax work −

At the top of the file add −

And in the <head> section of our dreamreal.html template add −

We are using the JQuery library for this example, so add −

The Ajax function that will be called on click −

Note that you need the “jquery-1.11.3.min.js” in your static files directory, and also the jquery.dajax.core.js. To make sure all dajax static files are served under your static directory, run −

Note − Sometimes the jquery.dajax.core.js can be missing, if that happens, just download the source and take that file and put it under your static folder.

You will get to see the following screen, upon accessing /myapp/dreamreal/ −

How to create file download links in Django?

In order to create a download link, we need to create a Django view that would serve the files:

Once you've done this, add a line to urlpatterns in urls.py , which references the view.

How it works?

It works because we send the following HTTP header to a browser:

It tells a browser to treat a response as a downloadable file. Have you noticed that we also include Content-Type header? This one tells what kind of file we are sending to the browser or in other words its mime type . If the header is not set Django will set it to text/html . mimetypes.guess_type is a handy function which tries to guess the mime type of the file, however if you know the mime type of your file(s) beforehand, it is better to set this manually.

3 reasons why you should learn function-based views first.

You're fairly new to Django, and there're a lot of unfamiliar concepts you need to wrap your head around. A common source of confusion for beginners is different kinds of views: function-based views, class-based views, generic views. You might be confused about which type of views would be best for.

How to use Django with an existing database?

In order to use an existing database in Django, you need to have a model for each table. Creating models for existing tables manually is just too much work. However, there's no need to do that, since Django has a builtin tool to solve this exact problem. Note: this guide.

Django Rest Framework OpenAPI 3 support.

OpenAPI 3 support in Django Rest Framework is still a work in progress. Things are moving quickly so there's not a lot of up to date info about this topic. It's not clear which features of OpenAPI 3 spec are supported in DRF and researching this info on your own.

Django Image And File Upload Using Ajax¶

Let's learn about how to work with Django image and file upload using Ajax. In most of the cases I personally use django file or image uploads using ajax . If we use normal upload then browser will reload the page in order to load the requested response and we will not have the files in the file fields due to page refreshed by the browser, again user has to browse the files if any error occurs while validating the form. If we use the ajax upload then the control is taken by the javascript. So, the page will not be refreshed and we will have the files in the fields selected by the user. If we use normal upload then browser will also send request to get the static resources so, It will have an effect on the page speed. Django image and file upload using Ajax will only requests for form validation so, other resources like static(css/js/images) will not be requested.

Lets see the code of Django image and file upload using Ajax¶

Points to be noted when using Django image and file upload using Ajax¶

In template we must define enctype="multipart/form-data"otherwise user browsed files (in our case image and attachment) will not sent in the request. So, Django will return errors like "This filed is required".

In views.py return only JsonResponse. Do not return HttpResponse. By default in HttpResponse content_type is set to "html/text" where as in JsonResponse the content_type is set to "json/application". When content type is set to "json" then the ajax will convert the response body into JSON data. Otherwise we have to manually parse response body to json using "JSON.parse"

We must define STATIC settings and MEDIA settings in settings.py file if these are not configured properly then we may get errors.

Related Blog Articles.

How To Add Custom Views To Django Admin.

Django provides the awesome inbuilt functionality with django admin. We can also add custom views to django admin to make it more powerful.

django-ajax-validation 0.1.3.

Provides support for doing validation using Ajax(currently with jQuery) using your existing Django forms.

Navigation.

Project links.

Statistics.

Stars: Forks: Open issues/PRs:

View statistics for this project via Libraries.io, or by using our public dataset on Google BigQuery.

License: BSD License (UNKNOWN)

Maintainers.

Classifiers.

Development Status 3 - Alpha Web Environment Django Developers OSI Approved :: BSD License OS Independent Python.

Project description.

Project details.

Project links.

Statistics.

Stars: Forks: Open issues/PRs:

View statistics for this project via Libraries.io, or by using our public dataset on Google BigQuery.

License: BSD License (UNKNOWN)

Maintainers.

Classifiers.

Development Status 3 - Alpha Web Environment Django Developers OSI Approved :: BSD License OS Independent Python.

Release history Release notifications | RSS feed.

Download files.

Download the file for your platform. If you're not sure which to choose, learn more about installing packages.

Files for django-ajax-validation, version 0.1.3 Filename, size File type Python version Upload date Hashes Filename, size django-ajax-validation-0.1.3.tar.gz (5.6 kB) File type Source Python version None Upload date Jan 30, 2009 Hashes View.

Hashes for django-ajax-validation-0.1.3.tar.gz.

Hashes for django-ajax-validation-0.1.3.tar.gz Algorithm Hash digest SHA256 961f91c7eb1ae983db47faed4da12e6cf01a9913967bfd5f1b29c4709e2ea27e Copy MD5 96ee5c948c21c3cdc1984d42b3120906 Copy BLAKE2-256 25fdf866629e1b894c430a12ec7f13a1386b50e46e250962b30382475c22c078 Copy.

About PyPI.

Contributing to PyPI.

Using PyPI.

Developed and maintained by the Python community, for the Python community. Donate today!