

**By Vitor Freitas**

I'm a passionate software developer and researcher from Brazil, currently living in Finland. I write about Python, Django and Web Development on a weekly basis.

[Read more.](#)



Limited time offer: Get 10 free Adobe Stock images.

ads via Carbon

TUTORIAL

How to Work With AJAX Request With Django

📅 Aug 29, 2016 ⌚ 8 minutes read 💬 65 comments 👁 292,999 views



There are many scenarios where you may want to use AJAX requests in your web application. It is a great resource that enables web applications to be faster and more dynamic. For this short tutorial I will be using jQuery to ease the implementation.

You can choose a different framework, or even implement it using JavaScript only. But the concept will remain the same.

Initial Setup

Here is how my **base.html** template looks like:

```
{% load static %}<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>{% block title %}Default Title{% endblock %}</title>
    <link rel="stylesheet" type="text/css" href="{% static 'css/app.css' %}">
    {% block stylesheet %}{% endblock %}
  </head>
  <body>
```

```
<header>
...
</header>
<main>
    {% block content %}
    {% endblock %}
</main>
<footer>
...
</footer>
<script src="https://code.jquery.com/jquery-3.1.0.min.js"></script>
<script src="{% static 'js/app.js' %}"></script>
{% block javascript %}{% endblock %}
</body>
</html>
```

The jQuery library and all the JavaScript resources stays in the end of the HTML page for two reasons: to guarantee the DOM will be loaded whe the script is executed and to avoid inline scripts (at least scripts that uses jQuery).

All the extra or page specific JavaScript goes inside the

```
{% block javascript %}{% endblock %}
```

 block.

Sample Scenario

Let's say you want to validate the **username** field in a sign up view, as soon as the user finish typing the desired username. You want just to do a simple check, if the **username** is already taken or not.

views.py

```
from django.contrib.auth.forms import UserCreationForm
from django.views.generic.edit import CreateView

class SignUpView(CreateView):
    template_name = 'core/signup.html'
    form_class = UserCreationForm
```

urls.py

```
from django.conf.urls import url
from core import views

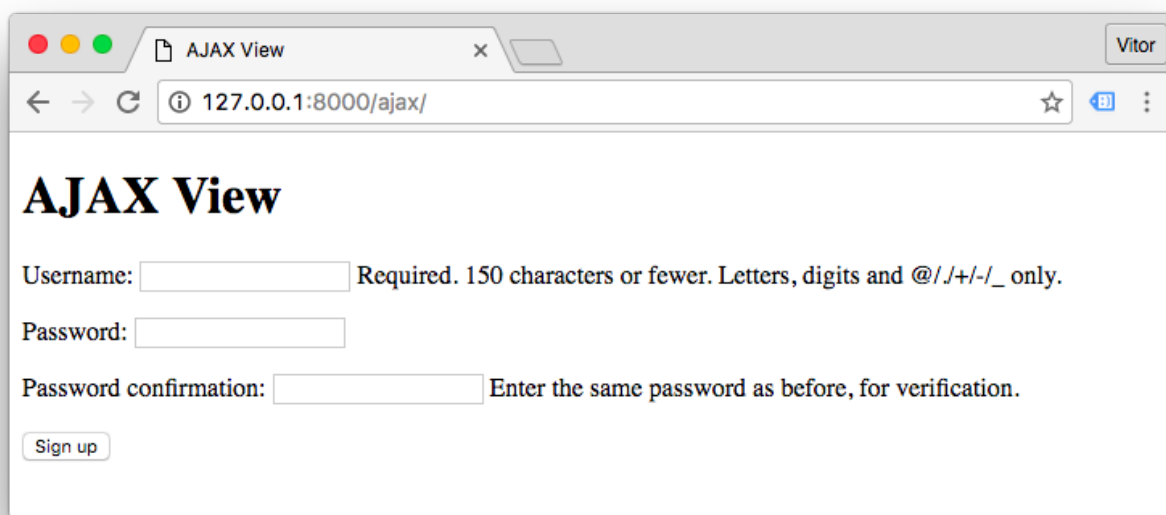
urlpatterns = [
    url(r'^signup/$', views.SignUpView.as_view(), name='signup'),
]
```

signup.html

```
{% extends 'base.html' %}

{% block content %}
    <form method="post">
        {% csrf_token %}
        {{ form.as_p }}
        <button type="submit">Sign up</button>
    </form>
{% endblock %}
```

The view looks like that:



A screenshot of a web browser window titled "AJAX View". The address bar shows "127.0.0.1:8000/ajax/". The page content includes a heading "AJAX View" and a form with the following fields and labels:

- Username: Required. 150 characters or fewer. Letters, digits and @/./+/_ only.
- Password:
- Password confirmation: Enter the same password as before, for verification.

At the bottom left of the form is a button labeled "Sign up".

Ajax Request

Let's implement an asynchronous request to validate if the **username** is already taken or not.

First we gotta have a look on the HTML generated by the `{{ form.as_p }}`. We want to inspect the **username** field, which looks like that:

```
<input type="text" required="" name="username" maxlength="150" id="id_username" al
```

What we need here is its ID, which is **id_username**. Let's create a listener for the **username**'s field change event:

signup.html

```
{% extends 'base.html' %}

{% block javascript %}
<script>
    $("#id_username").change(function () {
        console.log( $(this).val() );
    });
</script>
{% endblock %}

{% block content %}
<form method="post">
    {% csrf_token %}
    {{ form.as_p }}
    <button type="submit">Sign up</button>
</form>
{% endblock %}
```

This is something I still like to do nowadays, even working with jQuery every single day. Before moving forward, make sure you got the listener right. Make sure the

event is firing correctly. In this case, as the name suggests, the **change** event will occur every time the value of the **username** field changes.

You can check the output of the `console.log()` function in your web browser console. Generally right clicking in the page and clicking on a menu that says **Inspect** or something similar.

Let's create a view that checks if a given **username** is taken, and return a response as **JSON**.

views.py

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

def validate_username(request):
    username = request.GET.get('username', None)
    data = {
        'is_taken': User.objects.filter(username__iexact=username).exists()
    }
    return JsonResponse(data)
```

Add a route to this view:

urls.py

```
from django.conf.urls import url
from core import views

urlpatterns = [
    url(r'^signup/$', views.SignUpView.as_view(), name='signup'),
    url(r'^ajax/validate_username/$', views.validate_username, name='validate_user')
]
```

And then the simplest implementation would be:

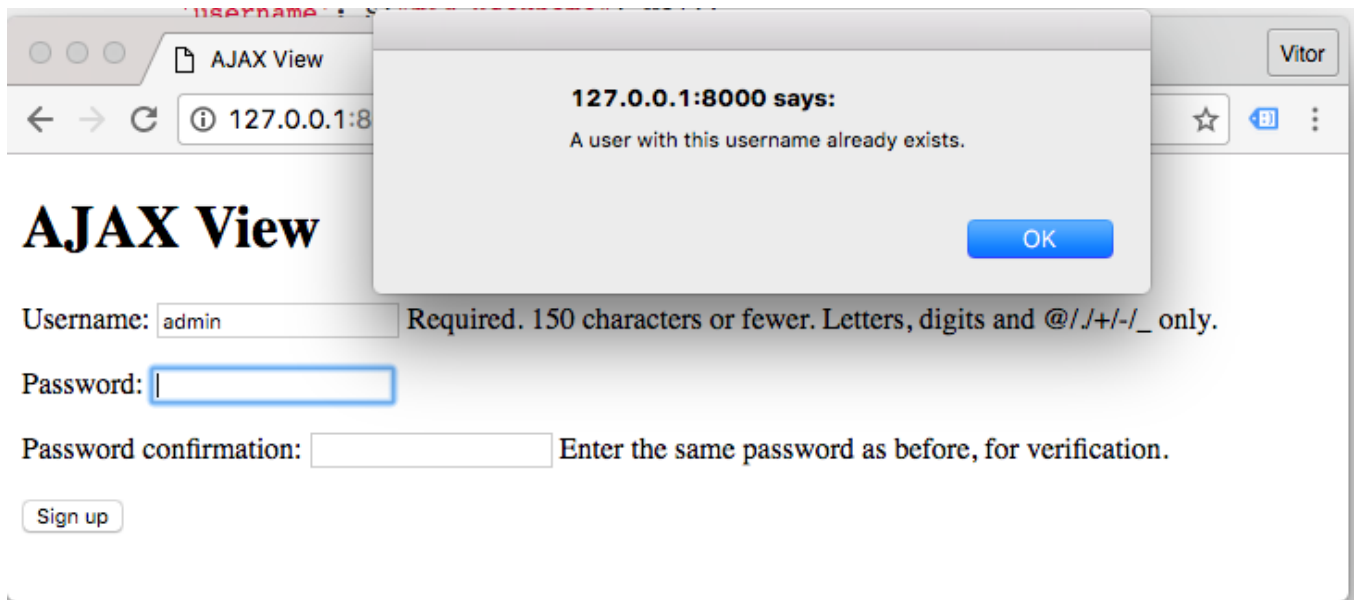
signup.html

```
{% extends 'base.html' %}

{% block javascript %}
<script>
    $("#id_username").change(function () {
        var username = $(this).val();

        $.ajax({
            url: '/ajax/validate_username/',
            data: {
                'username': username
            },
            dataType: 'json',
            success: function (data) {
                if (data.is_taken) {
                    alert("A user with this username already exists.");
                }
            }
        });
    });
</script>
{% endblock %}

{% block content %}
<form method="post">
    {% csrf_token %}
    {{ form.as_p }}
    <button type="submit">Sign up</button>
</form>
{% endblock %}
```



Improving the Example

The example is working, and that's great. But there are a few details we can improve:

- Hardcoded URL inside the Ajax function
- User message embeded in the JavaScript

Since we are inside a Django template, we could do something like that:

```
$.ajax({  
  url: '{% url "validate_username" %}',  
  ...  
});
```

It is already better, but, I don't really like this strategy. Two reasons: You won't be able to extract this JavaScript code to a external file. We have an external script modifying our JavaScript. If you find yourself writing code to write a JavaScript code, give it another thought. There might be a better solution.

Now, what I usually like to do:

views.py

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

def validate_username(request):
    username = request.GET.get('username', None)
    data = {
        'is_taken': User.objects.filter(username__iexact=username).exists()
    }
    if data['is_taken']:
        data['error_message'] = 'A user with this username already exists.'
    return JsonResponse(data)
```

Extract the user message to the Python code, this way is easier to work with translations.

signup.html

```
{% extends 'base.html' %}

{% block javascript %}
<script>
    $("#id_username").change(function () {
        var form = $(this).closest("form");
        $.ajax({
            url: form.attr("data-validate-username-url"),
            data: form.serialize(),
            dataType: 'json',
            success: function (data) {
                if (data.is_taken) {
                    alert(data.error_message);
                }
            }
        });
    });
</script>
{% endblock %}

{% block content %}
```

```
<form method="post" data-validate-username-url="{% url 'validate_username' %}">
    {% csrf_token %}
    {{ form.as_p }}
    <button type="submit">Sign up</button>
</form>
{% endblock %}
```

Ideally I would put the `data-validate-username-url` attribute directly in the **username** field. But here we would need to expose all the fields in the `{{ form.as_p }}`. So in this case I can live with the attribute in the **form** tag.

Generally speaking: try to avoid modifying JavaScript code with Python code, keep the URL references in the HTML and manage user messages in the Python code.

Another tip: if you have direct access to the HTML field, prefer adding a class name like this:

```
<input type="text" name="username" id="id_username" class="js-validate-username">
```

And then you hook the change event to the class `js-validate-username` instead. The `js-` prefix suggests that there is a JavaScript code that interacts with this element. And then use this prefix for JavaScript only, never use it for styling the component as well via css.

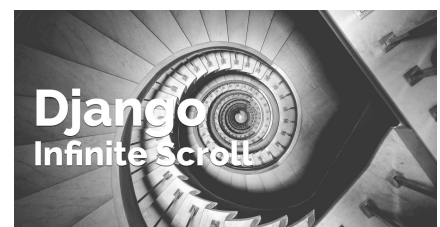
Related Posts



[How to Integrate Highcharts.js with Django](#)



[How to Implement Dependent/Chained](#)



[How to Create Infinite Scroll With Django](#)

[Dropdown List with Django](#)

[django](#) [ajax](#) [jquery](#)

Share this post



65 Comments

Simple is Better Than Complex



Recommend 34

Tweet

Share

Sort by Best ▾

Join the discussion...



Farzam Taghipour • a year ago

Thanks for the document but I think there is a problem:

in the [view.py](#) when we want to get posted element "username", we POST the username to the view NOT GET.

I tried and it is ok.

Code:

```
def validate_username (request):
    if request.method == 'POST':
        username = request.POST['username']
        data = {
            'status': 200,
            'is_taken':
                User.objects.filter(username__iexact=username).exists(),
        }
        return JsonResponse(data)
```

Best Regards.

2 ^ | ▾ • Reply • Share ›



Magdiel XD → Farzam Taghipour • 4 months ago • edited

you need remove the conditional bro

^ | ▾ • Reply • Share ›



PiRaTe HiStory • 3 years ago

pls help me, i make everything like you, but it's not working

<https://pastebin.com/KGmuG9tE>

i gives error with csrf_token

my code on pastebin

3 ^ | v 1 • Reply • Share ›



hoosho ➔ PiRaTe HiStory • 2 years ago • edited

<https://gist.github.com/gas...>

corrects the problem automatically

1 ^ | v • Reply • Share ›



gear ➔ PiRaTe HiStory • 2 years ago

You have to serializeJSON() the data, search on google.

I'll leave a link: <https://github.com/macek/jq...>

^ | v • Reply • Share ›



arsenalacid ➔ gear • 2 years ago

Can you help me with this further?

1 ^ | v 1 • Reply • Share ›



Tiago Silva ➔ PiRaTe HiStory • 3 years ago

Have you tried to post the action like this <form

method="post" action="." data-validate-username-url="

{% url 'validate_username' %}"></form>?

^ | v • Reply • Share ›



Farzam Taghipour ➔ Tiago Silva • a year ago

This is not totally correct but works!

^ | v • Reply • Share ›



Farzam Taghipour ➔ PiRaTe HiStory • a year ago

take a look at my code!

^ | v 1 • Reply • Share ›



Ankit Sinha • 6 months ago • edited

request.GET.get('username', None)

In above request what is 'username' is it name=username from

Inspect element or some thing else

1 ^ | v • Reply • Share ›



Javed Khan Siddiqui ➔ Ankit Sinha • a month ago

yupp

in `models.py`

```
fname = models.CharField(verbose_name="FIRST  
NAME", max_length=20,  
help_text="Please Provide First Name",  
db_column="fname")
```

in browser after inspecting element

```
<input type="text" <b="">name="fname" maxlength="20"  
id="id_fname">
```

```
request.Get.get["field name which is in your browser  
after / are models.py fields(column in table/model)"]
```

^ | v • Reply • Share ›



Fon Desmond • a year ago

Great article Vitor, like to point a little spelling typo at you



misspelled when and wrote whe

1 ^ | v • Reply • Share ›



Phuong Dinh • a year ago

your tutorial is helping lots of people including me. Could you please guide after successfully sign up, how does it redirect to index page or welcome page? Since you used JsonResponse, I dont know how to use combined JsonResponse along with return render in order to direct to other views function to invoke html page. Please excuse, Im pretty new to Django. Im looking forward to hearing your answer.

1 ^ | v • Reply • Share ›



dimitrispalasis ➔ Phuong Dinh • a year ago

disqus_gll1QceRKK easy

6 ^ | v • Reply • Share ›



WarpspeedScp ➔ Phuong Dinh • a year ago

you can use javascript to redirect the page after validation instead of doing it through python.

2 ^ | v • Reply • Share ›



Martin → Phuong Dinh • 2 months ago

Not hard just add render to the view but outside the if statement

^ | v • Reply • Share ›



Chris Berberich • 3 years ago

Thank you for your help, You're examples work and are perfect for what ails my django project...

1 ^ | v 1 • Reply • Share ›



Ankit Sinha • 6 months ago

How we can put validation to non Form fields I mean to say fields which we created not provided by django

^ | v • Reply • Share ›



pankaj mishra • 9 months ago

Can you help me on this question

<https://stackoverflow.com/q...>

^ | v • Reply • Share ›



Sai Teja Limbagiri • 9 months ago

How to handle ajax call if the form consist of inputs csv file and some text inputs and also load the graph which return from the django views of data into a <div>.

^ | v • Reply • Share ›



SP hemanth • 10 months ago

hi

^ | v • Reply • Share ›



Nguyen • a year ago

I tried but always got: {user_is_taken: false} although I did key- in existing username. Js and [Views.py](#) copy exactly the same with yours.

Any advises?

^ | v • Reply • Share ›



Javed Khan Siddiqui → Nguyen • a month ago

are filter is working properly

do a test with shell

python [manage.py](#) shell

check your filter , and then implement in [view.py](#)

untill we see the code we can't say anything

^ | v • Reply • Share ›



jesvane • a year ago

Very nice. Thank you!

^ | v • Reply • Share ›



Rodrigo Oliveira • a year ago

Vou escrever em português pq vc é nosso! rsrs

Obrigado pelas suas postagens, ja me salvou varias vezes e continha salvando! Parabéns esta tudo muito bem explicado.

^ | v • Reply • Share ›



Diego Avila • a year ago

Hi, thanks by your tutorial is excelent.. i suggest change for the new method on ajax... replace the succes and error by ..done and fail.... Deprecation Notice: The jqXHR.success(), jqXHR.error(), and jqXHR.complete() callbacks are removed as of jQuery 3.0. You can use jqXHR.done(), [jqXHR.fail\(\)](#), and jqXHR.always() instead.....link....: <http://api.jquery.com/jquer...>

^ | v • Reply • Share ›



Sachin Salve • 2 years ago

hello, it it possible for you for you make article django-autocomplete with jquery.?

^ | v • Reply • Share ›



carlos • 2 years ago

hey!,

thanks a lot for your tutorials! im your follower!

i have a question... how i can filter information of data with django and ajax in the same template!?

^ | v • Reply • Share ›



arsenalacid • 2 years ago

Hi i tried this and it seems as thought my data is being serialized as I am getting GET request in CMD with the data but im not getting alert. Any help?

^ | v • Reply • Share ›



arsenalacid ➔ arsenalacid • 2 years ago

If anyone is having trouble with this :

<https://stackoverflow.com/q...>

^ | v • Reply • Share ›



arsenalacid • 2 years ago

Can someone tell me what app.js is? thank you for the article

^ | v • Reply • Share ›



BareTech • 2 years ago

(urgent) Need help submitting Django form (using django-crispy-forms) with Ajax How do I:

1 validate input

2 submit without reloading

3 display errors in case of validation failure

code here:

<https://pastebin.com/VpQRuG4D>

^ | v • Reply • Share ›



Javed Khan Siddiqui → BareTech • a month ago

not seeing anything in pastebin...

hope its done now

^ | v • Reply • Share ›



robertf57 • 2 years ago

It would be helpful if you would print out a directory tree listing so we can see where each file is relative to the others.

^ | v • Reply • Share ›



elprofe • 2 years ago

Thank you man, I've been following your Django tutorials, this is some high quality stuff. Thanks again!

^ | v • Reply • Share ›



April Kroll • 2 years ago

What would the class based view (and associated url) for this look like ?

I'm really just trying to implement a simple search. I think this will help, but I'm not sure. Thanks. Sorry to bug you.

^ | v • Reply • Share ›



Alejandro santibañez • 2 years ago

very good you work about javascript with django

^ | v • Reply • Share ›



md tuhin • 2 years ago

Little bit of confusion ! You use post method in form but for retrieving data you write

`username = request.GET.get('username', None)` Should not be it
`request.POST.get('username')`

please someone help :)

^ | v • Reply • Share ›



Jarad Collier → md tuhin • 2 months ago

Yo, to actually understand this, in your view, just `print(request)` and you will see something like:
`<wsgirequest: get="" '="" videos="" ajax="" assign_class="" ? class_chosen="ASMR&video_id=980">` . Notice the question mark and parameters? Now you understand why he's using `request.GET.get ...` it's to get the username (which is a parameter) from the URL.

^ | v • Reply • Share ›



Liran → md tuhin • a year ago

If in AJAX, type "POST" is used, It should be `request.POST.get`.

^ | v • Reply • Share ›



Ujjawal Malik → md tuhin • 2 years ago

`request.GET.get('username',None)` will return username if it's there else it will return None just to prevent any error.

This is the correct approach to call `get()` on any request method as you handle the error by returning None.

^ | v • Reply • Share ›



osama abumar • 2 years ago

thanks for this great post

^ | v • Reply • Share ›



Pranjal Singh • 3 years ago

I am using a separate .js file. While doing `{% url 'abc' %}`, I am getting an error.

Is there an alternative to it?

^ | v • Reply • Share ›



Minjun • 3 years ago

Great post! since I get a lot of help from your blog, I added this on my bookmark and the bookmark name is "Django Master"

^ | v • Reply • Share ›



Is Fernandez • 3 years ago

Great post, but the real problem with the ajax is using HTTP method POST. How to work with ajax POST in django?

^ | v • Reply • Share ›



Mihai • 3 years ago

Very interesting post, it helps a lot.

But, I can't figure it out how to validate all the fields without bothering myself repeating the same thing for all fields.

Is there a DRY method in order to do that?

Thanks again.

^ | v • Reply • Share ›



Carlos Tenorio Pérez • 3 years ago

Great Post! But inserting the data in this way may involve an attack xss :o

^ | v • Reply • Share ›



Vitor Freitas Mod ➔ Carlos Tenorio Pérez • 3 years ago

Hey, Carlos!

Can you elaborate a little bit more on that subject?

Because the ajax request that's being used here with GET request is not modifying any data

The data is modified via POST and is protected with the csrf token

Thanks for the comment! :-)

^ | v • Reply • Share ›



Carlos Tenorio Pérez ➔ Vitor Freitas
• 3 years ago

For example:

With the preconditions:

✈ Subscribe to our Mailing List

Receive updates from the Blog!

Popular Posts



[How to Extend Django User Model](#)



[How to Setup a SSL Certificate on Nginx for a Django Application](#)



[How to Deploy a Django Application to Digital Ocean](#)

© 2015-2019 simple is better than complex cc by-nc-sa 3.0 // [about](#) [contact](#) [faq](#) [cookies](#) [privacy](#)
[policy](#)