

CM3035 Advanced Web Development

Lesson L11

Created by Ben Gay

Introduction to AJAX

AJAX stands for Asynchronous **JavaScript** And **XML**

update **asynchronously** by **exchanging** data to and from the **server**

update parts of a web page **without** reloading the complete web page

How does it work?

1. An event occurs on a web page example button click, etc.
2. An XMLHttpRequest object is created and sends the request to the server .
3. The server responds to the request.
4. The response is captured and then server respond back with response data.

L11app/templates/base.html (main / front end html page)

localhost:8000/L11app/

OEM USB Drivers... Good book for be... 127.0.0.1:8000/pay/ Move WordPress f... http://192.168.0.1... hdb www.facebooklike... D-Link real-estate DlinkHomeCam php helps »

Nick name

First name

Last name

Likes

What is your birth date?

January

1

1980

Lives in

Create Friend

Nick name	First name	Last name	Likes	DOB	lives in
tall guy	john	tan	baseball	1980-01-01	usa
funny guy	peter	wong	billard	1983-03-03	Singapore
smart guy	jim	gan	tennis	1984-04-04	canada
happy guy	roger	cheng	swimming	1985-05-05	hongkong
sporty guy	jack	lim	gym	1986-06-06	malaysia

L11app/templates/base.html (main / front end html page)

```
views.py  settings.py  urls.py  index.html  base.html
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5      <meta charset="utf-8">
6      <meta http-equiv="X-UA-Compatible" content="IE=edge">
7      <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
8      <title>{% block title %}{% endblock title %}</title>
9      <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css">
10     {% block style %}
11     {% endblock style %}
12 </head>
13
14 <body>
15
16     {% block content %}
17     {% endblock content %}
18
19     <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
20     <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"></script>
21     <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>
22
23     {% block javascript %}
24     {% endblock javascript %}
25
26 </body>
27
28 </html>
29
```

L11proj/settings.py

```
30
31 # Application definition
32
33 INSTALLED_APPS = [
34     'django.contrib.admin',
35     'django.contrib.auth',
36     'django.contrib.contenttypes',
37     'django.contrib.sessions',
38     'django.contrib.messages',
39     'django.contrib.staticfiles',
40     'L11app.apps.L11AppConfig',
41 ]
42
```

L11proj/urls.py

```
4         2. Add a URL to urlpatterns: path('blog/', include
5         """
6         from django.contrib import admin
7         from django.urls import path , include
8
9         urlpatterns = [
10             path('admin/', admin.site.urls),
11             path('L11app/', include('L11app.urls')),
12
13         ]
14
```

L11app/models.py

```
views.py  urls.py — L11proj  models.py  urls.py — L11app

from django.db import models

# Create your models here.
class Friend(models.Model):
    # NICK NAME should be unique
    nick_name = models.CharField(max_length=150, unique = True)
    first_name = models.CharField(max_length=150)
    last_name = models.CharField(max_length=150)
    likes = models.CharField(max_length = 300)
    dob = models.DateField(auto_now=False, auto_now_add=False)
    lives_in = models.CharField(max_length=200, null = True, blank = True)

    def __str__(self):
        return self.nick_name
```


Prepare to run server

Python3 manage.py makemigrations

Python3 manage.py migrate

Python3 manage.py runserver

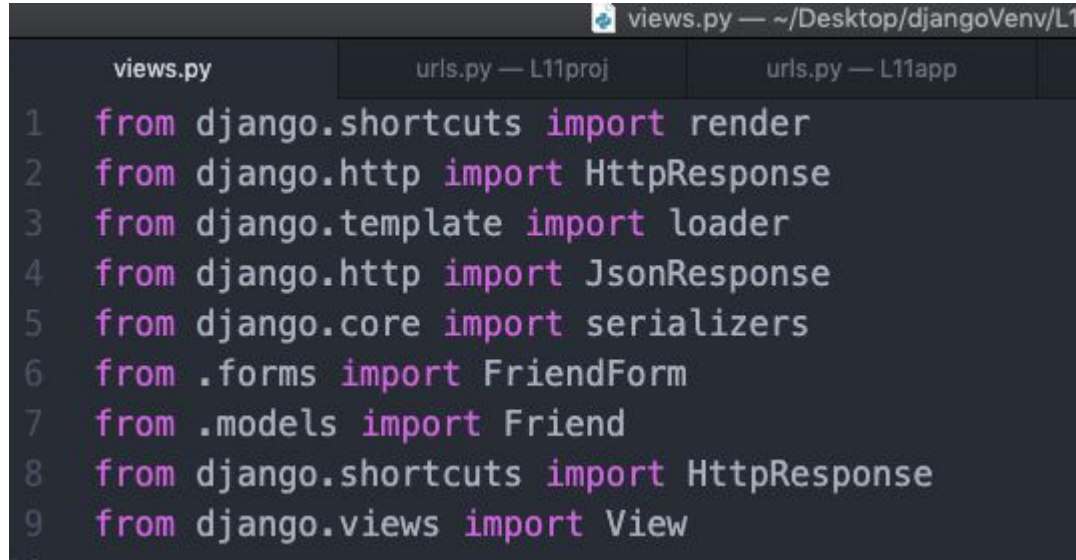
Let's Begin with POST Request

L11app/forms.py

```
forms.py — ~/Desktop/djangoven/L11proj/L11app
views.py  urls.py — L11proj  models.py  forms.py  urls.py — L11app

1  from .models import Friend
2  from django import forms
3  import datetime
4
5  class FriendForm(forms.ModelForm):
6      ## change the widget of the date field.
7      dob = forms.DateField(
8          label='What is your birth date?',
9          # change the range of the years from 1980 to currentYear - 5
10         widget=forms.SelectDateWidget(years=range(1980, datetime.date.today().year-5))
11     )
12
13     def __init__(self, *args, **kwargs):
14         super(FriendForm, self).__init__(*args, **kwargs)
15         ## add a "form-control" class to each form input
16         ## for enabling bootstrap
17         for name in self.fields.keys():
18             self.fields[name].widget.attrs.update({
19                 'class': 'form-control',
20             })
21
22     class Meta:
23         model = Friend
24         fields = ("__all__")
25
```

L11app/views.py



The image shows a code editor window with a dark theme. The title bar at the top reads "views.py — ~/Desktop/djangoVenv/L1". Below the title bar, there are three tabs: "views.py", "urls.py — L11proj", and "urls.py — L11app". The "views.py" tab is active, displaying the following Python code:

```
1 from django.shortcuts import render
2 from django.http import HttpResponse
3 from django.template import loader
4 from django.http import JsonResponse
5 from django.core import serializers
6 from .forms import FriendForm
7 from .models import Friend
8 from django.shortcuts import HttpResponse
9 from django.views import View
```

L11app/views.py

```
25
26 def indexView(request):
27     form = FriendForm()
28     friends = Friend.objects.all()
29     return render(request, "index.html", {"form": form, "friends": friends})
30
31 def postFriend(request):
32     # request should be ajax and method should be POST.
33     #if request.is_ajax and request.method == "POST":
34     if request.headers.get('x-requested-with') == 'XMLHttpRequest' and request.method == "POST":
35         # get the form data
36         form = FriendForm(request.POST)
37         # save the data and after fetch the object in instance
38         if form.is_valid():
39             instance = form.save()
40             # serialize in new friend object in json
41             ser_instance = serializers.serialize('json', [instance, ])
42             # send to client side.
43             return JsonResponse({"instance": ser_instance}, status=200)
44         else:
45             # some form errors occured.
46             return JsonResponse({"error": form.errors}, status=400)
47
48     # some error occured
49     return JsonResponse({"error": ""}, status=400)
50
```

L11app/urls.py

```
views.py  urls.py — L11proj  models.py  forms.py  urls.py — L11app
1  from django.urls import path
2  from . import views
3  from django.urls import include, path
4  from django.urls import path
5  from L11app.views import (indexView , postFriend , checkNickName , FriendView)
6
7
8
9  urlpatterns = [
10     #path('', views.index, name='index'),
11     path('', indexView),
12     path('post/ajax/friend', postFriend, name = "post_friend"),
13     path('get/ajax/validate/nickname', checkNickName, name = "validate_nickname"),
14     path("", FriendView.as_view(), name = "friend_cbv"),
15
16 ]
17
```


L11app/templates/index.html (content)

```
views.py  urls.py — L11proj  index.html  models.py  forms.py
31  <tbody>
32  {% for friend in friends %}
33  <tr>
34      <td>{{friend.nick_name}}</td>
35      <td>{{friend.first_name}}</td>
36      <td>{{friend.last_name}}</td>
37      <td>{{friend.likes}}</td>
38      <td>{{friend.dob | date:"Y-m-d"}}</td>
39      <td>{{friend.lives_in}}</td>
40  </tr>
41  {% endfor %}
42  </tbody>
43  </table>
44
45  </div>
46  {% endblock content %}
47
48
49  {% block javascript %}
50  <script>
51      /*
52       On submitting the form, send the POST ajax
53       request to server and after successfull submission
54       display the object.
55      */
56      $("#friend-form").submit(function (e) {
57          // preventing from page reload and default actions
58          e.preventDefault();
59          // serialize the data for sending the form data.
60          var serializedData = $(this).serialize();
```


L11app/templates/index.html (javascript)

```
views.py  urls.py — L11proj  index.html  models.py  forms.py
49 {% block javascript %}
50 <script>
51     /*
52         On submitting the form, send the POST ajax
53         request to server and after successfull submission
54         display the object.
55     */
56     $("#friend-form").submit(function (e) {
57         // preventing from page reload and default actions
58         e.preventDefault();
59         // serialize the data for sending the form data.
60         var serializedData = $(this).serialize();
61         // make POST ajax call
62         $.ajax({
63             type: 'POST',
64             url: "{% url 'post_friend' %}",
65             data: serializedData,
66             success: function (response) {
67                 // on successfull creating object
68                 // 1. clear the form.
69                 $("#friend-form").trigger('reset');
70                 // 2. focus to nickname input
71                 $("#id_nick_name").focus();
72
73                 // display the newly friend to table.
74                 var instance = JSON.parse(response["instance"]);
75                 var fields = instance[0]["fields"];
76                 $("#my_friends tbody").prepend(
77                     `<tr>
```

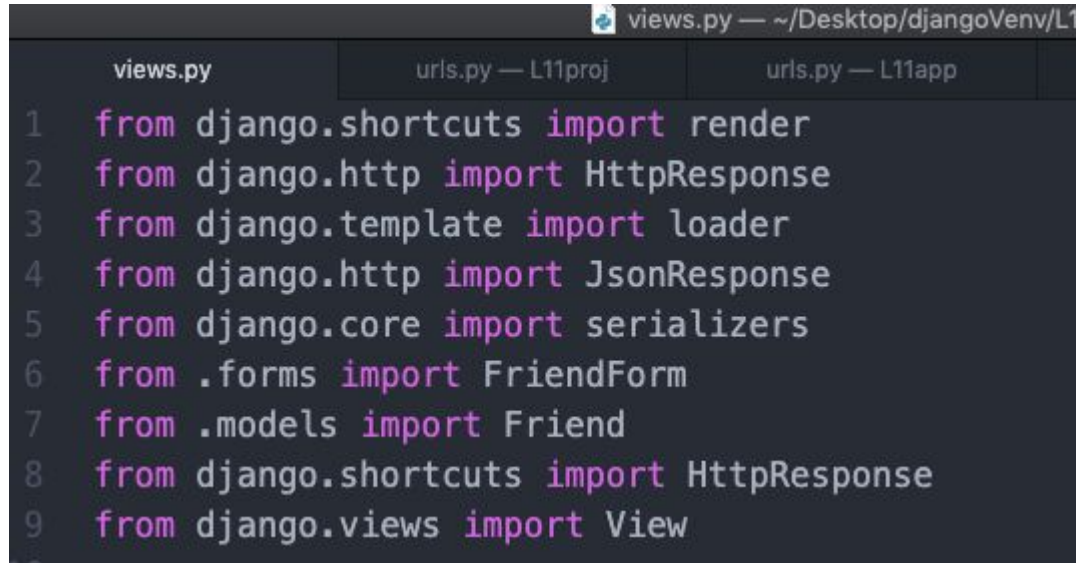
L11app/templates/index.html (javascript)

```
views.py  |  urls.py — L11proj  |  index.html  |  models.py
69      $('#friend-form').trigger('reset');
70      // 2. focus to nickname input
71      $('#id_nick_name').focus();
72
73      // display the newly friend to table.
74      var instance = JSON.parse(response["instance"]);
75      var fields = instance[0]["fields"];
76      $('#my_friends tbody').prepend(
77      `<tr>
78          <td>${fields["nick_name"]} | ""</td>
79          <td>${fields["first_name"]} | ""</td>
80          <td>${fields["last_name"]} | ""</td>
81          <td>${fields["likes"]} | ""</td>
82          <td>${fields["dob"]} | ""</td>
83          <td>${fields["lives_in"]} | ""</td>
84      </tr>`
85      )
86  },
87  error: function (response) {
88      // alert the error if any error occurred
89      alert(response["responseJSON"]["error"]);
90  }
91  })
92  })
93  </script>
94
95  <script>
96      /*
```

GET Request

L11app/views.py

Checks if Nickname is Unique

A screenshot of a code editor window. The title bar shows 'views.py — ~/Desktop/djangoVenv/L11'. The editor has three tabs: 'views.py', 'urls.py — L11proj', and 'urls.py — L11app'. The 'views.py' tab is active, showing the following Python code:

```
1 from django.shortcuts import render
2 from django.http import HttpResponse
3 from django.template import loader
4 from django.http import JsonResponse
5 from django.core import serializers
6 from .forms import FriendForm
7 from .models import Friend
8 from django.shortcuts import HttpResponse
9 from django.views import View
```

L11app/views.py

Checks if Nickname is Unique

```
views.py | urls.py — L11proj | index.html | models.py | forms.py | urls.py — L11app
51
52 def checkNickName(request):
53     # request should be ajax and method should be GET.
54     if request.headers.get('x-requested-with') == 'XMLHttpRequest' and request.method == "GET":
55         # get the nick name from the client side.
56         nick_name = request.GET.get("nick_name", None)
57         # check for the nick name in the database.
58         if Friend.objects.filter(nick_name = nick_name).exists():
59             # if nick_name found return not valid new friend
60             return JsonResponse({"valid":False}, status = 200)
61         else:
62             # if nick_name not found, then user can create a new friend.
63             return JsonResponse({"valid":True}, status = 200)
64
65     return JsonResponse({}, status = 400)
66
```

L11app/urls.py

```
views.py  urls.py — L11proj  urls.py — L11app  index.html  models.py  f
1  from django.urls import path
2  from . import views
3  from django.urls import include, path
4  from django.urls import path
5  #from L11app.views import (indexView , postFriend , checkNickName , FriendView)
6  from L11app.views import (indexView , postFriend , checkNickName)
7
8
9  urlpatterns = [
10     #path('', views.index, name='index'),
11     path('', indexView),
12     path('post/ajax/friend', postFriend, name = "post_friend"),
13     path('get/ajax/validate/nickname', checkNickName, name = "validate_nickname"),
14     #path("", FriendView.as_view(), name = "friend_cbv"),
15
16 ]
17
```

L11app/templates/index.html (javascript)

```
index.html — ~/Desktop/djangoVenv/L11proj/L11app/templates
views.py  urls.py — L11proj  urls.py — L11app  index.html  models.py  forms.py  base

95 <script>
96  /*
97   On focus out on input nickname, call AJAX get request to check if the nickName already exists or not.*/
98   $("#id_nick_name").focusout(function (e) {
99       e.preventDefault();
100       // get the nickname
101       var nick_name = $(this).val();
102       // GET AJAX request
103       $.ajax({
104           type: 'GET',
105           url: "{% url 'validate_nickname' %}",
106           data: {"nick_name": nick_name},
107           success: function (response) {
108               // if not valid user, alert the user
109               if(!response["valid"]){
110                   alert("You cannot create a friend with same nick name");
111                   var nickName = $("#id_nick_name");
112                   nickName.val("")
113                   nickName.focus()
114               }
115           },
116           error: function (response) {
117               console.log(response)
118           }
119       })
120   })
121 </script>
122 {% endblock javascript %}
123
```

End of Lesson 11