

Topic: Car Listing: Django Messages Part 12B

Speaker: Udemy Instructor Rathan Kumar | Notebook: Django Project: Car Listing



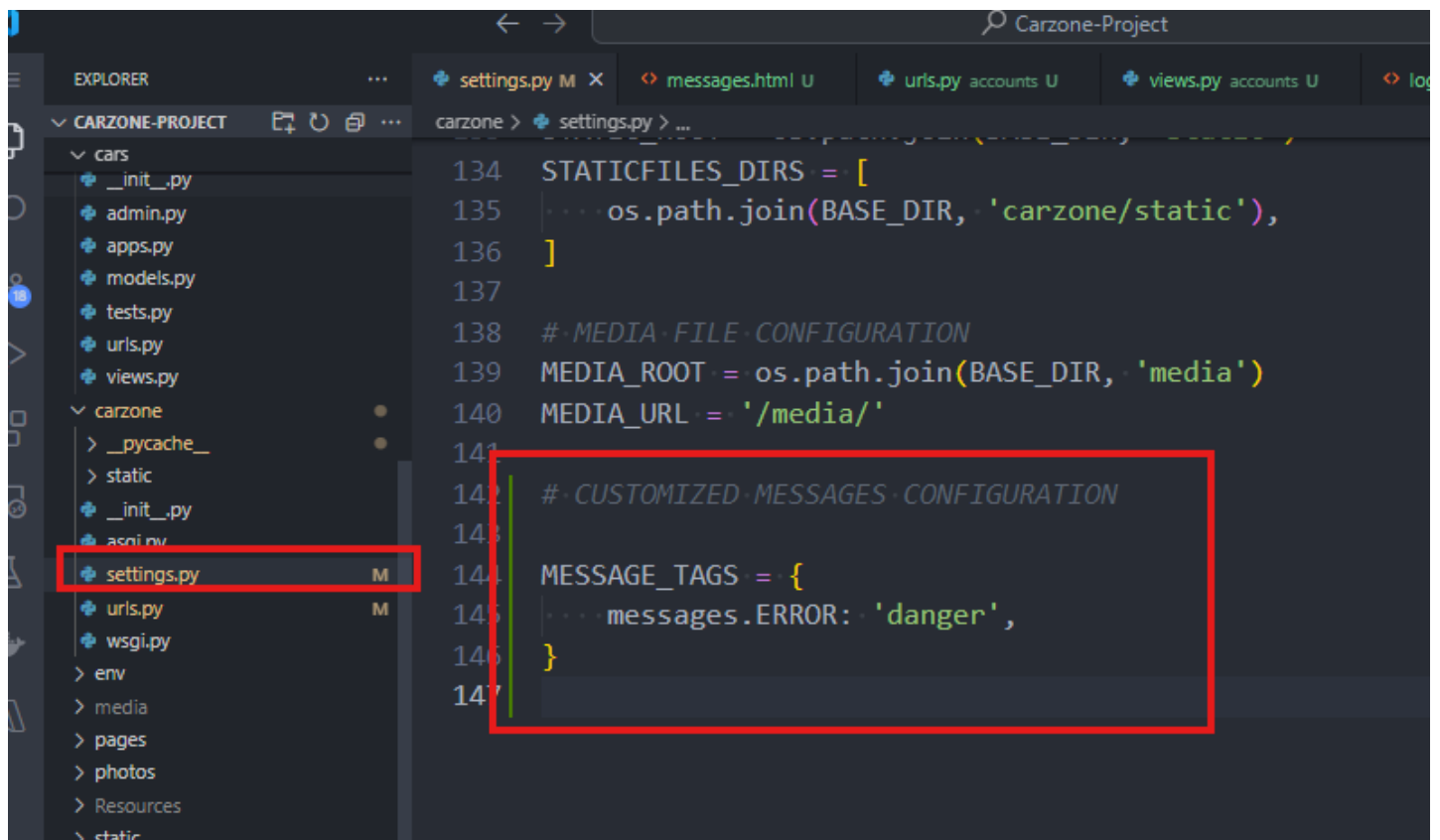
1. See the [DJANGO MESSAGES FRAMEWORK documentation here.](#)
2. We copy this block into SETTINGS.PY and modify this.

FROM:

```
from django.contrib.messages import constants as messages

MESSAGE_TAGS = {
    messages.INFO: "",
    50: "critical",
}
```

TO:



3. Create a new file MESSAGES.HTML in the TEMPLATES\INCLUDES folder. This can be used in ANY Django with Bootstrap projects.

```

<!-- this is for flash messages we show on our web pages
classes here are from the bootstrap.
-->
{% if messages %}
{% for message in messages %}
<div id="message" class="container">
  <div class="alert alert-{{ message.tags }}" alert-dismissible role="alert">
    <button type="button" class="close" data-dismiss="alert"><span aria-hidden="true">&times;</span></button>
    <strong>
      {% if message.level == DEFAULT_MESSAGE_LEVELS.ERROR %}
      Error:
      {% else %}
      {{message.tags|title}}
      {% endif %}
    </strong>
    {{message}}
  </div>
</div>
{% endfor %}
{% endif %}

```

The screenshot shows the Visual Studio Code interface with the following details:

- Explorer:** The project structure is visible, including folders like 'accounts', 'templates', and 'static'. The file 'messages.html' is selected and highlighted in red.
- Code Editor:** The content of 'messages.html' is displayed, showing the Django template code for rendering flash messages. The code is highlighted with a red box. It includes a loop for messages, a conditional for error messages, and a close button for each message.

4. To use the message framework, go to ACCOUNTS\VIEWS.PY and update as

```

1 from django.shortcuts import render, redirect
2 from django.contrib import messages
3
4 # Create your views here.
5
6
7 def login(request):
8     ... return render(request, 'accounts/login.html')
9
10
11 def register(request):
12     ... if request.method == 'POST':
13         ... # for testing
14         ... # print('this is the post method')
15         ... # this calls the error message functionality with INCLUDES\MESSAGES.HTML
16         ... messages.error(request, 'This is error message')
17         ... return redirect('register')
18     else:
19         ... return render(request, 'accounts/register.html')
20
21
22
23 def dashboard(request):
24     ... return render(request, 'accounts/dashboard.html')
25
26
27 def logout(request):
28     ... return redirect('home')
29

```

5. Now to display this message on our webpage like REGISTER.HTML, update the REGISTER.HTML. You have to write the line for the error message in the location you want to see it.

The browser screenshot shows the 'REGISTER AN ACCOUNT' page for 'Car Zone'. A red box highlights an error message: "ERROR: THIS IS ERROR MESSAGE". Below the error message is a registration form with fields for First Name, Last Name, carzone_admin, Email Address, Password, and Confirm Password.

The code editor shows the HTML template for 'register.html'. A red box highlights the line: `{% include 'includes/messages.html' %}`. The code also shows the form structure and the inclusion of the messages.html template.

```

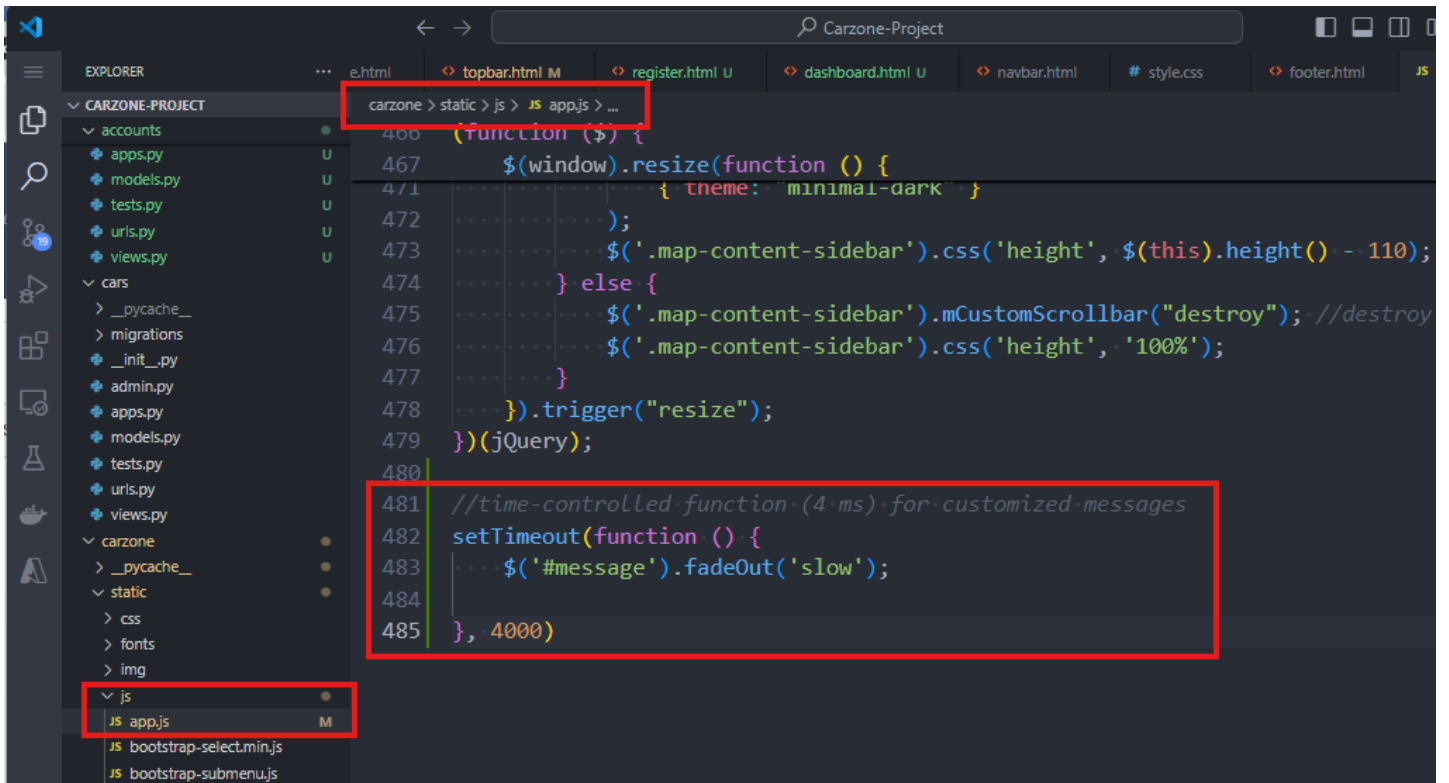
19 end -->
20
21 <!-- This displays our customized error message -->
22 <!-- This displays our customized error message -->
23 <!-- This displays our customized error message -->
24 <!-- This displays our customized error message -->
25 <!-- This displays our customized error message -->
26 <!-- This displays our customized error message -->
27 <!-- This displays our customized error message -->
28 <!-- This displays our customized error message -->
29 <!-- This displays our customized error message -->
30 <!-- This displays our customized error message -->
31 <!-- This displays our customized error message -->
32 <!-- This displays our customized error message -->
33 <!-- This displays our customized error message -->
34 <!-- This displays our customized error message -->
35 <!-- This displays our customized error message -->
36 <!-- This displays our customized error message -->
37 <!-- This displays our customized error message -->
38 <!-- This displays our customized error message -->
39 <!-- This displays our customized error message -->
40 <!-- This displays our customized error message -->
41 <!-- This displays our customized error message -->
42 <!-- This displays our customized error message -->
43 <!-- This displays our customized error message -->
44 <!-- This displays our customized error message -->
45 <!-- This displays our customized error message -->
46 <!-- This displays our customized error message -->
47 <!-- This displays our customized error message -->
48 <!-- This displays our customized error message -->
49 <!-- This displays our customized error message -->
50 <!-- This displays our customized error message -->
51 <!-- This displays our customized error message -->
52 <!-- This displays our customized error message -->

```

6. If you want your ERROR MESSAGE TO DISAPPER IN A GIVEN TIME instead of closing it, update your CARZONE\STATIC\JS\APPS.JS file as:

```
//time-controlled function (4 ms) for customized messages
```

```
setTimeout(function () {  
    $('#message').fadeOut('slow');  
}, 4000)
```



7. Run a COLLECTSTATIC for all updates made from the CARZONE\STATIC FILES AND FOLDERS.

```
$ python manage.py collectstatic
```

8. Run the server again. If the message does not disappear in 4 milliseconds, reload your REGISTER.HTML, VIEW THE PAGE SOURCE and open the APPS.JS to see if it has our SETTIMEOUT function. Reload again the webpage and test.

It should disappear after 4MS.

9. You may now continue with your USER REGISTRATION.