Topic: Custom Management Command Part 6: Adding Comments or Messages

Speaker: Udemy Instructor Rathan Kumar | Notebook: Django: Automating Common Tasks



Django has built-in MESSAGES FRAMEWORK that we can use to show ERROR OR SUCCESS MESSAGES. See Django's documentation.

SETTINGS.PY has already the configurations needed to support the messages.

1. So, we just need to include the following in our SETTINGS.PY.

From Django documentation, we copy this and modify based on our needs.

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

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

in our SETTINGS.PY, we modify it to where 'danger' is a bootstrap name.

```
EXPLORER

◆ settings.py M X  
◆ models.py uploads U

                                                                                     o importdata.html U
                                                                                                          helloworld.py
                                                                admin.py uploads U
AUTOMATINGCOMMONTASKS
                         autocommontasks_main > 💠 settings.py > ...
 🗸 autocommontasks_m... 🌘
                                 DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'
  __init__py
 settings.py
  views.py
                                 MEDIA_URL = '/media/'
 wsgi.py
                                 MEDIA_ROOT = BASE_DIR / 'media'

∨ management\commands

                                 MESSAGE_TAGS = {
  exportdata.py
                                       messages.ERROR: "danger",
   🗣 greeting.py
                                       50: "critical",
    importdata.py
   insertdata.py
                           138
  > migrations
```

2. Then in the TEMPLATES FOLDER, create a new HTML, ALERT.HTML and design as:

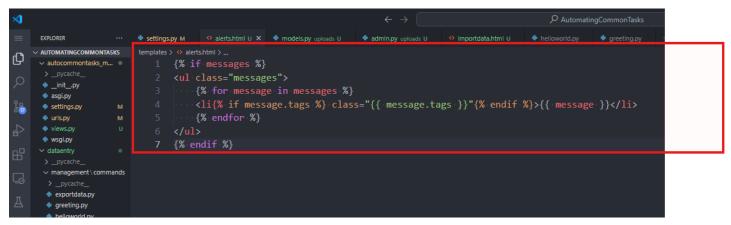
from our documentation:

Displaying messages ¶

```
get_messages(request)[source]
```

In your template, use something like:

So, in our ALERTS.HTML, we add the same code:



3. Now you can call this ALERTS.HTML in ANY HTML that will require messages or comments.

So, in DATAENTRY\IMPORTDATA.HTML, we add

```
<div class="container">
   <h3 class="text-center">Import Data From CSV File to Database Tables</h3>
    <form action="{% url 'import_data' %}" method="POST" enctype="multipart/form-d
        style="max-width: 600px;margin:auto;padding-top: 50px; ">
       {% csrf_token %}
       <div class="form-group">
            <label for="file path">Upload CSV File</label>
            <input type="file" name="file path" class="form-control" required>
       </div>
       <div class="form-group">
            <label for="model_name">Select Database Table</label>
            <select name="model name" class="form-control" required>
                <option value="" disabled selected>Select</option>
                {% for model in custom models %}
                <option value="{{model}}">{{model}}</option>
                {% endfor %}
            </select>
       </div>
        <input type="submit" value="Import Data" class="btn btn-primary">
       {% include 'alerts.html' %}
    </form>
```

4. To add our custom messages, we go to DATAENTRY\VIEWS.PY

```
uris.pv dataentry U
                                                                                                      views.pv dataent

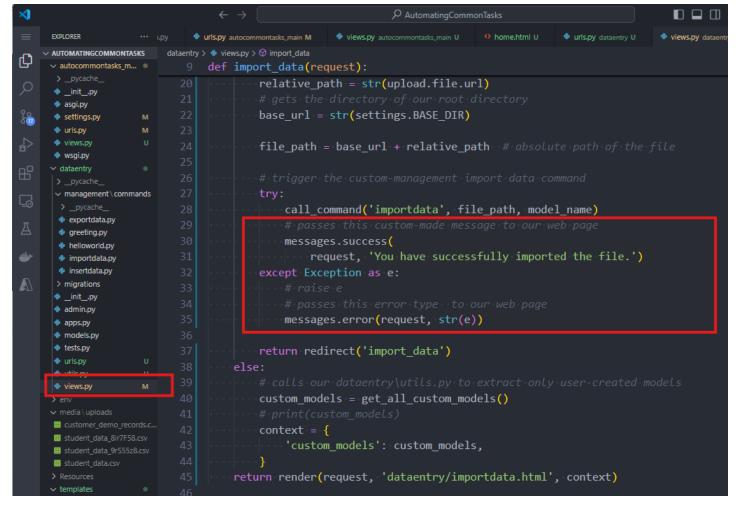
∨ AUTOMATINGCOMMONTASKS

                    dataentry > 🌵 views.py > ..
                          from django.shortcuts import render, redirect
                          from .utils import get_all_custom_models
 __init__.py
                          from uploads.models import Upload
 asgi.py
                          from django.conf import settings
 settings.py
                          from diango.core.management import call command
 urls.py
 views.py
                          from django.contrib import messages
 wsgi.py

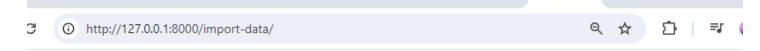
✓ dataentry

∨ management \ commands

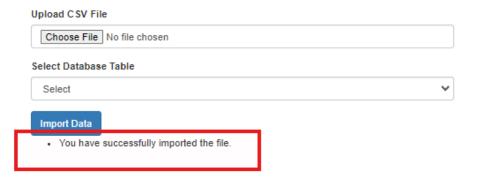
                           def import_data(request):
  exportdata.py
  greeting.py
                                   file_path = request.FILES.get('file_path')
  importdata.py
  insertdata.py
                                   model_name = request.POST.get('model_name')
  > migrations
 __init__.py
  admin.py
                                   upload = Upload.objects.create(file=file_path, model_name=model_name)
 apps.py
 tests.py
 · urls.py
                                   relative_path = str(upload.file.url)
```



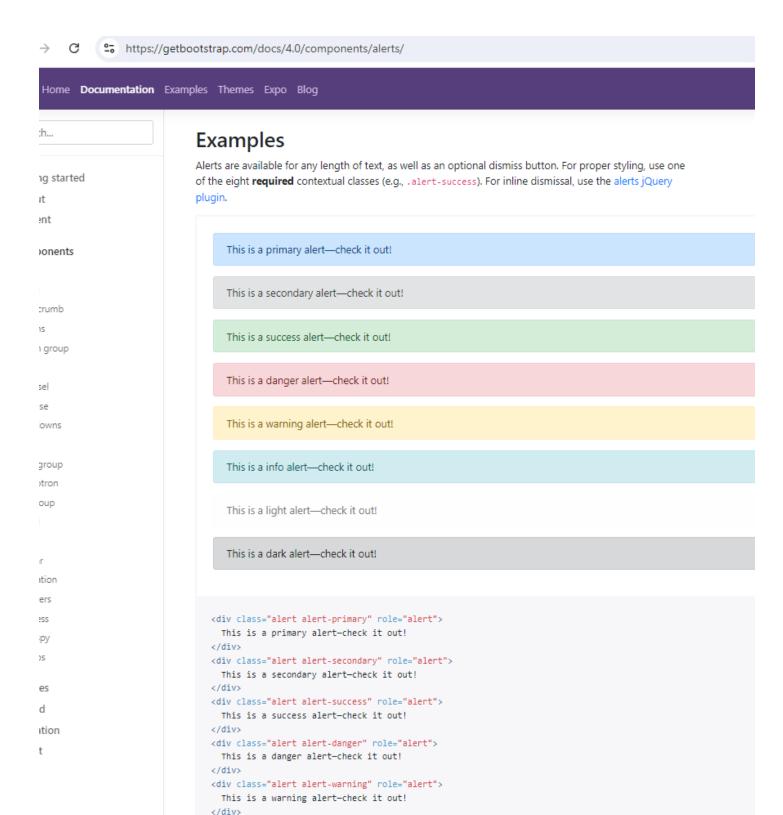
5. Now when you add a CSV file again, you should be able to see an error message.



Import Data From CSV File to Database Tables



6. Now to make the style of the message that follows the bootstrap style, go to this BOOTSTRAP ALERT documentation,



ALERTS.HTML BEFORE:

ALERTS.HTML AFTER:

7. Run the server again and check the message shown.

Saving Student.CSV to Student Model.

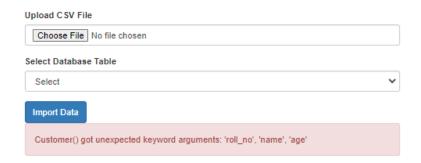




Saving Student.CSV to Customer Model



Import Data From CSV File to Database Tables



Or a different file to CUSTOMER model.

8. To modify the error message into something user-friendly especially the user has used a different file with a different model, we update our IMPORTDATA.PY

To get the fields of our model, in our IMPORTDATA.PY, we add this:

*#*compare*CSV*header*with*the*model's*field*names*and*throw*appropriate*message
*#*get*all*the*field*names*of*the*model*that*we*found
model_fields=*[field.name*for*field*in*target_model._meta.fields]
*print(model_fields)

So, when we upload our STUDENT.CSV FILE, we see this in our terminal:

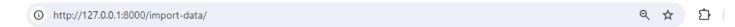
```
[11/Aug/2024 15:41:11] "GET /import-data/ HTTP/1.1" 200 1907 ['id', 'roll_no', 'name', 'age']
Data imported from the CSV file successfully.
```

CUSTOMER MODEL:

```
Django version 4.2.14, using settings 'autocommontasks_main.setting:
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.

['id', 'customer_name', 'country']
[11/Aug/2024 16:07:29] "POST /import-data/ HTTP/1.1" 302 0
[11/Aug/2024 16:07:29] "GET /import-data/ HTTP/1.1" 200 2251
```

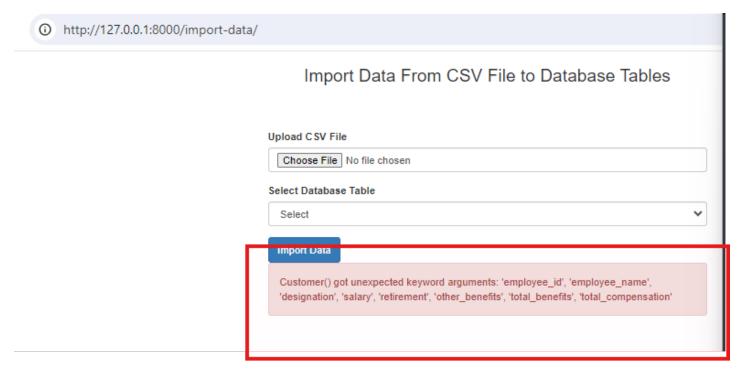
So, when we import a different file like EMPLOYEE.CSV into CUSTOMER Model, we get this user- friendly message:



Import Data From CSV File to Database Tables



Instead of this:



9. Your IMPORTDATA.PY should be:

```
EXPLORER

    # urls.py autocommontasks_main M
    # views.py auto-

✓ AUTOMATINGCOMMONTASKS dataentry > management > commands > 

importdata.py >

Ð
                           13 class Command(BaseCommand):
     dataentry
                                  def handle(self, *args, **kwargs):
                                          target model = None
                                           for my_app_config in apps.get_app_configs():
       exportdata.pv
      greeting.py
                                                   target_model = apps.get_model(my_app_config.label, model_name)
      > migrations
      __init_.py
      apps.pv
      models.py
                                          if not target_model:
      views.py
                                          model_fields = [field.name for field in target_model._meta.fields]
                                          print(model fields)
      .gitignore
     exported_Customer_data_2...
                                          with open(file_path, 'r') as file:
     material exported students data 20...
                                               reader = csv.DictReader(file)
     manage.py
                                                   raise DataError(
                                                        f'CSV file does not match with the {model_name} table fields')
                                                    target_model.objects.create(**row)
    > OUTLINE
    > TIMELINE
```

10. But this will give us an error message since we have the field 'ID' in our table but not in our CSV file, so we update our IMPORTDATA.PY as

11. IF WE WANT TO HIDE THE ERROR MESSAGES RIGHT AWAY AFTER IT WAS DISPLAYED, we create a JAVASCRIPT. But we need to update our SETTINGS.PY STATIC FILES.

FROM:

```
# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/4.2/howto/static-files/
STATIC_URL = 'static/'
```

TO:

```
STATIC_URL = 'static/'

STATICFILES_DIRS = ['autocommontasks_main/static'] * # * this * is * the * source * folder # * this * is * the * destination * folder * after * COLLECTSTATIC * command STATIC_ROOT * = * BASE_DIR * / * 'static'
```

12. Create an ID FOR YOUR ALERTS.HTML

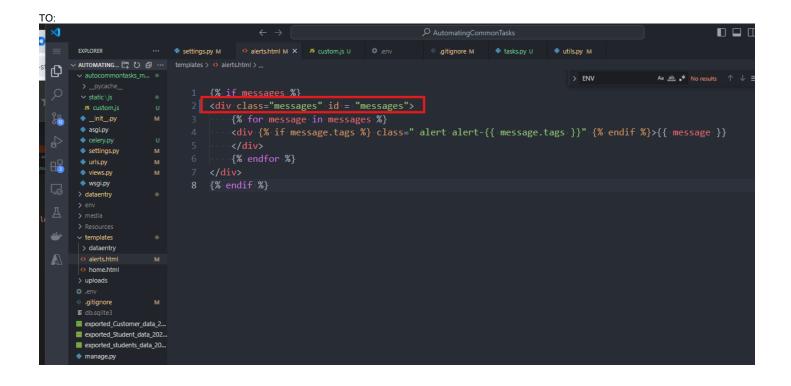
FROM:

```
×
                                                                                                                                                settings.py M
                                                                                                           utils.py M
                                                                                              tasks.py U
                                                                                .gitignore M

✓ AUTOMATINGCOMMONTASKS

Ф
                                                                                                                > ENV
                                                                                                                                   Aa ab * No results
                                  {% if messages %}
                                 <div class="messages">
                                       {% for message in messages %}
                                      <div {% if message.tags %} class=" alert alert-{{ message.tags }}" {% endif %}>{{ message }}
      settings.py
                                      {% endfor %}
      views.py
      wsgi.py
                                  {% endif %}
      > dataentry

    alerts.html
```



13. In our main project, we create a new folder and JS file, so in AUTOCOMMONTASKS_MAIN\STATIC\JS\CUSTOM.JS



14. Since we have our own static files like CUSTOM.JS, we need to run COLLECTSTATIC COMMAND in our DJANGO-SERVER bash terminal. This will create a new folder STATIC in the root directory.

```
V AUTOMATINGCOMMONTASKS
O
                                                                                                                                                                                                                                                                                   > ENV
                                                                                                                                                                                                                                                                                                                                 Aa <u>ab</u> * 1 of 1
                                                                                   LANGUAGE CODE = 'en-us'
               asgi.py
                                                                                   TIME_ZONE = 'UTC'
                                                                                  USE_I18N = True
               urls.py
               views.py
               wsgi.py
                                                                     118 USE TZ = True
              > dataentry
                                                                                   STATIC_URL = 'static/'
                                                                                   STATICFILES_DIRS = ['autocommontasks_main/static'] * # this is the source folder
                > dataentry
                                                      м
                o alerts.htm
               o home.html
                                                                                   STATIC_ROOT = BASE_DIR / 'static'
              > uploads
                 .gitignore
             exported_Customer_data_2...
             exported_Student_data_202...
                                                                                   DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'
             exported_students_data_20...
              manage.pv
                                                                     136 MEDIA_URL = '/media/'
                                                                                  MEDIA ROOT = BASE DIR / 'media'
                                                                   139 # Message configuration for TMEO FRROR or SUCCESS messages
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS AZURE
                                                                           if self.storage.exists(prefixed path):
                                                                      File "C:\Users\Rosilie\AppData\Local\Programs\Python\Python39\lib\site-packages\django\core\files\storage\filesystem.py", line 165, in exists
                                                                          return os.path.lexists(self.path(name))
                                                                      File \ "C:\Users\Rosilie\AppData\Local\Programs\Python\Python39\lib\site-packages\django\contrib\staticfiles\storage.py", line 39, in path and the packages of the packages 
                                                                         raise ImproperlyConfigured(
                                                                   django.core.exceptions.ImproperlyConfigured: You're using the staticfiles app without having set the STATIC_ROOT setting to a filesystem path.
                                                                  Rosilie@DELL MINGW64 ~/OneDrive/Desktop/LEARNING DJANGO PROJECTS/AutomatingCommonTasks (main) $ python manage.py collectstatic
                                                                   126 static files copied to 'C:\Users\Rosilie\OneDrive\Desktop\LEARNING DJANGO PROJECTS\AutomatingCommonTasks\static'.
          > OUTLINE
                                                                 Rosilie@DELL MINGw64 ~/OneDrive/Desktop/LEARNING DJANGO PROJECTS/AutomatingCommonTasks (main)
   Rosilie@DELL MINGW64 ~/OneDrive/Desktop/LEARNING DJANGO PROJECTS/AutomatingCommonTasks (main)
```

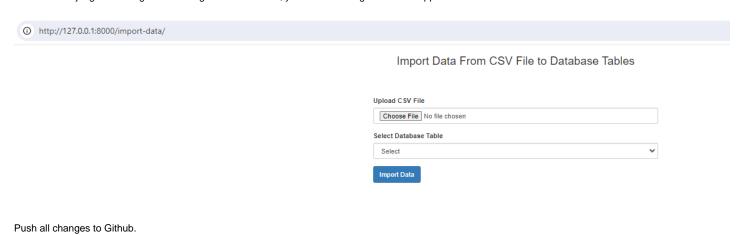
126 static files copied to 'C:\Users\Rosilie\OneDrive\Desktop\LEARNING DJANGO PROJECTS\AutomatingCommonTasks\static'.

15. Include this CUSTOM.JS JAVASCRIPT FILE in our DATAENTRY.HTML

\$ python manage.py collectstatic

```
→ importdata.html M X → alerts.html M JS custom.js U → .env → .gitignore M
                 EXPLORER.
                                                                  ··· 🌵 settings.py M
                                                                                                                                                                                                                                                                                                                                                            tasks.py U tils.py M
            Ф
                                                                                                                                                                                                                                                                                                                                                        > ENV Aa ab * No results ↑ ↓ = X
                                                                                           1 {% load static %}
                                                                                                       <html·lang="en">
                  _init_.py
                  asgi.py
                  settings.py
                                                                                                                       <meta charset="UTF-8">
                  urls.py
e#
                  views.py
                   wsqi.py
                                                                                                                       <title>Automate The Common Boring Stuffs Using Django</title>
                 > dataentry
                                                                                                                      \verb|\climber| com/bootstrap| stylesheet" | href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css"> (link rel="stylesheet" | href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css") | link rel="stylesheet" | href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css" | link rel="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css" | link rel="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css" | link rel="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css" | link rel="https://maxcdn.bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.com/bootstrapcdn.co
                                                                                                                       <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
                                                                                                                        <script src="https://maxcdn_bootstrapcdn_com/bootstrap/3_4.1/js/bootstrap.min.js"></script>
                <script src="{% static 'js/custom.js' %}"></script>
                    o home.html
```

16. Run the Django server again for testing. After 5 seconds, your alert message should disapper.



Copyright © Personal Digital Notebooks | By Rosilie | Date Printed: Oct. 21, 2025, 11:17 p.m.