Django Forms

SoftUni Team

Technical Trainers

Software University

SoftUni

Software University

https://softuni.bg

Table of Contents

Have a Question?

Software University

sli.do

#python-web

Validating Forms in Django

Django Validators (1)

- A validator is a function (or a class) that tries to meet some criteria:
  - If it does - **returns nothing**
  - Otherwise - it raises a **ValidationError**

validators.py

```
from django.core.exceptions import ValidationError

def validate_function(value):
    # if not valid:
        raise ValidationError("Some Error Message")
```

Django Validators (2)

- You can **reuse validation logic** between different types of **fields**
- Also, you can reuse it between
  - A **Model**
  - A **Form**
  - A **ModelForm**

Validating Forms

- **Form validation** happens when the **data is cleaned**
  - Raises **ValidationError** and passes the relevant **information**
  - Returns the **normalized** (cleaned) **data** as a Python object
- Each **form field** has a custom **validation logic**

Form Validators

- You can **pass additional validators** to a Form field

```
class NameForm(forms.Form):
    name = forms.CharField(
        validators=[validator_one, validator_two, ...]
    )
```

- You can use both:
  - Custom validators
  - Built-in Django validators

Built-in Validators (1)

- The Django **validators module** contains a collection of validators for use

```
from django.core import validators

class NameForm(forms.Form):
    name = forms.CharField(
        validators=[
            validators.MinLengthValidator(2),
            validators.MinLengthValidator(50),
        ])
```

Built-in Validators (2)

- EmailValidator
- URLValidator
- MinValueValidator / MaxValueValidator
- MinLengthValidator / MaxLengthValidator
- RegexValidator

More built-in validators: https://docs.djangoproject.com/en/4.1/ref/validators/#built-in-validators

ModelForm Validators

- You can make validation in a ModelForm by:
  - Validating the **Model**
  - Validating the **Form**

Validating the Model

- Pass **additional validators** to the **Model field** constructor

```
class Name(models.Model):
    first_name = models.CharField(
        max_length=20,
        validators=[
            validator_one,
            validator_two,
            ...
        ]
    )
```

Validating the ModelForm

- Override the **`clean()`** method in the ModelForm class

```
def clean(self):
    super(NameForm, self).clean()

    first_name = self.cleaned_data.get('first_name')
    if len(first_name) < 2:
        self._errors['first_name'] = self.error_class([
            'Minimum 2 characters required!'])
    return self.cleaned_data
```

Error Messages in Forms

- You can **override** the default **error messages** on an existing validator
  - Each validator has a **list of error message keys**
- Pass in a **dictionary** with keys and error messages

```
class NameForm(forms.Form):
    name = forms.CharField(
        error_messages={
            'required': 'Please enter your name'
        })
```

Error Messages in Models

- You can directly override the error messages in the model

```
class UserName(models.Model):
    username = models.CharField(
        max_length=50,
        unique=True,
        error_messages={
            "unique": "The name is already taken."
        })
```

Error Messages in ModelForms

- You can override the error messages in the model form

```
class Name(models.ModelForm):
    class Meta:
        ...
        error_messages = {
            'name': {
                'max_length': "The name is too long."
            }
        }
```

Working with Media Files

Live Demo

What are Media Files?

- Media files are **pictures**, **music**, **audios**, **videos**, and **documents**
- Computer programs or applications can read and work with a digital file after it is **encoded** during the **saving** process
- For instance, document formats can be read and edited in word-processing programs like Microsoft Word

Most Common Media Files

- **Photo** file formats: JPEG, GIF, TIFF, BMP
- **Music** file formats: AAC, MP3, WAV, WMA, DOLBY DIGITAL, DTS
- Other available **music** file formats: AIFF, ASF, FLAC, ADPCM, DSD, LPCM, OGG
- **Video** file formats: MPEG-1, MPEG-2, MPEG-4, AVI, MOV, AVCHD, H.264, H.265
- Other available **video** formats: DivX and DivX HD, Xvid HD, MKV, RMVB, WMV9, TS/TP/M2T, WMV

Pillow - Python Imaging Library

- Python Imaging Library (abbreviated as PIL) (in newer versions known as Pillow) is a free library
- It adds support for opening, manipulating, and saving many different image file formats
- It is available for Windows, Mac OS X and Linux
- Some of the file formats supported are PPM, PNG, JPEG, GIF, TIFF, and BMP

Installing Pillow

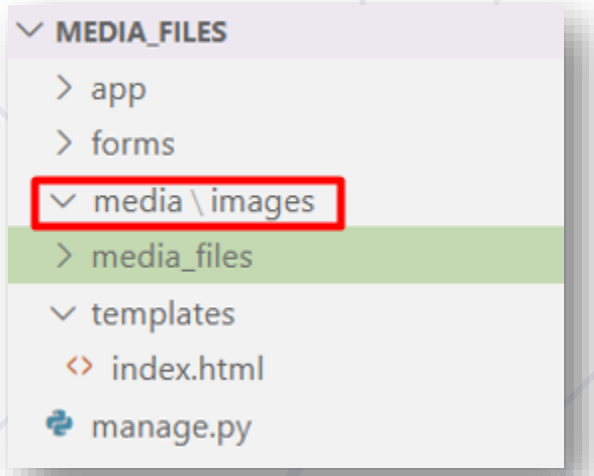
- To install pillow, we can use the python package manager (pip)

```
pip install pillow
```

- Warnings
  - Pillow and PIL **cannot** co-exist in the same environment
  - Before installing Pillow, please **uninstall PIL**

Configure Media Folder

- Create a media folder and configure it in the settings.py file




```
117
118   USE_L10N = True
119
120   USE_TZ = True
121
122
123   # Static files (CSS, JavaScript, Images)
124   # https://docs.djangoproject.com/en/3.0/howto/static-files/
125
126   STATIC_URL = '/static/'
127
128   MEDIA_ROOT = os.path.join(BASE_DIR, 'media/')
129   MEDIA_URL = '/media/'
130
```

Create Image Field in a Model

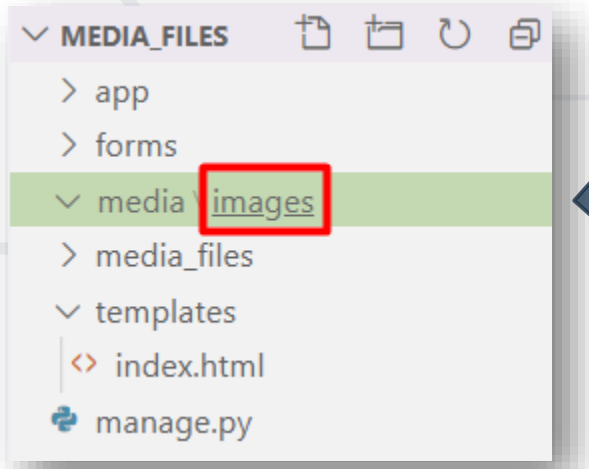
Software University

app > models.py > Image

```python
from django.db import models

# Create your models here.
class Image(models.Model):
    image = models.ImageField(upload_to="images")
```

Name of the folder where the images will be stored



Create a Model Form

Software University

forms > image_form.py > ImageForm > Meta

```python
from django.forms import ModelForm
from app.models import Image

class ImageForm(ModelForm):
    class Meta:
        model = Image
        fields = '__all__'
```


```html
<form enctype="multipart/form-data", method="POST">
    {% csrf_token %}
    {{ form.as_p }}
    <input type="submit">
</form>
```

Handling the POST Request

```
11      elif req.method == "POST"
12          form = ImageForm(req.POST, req.FILES)
13          if form.is_valid():
14              image = form.save()
15              image.save()
16              return render(req, 'index.html', {'images': images})
```

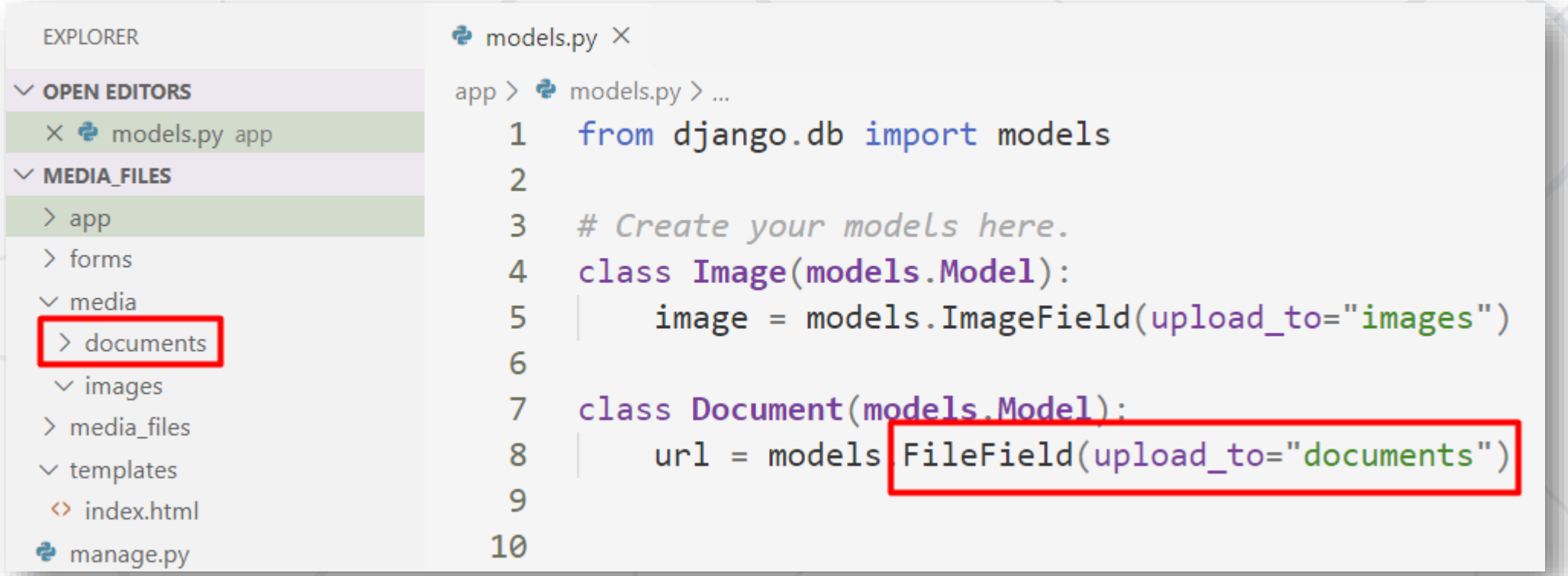
```
3   from django.conf import settings
4   from django.conf.urls.static import static
5
6
7   urlpatterns = [
8       path('', index, name="index")
9   ] + static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

Configure the URLs for media

Displaying the Image

```
1  <div class="card m-3" style="width: 18rem;">
2      <img class="card-img-top" src="{{ image.url }}" alt="Card image cap">
3  </div>
```

Create Documents Folder



Create Model Form

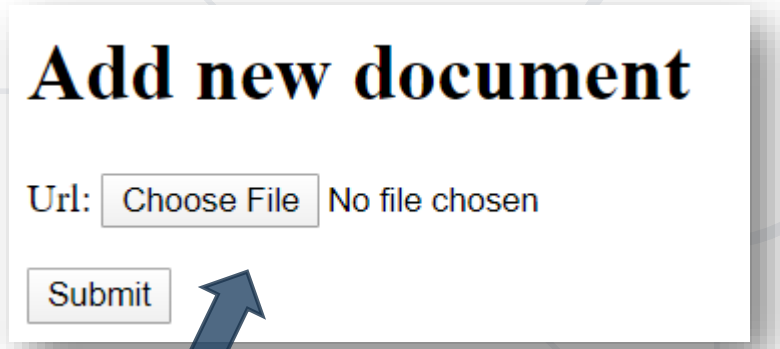
Software University

document_form.py

forms > document_form.py > ...

```python
from django.forms import ModelForm
from app.models import Document

class DocumentForm(ModelForm):
    class Meta:
        model = Document
        fields = '__all__'
```

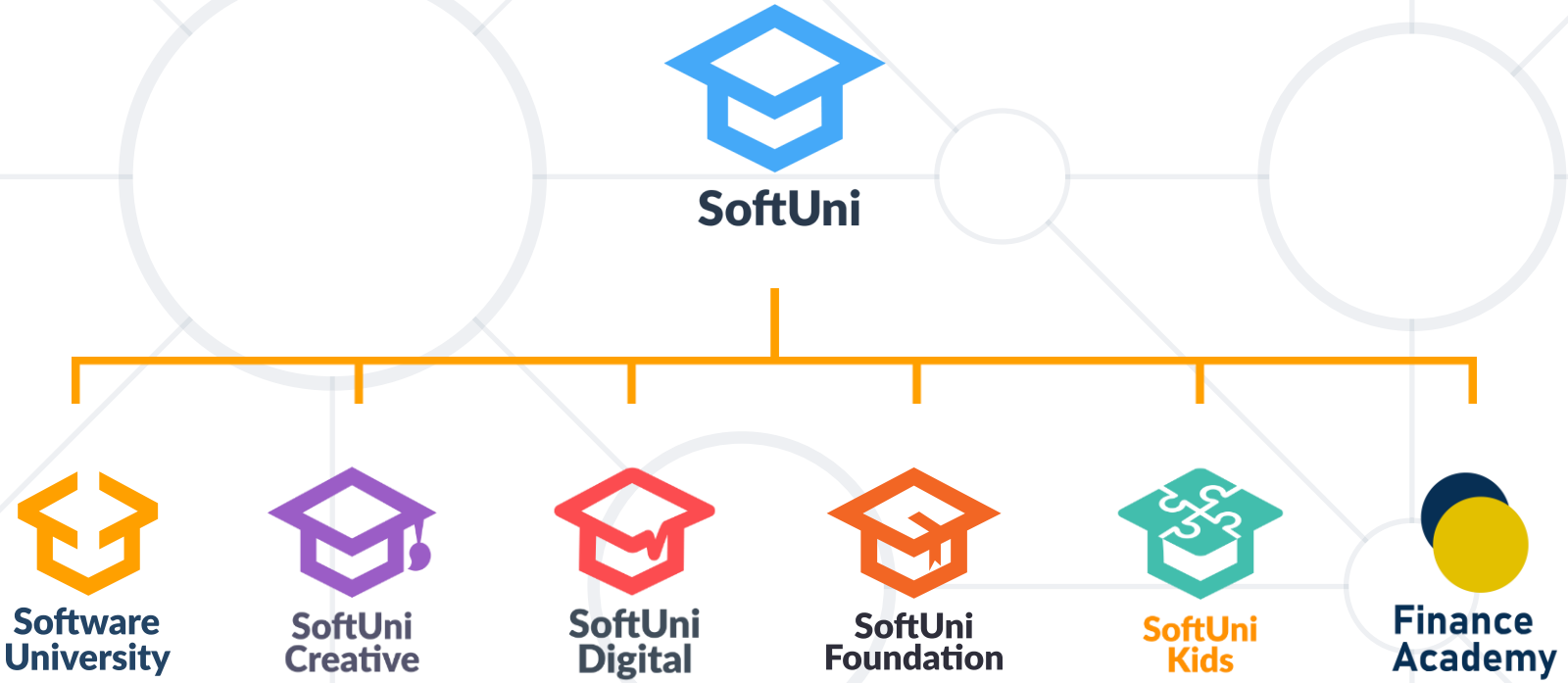



```python
<form enctype="multipart/form-data", method="POST">
    {% csrf_token %}
    {{ form.as_p }}
    <input type="submit">
</form>
```

Summary

- We can let Django handle the performance and use the prebuilt Django tools
- We can override Django methods, classes, or functions
- We can customize the built-in ones

Questions?



SoftUni Diamond Partners

Software University

SUPER HOSTING .BG

Coca-Cola HBC Bulgaria

POKERSTARS
POKER | CASINO | SPORTS
a Flutter International brand

INDEAVR
Serving the high achievers

AMBITIONED

DRAFT KINGS

SOFTWARE GROUP

createX

Postbank
Решения за твоето утре

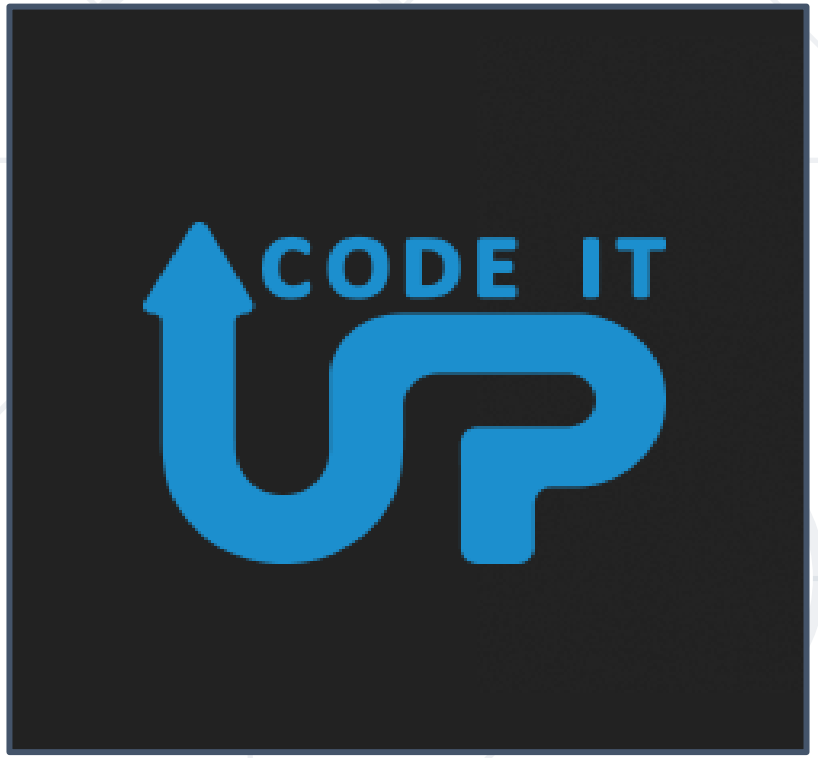
BOSCH

DXC TECHNOLOGY

SmartIT

Educational Partners

Software University



Trainings @ Software University (SoftUni)

- Software University – High-Quality Education, Profession and Job for Software Developers
  - softuni.bg, softuni.org
- Software University Foundation
  - softuni.foundation
- Software University @ Facebook
  - facebook.com/SoftwareUniversity
- Software University Forums
  - forum.softuni.bg

Software University

FORUM

License

- This course (slides, examples, demos, exercises, homework, documents, videos, and other assets) is **copyrighted content**
- Unauthorized copy, reproduction, or use is illegal
- © SoftUni – https://softuni.org
- © Software University – https://softuni.bg