Django Framework

CS253 (Software Development and Operations)

Nikhil Kumar Singh
Department of Computer Science and Engineering
Indian Institute of Technology, Kanpur
Email: nksingh@cse.iitk.ac.in

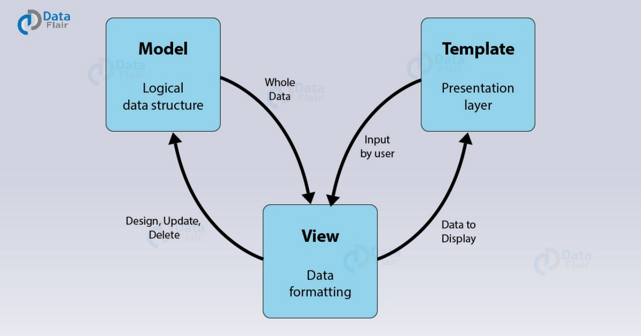
March 25, 2022

Introduction

- create efficient web applications
- To build a website, we need a similar set of components: a way to handle user authentication (signing up, signing in, signing out), a management panel for your website, forms, a way to upload files, etc.
- Huge library and features such as Web Scrapping, Machine Learning, Image Processing, Scientific Computing, etc. One can integrate it all this with web application

Architecture

- Model-View-Template architecture



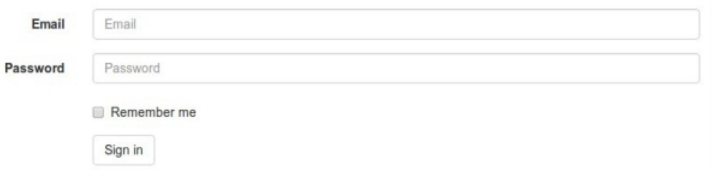
Features

- Versatility - Multiple formats
- Security - Hashed password
- Scalability - No stored state
- Portability - Multiple platforms

Demo

Forms

- What is a form?
- How to create a **Form** class?



Forms

- Defining a field in Django

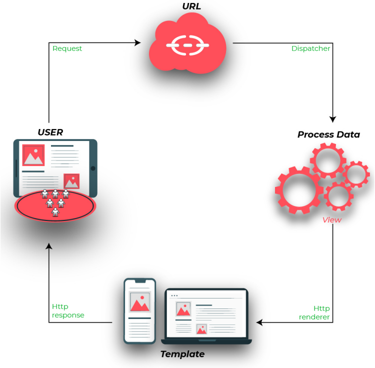
```
field_name = forms.FieldType(**options)
```

Example

```
class InputForm(forms.Form):
    first_name = forms.CharField(max_length = 200)
    last_name = forms.CharField(max_length = 200)
    roll_number = forms.IntegerField()
    password = forms.CharField(widget = forms.PasswordInput
```

Views

- a Python function that takes a Web request and returns a Web response.
- They render the HTML/CSS/Javascript in your Template files

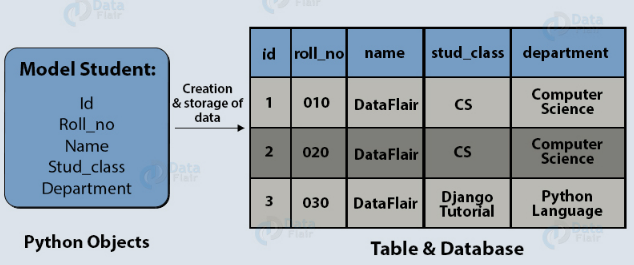


Example

```
def form_view(request):
    context ={}
    context['form']= InputForm()
    return render(request, "home.html", context)
```

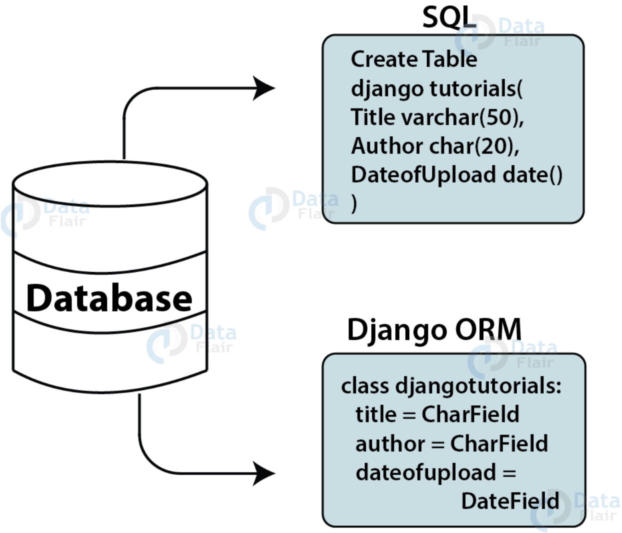
Models

- To create tables
- Each model maps to a single database table.
- ORM (object-relational mapper) transmit data between a relational database and application model.



id	roll_no	name	stud_class	department
1	010	DataFlair	CS	Computer Science
2	020	DataFlair	CS	Computer Science
3	030	DataFlair	Django Tutorial	Python Language

ORM v/s SQL



SQL

```
Create Table
django tutorials(
Title varchar(50),
Author char(20),
DateofUpload date()
)
```

Django ORM

```
class djangotutorials:
  title = CharField
  author = CharField
  dateofupload =
                    DateField
```

Templates

- Written in HTML, CSS, and Javascript in a .html file.
- Dynamically HTML web pages that are visible to the end-user.
- Django template language.
  - Consists of Variables, Tags, Filters, and Comments.
  - Variables output a value from the context.
  - Tags provide arbitrary logic in the rendering process.
  - Filters transform the values of variables and tag arguments

Example

```
TEMPLATES = [
{        'BACKEND': 'django.template.backends.django.DjangoTemplates',
         'DIRS': [<path to template>],
},
]
```

```
<!DOCTYPE html>
<html>
<head>
  <title>{{ title }}</title>
</head>
<body>
  <h1>About {{ title }}</h1>
  <p>This Website was developed by {{ author }}.</p>
  <p>Now using the Django's Template Engine.</p>
  <p><a href="{% url 'homepage' %}">Return to the homepage</a>.</p>
</body>
</html>
```

Thanks for your attention!!

Code: `https://github.com/ciniks117/CS253_TA`

References: `https://geeksforgeeks.org`
`https://docs.python.org`