KAZAKH KBTU BRITISH
TECHNICAL
UNIVERSITY

WEB DEVELOPMENT

Lesson 10

Regular Expressions in URL

Django 1.*	**Django 2.***
url	path
url	re_path

Regular Expressions in URL

. — (dot) Any character

\d — Any digit

[A-Z] — Any character A-Z (uppercase)

[a-z] — Any character, a-z (lowercase)

[A-Za-z] — Any character, a-z (case insensitive)

+ — One or more of the previous expression

[^/]+ — All characters except forward slash

? — Zero or more of the previous expression

{1,3} — Between one and three (inclusive) of the previous expression

https://docs.djangoproject.com/en/2.1/topics/http/urls/

How Django Processes a Request

- `python manage.py runserver`
  - `settings.py`
  - `ROOT_URLCONF`
- `/time/`
  - look through all urls patterns and compare
  - `HTTPRequest` object as first parameter
  - view function responsible to return `HTTPResponse`

404 Errors

Word About Pretty URLs

```
(r'^time/plus/\d+/$', hours_ahead),
```

- /time/plus/1/
- /time/plus/2/
- /time/plus/3/
- /time/plus/1000000/

Word About Pretty URLs

```
(r'^time/plus/\d{1,2}/$', hours_ahead),
```

- /time/plus/1/
- /time/plus/2/
- /time/plus/3/
- /time/plus/1000000/

Passing that data to the view function

parentheses around the data

```
(r'^time/plus/(\d{1,2})/$', hours_ahead),
```

Coding Order

1. views —> urls

2. urls —> views

Django new App

The Django Template System

```
(r'^time/plus/(\d{1,2})/$', hours_ahead),
```

MTV

- `M (Model)` — data access layer
- `T (Template)` — presentation layer
- `V (View)` — business logic layer

Configuring the Database

Configuring the Database

settings.py

- `DATABASE_ENGINE = ''`
- `DATABASE_NAME = ''`
- `DATABASE_USER = ''`
- `DATABASE_PASSWORD = ''`
- `DATABASE_HOST = ''`
- `DATABASE_PORT = ''`

Configuring the Database

`settings.py`

- `DATABASE_ENGINE = ''`

Settings	Database	Required Adapter
`postgresql`	PostgreSQL	`psycopg` version 1.x
`postgresql_psycopg2`	PostgreSQL	`psycopg` version 2.x,
`mysql`	MySQL	`MySQLdb`
`sqlite3`	SQLite	`pysqlite`
`ado_mssql`	Microsoft SQL Server	`adodbapi` version 2.0.1+
`oracle`	Oracle	`cx_Oracle`

Shop —new Django app

Defining Models in Python

Python model  SQL **CREATE TABLE**

Create new migration file if necessary

```
>>> python manage.py makemigrations
```

Execute created migration file

(change database schemas)

```
>>> python manage.py migrate
```

Inserting and Updating Data

- `Create instance of model`
  - `save() // INSERT INTO “table_name” …`
- `Find object and update class fields`
  - `save() // UPDATE “table_name” SET ...`

Selecting Objects

```
>>> products = Product.objects.all()
```

```
SELECT * FROM products;
```

Retrieving Single Objects

```
>>> p = Product.objects.get(id=3)
```

```
SELECT * FROM product WHERE id=3;
```

Questions?