

Django

SPJA – Django

Radek Simkanič

URL – materiály, tutoriály, info

- <http://www.djangoproject.cz/>
- <https://www.djangoproject.com>
- Verze Django:
`python3 -m django --version`

Architektura MVC/MVT

- MVC – Model View Controller
- MVT / MTV – Model View Template (django)
- Model – reprezentace informací
- View – interaktivní reprezentace informací (GUI)
- Controller – reaguje na události (např. klik) a vyvolává načítání či změnu dat v Modelu a změnu View
- Template – (X)HTML šablona

Vytvoření projektu

`django-admin startproject <název projektu>`

- Název projektu nepoužívat: django, test apod.
- `__init__` - python modul/balíček
- Settings – nastavení projektu
- Urls – routování url
- ~~Wsgi – nastavení wsgi serveru~~
- Manage – utilita pro správu projektu

`python3 manage.py runserver`

`python3 manage.py migrate`

Název	Velikost	Typ
prvni	4 položky	Složka
<code>__init__.py</code>	0 bajtů	Text
<code>settings.py</code>	3,1 kB	Text
<code>urls.py</code>	762 bajtů	Text
<code>wsgi.py</code>	388 bajtů	Text
<code>manage.py</code>	803 bajtů	Text

Aplikace

- Projekt může obsahovat více (zásuvných) aplikací. - DRY
- Jedna aplikace může být ve více projektech. - DRY
- DRY (Dont Repeat Yourself)
- Vytvoření aplikace

```
python3 manage.py startapp <název aplikace>
```
- Aplikace obsahuje model i view
- Aplikace je potřeba nainstalovat v `INSTALLED_APPS` (`settings.py`)

View Hello World

- `HttpResponse()` - vrátí text který se mu dá – v reálu se moc nepoužívá

URL

- Lze definovat lokálně v konkrétní aplikaci
- Lze definovat globálně v projektu
- V aplikaci je potřeba vytvořit v souboru:
`urlpatterns = []`
- Propojení URL z aplikace do projektu:
`include() #django.conf.urls.include`
- Je potřeba znát regulární výrazy (regex)
https://en.wikipedia.org/wiki/Regular_expression

Model

- Model je reprezentace typů dat a propojuje DB
- Vytváří se v souboru model.py
- Každý model (třída) dědí z model.Model
- CharField() – Textové pole
- DateTimeField() – Datum a čas
- IntegerField() – Číselné pole
- ForeignKey() – Propojení pomocí cizího klíče
- Provede-li se migrace – vytvoří/upraví se DB tabulky

Migrace

- Vytvoření DB tabulek (vytvoření a spuštění migrace) – až po vytvoření/úpravě modelů
- Příprava migrace:
`python3 manage.py makemigrations`
- Výpis migrace v SQL:
`python3 manage.py sqlmigrate <název aplikace> <id migrace>`
- Provedení migrace:
`python3 manage.py migrate`
- Migrace se nacházejí v podadresáři **migrations**

Shell

- Spuštění upraveného Ipython
`python3 manage.py shell`
- S čím chceme pracovat tak je potřeba nainportovat
- Při práci s shellem s modely, je vhodné mít implementovanou metodu `__str__` pro lepší přehlednost
- U modelů je možné využívat například:
 - `<model>.objects.all()`
 - `<model instance>.save()`
 - `<model instance>.delete()`
 - `<model instance>.id`
 - `<model>.objects.filter()`
 - `<model>.objects.get()`
 - `<model instance>.<název sloupce cizího klíče>.all()`
 - `<model instance>.<název sloupce cizího klíče>.create()`
 - `<model instance>.<název sloupce cizího klíče>.count()`

Superuser

- Vytvoření admina:
`python3 manage.py createsuperuser`
- Registrace modelu aplikace do administrace (admin.py):
`admin.site.register(<Třída modelu>)`