

Evaluation - Web Framework

Projekt: DigitalSchoolNotes

Projekt Team: Adler, Brinnich, Hohenwarter, Karic, Stedronsky

Version 2.0

12.10.2015

Status: [RELEASE]

	Datum	Name	Unterschrift
Erstellt	29.09.2015	Selina Brinnich	
Geprüft	12.10.2015	Adin Karic	
Freigegeben			
Git-Pfad: /doc/evaluation Dokument: evaluation_webframework.doc			

1 Changelog

Version	Datum	Status	Bearbeiter	Kommentar
0.1	2015-09-29	Erstellt	Selina Brinnich	Dokument erstellt
1.0	2015-10-07	Geprüft	Niklas Hohenwarter	QA
1.1	2015-10-12	Bearbeitet	Thomas Stedronsky	Rechtschreibfehler
2.0	2015-10-12	Geprüft	Adin Karic	QA

2 Vergleich

	Flask [1]	Django [2]	Play [3]
Dokumentation	9/10 Funktionalität des Frameworks und einiger Erweiterungen gut dokumentiert. [4]	10/10 Sehr ausführliche Dokumentation für alle Funktionalitäten des Frameworks vorhanden. [5]	6/10 Offizielle Dokumentation vorhanden, allerdings nicht sehr ausführlich. [6]
Community	2/10 Sehr wenig Community. (11.772 Fragen in Stack-Overflow [7])	8/10 Ziemlich gute Community, viele Fragen und Antworten vorhanden. (113.909 Fragen in Stack-Overflow [7])	10/10 Sehr gute Community. Antworten auf diverse Probleme können schnell gefunden werden. (196.580 Fragen in Stack-Overflow [7])
Built-In Funktionalität	6/10 Kleines Framework, eher weniger Built-In Funktionalität.	9/10 Sehr viele Built-In Funktionalitäten im Framework.	8/10 Relativ viel Built-In Funktionalität.
Erweiterbarkeit	9/10 Einfach erweiterbar, viele Erweiterungen in offizieller Flask-Dokumentation vorhanden. Bei Bedarf kann eine eigene Erweiterung entwickelt und eingebunden werden.	9/10 Einfach erweiterbar, viele Erweiterungen in offizieller Django-Dokumentation vorhanden. Bei Bedarf kann eine eigene Erweiterung entwickelt und eingebunden werden.	6/10 Einbindung von zusätzlichen Modulen möglich aber etwas kompliziert. Einige Module von der Play Community zur Verfügung gestellt, eigene Module können bei Bedarf entwickelt und eingebunden werden.
Lizenzierung	10/10 Frei verfügbar. (BSD-License)	10/10 Frei verfügbar. (BSD-License)	10/10 Frei verfügbar. (Apache-License)

Installation/Konfiguration	10/10 Sehr schnell und einfach installiert und konfiguriert. Kein großer Aufwand.	10/10 Sehr schnell und einfach installiert und konfiguriert. Kein großer Aufwand.	6/10 Installation deutlich aufwendiger und dauert sehr lange.
Prototyp	10/10 "Hello World"-Beispiel in offizieller Dokumentation gut beschrieben und sehr einfach umzusetzen.	10/10 "Hello World"-Beispiel sehr einfach, schnell und ohne Probleme umzusetzen.	7/10 "Hello World"-Beispiel etwas aufwendiger zu erstellen, als bei den anderen beiden Produkten. Zeitaufwand deutlich höher.
Summe	56/70	66/70	53/70

3 Flask

Installiert wird Flask über pip [1]:

```
pip3 install Flask
```

Für ein einfaches "Hello World"-Programm für Flask wird folgender Code in ein File "hello.py" geschrieben [1]:

```
from flask import Flask

app = Flask(__name__)

@app.route("/")
def hello():
    return "Hello Flask World!"

if __name__ == "__main__":
    app.run()
```

Anschließend wird folgender Befehl ausgeführt um das Programm zu starten:

```
python3 hello.py
```

Öffnet man nun einen Browser und zeigt die Seite an sieht man folgenden Output:

Hello Flask World!

4 Django

Django wird folgendermaßen installiert [8]:

```
apt-get install python3-django
```

Für ein einfaches “Hello World”-Programm für Django muss zunächst mit folgendem Befehl ein neues Projekt erstellt werden [9]:

```
django-admin startproject app
```

Mit folgendem Befehl wird nun die Datenbank entsprechend der Project-Settings konfiguriert:

```
python manage.py migrate
```

Anschließend wird im Unterordner app ein neues File “hello.py” mit folgendem Code erstellt [10]:

```
from django.http import HttpResponse

def hello_world(request):
    return HttpResponse("Hello Django World!")
```

Im File “url.py” wird folgender Code hinzugefügt:

```
url(r'^$', 'app.hello.hello_world'),
```

Anschließend wird folgender Befehl ausgeführt um das Programm zu starten [9]:

```
python manage.py runserver
```

Öffnet man nun einen Browser und ruft die Seite auf sieht man folgenden Output:

Hello Django World!

5 Play

Um Play zu installieren muss das Framework zunächst von folgender Seite heruntergeladen werden:

<https://www.playframework.com/download> [11]

Anschließend muss das zip-File entpackt werden der enthaltene Activator zum PATH hinzugefügt werden. Um den Activator zu starten wird folgender Befehl ausgeführt [12]:

```
activator ui
```

Für ein einfaches “Hello World”-Programm muss zunächst ein neues Projekt (mit “play-java” als vordefiniertes Template) erstellt werden:

```
activator new hello play-java
```

Anschließend muss im File app/controllers/Application.java folgende Methode hinzugefügt werden:

```
public static Result hello() {
    return ok(main.render("Hello World", new
        play.twirl.api.Html("Hello Play World!")));
}
```

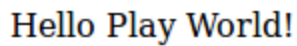
Im File conf/routes wird die Route für “/” auf folgendes umgeändert:

```
GET / controllers.Application.hello()
```

Um das neu erstellte Projekt zu starten muss nun im Projekt-Verzeichnis folgender Befehl ausgeführt werden:

```
activator run
```

Öffnet man nun einen Browser und zeigt die Seite an sieht man folgenden Output:



Hello Play World!

6 Endergebnis

Aufgrund der Ergebnisse der Evaluierungstabelle sowie den einzelnen Prototypen der drei verglichenen Produkte entscheidet das Team sich dazu das **Web-Framework Django** zu verwenden. Django bietet laut Tabelle eine gute Dokumentation und Community, was für das Team sehr wichtig ist, da die Kenntnisse für die Programmierung mit diesem Framework noch nicht sehr ausgeprägt sind. Zusätzlich hat der erstellte Prototyp einen sehr guten Eindruck hinterlassen, da die Installation, Konfiguration, sowie Programmierung des ersten "Hello World"-Programmes sehr schnell, einfach und ohne Probleme abgelaufen sind.

7 Quellen

- [1], "Flask", Armin Ronacher,
<http://flask.pocoo.org/>, zuletzt besucht: 30.09.2015
- [2], "Django", Django Software Foundation,
<https://www.djangoproject.com/>, zuletzt besucht: 30.09.2015
- [3], "Play Framework", Play,
<https://www.playframework.com/>, zuletzt besucht: 30.09.2015
- [4], "Flask Documentation", Armin Ronacher,
<http://flask.pocoo.org/docs/0.10/>, zuletzt besucht: 30.09.2015
- [5], "Django documentation contents", Django Software Foundation,
<https://docs.djangoproject.com/en/1.8/contents/>, zuletzt besucht: 30.09.2015
- [6], "Play 2.4.x documentation", Play,
<https://www.playframework.com/documentation/2.4.x/Home>, zuletzt besucht: 30.09.2015
- [7], "Stack Overflow", Stack Exchange Inc,
<http://stackoverflow.com/>, zuletzt besucht: 30.09.2015
- [8], "Third-party distributions of Django", Django Software Foundation,
<https://code.djangoproject.com/wiki/Distributions>, zuletzt besucht: 30.09.2015
- [9], "Writing your first Django app, part 1", Django Software Foundation,
<https://docs.djangoproject.com/en/1.8/intro/tutorial01>, zuletzt besucht: 30.09.2015

[10], "Hello World in Django [Python]", sundip,
<http://code.runnable.com/UWRVp6lLuONCAABD/hello-world-in-django-for-python>, zuletzt besucht:
30.09.2015

[11], "Downloads", Play,
<https://www.playframework.com/download>, zuletzt besucht: 30.09.2015

[12], "Installing Play", Play,
<https://www.playframework.com/documentation/2.4.x/Installing#Quick-Start>, zuletzt besucht:
30.09.2015