



BA-Praktikum WS2019/2020

February 18, 2020



Universität Hamburg

DER FORSCHUNG | DER LEHRE | DER BILDUNG

Eugen Ruppert

BA-PRAKTIKUM WS2019/2020

WEB SERVER

JSON

Spring und Spring Boot

Flask

Software Engineering

Beispielprojekt

Lösungsansätze

Weiterer Ablauf

Qualifizierte Fragen?

JSON

JSON – JavaScript Object Notation

Übersicht



Universität Hamburg

DER FORSCHUNG | DER LEHRE | DER BILDUNG

- einfaches, leicht erweiterbares Format
- maschinenlesbar, Bibliotheken existieren für sehr viele Programmiersprachen
- unterstützt: Key-Value-Paare, Listen und Objekte – beliebige Verschachtelung
- de-facto Standard zur Übertragung struktureller Daten im Internet

JSON – JavaScript Object Notation

Beispiel



Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

```
{  
  "name1": "value1",  
  "name2": "value2",  
  "list1": [  
    "a",  
    "b",  
    "c"  
  ],  
  "object1": {  
    "key": "value"  
  }  
}
```

Spring und Spring Boot

Spring und Spring Boot

Spring



Universität Hamburg

DER FORSCHUNG | DER LEHRE | DER BILDUNG

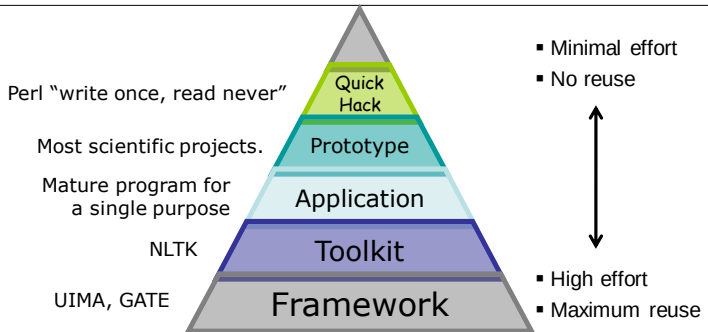
- ausgereiftes open-source Framework
<https://spring.io/>
- Inversion of Control: "man sagt, was man braucht, das Framework kümmert sich darum"
- Objekte und Typen werden zur Laufzeit entschieden oder konfiguriert
- Konfiguration über Annotationen oder XML
- verwandte Projekte für Daten oder Cloud-Zugriffe
- <https://spring.io/>



Spring und Spring Boot

Wozu Frameworks?

Productivity Pyramid



"With great power comes ...

... less responsibility, but high complexity."

after Leidner, 2003

Spring und Spring Boot

Spring Boot



Universität Hamburg

DER FORSCHUNG | DER LEHRE | DER BILDUNG

- rapid-prototyping Framework
<https://spring.io/projects/spring-boot>
- sehr schnelle Umsetzung von Web-Apps
- low effort, maximum reuse
- keine XML-Konfigurationen (meistens automatisch)
- Standalone Spring Apps
(Tomcat-Server ist automatisch dabei)
- Templating über Thymeleaf
<https://www.thymeleaf.org/>

Flask

Flask

Python Web Server



Universität Hamburg

DER FORSCHUNG | DER LEHRE | DER BILDUNG

- light-weight Webserver
<https://palletsprojects.com/p/flask/>
- schneller Start
- Templating wird unterstützt
https://www.tutorialspoint.com/flask/flask_templates.htm
- JSON-Output über jsonify()-Funktion:
<https://flask.palletsprojects.com/en/1.1.x/api/#flask.json jsonify>

Flask

Beispiel



```
from flask import Flask, escape, request

app = Flask(__name__)

@app.route('/')
def hello():
    name = request.args.get("name", "World")
    return f'Hello, {escape(name)}!'
```

Examples

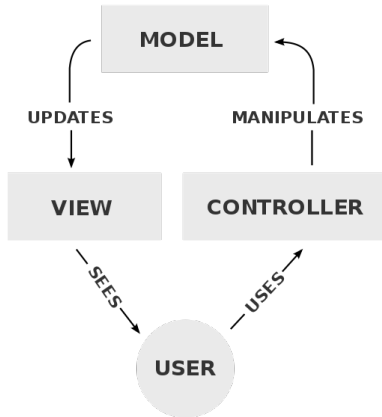
```
$ export FLASK_APP=hello.py  
$ flask run  
* Running on http://127.0.0.1:5000/
```

Software Engineering

- Unterteilung des Programms in 3 unabhängige Teile:
 - Model** Das Datenmodell; alle Datenoperationen werden vom Modell durchgeführt
 - View** Representation des Modells und das User Interface
 - Controller** Verarbeitung von Inputs, Schnittstelle zwischen Model und View
- Spring Boot ist ein MVC-Framework
- MVC ermöglicht wiederverwendbaren Code ohne große Abhängigkeiten

Software Engineering

MVC



<https://en.wikipedia.org/wiki/Model-view-controller>

- IntelliJ IDEA
<https://www.jetbrains.com/idea/>
- moderne Entwicklungsumgebung
- Linux, Mac und Windows
- unterstützt Git, GitHub und GitLab
- Projektmanagement mit Maven
<https://maven.apache.org/>
- IDEs für Python und andere Programmiersprachen vorhanden

- Projektmanagement
<https://maven.apache.org/>
- Konzept: Project Object Model (POM)
- definiert Abhängigkeiten sehr genau
- definiert den Build-Prozess
- schneller Einstieg über POM-File
- Maven Central als zentrale Bibliothek von Java Libraries
<https://search.maven.org/>

Software Engineering

Beispiel POM



```
<project>
  <groupId>basecamp</groupId>
  <artifactId>ba19-service-backend</artifactId>
  <version>0.0.1-SNAPSHOT</version>

  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>1.3.3.RELEASE</version>
  </parent>

  <dependencies>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter</artifactId>
    </dependency>
  </dependencies>

  <build>
    <plugins>
      <plugin>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-maven-plugin</artifactId>
      </plugin>
    </plugins>
  </build>
</project>
```

Beispielprojekt

- ein Backend Server mit Thymeleaf
- Projekt befinden sich im base.camp-Repository:
`https://git.informatik.uni-hamburg.de/base.camp`
- nächste Schritte
 - Login prüfen
 - Projekt "forken" (gerne auch 1x pro Gruppe; Mitglieder können zum Projekt hinzugefügt werden)
 - Projekt auschecken und bearbeiten

Beispielprojekt

REST Server



Universität Hamburg

DER FORSCHUNG | DER LEHRE | DER BILDUNG

- IntelliJ IDEA öffnen
- VCS -> Checkout from Version Control -> Git
- GitLab-Link einfügen (vom eigenen Projekt)
- Account-Daten eingeben und speichern
- Clone -> Open -> This Window
- Server kann gestartet werden, Main Klasse:
`src/main/java/basecamp/project/server/ServerApplication.java`
- Server läuft auf `http://localhost:8080/`
- Dokumentation:
`http://projects.spring.io/spring-boot/`



Lösungsansätze

- Daten sind immer verrauscht!
- auf korrekte Kodierung achten: UTF-8
- Daten im System anschauen (Umlaute, Satztrennung, ...)
- auf gleiche Verarbeitung aller Daten achten (z.B. Input/Output)
- prüfen, ob Deutsch sauber unterstützt wird



Lösungsansätze

Best Practices



Universität Hamburg

DER FORSCHUNG | DER LEHRE | DER BILDUNG

- keine Daten in Git veröffentlichen!
- keine Service-Login Daten veröffentlichen!
besser: Logins aus Dateien auslesen
- bei den APIs auf Preis und Leistung achten
- nice-to-have: korrekte Lizenzen ausweisen

Lösungsansätze

Weitere Ressourcen



Universität Hamburg

DER FORSCHUNG | DER LEHRE | DER BILDUNG

- JSON Lint

<http://jshint.com/>

- JSON Validator/Formatter

<https://jsonformatter.curiousconcept.com/>

Weiterer Ablauf

Weiterer Ablauf

Scrum



Universität Hamburg

DER FORSCHUNG | DER LEHRE | DER BILDUNG

- agile Softwareentwicklung
<https://de.wikipedia.org/wiki/Scrum>
- schnelle Iterationen
- Kundenwünsche entstehen durch die Interaktion
- wöchentliche kurze Meetings:
 - was wurde gemacht?
 - was mache ich heute?
 - gibt es aktuell Blocker?
- ER als Scrum-Master, immer auf dem aktuellen Stand

Weiterer Ablauf

Bewertung

- 50% Qualität der Programme (Funktionalität, Benutzbarkeit, Code, Dokumentation)
- 25% Teilnahme an Meetings und Präsentationen
- 25% Abschlussbericht (Teilnehmer kennzeichnen, welcher Teil jeweils von ihnen gemacht wurde)

Weiterer Ablauf

Hilfe

- per Mail so gut wie immer erreichbar
- Mattermost, @eugen
- qualifizierte Fragen:
<https://blog.codinghorror.com/rubber-duck-problem-solving/>

Qualifizierte Fragen?