

# *Projektantrag Mash-Up*

## *Modul 151*

14.12.2018  
BBW (5IA16b)  
Romina Ferrario



## Inhalt

Aufgabe Mash-Up.....	2
Bemerkungen .....	2
1 Mashup Aufgabe (bewertet).....	2
1.1 Einleitung und Aufgabe.....	2
1.2 Bewertungselemente der Lösung (Pauschal) .....	3
1.3 Muster Ideen .....	3
Datenquellen und Hinweise.....	4
2.1 Liste von APIs.....	4
2.2 Beispiele von Internetseiten und interessanten Inhalten (HTML) .....	4
Projektidee.....	5
API 1 - Transport API.....	5
API 2 – The Internet Chuck Norris Database API .....	5
Datenbank .....	5
Account.....	5
Stundenplan.....	5
Risikoanalyse.....	6
Projektziele.....	6
Projektorgane und ihre Aufgaben und Kompetenzen .....	6
Planung und Organisation.....	7
Meilensteine .....	7
Projektstrukturplan: Handlungsschritte .....	7
Mittelbedarf.....	7





## Aufgabe Mash-Up

### Bemerkungen

Mashup (von englisch to mash für vermischen) bezeichnet die Erstellung neuer Medieninhalte durch die nahtlose (Re-)Kombination bereits bestehender Inhalte. Der Begriff stammt aus der Welt der Musik und bedeutet dort im Englischen so viel wie Remix (vgl. Mashup Musik). In den deutschen Sprachraum wurde der Begriff rund um das Schlagwort Web 2.0 importiert, da Mashups als ein wesentliches Beispiel für das Neue an Web 2.0 angeführt werden: Inhalte des Webs, wie Text, Daten, Bilder, Töne oder Videos, werden z. B. collagenartig neu kombiniert. Dabei nutzen die Mashups die offenen Programmierschnittstellen (APIs), die andere Webanwendungen zur Verfügung stellen.

Quelle: Wikipedia September 2016

### 1 Mashup Aufgabe (bewertet)

#### 1.1 Einleitung und Aufgabe

Das Ziel ist, eine Mashup-Anwendung mit Java zu erstellen, welche die Inhalte von anderen Webseiten und anderen Datenquellen abrufen und zu einem neuen Content zusammenstellt. Dabei sollen Daten aus HTML Webseiten, Daten von JSON, CSV und/oder XML Datenquellen geladen werden. Wahlweise dürfen Sie eine Web-Applikation mit JSF und Java erstellen oder Desktop-Applikation mit JavaFX. Diverse Beispiele von Datenquellen finden sich weiter hinten in diesem Dokument. Überlegen Sie sich eine kleine Mashing-Anwendungs-Idee und fassen Sie Ihre Idee in einer kurzen Beschreibung zusammen. Zeigen Sie bevor Sie beginnen die schriftliche Beschreibung Ihrer der Lehrperson, die das Projekt bewilligen soll.

<b>Rahmenbedingungen:</b>	
<b>Zeit:</b>	Start 16.11.2018, Abgabe 14.12.2018, ca. 10 Lektionen Zwischenabgabe am 30.11.2018
<b>Sozialform:</b>	Einzelarbeit
<b>Ort:</b>	Im Unterricht Modul 151

Die Applikation ist im GitHub oder GitLab zu verwalten. Der Lehrperson muss Zugriff auf das Repository gewährt werden.

#### Kompetenzen:

Die Inhalte der vorliegenden Aufgabe richten sich nach den Kompetenzen aus den folgenden Handlungszielen. Informieren Sie sich über die zu erreichenden Kompetenzen.

Handlungsziel 1 - Anforderungen an eine Web-Applikation und Gegebenheiten der einzubindenden Datenbank bzw. des Datenbestandes analysieren, Anbindungstechnik definieren und dokumentieren.

Handlungsziel 3 - Anbindung der Web-Applikation an die Datenbank bzw. an den Datenbestand realisieren und dabei Transaktionen, Datensicherheit und Datenschutz beachten.





## 1.2 Bewertungselemente der Lösung (Pauschal)

Hinweise, Bewertungspunkte Kompetenzraster HZ1 und HZ3	Punkte Lehrperson	Punkte Selbstbewertung
Architektur der Applikation (JSF, Beans, Java, Schichten)	5	
Datenbestände aus externen Quellen (Anzahl)	5	
Datenbestände aus externen Quellen (Kombination, Komplexität)	5	
Mashing Idee, Originalität und Umsetzung, sowie Verknüpfung der eingebundenen externen Daten	5	
Lokale Speicherung von Daten (MySQL, JDBC, JPA)	5	
Subjektiver Eindruck der Lehrperson	5	
<b>Total</b>	<b>30</b>	

## 1.3 Muster Ideen

### 1.3.1 Mein Schulweg

In dieser Anwendung werden Beispielsweise Zuginformationen wie: wann fährt der nächste Zug nach Winterthur (oder Zurück), welches sind meine nächsten Verbindungen, etc. angezeigt. Diese Informationen sollen aktuell sein und sich nach dem aktuellen Datum und Uhrzeit richten. Die Anwendung könnte beispielsweise auch das Wetter am Start und Zielort anzeigen oder Informationen Ihres Lieblingsrestaurants anzeigen, wie beispielsweise das Tages Sandwich bei [www.subway-sandwiches.ch](http://www.subway-sandwiches.ch).

### 1.3.2 Tages und/oder Wochen News

Aktuelle Tagesnews wie beispielsweise die letzten News bei 20min und/oder die Tages Aktionen bei digitec und/oder Tages Sandwich bei Subway und/oder die Filme die am Abend im TV laufen, werden in dieser Anwendung zusammengestellt. Eine History kann gespeichert werden. Die App könnte auch eine Vorschau einiger Inhalte der nächsten Tage anzeigen.

### 1.3.3 Serien Forecast

Diese Anwendung sucht nach den Austrahlungsterminen einer bestimmten Serie und zeigt diese an. In einer Art Suchauftrag kann nach einer bestimmten Serie gesucht werden. Die Anwendung sucht dann auf TV-Seiten, Serien-Seiten, etc. nach den gewünschten Informationen. Bereits gesehene Serien können gespeichert werden. Die Anwendung könnte Hintergrundinformationen zu Serien bereitstellen.





## Datenquellen und Hinweise

### 2.1 Liste von APIs

opendata.swiss ist das Portal für Schweizer Open Government Data (OGD). Hier können Sie Schweizer Behördendaten kostenlos herunterladen.

<https://opendata.swiss/de>

The Transport API allows interested developers to build their own applications using public timetable data, whether they're on the web, the desktop or mobile devices. The aim of this unofficial API is to cover public transport within Switzerland. The API uses a web service provided by search.ch.

<https://transport.opendata.ch>

#### Google Maps API, Google Geo Location API

<https://developers.google.com/maps/?hl=de>

<https://developers.google.com/maps/documentation/geolocation/intro?hl=de>

The Netflix API lets you access data for 100,000 movie and TV episode titles on DVD as well as Netflix account access on a user's behalf.

<https://www.programmableweb.com/api/netflix>

Facebook API, Instagram API, Open Weather Map

Diverse RSS Newsfeeds: 20min, NZZ, Tagesanzeiger, Radio-Top, Blick, etc.

CitySDK is a set of API for accessing Open Data, provided by cities' local governments, in the domains of Tourism, Participation and Mobility. This website provides information on the Tourism API and on how to make use of it, whether you are part of the city government or a developer looking to develop a tourism application.

<http://tourism.citysdk.eu>

### 2.2 Beispiele von Internetseiten und interessanten Inhalten (HTML)

20min bietet ein Display für News

<http://www.20min.ch/share>

Datenbank mit Filmen

<https://www.themoviedb.org/documentation/api?language=ch>

Aktionen auf digitec.ch

<http://www.digitec.ch/>

Fernsehprogramm auf sfdrs.ch

<http://tvprogramm.sf.tv>

Informationen zu Serien auf [www.serienjunkies.de](http://www.serienjunkies.de)

<http://www.serienjunkies.de/serien/sender/netflix.html> → Programm

Die aktuellsten Informatik Jobs (java)

[www.stellen.ch](http://www.stellen.ch), [www.jobs.ch](http://www.jobs.ch) → Informatik, java, etc





## Projektidee

Das Projekt sollte bei der Planung des Tages helfen. Man sollte auf einen Blick den Stundenplan vor sich sehen, sowie die nächste Verbindung von Zuhause in die Schule. Um den Tag der Person zu verbessern wird nach der Verbindung noch einen Witz angezeigt.

## API 1 - Transport API

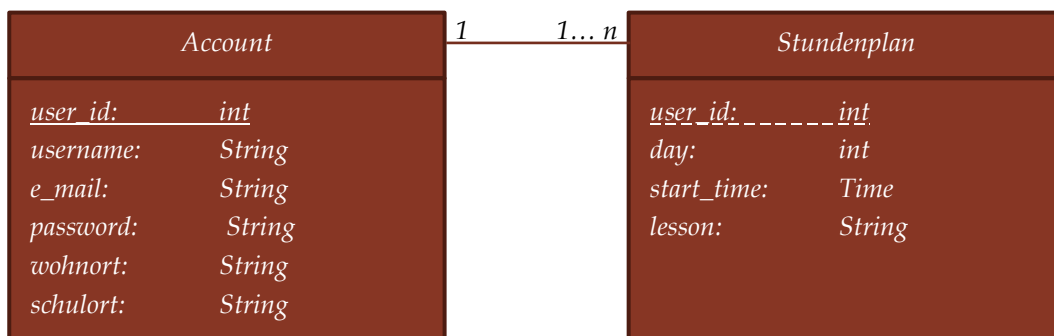
Um die aktuellen Verbindungen herauszusuchen, habe ich mich dafür entschieden die Transport API aufzurufen. Hierbei kann ich beim Aufruf den Wohnort und den Ort der Schule (→ Informationen aus der Datenbank) mitgeben um die Verbindungen vom Wohnort zur Schule zu finden.

## API 2 - The Internet Chuck Norris Database API

Um den Tag der Person etwas besser zu machen habe ich mich entschieden, dass ich noch einen Witz ausgabe. Damit auch nicht immer wieder der selbe Witz angezeigt wird, entschied ich mich dafür, die «The Internet Chuck Norris Database»-API einzubinden

## Datenbank

Um für jede Person den richtigen Stundenplan und die richtige Verbindung ausgeben, habe ich mich für eine lokale Speicherung der Daten der Person und des Stundenplans. Dafür verwende ich XAMPP (phpMyAdmin Datenbank → MySQL). Ich brauche dafür zwei Tabellen (Account und Stundenplan).



## Account

Beispieldaten:

user_id	username	e_mail	password	wohnmort	schulort
1	romina5277	<a href="mailto:saemyferrario@outlook.com">saemyferrario@outlook.com</a>	romina5277	Bülach	Winterthur
2	blub	<a href="mailto:blub@blub.blub">blub@blub.blub</a>	Blub	Zürich	Winterthur
3	username	e_mail	password	Lausanne	Zürich

## Stundenplan

Beispieldaten:

user_id	day	start_time	lesson
1	4	08:25:00	Mathematik
1	4	09:15:00	Mathematik
1	4	10:20:00	Wirtschaft
1	4	11:10:00	Wirtschaft





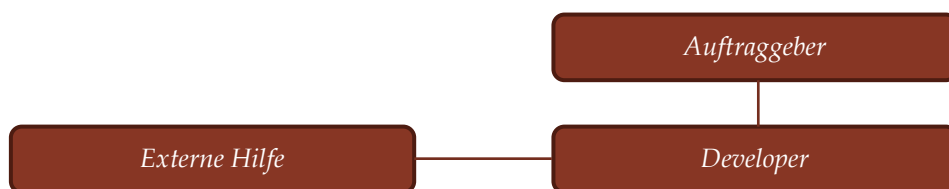
## Risikoanalyse

<b>Finanzielle Risiken</b>	<ul style="list-style-type: none"> <li>bei diesem Projekt gibt es keine finanziellen Risiken, da keine Einkäufe getätigt werden müssen</li> </ul>								
<b>Personelle Risiken</b>	<ul style="list-style-type: none"> <li>nichtausreichendes Knowhow</li> </ul>								
<b>Konsequenzen bei verspäteter oder Nicht-Realisierung des Projektes</b>	<ul style="list-style-type: none"> <li>schlechte Schulnote</li> </ul>								
<b>SWOT-Analyse</b>	<table border="1"> <tr> <td rowspan="2">Gegenwart</td><td> <b>Stärken</b>                      - Frontend                 </td><td> <b>Chancen</b>                      - Knowhow-Aufbau                 </td><td rowspan="2">Zukunft</td></tr> <tr> <td> <b>Schwächen</b>                      - Backend &lt;--&gt; Frontend Verbindung                 </td><td> <b>Gefahren</b>                      - Verzweiflung                      - Stress                 </td></tr> </table>	Gegenwart	<b>Stärken</b> - Frontend	<b>Chancen</b> - Knowhow-Aufbau	Zukunft	<b>Schwächen</b> - Backend <--> Frontend Verbindung	<b>Gefahren</b> - Verzweiflung - Stress		
Gegenwart	<b>Stärken</b> - Frontend		<b>Chancen</b> - Knowhow-Aufbau	Zukunft					
	<b>Schwächen</b> - Backend <--> Frontend Verbindung	<b>Gefahren</b> - Verzweiflung - Stress							

## Projektziele

- Ich möchte in meinem Projekt 2 API's einbinden (Transport-API und «The Internet Chuck Norris Database»-API)
- Ich möchte eine Datenbankverbindung haben, um Informationen über die Person und den Stundenplan abzuspeichern

## Projektorgane und ihre Aufgaben und Kompetenzen



- Auftraggeber** - Herr R. Lanza
- Auftragserteilung
  - Unterstützung bei Problemen
  - Benotung
- Developer** - Romina Ferrario
- Ideenfindung
  - Entwicklung
  - Testing
- Externe Hilfe** - Mitschüler / (Ehemalige) Oberstiften
- Unterstützung bei Problemen
  - Tipps





## Planung und Organisation

### Meilensteine

MS	Ziel / Aufgabe	Endtermin
1	Erste API einbinden	23.11.2018
2	Zweite API einbinden	27.11.2018
3	Datenbankeinbindung	30.11.2018
4	Frontend	01.12.2018
5	Backend <--> Frontend Verbindung	07.12.2018

### Projektstrukturplan: Handlungsschritte

Teilaufgabe	Arbeitspaket 1	Arbeitspaket 2	Arbeitspaket 3
<b>1. Erste API einbinden</b>			
	API-Dokumentation studieren	Request erstellen	Daten abfüllen
<b>2. Zweite API einbinden</b>			
	API-Dokumentation studieren	Request erstellen	Daten abfüllen
<b>3. Datenbankeinbindung</b>			
	Datenbank erstellen	Datenbank aufrufen	Daten abfüllen
<b>4. Frontend</b>			
	Login-Seite erstellen	Home-Seite erstellen	xml einbinden
<b>5. Backend &lt;--&gt; Frontend Verbindung</b>			
	Login View (Backend)	HomeView (Backend)	Ausgabe (Frontend)

### Mittelbedarf

<b>Sachmittel</b>	<ul style="list-style-type: none"> <li>Entwicklungsumgebung (IntelliJ)</li> <li>XAMPP</li> <li>Wildfly-Server</li> </ul>
<b>Personelle Ressourcen</b>	<ul style="list-style-type: none"> <li>Developer (Romina Ferrario)</li> <li>Externe Hilfe (Luca Tschanz, Leandro Meleti, etc.)</li> </ul>
<b>Finanzmittel</b>	<ul style="list-style-type: none"> <li>Es besteht keinen Bedarf an Finanzmitteln, da dieses Projekt keine Kosten verursachen wird</li> </ul>

