



**Hochschule für Technik  
und Wirtschaft Berlin**

*University of Applied Sciences*

# Web Applications Documentation II

Hochschule für Technik und Wirtschaft Berlin  
15.01.2018

Stephan Nordheim

Kristine Schwabauer

<b>Intro</b>	<b>2</b>
<b>Requirements</b>	<b>3</b>
<b>Setting Up</b>	<b>4</b>
<b>How-To-Use</b>	<b>5</b>
<b>Development</b>	<b>6</b>
Libraries	6
API	6
Datenbank	6
<b>Fazit</b>	<b>7</b>
<b>Abbildungsverzeichnis</b>	<b>8</b>

# Intro

Im Rahmen der Web Application Veranstaltung bei Prof. **Gefei Zhang** wurde uns die Aufgabe gestellt, eine Web Applikation, die die Auswertung der Bundestagswahl darstellen soll, zu entwickeln.

Im Kapitel 2 werden die Anforderungen an die Applikation beschrieben.

# Requirements

- Die Daten der btw17\_kerg.csv-Datei soll in eine Sqlite-Datenbank importiert werden.
- Beim Starten der Applikation sollen alle Bundesländer angezeigt werden.
- Beim Anklicken des einzelnen Bundeslandes soll eine Liste der zugehörigen Wahlkreise erscheinen.
- Sowohl die Erststimme als auch die Zweitstimme jeder Partei müssen beim Anklicken auf den Wahlkreis angezeigt werden.
- Das Resultat der Wahlkreise muss grafisch dargestellt werden.
- Der User soll mithilfe einer Suchfunktion nach Wahlkreisen suchen können.
- Die Applikation soll eine Single-Page-App sein, dh. aus einer einzigen Seite bestehen.
- Die Applikation soll in Python implementiert werden und die Frameworks Flask und AngularJS verwenden.

# Setting Up

Bevor die Flask Web Applikation genutzt werden kann, muss das richtige Set Up existieren. Hierzu kann das Projekt direkt von Github <sup>1</sup> gecloned bzw. heruntergeladen werden.

Anschließend müssen folgende Schritte in der Konsole ausgeführt werden, welche nochmal in der ReadMe.md des Projektes nachgelesen werden können:

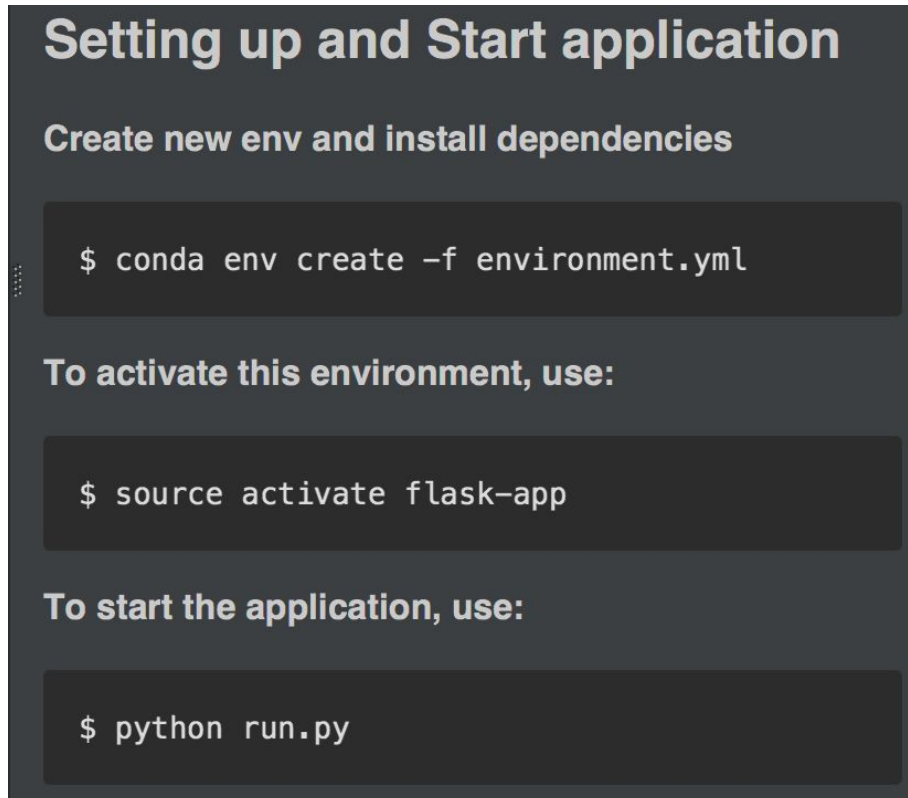


Abbildung 1: Befehl zur Erstellung und Aktivierung der richtigen Umgebung mit Anaconda sowie zum Starten der App

---

<sup>1</sup> <https://github.com/krstnschwpwr/bundestagswahl>

# How-To-Use

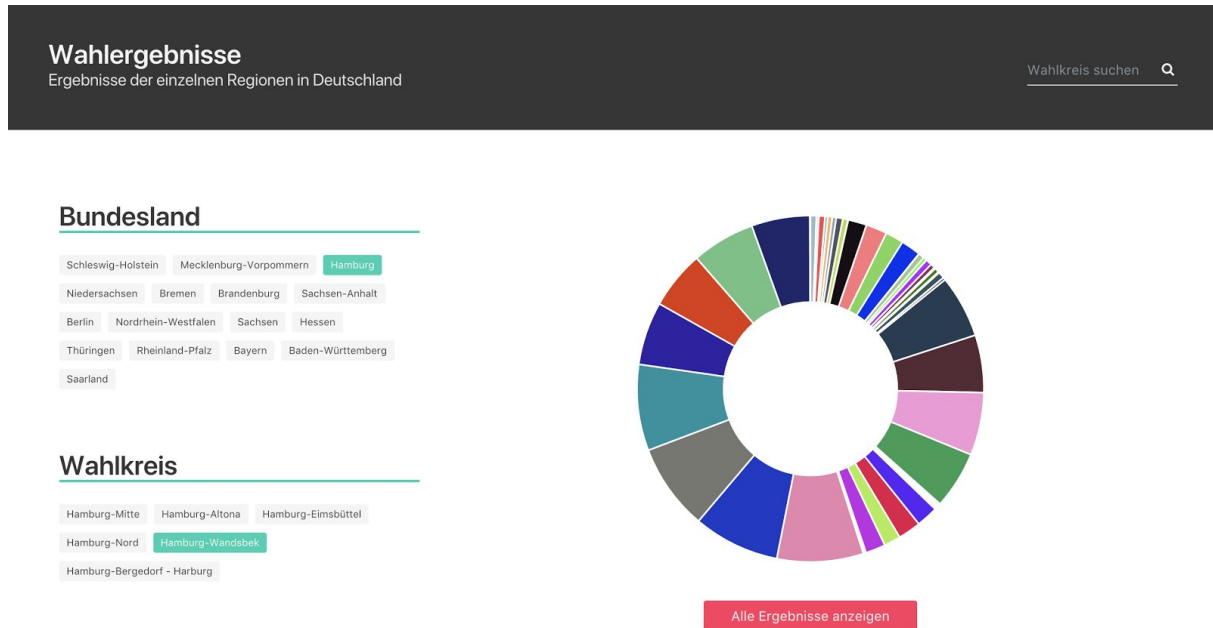


Abbildung 2: Darstellung der Single-Page-Applikation

Da die Applikation laut der Anforderung aus einer einzigen Seite bestehen soll, hat der User auf dieser die Möglichkeit alle Funktionalitäten zu testen und sofort das Ergebnis anzeigen zu lassen. Dieses ist eingeteilt in vier Hauptkomponenten: Bundesland, Wahlkreis, graphische Auswertung sowie eine Tabelle mit den detaillierten Daten.

## Bundesland:

Hier werden beim Aufrufen der Seite alle Bundesländer aufgelistet.

## Wahlkreis:

Beim Anklicken eines Bundeslandes werden alle Wahlkreise des jeweiligen Bundeslandes direkt angezeigt. Um sich einen bestimmten Wahlkreis anzeigen zu lassen kann im Suchfeld "Wahlkreis suchen" in der Headerleiste der Name eingegeben werden. Die Wahlkreise werden dann direkt bei der Eingabe gefiltert und angezeigt.

## Graph:

Nach der Auswahl des jeweiligen Wahlkreises wird ein Donut Graph angezeigt, der alle Daten des ausgewählten Wahlkreises anzeigt. Beim Hovern der einzelnen Bereiche werden die Parteien und entsprechenden Stimmen (Erststimme und Zweitstimme) angezeigt.

**Ergebnisse:**

Um sich die Ergebnisse im Detail anzeigen zu lassen, klickt man auf den “Alle Ergebnisse anzeigen” Button und bekommt eine Tabelle mit allen Parteien/Daten in alphabetischer Reihenfolge und der zugehörigen Stimmenanzahl.

## Development

Im Folgenden werden benutzte Patterns und Methoden sowie Schwierigkeiten beschrieben und genauer erläutert.

### Libraries

Folgende Libraries wurden für die Entwicklung dieser Anwendung verwendet:

- bulma (CSS-Framework)
- jQuery
- Flask-Marshmallow
- AngularJS

### API

**Routen:**

- alle Routen liefern Daten im JSON Format

**Struktur:**

- /api/countries -> gibt alle Bundesländer zurück
- /api/countries/<int:wahlkreis\_id> -> liefert die entsprechenden Wahlkreise eines Bundeslandes
- /api/countries/detail/<int:wahlkreisDetail\_id> -> gibt die Detaildaten eines Wahlbezirks zurück

### Datenbank

Die Datenbank der Wahlergebnisse Applikation besteht insgesamt aus drei Tabellen und hat folgende Felder:

- **Parteien:**
  - id(INTEGER)
  - name(String)
- **Gebiete:**

- id(INTEGER)
- gebiet(String)
- gehoert\_zu(INTEGER)
- **Stimmen:**
  - id (INTEGER)
  - p1 - p48 (INTEGER) - wobei jede Partei bzw. Datensatz einer Spalte entspricht

## Fazit

Die uns gestellte Aufgabe mit dem im ersten Kapitel aufgelisteten Anforderungen, eine Single-Page-Applikation in Python, Flask und AngularJS zur Darstellung der Bundestagswahlen zu entwickeln, konnte erfüllt sowie alle Anforderungen umgesetzt werden. Dabei stellte der Aufbau der Datenbank eine erhebliche Schwierigkeit dar.



# Abbildungsverzeichnis

Abbildung 1: Befehl zur Erstellung und Aktivierung der richtigen Umgebung mit Anaconda sowie zum Starten der App

Abbildung 2: Darstellung der Single-Page-Applikation