

# Dokumentation

## Beschreibung der Implementierung

---

### Konzept

Wir bauen eine Responsive Web App, d.h. sie soll auf allen Bildschirmgröße und Gerätetypen gut anwendbar sein. Dafür verwenden wir Bootstrap. Darüber hinaus setzen wir das Konzept der Single Page Application mittels MEAN Stack um.

Die Web App besteht aus 2 Abschnitten:

- den Fragebogen
- die Karte mit Benchmark-Grafiken

### Anwendung

Die Web App kann somit von einem Rechenzentrum selbst genutzt werden um an dem Benchmark teilzunehmen oder zum anderen von Auditoren um anhand des Fragebogens eine Beurteilung oder . Zertifizierung durchzuführen.

### Kompetenzen

- MVC-Pattern : [Info \(http://de.wikipedia.org/wiki/Model\\_View\\_Controller\)](http://de.wikipedia.org/wiki/Model_View_Controller)
- Dependency Injection & Inversion of Control : [Info \(http://www.itwissen.info/definition/lexikon/Dependency-Injection-dependency-injection-DI.html\)](http://www.itwissen.info/definition/lexikon/Dependency-Injection-dependency-injection-DI.html)
- Responsive Web Design mit Bootstrap : [Bootstrap Website \(http://getbootstrap.com/\)](http://getbootstrap.com/)
- Sass: [SASS \(http://sass-lang.com/\)](http://sass-lang.com/)
- Jade: [JADE \(http://jade-lang.com/\)](http://jade-lang.com/)
- Bourbon [Bourbon \(http://bourbon.io/\)](http://bourbon.io/)
- Angular.js : [Angular.js Website \(https://docs.angularjs.org/guide\)](https://docs.angularjs.org/guide)
- Jasmine : [Jasmine on GitHub \(https://github.com/jasmine/jasmine\)](https://github.com/jasmine/jasmine)
- Node.js : [Node.js Website \(http://nodejs.org/\)](http://nodejs.org/), [Node.js Tutorial \(http://nodeschool.io/\)](http://nodeschool.io/), [express.js framework \(http://expressjs.com/\)](http://expressjs.com/), [node-inspector \(https://github.com/node-inspector/node-inspector\)](https://github.com/node-inspector/node-inspector)
- MongoDB : [MongoDB Website \(http://www.mongodb.org/\)](http://www.mongodb.org/), [MongoLab \(https://mongolab.com/\)](https://mongolab.com/)
- RESTful Services : [Konzept \(https://www.youtube.com/watch?v=YCcAE2SCQ6k\)](https://www.youtube.com/watch?v=YCcAE2SCQ6k), [RESTful API erstellen \(https://www.youtube.com/watch?v=MMOlr\\_VwwAk\)](https://www.youtube.com/watch?v=MMOlr_VwwAk), [FYI \(https://www.youtube.com/watch?v=7YcW25PHnAA\)](https://www.youtube.com/watch?v=7YcW25PHnAA)

### Technologien

#### Single Page Application (SPA)

Single Page Applications sind die zurzeit beliebteste Art von Webseiten. Sowohl bei den Entwicklern als auch bei den Besuchern. Bei SPA's wird nur eine Seite aufgerufen der weitere Inhalt dynamisch via AJAX nachgeladen. Alle visuellen Elemente sind entweder von Beginn an vorhanden (wie bei endlos scrollbaren Seiten) oder werden durch Interaktion in vorgesehene Platzhalter geladen oder ausgetauscht, wobei sich die URL jedoch nie ändert bzw die Seite neu geladen werden muss! Single Page Application kommunizieren über RESTful Services mit dem Server. Sie tauschen dabei Daten im JSON oder XML Format aus. Die eigentliche Anwendungslogik steckt im Client. SPA's fokussieren Frontend-Entwicklung mit HTML Templates, MVC-Frameworks und JavaScript.

#### MEAN Stack

Als MEAN Stack werden die Technologien *MongoDB*, *Express.js*, *Angular.js* und *Node.js* zusammengefasst. Das Paket eignet sich im Besonderen für die Entwicklung von Single Page Applications. Jede oben genannte Eigenschaft wird durch eine der Technologien abgedeckt. Das komplette Webprojekt wird mit HTML, CSS und JavaScript realisiert.

#### MongoDB

MongoDB ist eine der führenden NoSQL Datenbanken. Es handelt sich also nicht um eine klassische relationale Datenbank. MongoDB ist eine dokumenten-orientierte Datenbank. Attribute eines Dokuments werden unter Verwendung von JSON als Key-Value-Paare gespeichert. Dabei sind Dokumente nicht auf die Zweidimensionalität von relationalen

Datenbanken beschränkt, sondern können theoretisch unendlich tief verschachtelte Unterdokumente tragen. Objekte lassen sich so komplex darstellen wie sie in der wahren Welt sind. Da MongoDB objekt-orientiert ist, spart man sich eine ORM-Schicht (Object Relational Mapping). Dabei ist MongoDB performanter als klassische relationale Datenbanken. Die Daten liegen nicht nur logisch sondern auch physisch im Arbeitsspeicher näher beisammen, wodurch Selektionen und Filter schneller ausgewertet werden als durch *JOINS* und mehrfach *SELECT*.

## **Express.js**

Express.js ist ein Framework das auf Node.js aufsetzt. Es erleichtert die Entwicklung von serverseitigem Code und im speziellen Routing und REST-API Entwicklung.

## **Angular.js**

Angular.js ist ein von Google entwickeltes JavaScript MVC-Framework. Die Trennung von Business-, Verhaltens- und Oberflächenlogik ermöglichen agile Programmierung. Der modulare Ansatz fördert re-use von Modulen/Services/Komponenten.

## **Node.js**

Node.js ist ein vollwertiger, in JavaScript geschriebener Webserver. Anders als traditionelle Server, die für jede Verbindung einen neuen Thread mit mindesten 2 MB Arbeitsspeicher anlegen und somit eine maximale Anzahl an simultanen Verbindungen je nach Größe des Arbeitsspeichers haben, erstellt Node.js Prozesse die I/O operationen nicht blocken. Dadurch verschiebt sich der Flaschenhals von der maximalen Anzahl an Verbindungen zum maximalen Kapazität an Traffic, den das System leisten kann. Durch diesen Paradigmenwechsel kann ein Node.js Server allein sehr viel mehr Useranfragen verarbeiten. Zudem lassen sich mehrere Node.js Server problemlos zu einem Cluster zusammenfügen. Eine Webarchitektur basierend auf Node.js ist somit höchst flexibel und skalierbar. Alle Serveranfragen landen in der Event-Loop. -- anhand der Grafik erläutern.

## **Frontend**

### **Preprocessor**

Wir nutzen *SASS* und *JADE* um den Workflow von Markup und Styling zu beschleunigen. Die beiden Preprocessors zeichnen sich durch eine verkürzte Syntax und intuitive Methode aus. Mehrere Zeilen Code werden so eingespart. Außerdem bekommen Entwickler die geplante Arbeit schneller erledigt. Die erstellten .jade und .sass Dateien müssen vorab allerdings zu gewöhnlichen .html und .css Dateien kompiliert werden. Viele IDE's und andere Programme machen dies automatisch durch "File Watcher" oder man kompiliert die Dateien selbst über einen Aufruf in der Konsole. Preprocessors sind ein Standard auf den Webentwickler nicht mehr verzichten.

### **Bootstrap**

Bootstrap ist ein von Twitter entwickeltes Grid-Layout. Es nach dem Paradigma "Mobile first" entstanden. Dieses besagt, dass Webseiten und Anwendungen vorrangig für Mobile Geräte gestaltet werden sollen. Dazu bietet Bootstrap die Möglichkeit Elemente reihenweise über eine feste Anzahl von 12 Spalten zu verteilen. HTML-Elemente können mit vordefinierten CSS-Klassen versehen werden. Die Klassen bestimmen so über wie viele Spalten ein HTML-Element bei einer bestimmten Bildschirmgröße (xs, sm, md und lg) einnehmen soll. So wird das Layout responsive.

## **Backend**

### **Express.js**

-- Aufbau des Servers. Wie werden Anfragen verarbeitet? URL-Routing. vielleicht eigene REST API entwickeln

### **Mongoose**

Mongoose ist eine Node.js Module, welches die Interaktion mit MongoDB vereinfacht.

### **Angular Module**

-- Businesslogik in eigene Module packen. Controller für bestimmte Views.

### **Angular Services**

-- Mongoose Service -- Angular.Ressource Service