

CIC-Challenge

Entsprechend der Fähigkeiten sollte eine der folgenden Varianten gewählt werden:

1. **Back-end:** Nur die Back-end Challenge unten implementieren. Evtl. minimales Front-end hinzufügen (Statischer View oder API Docs)
2. **Front-end:** Nur die Front-end Challenge unten implementieren. Entweder die Back-end Rest-Ressourcen statisch hinzufügen oder ein beliebiges minimales Back-end Service bauen (z.B. NodeJS)
3. **Fullstack:** Front-end und Back-end Challenge implementieren.

Allgemeines

- Der erstellte Sourcecode sollte in einem Source-Repository (github, bitbucket etc.) eingereicht werden.
- Das Ergebnis sollte public gehostet sein (z.B.: Bluemix oder ähnlich).
- Nicht mehr als maximal 5 Stunden für die Challenge verwenden.

Themenvorschläge

1. **Drehorte in San Francisco**
 - Der Service soll von allen Filmen, die in San Francisco gedreht worden sind, den Namen des Films und den Drehort liefern.
 - Daten gibt es hier: <https://data.sfgov.org/Culture-and-Recreation/Film-Locations-in-San-Francisco/yitu-d5am>
 - Im erstellten Service sollte nur der Drehort und der Name des Filmes sein (nicht alle Daten)
2. **CO2 Emissionen von öffentlichen Gebäuden in San Francisco**
 - Der Service soll von allen öffentlichen Gebäuden in San Francisco die CO2 Emissionen liefern.
 - Der erstellte Service sollte nur das Gebäude, den Energieträger und die CO2 Emissionen liefern.
 - Wenn der CO2 Wert 0 ist, sollte der Datensatz ignoriert werden.
 - Die Daten gibt es hier: <https://data.sfgov.org/Energy-and-Environment/San-Francisco-Municipal-Greenhouse-Gas-Inventory/pxac-sadh>

Back-end: REST-Service

Aufgabe ist es, einen REST-Service Server zu implementieren.

Das Ergebnis soll dann etwa so aussehen:

<http://cic-challenge.eu-gb.mybluemix.net/ServiceA>

Anleitung

- Die externen Daten sollte direkt im Service während des Zugriffs gelesen und modifiziert werden.
- Programmiersprache: Java / JEE
- Der zu erstellende Service sollte das JSON Format (oder optional SOAP Format) unterstützen
- Die Wahl der Json (oder SOAP) Library ist frei.
- Filter: Über einen Eingabeparameter unterstützt der Service einen Filter (z.B. Filmmnamen)
- Automatisiertes Testing (optional)
- Die Funktionalität des Services sollte in einem Markdown File dokumentiert sein (als Teil der Sourcen).

Front-end

- Das User-Interface sollte optisch und funktional zum Thema passen.
- Das Front-end sollte eine 'Single Page Application' sein. Ein statischer Entry-Point (index.html) sollte alle nötigen externen Ressourcen (JS/CSS) inkludieren.
- Ein JS Framework (AngularJS, React, Polymer etc.) und JS Libraries können frei gewählt werden .
- Es sollten die CSS3, HTML5 und JS Kenntnisse demonstriert werden.
- In einem Readme.md sollte die wesentliche Funktionalität dokumentiert sein.