

Backend Coding Challenge

team neusta Wiki

Exported on 06/07/2020

Table of Contents

1	Wer sitzt wo?	3
1.1	Gesamtbeschreibung:.....	3
1.1.1	Import Akzeptanzkriterien	3
1.1.2	Import Error Handling.....	3
1.1.3	Raumdetail Akzeptanzkriterien.....	3
1.1.4	Raumdetail Error Handling.....	3
1.1.5	Generelle Akzeptanzkriterien:	3
1.1.5.1	Endpunkt: "POST /api/import"	4
1.1.5.2	Endpunkt: "GET /api/room/{number}"	4
1.1.6	Appendix.....	5
1.1.6.1	Error-Handling	5
1.1.6.2	sitzplan.csv https://portal.neusta.de/confluence/download/attachments/20643910/sitzplan.csv?api=v2&modificationDate=1554379858000&version=1	5

1 Wer sitzt wo?

1.1 Gesamtbeschreibung:

Um schneller herauszubekommen, wer wo sitzt und wie man Leute im Schuppen findet, wollen wir eine Anwendung bauen, die Sitzplätze ausgeben kann. Dazu wollen wir im ersten Schritt eine REST-API erstellen, die uns die Daten zur Verfügung stellt. In einem späteren Schritt soll vor diese API natürlich eine Web-GUI mit entsprechenden Suchfeldern gelegt werden.

Die Rohdaten werden als CSV zur Verfügung gestellt (siehe [sitzplan.csv¹](https://portal.neusta.de/confluence/download/attachments/20643910/sitzplan.csv?api=v2&modificationDate=1554379858000&version=1)). Diese sollen per POST-Request aus einer Datei (multipart/form-data) importiert werden können.

1.1.1 Import Akzeptanzkriterien

1. Es gibt einen Import Endpunkt (siehe **Endpunkt: POST /api/import**)
2. Es gibt einen Endpunkt um Raumdetails abzufragen.
 - a. Details: Raumnummer (room:string), Bewohner (people:[Person])
 - b. Person: (siehe **Endpunkt: GET /api/room/{number}**)
3. Im Fehlerfall wird nicht importiert.
4. Der Datenbestand wird mit einem Import immer überschrieben.

1.1.2 Import Error Handling

1. Räume dürfen nur einmalig in der Importdatei vorkommen, sonst Fehlercode 2 (HTTP 400)
2. Bewohner dürfen nur einmalig in der Importdatei vorkommen, sonst Fehlercode 3 (HTTP 400)
3. CSV Zeile valide (siehe **Endpunkt: POST /api/import**), sonst Fehlercode 4 (HTTP 400)

1.1.3 Raumdetail Akzeptanzkriterien

1. Es gibt einen Raumdetail Endpunkt (siehe **Endpunkt: GET /api/room/{number}**).
2. Wenn keine Raumnummer angegeben wird, werden alle Räume zurückgegeben.

1.1.4 Raumdetail Error Handling

1. Wenn Raum mit angegebener Nummer nicht vorhanden, Fehlercode 5 (HTTP 404)
2. Wenn Raumnummer nicht vierstellig, Fehlercode 6 (HTTP 400)

1.1.5 Generelle Akzeptanzkriterien:

1. Endpunkt mit falscher Methode aufgerufen. (HTTP 405)
2. Rückgabety: application/json
3. Error Rückgabety: application/json

¹ <https://portal.neusta.de/confluence/download/attachments/20643910/sitzplan.csv?api=v2&modificationDate=1554379858000&version=1>

```
{
  "code": int,
  "message": string
}
```

1.1.5.1 Endpunkt: "POST /api/import"

Aufbau Datenfeld (Beispiel-Daten siehe Sitzplan.csv):

Raumnummer (Validierung: 4-stellig, beliebige Zeichen), ggf. Titel Vorname ggf. Zweitname(n) ggf. Namenszusatz Nachname (LDAP-Username), <weiterer Name>, <weiterer Name> ... \n

Erlaubte Titel: Dr.

Erlaubte Namenszusätze: van, von, de

Zweitname ist Teil des Vornamens

1.1.5.2 Endpunkt: "GET /api/room/{number}"

Beispiel-Ausgabe:

```
{
  "room": "1111",
  "people": [
    {
      "first name": "Frank",
      "last name": "Supper",
      "title": "Dr.",
      "name addition": "von",
      "ldapuser": "fsupper"
    },
    {
      "first name": "Dennis",
      "last name": "Fischer",
      "title": "",
      "name addition": "",
      "ldapuser": "dfischer"
    },
    {
      "first name": "Susanne",
      "last name": "Moog",
      "title": "",
      "name addition": "",
      "ldapuser": "smoog"
    }
  ]
}
```

1.1.6 Appendix

1.1.6.1 Error-Handling

- Wird in einem GET-Request auf einen nicht importierten Raum zugegriffen, so soll ein Fehler mit dem Status-Code „404“ und einer entsprechenden Meldung erzeugt werden.
- Fehlt im POST-Request das Daten-Feld oder ist es nicht valide, soll ein Status-Code „400“ mit einer entsprechenden Meldung erzeugt werden.

1.1.6.2 [sitzplan.csv](#)²

sitzplan.csv

```
1111,Dennis Fischer (dfischer),Dr. Frank von Supper (fsupper),Susanne Moog (smoog),
1110,Christina Hülsemann (chulsemann),Iftikar Ahmad Khan (ikhan),Mabelle Tengue (mtengue),Ralf Schmidt
(rschmidt)
1109,Stefanie Borcharding (sborcherding),Tobias Hahn (thahn),Dominik Elm (delm),
1108,Kai Wesling (kwesling),Thomas Kruse (tkruse),,
1107,Carsten Schütte (cschuette),Carsten Elfers (celfers),Nicole Dymini (ndymini),
1106,Jan Stockfisch (jstockfisch),Denis Sawazky (dsawazky),Simon Bled (sbled),
1105,Marina Bortin (mbortin),Svenja Hilken (shilken),Florenz Buhrke (fbuhrke),
1104,Sebastian Foot (sfoot),Alexander Häusler (ahaeusler),Renas Sheho (rsheho),Dirk Reuschling
(dreuschling)
1103,Leif Arne Genzmer (lgenzmer),Samin ölker (soelker),,
1102,,,
1101,Markus Nolte (mnolte),,,
1113,Claudia Fleuter (cfleuter),Sabine Strodthoff (sstrodthoff),,
1114,Malek Olabi (molabi),Alexander James Cole (acole),,
2114,Daniel Ott (dott),Katharina Tan (ktan),Arne Wagenfeld (awagenfeld), Janina Pinnhammer
(jpinnhammer),Janis Hau (jhau),Iaroslav Sheptykin (isheptykin),Ulf Tietze (utietze),Bernhard Leers
(bkatzmarski), Julian Nuß (jnuss), Thomas von Gostomski (tgostomski),,
2002,Janina Thomas (jthomas),Jonas Zimmermann (jzimmermann),Tobias Küttner (tkuettner), Sarah Fröhling
(sfroehling), Yvonne Asendorf (yasendorf), Etienne de Ruffray (eruffray),Dr. Dennis Krannich (dkrannich),
```

² <https://portal.neusta.de/confluence/download/attachments/20643910/sitzplan.csv?api=v2&modificationDate=1554379858000&version=1>