

Permanente evaluatie Web Expert

40% van de punten van het vak Web Expert worden bepaald door het resultaat van de permanente evaluatie. Let op dit cijfer blijft behouden in de tweede zitting.

In totaal zijn er 4 werkpakketten die verbeterd worden op 40 punten.

Werkpakket 1

Werkpakket 1 staat op 6 punten van de 40.

Je plaatst de code voor wp1 in de repository die je via de onderstaande link aanmaakt.

<https://classroom.github.com/a/R7B4bGcR>

Voor werkpakket1 maak je gebruik van Express om json-server (fake REST-API) na te bouwen. In onderstaande link kan je de documentatie van json-server terugvinden. Het is niet de bedoeling om deze repository te clonen.

<https://github.com/typicode/json-server>

Bij het opstarten van je toepassing worden de routes eigenschappen (properties) en gegevens (data) uit een JSON-configuratiebestand gelezen. Deze gegevens worden bewaard in het geheugen.

Een mogelijk voorbeeld van de inhoud van het configuratiebestand wordt op de volgende pagina gegeven. Je mag hiervan afwijken.

```
[
  {
    "route": "person",
    "properties": [
      "name",
      "age",
      "petIds"
    ],
    "data": [
      {
        "id": 1,
        "name": "tim",
        "age": 11,
        "petIds": [
          1,
          2
        ]
      },
      {
        "id": 2,
        "name": "sofie",
        "age": 8,
        "petIds": [
        ]
      }
    ]
  },
  {
    "route": "pet",
    "properties": [
      "name"
    ],
    "data": [
      {
        "id": 1,
        "name": "bliksem"
      },
      {
        "id": 2,
        "name": "duvel"
      }
    ]
  }
]
```

Je implementeert onderstaande routes:

(1) GET /route_i

Dit request geeft een array van alle objecten voor route_i terug.

Hint:

```
app.get('/:route/', (req, res, next)=>{  
    const {route} = req.params;  
    ...  
});
```

(2) GET /route_i/1

Dit request geeft het object met id=1 voor route_i terug.

```
app.get('/:route/:id', (req, res, next)=>{  
    const {route, id} = req.params;  
    // zoek het object in data van route op dat de waarde voor id heeft  
    ...  
});
```

(3) POST /route_i

Er wordt gecontroleerd of (in het json-object in de request-body) een waarde ingegeven wordt voor alle eigenschappen (properties). Indien dit niet zo is wordt een leeg object en statuscode 400 teruggegeven.

Er wordt een nieuw object aangemaakt.

Dit object wordt bijgehouden in het geheugen (een volgend GET request geeft de gewijzigde toestand terug).

Je denkt aan de juiste status-code. Ook voor niet-bestaande routes, niet-bestaande id's, invalid json in de request-body.

Verplicht te gebruiken technische aspecten:

- express
- callback om het configuratiebestand te lezen

Als je de opdracht zoals hierboven uitwerkt krijg je 10/20.

Extra punten:

- full text search

2pt

Full-text search

Add `q`

```
GET /posts?q=internet
```

- relationships

2pt

Relationships

To include children resources, add `_embed`

```
GET /posts?_embed=comments
GET /posts/1?_embed=comments
```

- operators `_gte`, `_lte`

2pt

Operators

Add `_gte` or `_lte` for getting a range

```
GET /posts?views_gte=10&views_lte=20
```

- gebruik yargs om bij het opstarten de lokatie van het configuratiebestand te specificeren

2pt

- zorg dat bij het afsluiten (ctrl-c) de nieuwe toestand bewaard wordt in het configuratiebestand

2pt