

# Programmeerimine II

## HKI5003.HK

MARTTI RAAVEL

MRT@TLU.EE

# Minust

- ▶ Rapla Gümnaasiumi IT-juht
- ▶ Haapsalu Kolledži viliestlane
- ▶ TLÜ magistrant
- ▶ Tarkvara arendaja väikeses ettevõttes

# Aine lühikirjeldus

- ▶ CRUD API. Node.js baasil töötava API loomine. NPM. Express. Middleware. API struktureerimine. API testimine. Autentimine ja autoriseerimine. Koodi haldamine github.com keskkonnas.
- ▶ Kasutades loengutes saadud teadmisi, loob üliõpilane iseseisva tööna eelnevalt nimetatud tehnoloogiaid kasutades töötava API, mille kood on koodihalduskeskkonnas, mis on testitav, sisaldab autentimist ja autoriseerimist. API peab võimaldama luua/lugeda/muuta/kustutada vähemalt viite erinevat tüüpi kirjeid. Üliõpilane peab olema võimeline oma projekti kaitse ja põhjendama oma valikuid.

# Hindamine

- ▶ Eksam
- ▶ 50% moodustab projekti kaitsmine
- ▶ 50% erinevate kriteeriumite täitmine (Kogu kood algusest peale githubis, API struktuur, autentimise ja autoriseerimise olemasolu, koodi testitavus jne)
- ▶ „A” - suurepärane 91-100%
- ▶ „B” - väga hea 81- 90%
- ▶ „C” - hea 71- 80%
- ▶ „D” - rahuldav 61-70%
- ▶ „E” - kasin 51- 60%
- ▶ „F” - puudulik 0 – 50%

# Loengute ajad

- ▶ 16. oktoober: sissejuhatus ainesse, github, nodejs, npm, Express, Postman, json, projekti tutvustus ja alustamine.
- ▶ 31. oktoober: API struktuur, moodulid, CRUD, REST, controller, service
- ▶ 14. november: middleware, autentimine, autoriseerimine
- ▶ 28. november: andmebaasiga liidestamine
- ▶ 18. detsember: testimine, automaattestid
- ▶ 9. jaanuar: eksam

# Kodused tööd

- ▶ Loengu lõpus väljapääsupilet
- ▶ Loengu lõpus antav kodune ülesanne
- ▶ Järgmises loengus 5 juhuslikult valitud üliõpilast näitavad ja tutvustavad oma kodust tööd

# Github

- ▶ Koodihalduskeskkond
- ▶ Ligipääs, kui kood on privaatses repositoariumis (mrttlu)
- ▶ Create repository
  - ▶ Add a README file
  - ▶ Add .gitignore (template: Node)

# CRUD API

- ▶ Application Programming Interface
  - ▶ Create
  - ▶ Read
  - ▶ Update
  - ▶ Delete

<https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods>

# Node.js – javascript, mis töötab serveris

- ▶ Lihtne alustada
- ▶ Asünkroonne
- ▶ JSON
- ▶ npm
- ▶ Kes kasutavad?
  - ▶ PayPal
  - ▶ eBay
  - ▶ LinkedIn
  - ▶ Netflix

<https://nodejs.org/en/>

```
const express = require('express');
const app = express();

const PORT = 3000;

app.get('/', (req, res) => {
  res.status(200).json({
    success: true,
    message: 'Hello world!'
  });
});

app.listen(PORT, () => {
  console.log('App is listening on port: ', PORT);
});
```

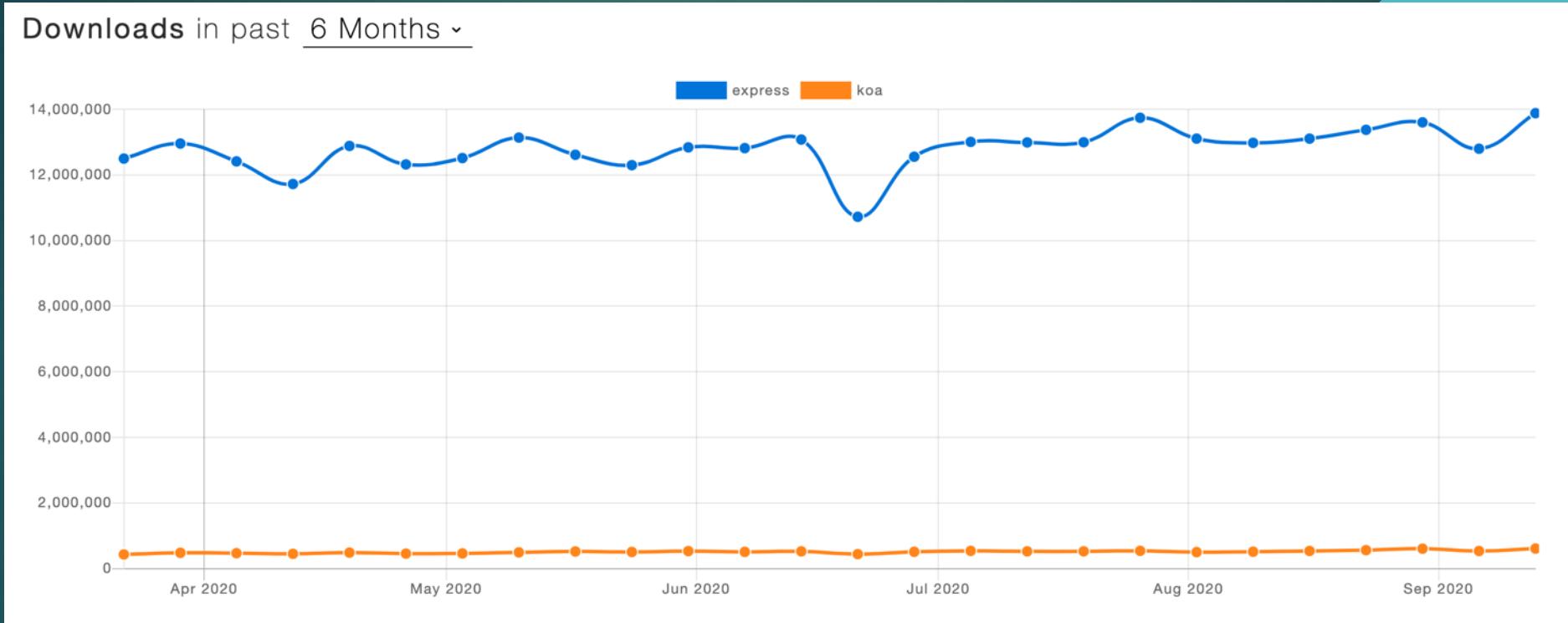
# npm – Node Package Manager

- ▶ Online repositoorium Node.js projektide jaoks
  - ▶ <http://npmjs.org/>
- ▶ Käsurealt kasutatav utiliit nimetatud repositooriu mig a suhtlemiseks (pakkide paigaldamiseks, versioonide ja sõltuvuste haldamiseks)
  - ▶ npm init
  - ▶ npm install
  - ▶ node\_modules
  - ▶ package.json

<https://nodejs.org/en/knowledge/getting-started/npm/what-is-npm/>

# Express

- ▶ Javascripti veebiraamistik node.js-le



<https://www.npmtrends.com/express-vs-koa>

<https://expressjs.com/>

# HTTP request methods

- ▶ POST - CREATE
- ▶ GET - READ
- ▶ PUT - UPDATE
- ▶ DELETE - DELETE

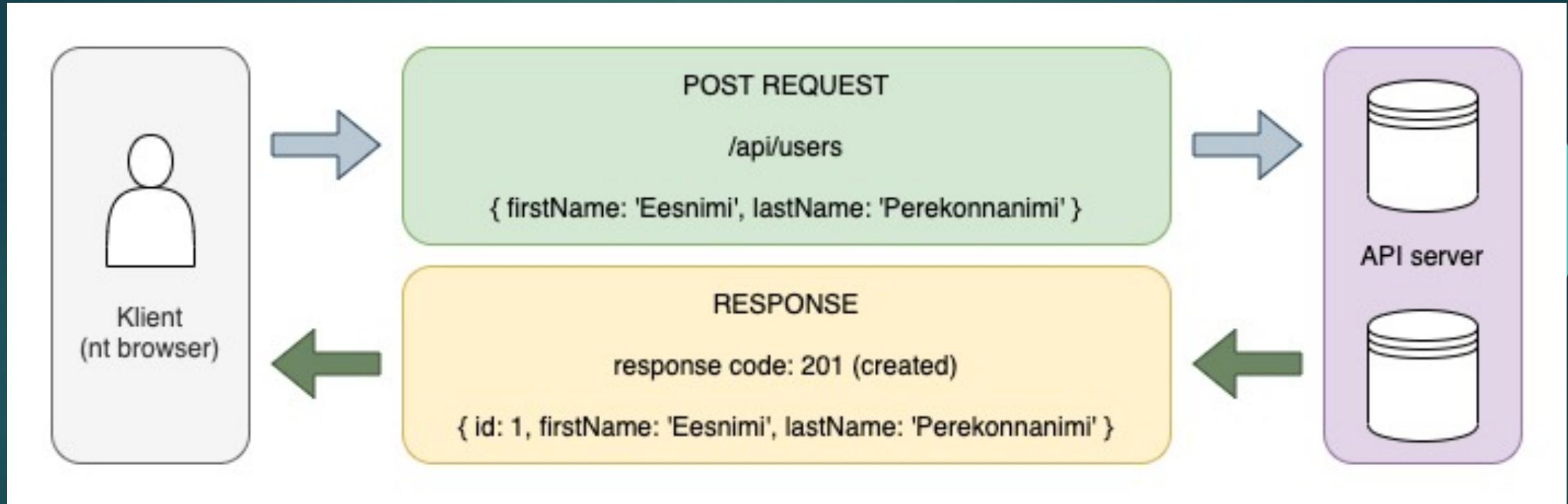
<https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods>

# HTTP response codes

- ▶ Edukad
  - ▶ 200 OK
  - ▶ 201 Created
- ▶ Vead
  - ▶ 400 Bad request
  - ▶ 401 Unauthorized
  - ▶ 403 Forbidden
  - ▶ 404 Not Found
- ▶ Server errors ...

<https://developer.mozilla.org/en-US/docs/Web/HTTP/Status>

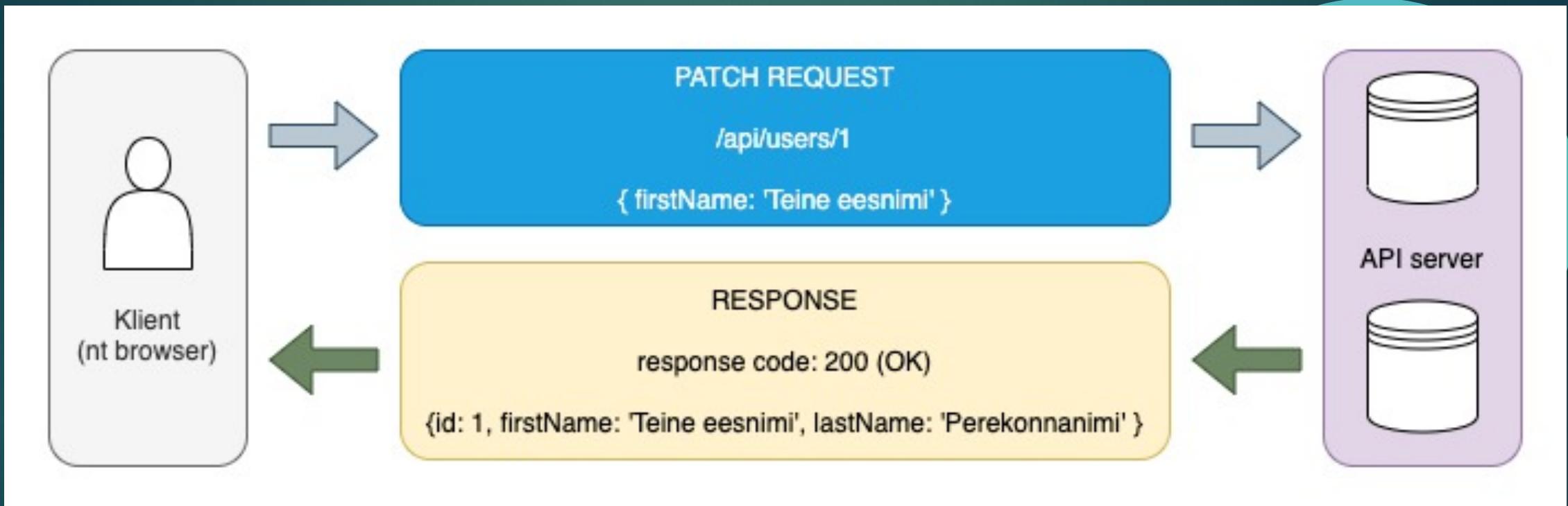
# CREATE



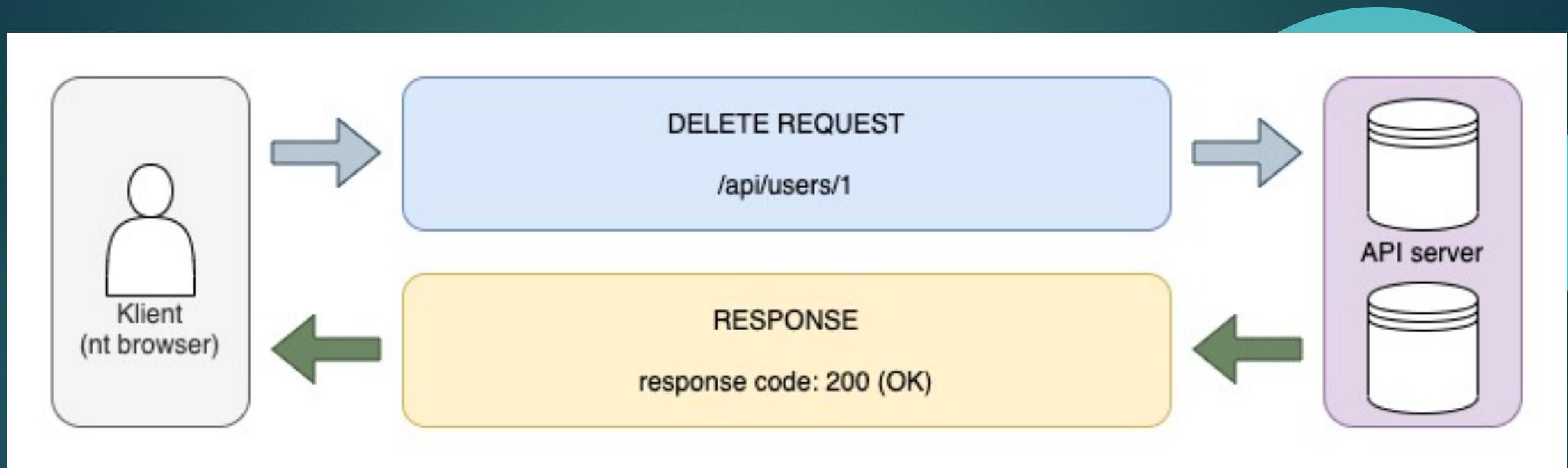
# READ



# UPDATE



# DELETE



# Postman - API arendamise platvorm

The screenshot shows the Postman application interface. At the top, there is a header bar with a 'GET' button, a URL input field containing 'localhost:3000', and a dropdown menu showing 'No Environment'. Below the header is a toolbar with 'Send' and 'Save' buttons. The main area is titled 'Untitled Request' and contains tabs for 'Params', 'Authorization', 'Headers (7)', 'Body', 'Pre-request Script', 'Tests', 'Settings', 'Cookies', and 'Code'. The 'Body' tab is selected, showing a table for 'Query Params' with one row: 'Key' (Value: Value) and 'Description' (Value: Description). Below the table, there are tabs for 'Body', 'Cookies', 'Headers (6)', and 'Test Results'. The 'Body' tab is active, displaying a JSON response with status 200 OK, time 61 ms, and size 253 B. The response body is shown in a JSONpretty-printed format:

```
1 {  
2   "success": true,  
3   "message": "Hello world!"  
4 }
```

<https://www.postman.com/downloads/>

# JSON - JavaScript Object Notation

```
{  
  "employees": [  
    {  
      "firstName" : "John",  
      "lastName" : "Doe"  
    },  
    {  
      "firstName" : "Anna",  
      "lastName" : "Smith"  
    }  
  ]  
}
```

<https://www.json.org/json-en.html>

# Kodune töö

- ▶ Väljapääsupilet:
- ▶ Alusta oma projektiga
  - ▶ Projekt Githubi
  - ▶ Tee API vähemalt 3 lõpp-punktiga (end-point)