

LIIGA API

Liigan sarjataulukko kaudelta 2023-24.
60 pelatun ottelun jälkeen

TEKNIIKAT

Tietokantana SQLite3

Käyttää Node.js ää

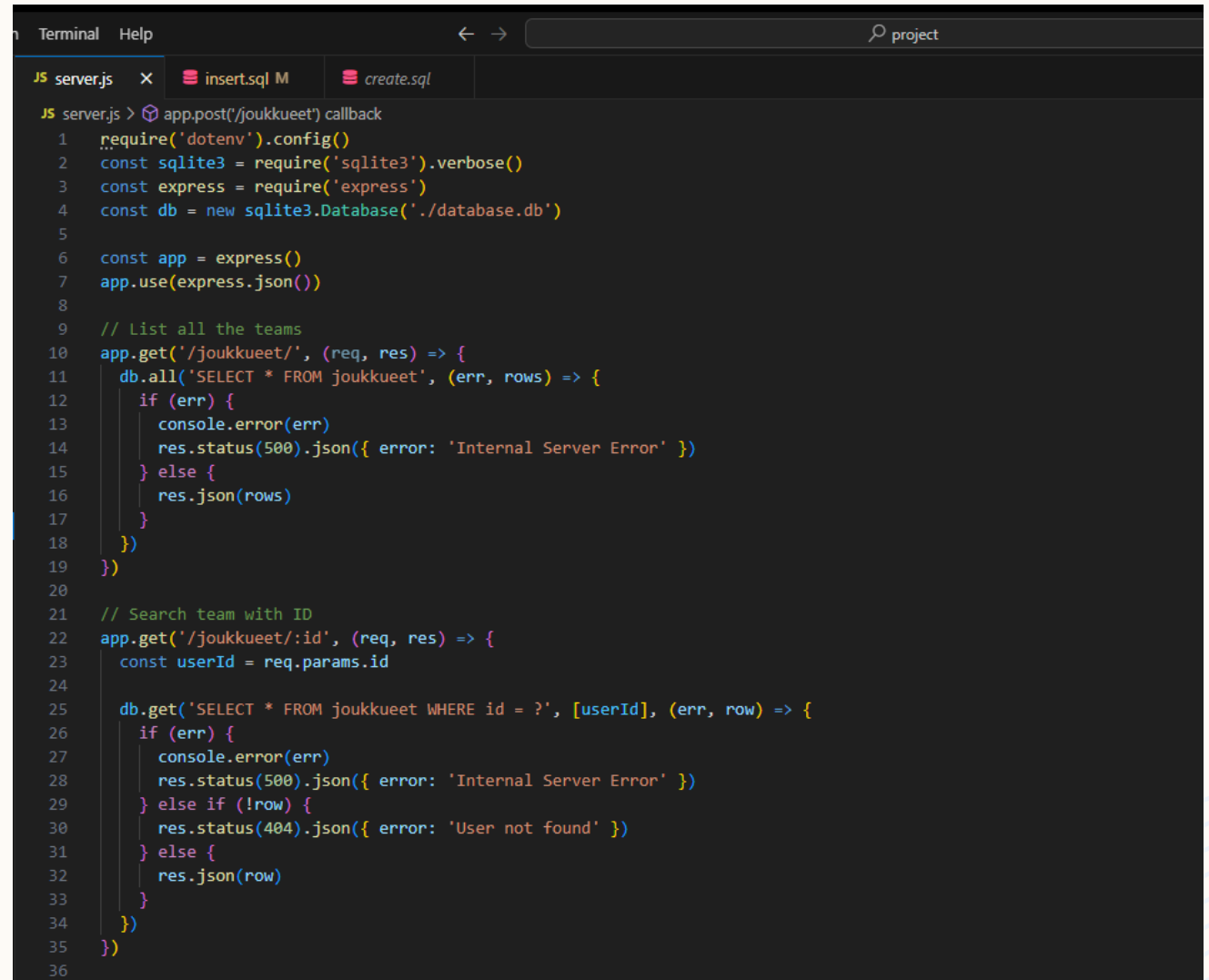
Muita node paketteja express ja dotenv

SERVER.JS

Moduulejen importoinnit

Serverin käynnistys

Ja reitityksen endpointit



```
JS server.js > app.post('/joukkueet') callback
1  require('dotenv').config()
2  const sqlite3 = require('sqlite3').verbose()
3  const express = require('express')
4  const db = new sqlite3.Database('./database.db')
5
6  const app = express()
7  app.use(express.json())
8
9  // List all the teams
10 app.get('/joukkueet/', (req, res) => {
11   db.all('SELECT * FROM joukkueet', (err, rows) => {
12     if (err) {
13       console.error(err)
14       res.status(500).json({ error: 'Internal Server Error' })
15     } else {
16       res.json(rows)
17     }
18   })
19 })
20
21 // Search team with ID
22 app.get('/joukkueet/:id', (req, res) => {
23   const userId = req.params.id
24
25   db.get('SELECT * FROM joukkueet WHERE id = ?', [userId], (err, row) => {
26     if (err) {
27       console.error(err)
28       res.status(500).json({ error: 'Internal Server Error' })
29     } else if (!row) {
30       res.status(404).json({ error: 'User not found' })
31     } else {
32       res.json(row)
33     }
34   })
35 })
36
```

TIETOKANTA

Koostuu 8 sarakkeesta ja 15 rivistä

SELECT * FROM joukkueet Schema Query Editor Auto Reload Find Other Tools...								
	id INTEGER PRIMARY KEY ↑	nimi VARCHAR(60) NOT NULL	Ottelut INT	Voitot INT	Tasapelit INT	Tappiot INT	Lisäpisteet INT	Pisteet INT
1	1	Tappara	60	34	13	13	8	123
2	2	Ilves	60	33	14	13	7	120
3	3	Pelicans	60	31	12	17	4	109
4	4	Kärpät	60	28	14	18	8	106
5	5	Jukurit	60	30	10	20	3	103
6	6	HIFK	60	26	18	16	7	103
7	7	KalPa	60	25	13	22	6	94
8	8	Lukko	60	22	16	22	10	92
9	9	TPS	60	17	19	24	14	84
10	10	Sport	60	21	15	24	5	83
11	11	Ässät	60	19	17	24	7	81
12	12	KooKoo	60	17	17	26	11	79
13	13	JYP	60	16	14	30	7	69
14	14	HPK	60	15	10	35	4	59
15	15	SaiPa	60	5	20	35	10	45
+								

```
server.js insert.sql M delete.sql  
sql > insert.sql  
Run on active connection | Select block  
1 INSERT INTO joukkueet (nimi, Ottelut, Voitot, Tasapelit, Tappiot, Lisäpisteet, Pisteet) VALUES  
2 ('Tappara', 60, 34, 13, 13, 8, 123);  
3  
4 INSERT INTO joukkueet (nimi, Ottelut, Voitot, Tasapelit, Tappiot, Lisäpisteet, Pisteet) VALUES  
5 ('Ilves', 60, 33, 14, 13, 7, 120);  
6  
7 INSERT INTO joukkueet (nimi, Ottelut, Voitot, Tasapelit, Tappiot, Lisäpisteet, Pisteet) VALUES  
8 ('Pelicans', 60, 31, 12, 17, 4, 109);  
9  
10 INSERT INTO joukkueet (nimi, Ottelut, Voitot, Tasapelit, Tappiot, Lisäpisteet, Pisteet) VALUES  
11 ('Kärpät', 60, 28, 14, 18, 8, 106);  
12  
13 INSERT INTO joukkueet (nimi, Ottelut, Voitot, Tasapelit, Tappiot, Lisäpisteet, Pisteet) VALUES  
14 ('Jukurit', 60, 30, 10, 20, 3, 103);  
15  
16 INSERT INTO joukkueet (nimi, Ottelut, Voitot, Tasapelit, Tappiot, Lisäpisteet, Pisteet) VALUES  
17 ('HIFK', 60, 26, 18, 16, 7, 103);  
18  
Response(4ms) create.sql  
sql > create.sql  
Run on active connection | Select block  
1 DROP TABLE IF EXISTS joukkueet;  
2 CREATE TABLE IF NOT EXISTS joukkueet (  
3     id INTEGER PRIMARY KEY,  
4     nimi VARCHAR(60) NOT NULL,  
5     Ottelut INT,  
6     Voitot INT,  
7     Tasapelit INT,  
8     Tappiot INT,  
9     Lisäpisteet INT,  
10    Pisteet INT  
11 );  
12
```

Tietokannan alustus oikealla
create.sql tiedostossa ja vasemman
puoleisessa insert.sql tiedostossa
datan syöttö tauluihin

OHJEET TIETOKANNAN ALUSTUKSEEN

```
1  Open or create database.db file
2  with command
3
4  <sqlite3 database.db>
5
6  copy statements from create.sql to create tables
7
8  copy statements from insert.sql to insert database
9
10
11  If you want to empty the database!
12  copy statements from delete.sql to empty database
```

JOUKKUEEN HAKEMINEN NIMELLÄ

```
35  })
36
37  // Search by team name
38  app.get('/search/', (req, res) => {
39    const teamName = req.query.team
40    const sql = 'SELECT * FROM joukkueet WHERE nimi LIKE \'%\'' || ? || \'%\''
41
42    db.all(sql, [`%${teamName}%`], (err, rows) => {
43      if (err) {
44        console.error(err.message)
45        return res.status(500).json({ error: 'Internal Server Error' })
46      }
47      res.status(200).json(rows)
48    })
49  })
50
51  // Query with amount of losses and points
```

```
localhost.rest > ...  
1  
2   ### Get all  
   Send Request  
3   GET http://localhost:8080/joukkueet/  
4  
5  
6   ### Get by id  
   Send Request  
7   GET http://localhost:8080/joukkueet/2  
8  
9  
10  ### Query with name  
   Send Request  
11  GET http://localhost:8080/search?team=ilves  
12  
13  
14  ### Query with amount of losses and points  
   Send Request  
15  GET http://localhost:8080/query?tappiot=15&pisteet=100  
16  
17  
18  ### DELETE by id  
   Send Request  
19  DELETE http://localhost:8080/joukkueet/11  
20  
21  
22  ### Add new team  
   Send Request  
23  POST http://localhost:8080/joukkueet/  
24  Content-Type: application/json  
25  
26  {  
27    "nimi": "Saipa",  
28    "Ottelut": 60,  
29    "Voitot": 5,  
30    "Tasapelit": 20,  
31    "Tappiot": 35,  
32    "Lisäpisteet": 10,  
33    "Pisteet": 45  
34  }
```

ESIMERKKI API KUTSUT LISTATTUNA REST CLIENTISSA

ESIMERKKI TULOSTUS GET ALL KUTSULLE

9

```
project
Response(6ms) X
1 HTTP/1.1 200 OK
2 X-Powered-By: Express
3 Content-Type: application/json; charset=utf-8
4 Content-Length: 1627
5 ETag: W/"65b-wmrgzERbhV5pTb6xxL9u51xia6Q"
6 Date: Sun, 24 Mar 2024 21:18:29 GMT
7 Connection: close
8
9 √ [
10 √ {
11   "id": 1,
12   "nimi": "Tappara",
13   "Ottelut": 60,
14   "Voitot": 34,
15   "Tasapelit": 13,
16   "Tappiot": 13,
17   "Lisäpisteet": 8,
18   "Pisteet": 123
19 },
20 √ {
21   "id": 2,
22   "nimi": "Ilves",
23   "Ottelut": 60,
24   "Voitot": 33,
25   "Tasapelit": 14,
26   "Tappiot": 13,
27   "Lisäpisteet": 7,
28   "Pisteet": 120
29 },
30 √ {
31   "id": 3,
32   "nimi": "Pelicans",
33   "Ottelut": 60,
34   "Voitot": 31,
35   "Tasapelit": 12,
36   "Tappiot": 17,
37   "Lisäpisteet": 4,
38   "Pisteet": 109
39 },
40 √ {
41   "id": 4,
42   "nimi": "Kärpät",
43   "Ottelut": 60,
44   "Voitot": 30,
45   "Tasapelit": 11,
46   "Tappiot": 18,
47   "Lisäpisteet": 3,
48   "Pisteet": 107
49 }
50 ]
```

API:N KUVAUS

```
26
27 API DESCRIPTION
28
29 The API provides endpoints for managing teams in Liiga sarjataulukko.
30 Below are the available endpoints along with their descriptions:
31
32 GET http://localhost:8080/joukkueet
33 | Description: Retrieves a list of all teams.
34
35 GET http://localhost:8080/joukkueet/{id}
36 | Description: Retrieves details of a specific team identified by their ID.
37 | Parameters:
38 |   - id (integer): The unique identifier of the team.
39
40 GET http://localhost:8080/search?team=ilves
41 | Description: Retrieves details of teams matching the provided team name query parameter.
42 | Parameters:
43 |   - team (string): The name of the team to search for.
44
45 GET http://localhost:8080/query?tappiot=15&pisteet=100
46 | Description: Retrieves a list of teams based on the specified criteria of maximum losses and minimum points.
47 | Parameters:
48 |   - tappiot (integer): The maximum number of losses allowed.
49 |   - pisteet (integer): The minimum number of points required.
50
51 DELETE http://localhost:8080/joukkueet/{id}
52 | Description: Deletes a team.
53 | Parameters:
54 |   - id (integer): The unique identifier of the team to be deleted.
55
56 POST http://localhost:8080/joukkueet
57 | Description: Creates a new team.
58 | Request Body:
59 |   - nimi (string)
60 |   - ottelut (string)
61 |   - voitot (int)
62 |   - tasapelit (int)
63 |   - tappiot (int)
64 |   - lisapisteet (int)
65 |   - pisteet (int)
66
```

ESIMERKKI CURL- KOMENNOT

```
66
67 API EXAMPLES
68
69 <curl .... command examples>
70 1. Get list of all teams:
71 curl --silent --include http://localhost:8080/joukkueet
72
73 2. Get details of a specific team with id:
74 curl --silent --include http://localhost:8080/joukkueet/2
75
76 3. Search team by name:
77 curl --silent --include http://localhost:8080/search?team=ilves
78
79 4. Query with amount of losses and points:
80 curl --silent --include http://localhost:8080/query?tappiot=15&pisteet=100
81
82 5. Delete team by id:
83 curl -X DELETE http://localhost:8080/joukkueet/10
84
85 6. Add new team:
86 curl -X POST http://localhost:8080/joukkueet/ -H "Content-Type: application/json" -d '{
87     "nimi": "Saipa",
88     "Ottelut": 60,
89     "Voitot": 5,
90     "Tasapelit": 20,
91     "Tappiot": 35,
92     "Lisäpisteet": 10,
93     "Pisteet": 45
94 }'
```

UUDEN JOUKKUEEN LISÄÄMINEN TIETOKANTAAN


```
88
89 // Add team into the table
90 app.post('/joukkueet', (req, res) => {
91   const newJoukkue = req.body
92
93   db.run('INSERT INTO joukkueet (nimi, Ottelut, Voitot, Tasapelit, Tappiot, Lisäpisteet, Pisteet) VALUES (?, ?, ?, ?, ?, ?, ?)',
94     [newJoukkue.nimi,
95       newJoukkue.Ottelut,
96       newJoukkue.Voitot,
97       newJoukkue.Tasapelit,
98       newJoukkue.Tappiot,
99       newJoukkue.Lisäpisteet,
100      newJoukkue.Pisteet],
101     function (err) {
102       if (err) {
103         console.error(err)
104         return res.status(500).json({ error: 'Internal Server Error' })
105       }
106       res.status(201).json({ message: 'New joukkue added successfully' })
107     })
108   })
109
110 const PORT = process.env.PORT || 8000
111
112 app.listen(PORT, () => {
113   console.log(`Server is running on http://localhost:${PORT}`)
114 })
115
```

JOUKKUEEN POISTAMINEN TIETOKANNASTA ID:LLÄ

```
1 // Delete team from the table
2 app.delete('/joukkueet/:id', (req, res) => {
3   const id = req.params.id
4
5   db.run('DELETE FROM joukkueet WHERE id = ?', [id], function (err) {
6     if (err) {
7       console.error(err.message)
8       return res.status(500).json({ error: 'Internal Server Error' })
9     }
10    if (this.changes === 0) {
11      return res.status(404).json({ error: 'Entry not found' })
12    }
13    console.log(`Entry with id ${id} has been deleted`)
14    res.status(204).send()
15  })
16 })
17 })
18
```

PACKAGE.JSON

14

```
{ } package.json > { } scripts >  dev
1  {
2    "name": "project",
3    "version": "1.0.0",
4    "description": "GENERAL",
5    "main": "server.js",
6    "scripts": {
7      "test": "standard && echo \"Error: no test specified\" && exit 1",
8      "start": "node server.js",
9      "dev": "nodemon server.js"
10   },
11   "author": "Timo Kempainen",
12   "license": "ISC",
13   "dependencies": {
14     "dotenv": "^16.4.5",
15     "express": "^4.18.3",
16     "md5": "^2.3.0",
17     "sqlite3": "^5.1.7"
18   },
19   "devDependencies": {
20     "nodemon": "^3.1.0",
21     "standard": "^17.1.0"
22   }
23 }
24
```

KIITOS