

18) Opintorekisteri, Arviointitaulu

Database.js tiedosto:

```
JS database.js > ...
1  const mysql = require('mysql2');
2  const connection = mysql.createPool({
3    host: 'localhost',
4    user: 'testi',
5    password: 'xxxx',
6    database: 'test_db2'
7  });
8  module.exports = connection;
```

Arviointi.js tiedosto

```
routes > JS arviointi.js > router.get('/:id') callback
1  const express = require('express');
2  const router = express.Router();
3  const arviointi = require('../models/arviointi_model');
4
5  router.get('/',
6    function (request, response) {
7      arviointi.getAll(function (err, dbResult) {
8        if (err) {
9          response.json(err);
10         } else {
11           console.log(dbResult);
12           response.json(dbResult);
13         }
14       });
15     });
16
17  router.get('/:id',
18    function (request, response) {}
19    arviointi.getById(request.params.id, function (err, dbResult) {
20      if (err) {
21        response.json(err);
22      } else {
23        response.json(dbResult);
24      }
25    });
26  });
27
28
29  router.post('/',
30    function(request, response) {
31      arviointi.add(request.body, function(err, dbResult) {
32        if (err) {
33          response.json(err);
34        } else {
35          response.json(dbResult);
36        }
37      });
38    });
39
40
41  router.delete('/:id',
42    function(request, response) {
43      arviointi.delete(request.params.id, function(err, dbResult) {
44        if (err) {
45          response.json(err);
46        } else {
47          response.json(dbResult);
48        }
49      });
50    });
51
52
53  router.put('/:id',
54    function(request, response) {
55      arviointi.update(request.params.id, request.body, function(err, dbResult) {
56        if (err) {
57          response.json(err);
58        } else {
59          response.json(dbResult);
60        }
61      });
62    });
63
64  module.exports = router;
```

Arviointi_model.js tiedosto:

```
models > JS arviointi_model.js > ...
1  const db = require('../database');
2
3  const arviointi = {
4    getAll: function(callback) {
5      return db.query('select * from arviointi', callback);
6    },
7    getById: function(id, callback) {
8      return db.query('select * from arviointi where idArviointi=?', [id], callback);
9    },
10   add: function(arviointi, callback) {
11     return db.query(
12       'insert into arviointi (idOpintojakso,idOpiskelija,arvosana,paivamaara) values(?,?,?,?)',
13       [arviointi.idOpintojakso, arviointi.idOpiskelija, arviointi.arvosana, arviointi.paivamaara],
14       callback
15     );
16   },
17   delete: function(id, callback) {
18     return db.query('delete from arviointi where idArviointi=?', [id], callback);
19   },
20   update: function(id, arviointi, callback) {
21     return db.query(
22       'update arviointi set idOpintojakso=?,idOpiskelija=?, arvosana=?, paivamaara=? where idArviointi=?',
23       [arviointi.idOpintojakso, arviointi.idOpiskelija, arviointi.arvosana, arviointi.paivamaara, id],
24       callback
25     );
26   }
27 };
28 module.exports = arviointi;
```

App.js tiedosto:

```
JS app.js > ...
1  var express = require('express');
2  var path = require('path');
3  var cookieParser = require('cookie-parser');
4  var logger = require('morgan');
5
6
7  var arviointiRouter = require('./routes/arviointi');
8
9
10 var app = express();
11
12 app.use(logger('dev'));
13 app.use(express.json());
14 app.use(express.urlencoded({ extended: false }));
15 app.use(cookieParser());
16 app.use(express.static(path.join(__dirname, 'public')));
17
18
19 app.use('/arviointi', arviointiRouter);
20 |
21
22 module.exports = app;
23
```

GET-metodi:

The screenshot shows a REST client interface with a GET request to `localhost:3000/arviointi/5`. The response is a JSON array containing one object with the following fields: `idArviointi` (5), `idOpintojakso` (3), `idOpiskelija` (3), `arvosana` (5), and `paivamaara` (2025-01-28T22:00:00.000Z).

GET `localhost:3000/arviointi/5`

Params Authorization Headers (7) Body Pre-request Script Tests Settings

Query Params

Key
Key

Body Cookies Headers (7) Test Results

Pretty Raw Preview JSON

```
1 [
2   {
3     "idArviointi": 5,
4     "idOpintojakso": 3,
5     "idOpiskelija": 3,
6     "arvosana": 5,
7     "paivamaara": "2025-01-28T22:00:00.000Z"
8   }
9 ]
```

POST-metodi:

POST localhost:3000/arviointi

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings

☐ none ☐ form-data ☒ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

	Key	Value
<input checked="" type="checkbox"/>	idOpintojakso	1
<input checked="" type="checkbox"/>	idOpiskelija	4
<input checked="" type="checkbox"/>	arvosana	3
<input checked="" type="checkbox"/>	paivamaara	2025-02-20

Body Cookies Headers (7) Test Results

Pretty Raw Preview JSON

```
1 {
2   "fieldCount": 0,
3   "affectedRows": 1,
4   "insertId": 14,
5   "info": "",
6   "serverStatus": 2,
7   "warningStatus": 0,
8   "changedRows": 0
9 }
```

```
1 [
2   {
3     "idArviointi": 14,
4     "idOpintojakso": 1,
5     "idOpiskelija": 4,
6     "arvosana": 3,
7     "paivamaara": "2025-02-19T22:00:00.000Z"
8   }
9 ]
```

PUT-metodi:

PUT localhost:3000/arviointi/14

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings

☐ none ☐ form-data ☒ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

	Key	Value
<input checked="" type="checkbox"/>	idOpintojakso	2
<input checked="" type="checkbox"/>	idOpiskelija	4
<input checked="" type="checkbox"/>	arvosana	3
<input checked="" type="checkbox"/>	paivamaara	2025-02-20

Body Cookies Headers (7) Test Results

Pretty Raw Preview JSON

```
1 {  
2   "fieldCount": 0,  
3   "affectedRows": 1,  
4   "insertId": 0,  
5   "info": "Rows matched: 1 Changed: 1 Warnings: 0",  
6   "serverStatus": 2,  
7   "warningStatus": 0,  
8   "changedRows": 1  
9 }
```

Opintojakso vaihdettu 1->2

```
1 [  
2   {  
3     "idArviointi": 14,  
4     "idOpintojakso": 2,  
5     "idOpiskelija": 4,  
6     "arvosana": 3,  
7     "paivamaara": "2025-02-19T22:00:00.000Z"  
8   }  
9 ]
```

DELETE-metodi:

DELETE localhost:3000/arviointi/14

Params Authorization Headers (9) Body ● Pre-request Script Tests

Query Params

Key
Key

Body Cookies Headers (7) Test Results

Pretty Raw Preview JSON ▾

```
1 {
2   "fieldCount": 0,
3   "affectedRows": 1,
4   "insertId": 0,
5   "info": "",
6   "serverStatus": 2,
7   "warningStatus": 0,
8   "changedRows": 0
9 }
```

idArviointi 14 poistettu:

GET localhost:3000/arviointi/14

Params Authorization Headers (9) Body ● Pre-request Script Tests

Query Params

Key
Key

Body Cookies Headers (7) Test Results

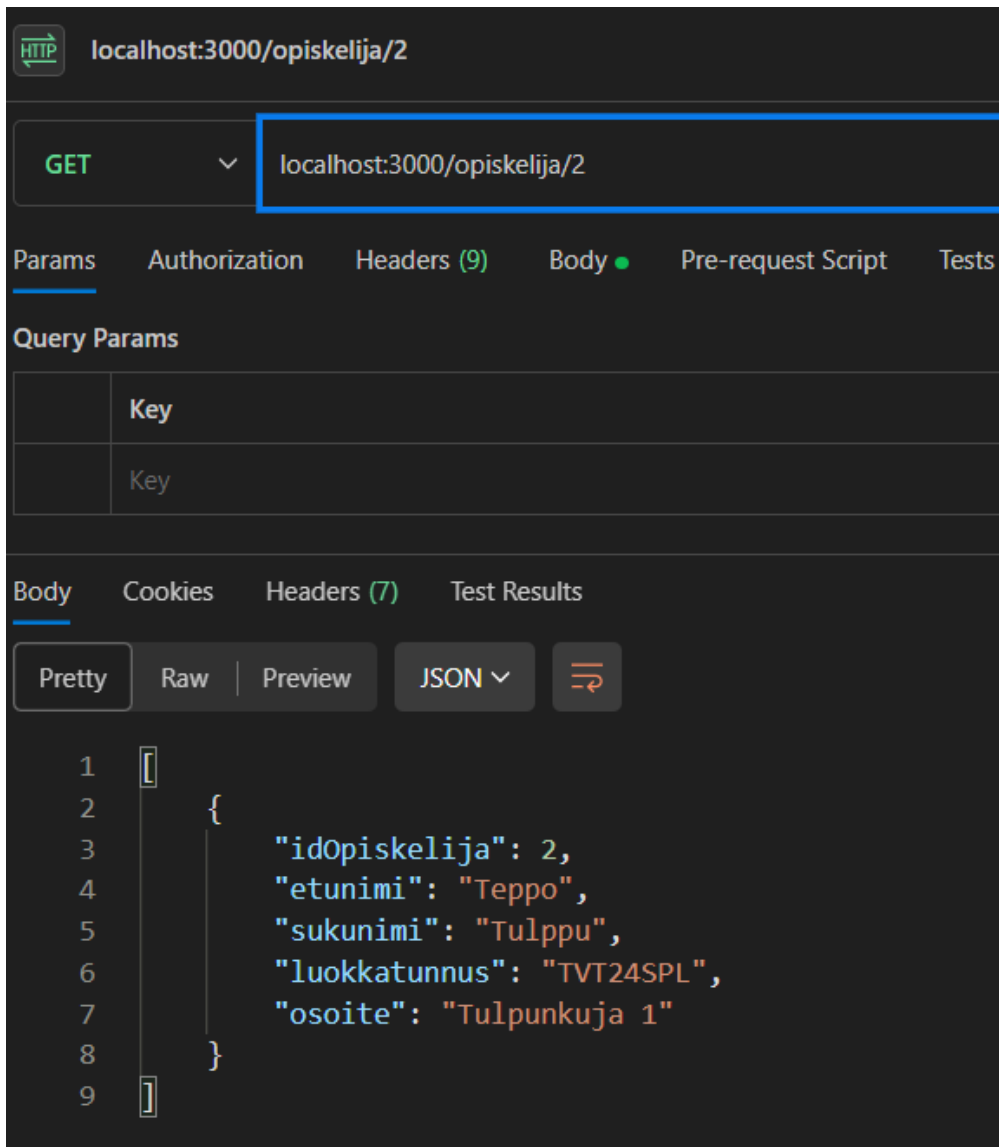
Pretty Raw Preview JSON ▾

```
1 []
```

Jotta documentista ei tule järjettömän pitkä, jätän screenshotit .js tiedostoista pois. Olen kuitenkin ne tehnyt (muutenhan seuraavat CRUD menetelmähän eivät toimisi).

Opiskelijataulu

GET-metodi:



The screenshot shows a web browser's developer tools interface. At the top, the address bar shows `localhost:3000/opiskelija/2`. Below it, the **GET** method is selected, and the URL `localhost:3000/opiskelija/2` is entered. The **Params** tab is active, showing a table with two rows, both labeled **Key**. Below the tabs, the **Body** tab is selected, showing a JSON response in **Pretty** format. The response is an array containing one object with the following properties:

```
[
  {
    "idOpiskelija": 2,
    "etunimi": "Teppo",
    "sukunimi": "Tulppu",
    "luokkatunnus": "TVT24SPL",
    "osoite": "Tulpunkuja 1"
  }
]
```

POST-metodi:

POST localhost:3000/opiskelija/

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings

☐ none ☐ form-data ☒ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

	Key	Value
<input checked="" type="checkbox"/>	etunimi	Milla
<input checked="" type="checkbox"/>	sukunimi	Magia
<input checked="" type="checkbox"/>	luokkatunnus	TVT24SPL
<input checked="" type="checkbox"/>	osoite	Vuoritie 1

Body Cookies Headers (7) Test Results

Pretty Raw Preview JSON

```
1 {
2   "fieldCount": 0,
3   "affectedRows": 1,
4   "insertId": 7,
5   "info": "",
6   "serverStatus": 2,
7   "warningStatus": 0,
8   "changedRows": 0
9 }
```

```
1 [
2   {
3     "idOpiskelija": 7,
4     "etunimi": "Milla",
5     "sukunimi": "Magia",
6     "luokkatunnus": "TVT24SPL",
7     "osoite": "Vuoritie 1"
8   }
9 ]
```


PUT-metodi:

PUT localhost:3000/opiskelija/7

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings

☐ none ☐ form-data ☒ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

	Key	Value
<input checked="" type="checkbox"/>	etunimi	Milla
<input checked="" type="checkbox"/>	sukunimi	Magia
<input checked="" type="checkbox"/>	luokkatunnus	TVT24SPO
<input checked="" type="checkbox"/>	osoite	Vuoritie 1

Body Cookies Headers (7) Test Results

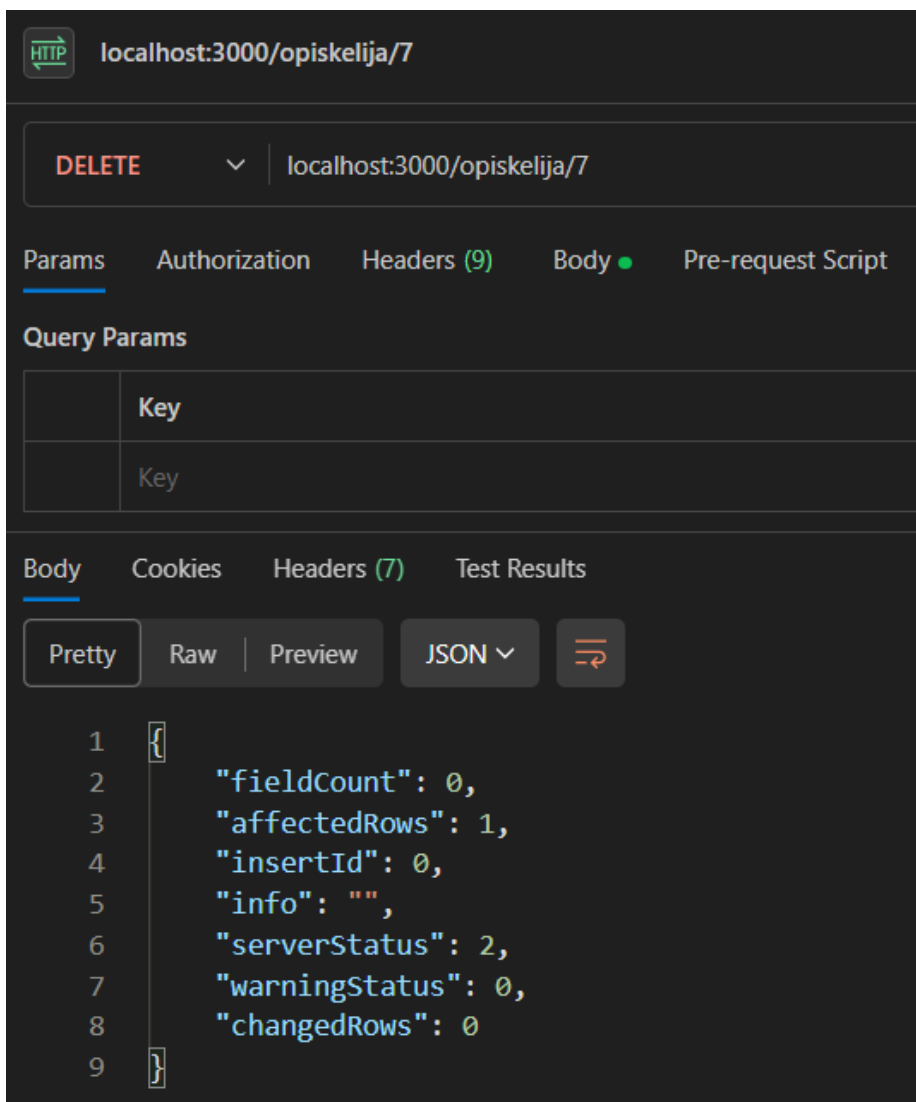
Pretty Raw Preview JSON

```
1 {
2   "fieldCount": 0,
3   "affectedRows": 1,
4   "insertId": 0,
5   "info": "Rows matched: 1 Changed: 1 Warnings: 0",
6   "serverStatus": 2,
7   "warningStatus": 0,
8   "changedRows": 1
9 }
```

luokkatunnus vaihdettu TVT24SPL -> TVT24SPO

```
1 [
2   {
3     "idOpiskelija": 7,
4     "etunimi": "Milla",
5     "sukunimi": "Magia",
6     "luokkatunnus": "TVT24SPO",
7     "osoite": "Vuoritie 1"
8   }
9 ]
```

DELETE-metodi:



localhost:3000/opiskelija/7

DELETE localhost:3000/opiskelija/7

Params Authorization Headers (9) Body ● Pre-request Script

Query Params

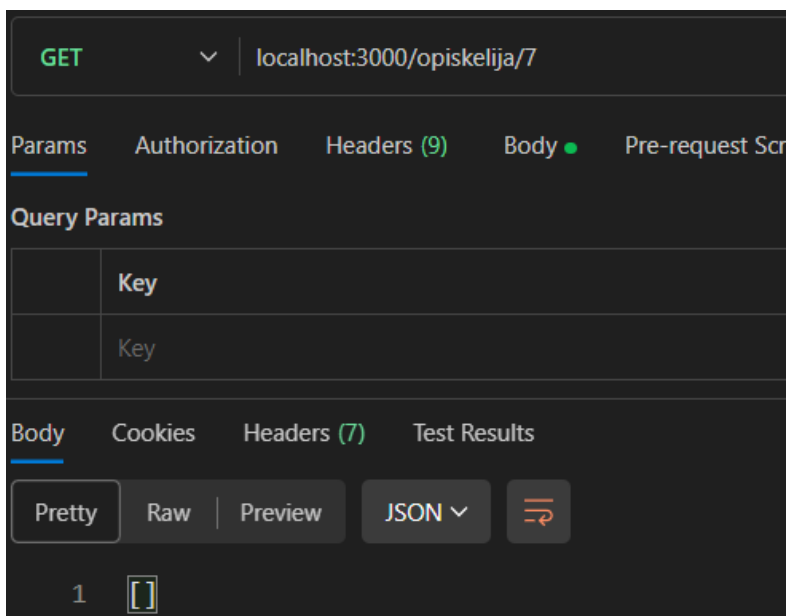
	Key
	Key

Body Cookies Headers (7) Test Results

Pretty Raw Preview JSON ↕

```
1 {
2   "fieldCount": 0,
3   "affectedRows": 1,
4   "insertId": 0,
5   "info": "",
6   "serverStatus": 2,
7   "warningStatus": 0,
8   "changedRows": 0
9 }
```

idOpiskelija 7 poistettu:



GET localhost:3000/opiskelija/7

Params Authorization Headers (9) Body ● Pre-request Script

Query Params

	Key
	Key

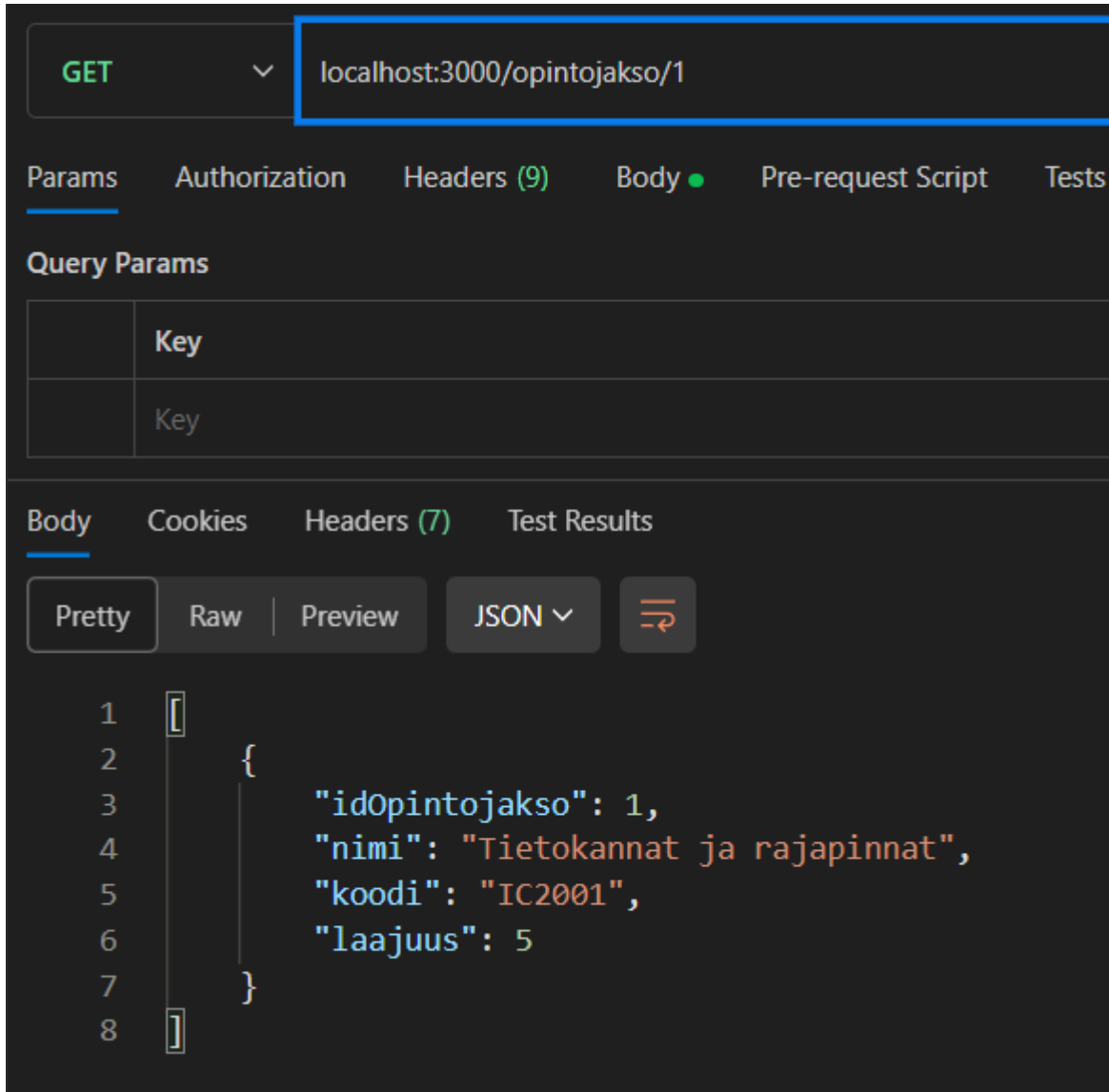
Body Cookies Headers (7) Test Results

Pretty Raw Preview JSON ↕

```
1 []
```

Opintojaksotaulu

GET-metodi:



The screenshot shows a REST client interface with the following components:

- Method:** GET
- URL:** localhost:3000/opintojakso/1
- Tabs:** Params, Authorization, Headers (9), Body ●, Pre-request Script, Tests
- Query Params:** A table with two rows, both labeled "Key".
- Body Tab:** Active, showing a JSON response.
- Body Viewers:** Pretty, Raw, Preview, JSON (selected), and a refresh icon.
- JSON Response:**

```
1 [
2   {
3     "idOpintojakso": 1,
4     "nimi": "Tietokannat ja rajapinnat",
5     "koodi": "IC2001",
6     "laajuus": 5
7   }
8 ]
```

POST-metodi:

POST localhost:3000/opintojakso/

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings

☐ none ☐ form-data ☒ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

	Key	Value
<input checked="" type="checkbox"/>	nimi	Robotiikan perusteet
<input checked="" type="checkbox"/>	koodi	IC2007
<input checked="" type="checkbox"/>	Laajuus	5

Body Cookies Headers (7) Test Results

Pretty Raw Preview JSON

```
1 {  
2   "fieldCount": 0,  
3   "affectedRows": 1,  
4   "insertId": 7,  
5   "info": "",  
6   "serverStatus": 2,  
7   "warningStatus": 0,  
8   "changedRows": 0  
9 }
```

```
1 [  
2   {  
3     "idOpintojakso": 7,  
4     "nimi": "Robotiikan perusteet",  
5     "koodi": "IC2007",  
6     "laajuus": 5  
7   }  
8 ]
```

PUT-metodi:

PUT localhost:3000/opintojakso/7

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings

☐ none ☐ form-data ☒ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

	Key	Value
<input checked="" type="checkbox"/>	nimi	Robotiikan perusteet
<input checked="" type="checkbox"/>	koodi	IC2007
<input checked="" type="checkbox"/>	laajuus	6

Body Cookies Headers (7) Test Results

Pretty Raw Preview JSON

```
1 {
2   "fieldCount": 0,
3   "affectedRows": 1,
4   "insertId": 0,
5   "info": "Rows matched: 1 Changed: 1 Warnings: 0",
6   "serverStatus": 2,
7   "warningStatus": 0,
8   "changedRows": 1
9 }
```

laajuus päivitetty 5 -> 6

```
1 [
2   {
3     "idOpintojakso": 7,
4     "nimi": "Robotiikan perusteet",
5     "koodi": "IC2007",
6     "laajuus": 6
7   }
8 ]
```

DELETE-metodi:

DELETE localhost:3000/opintojakso/7

Params Authorization Headers (9) Body ● Pre-request Script

Query Params

Key
Key

Body Cookies Headers (7) Test Results

Pretty Raw Preview JSON ▾

```
1 {
2   "fieldCount": 0,
3   "affectedRows": 1,
4   "insertId": 0,
5   "info": "",
6   "serverStatus": 2,
7   "warningStatus": 0,
8   "changedRows": 0
9 }
```

idOpintojakso 7 poistettu:

GET localhost:3000/opintojakso/7

Params Authorization Headers (9) Body ● Pre-request Script

Query Params

Key
Key

Body Cookies Headers (7) Test Results

Pretty Raw Preview JSON ▾

```
1 []
```