INFOSÜSTEEMIDE ARENDAMINE III - HAJUSRAKENDUSED

Loeng 7 – REST API dokumenteerimine, testimine ja LINQ, Lambda

Tarvo Treier
Tarkvarateaduse instituut

14.10.2024

PRAKTIKUMIS 1. ITERATSIOONI ESITLEMINE

- Kogu meeskond esitleb
- Esitlege valminud toodet
- Iga üks räägib, mis oli tema panus antud iteratsioonis
- Näidake projekti Wiki-t, Issues->Board-i ja Toggl-t
- Tutvustage järgmise iteratsiooni plaani
- Küsimused/kommentaarid teistelt meeskondadelt ja õppejõult

TAL TECH

- Punktid vastavalt „Projekti punktide jagunemine“ juhendile.

REST API DOKUMENTEERIMISE TÄHTSUS

- Kuna REST on pigem stiil kui standard, siis pole REST API-del sellist kohustuslikku kirjeldust nagu on SOAP API-del WSDL.

- Lisaks API loojatele, on oluline, et API-d oskaksid kasutada ka need, kes pole näinud API lähtekoodi.

TAL TECH

HEA API DOKUMENTATSIOON PAKUB:

- Selgitusi, mida meetod/ressurss teeb
- Olulist infot arendajale sh. tagastatavad tüübid, hoiatused ja veateated
- **Näidis väljakutse sõnumit**
- Kasutatavate parameetrite loetelu koos nende tüüpide, formaatide ja reeglitega (kohustuslikkus)
- **Näidis vastussõnumit**
- Interaktiivset proovi/testi väljakutsete võimalust

TAL TECH

- Viide: https://www.mulesoft.com/resources/api/guidelines-api-documentation

SWAGGER ON

- avatud lähtekoodiga tarkvara raamistik, mida kasutatakse REST API-de dokumenteerimisel.
- sarnane Javadoc-ga, mis suudab java koodi kommentaaride põhjal automaatselt dokumentatsiooni genereerida.
- Loodud aastal 2011 Tony Tam-i poolt.

TAL TECH

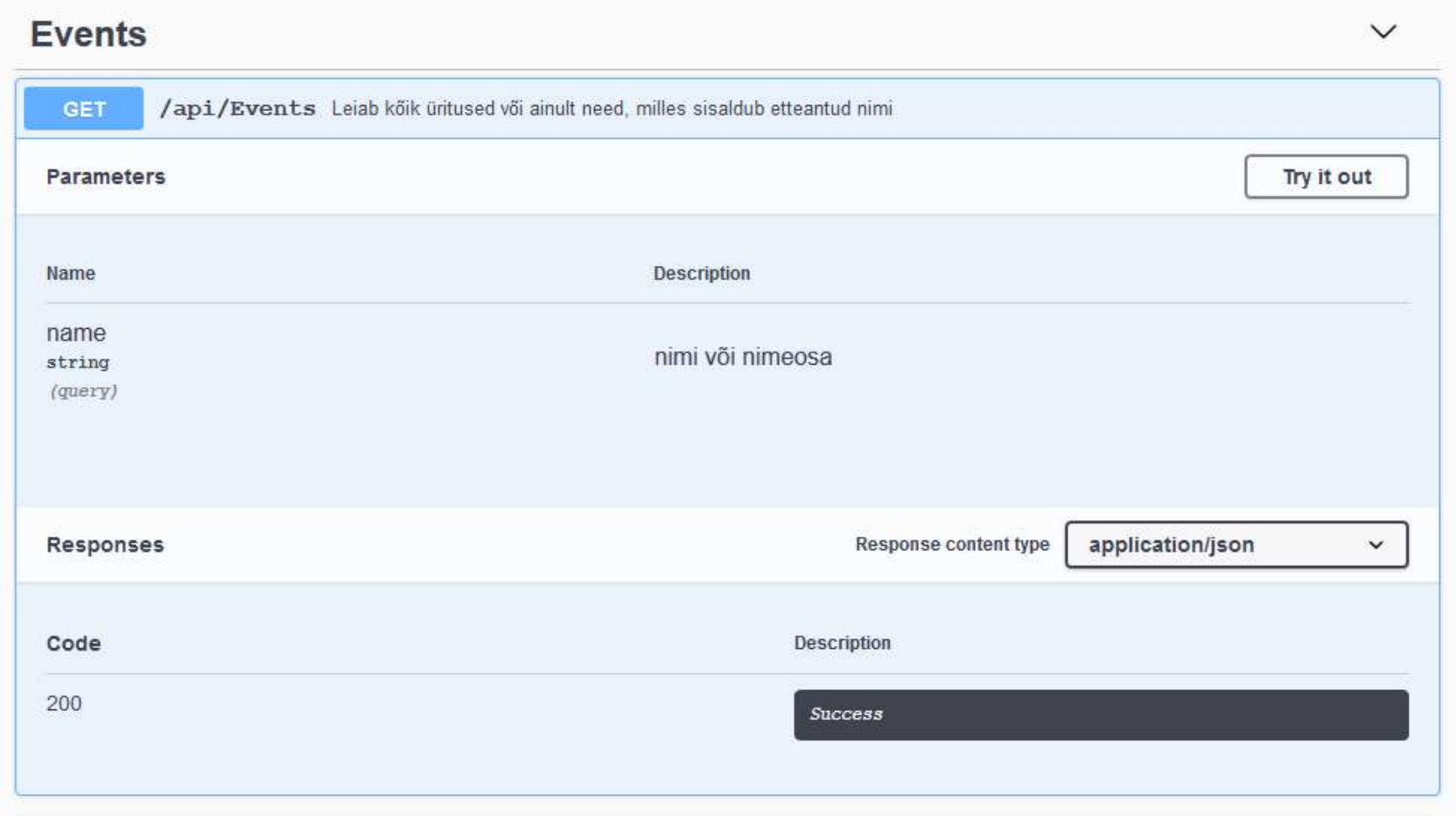
SWAGGERI LISAMINE PROJEKTI: KONTROLLERISSE ///

```
…
/// <summary>
/// Leiab kõik üritused või ainult need, milles sisaldub etteantud nimi
/// </summary>
/// <param name="name">nimi või nimeosa</param>
/// <returns>Ürituste nimekiri</returns>
// GET: api/<controller>
[HttpGet]
public IActionResult GetEvents([FromQuery]String name)
…
```

TAL TECH

- Viide: https://exceptionnotfound.net/adding-swagger-to-asp-net-core-web-api-using-xml-documentation/

SWAGGERI LISAMINE PROJEKTI: /// TULEMUS



TAL TECH

SWAGGERI LISAMINE PROJEKTI: /// VÕIMALIKUD ELEMENDID

- **summary**: A high-level summary of what the method/class/field is or does.
- **remarks**: Additional detail about the method/class/field.
- **param**: A parameter to the method, and what it represents.
- **returns**: A description of what the method returns.

- **MEESKONDLIKU PROJEKTI API tuleb dokumenteerida!!!**

TAL TECH

- Viide: https://exceptionnotfound.net/adding-swagger-to-asp-net-core-web-api-using-xml-documentation/

TESTIMINE

- Testitav objekt (funktsioon, API operatsioon...) - must kast

- Sisend
- Oodatav tulemus
- Tegelik tulemus

- **Mis nendest tuleb fikseerida enne testi läbiviimist?**

- Näide: Add(a, b)

TAL TECH

REST API TESTIMINE (FUNKTSIONAALNE)

Mis on

- testitav objekt,
- sisend?

Kuidas me saame

- oodatava tulemuse,
- tegeliku tulemuse?

TAL TECH

REST API TESTIMINE: MIDA TÄPSEMALT KONTROLLIDA?

Kontrolli kas

- Vastuse HTTP kood on korrektne
- Vastuse JSON-is nõutud väljad olemas, õiget tüüpi ja korrektse väärtusega
- Vastuse päises on nõutud väärtused olemas

- Positiivsed stsenaariumid + vabatahtlik sisend
- Negatiivsed stsenaariumid + korrektne/ebakorrektne sisend

TAL TECH

- Viide: https://www.sisense.com/blog/rest-api-testing-strategy-what-exactly-should-you-test/

REST API TESTIMINE: POSTMAN/NEWMAN

- https://www.guru99.com/postman-tutorial.html

TAL TECH

LANGUAGE INTEGRATED QUERY (LINQ) & LAMBDA

https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/concepts/linq/

https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/operators/lambda-expressions

Kasulik veebikursus:

https://www.codingame.com/playgrounds/213/using-c-linq---a-practical-overview/welcome

TAL TECH

MIKS LINQ?

https://www.tutorialsteacher.com/linq/why-linq

TAL TECH

NÄIDE ILMA LINQ-TA

```
Student[] studentArray = {
        new Student() { StudentID = 1, StudentName = "John", Age = 18 },
        new Student() { StudentID = 2, StudentName = "Steve",  Age = 21 },
        new Student() { StudentID = 3, StudentName = "Bill",  Age = 25 },
        new Student() { StudentID = 4, StudentName = "Ram" , Age = 20 },
        };
Student[] students = new Student[10];
int i = 0;
foreach (Student std in studentArray)
{
        if (std.Age > 12 && std.Age < 20)
        {
                students[i] = std;
                i++;
        }
}
```

TAL TECH

NÄIDE LINQ-GA

```
Student[] studentArray = {
            new Student() { StudentID = 1, StudentName = "John", age = 18 } ,
            new Student() { StudentID = 2, StudentName = "Steve",  age = 21 } ,
            new Student() { StudentID = 3, StudentName = "Bill",  age = 25 } ,
            new Student() { StudentID = 4, StudentName = "Ram" , age = 20 } ,
            };

// Use LINQ to find teenager students
Student[] teenAgerStudents = studentArray.Where(s => s.age > 12 && s.age < 20).ToArray();

// Use LINQ to find first student whose name is Bill
Student bill = studentArray.Where(s => s.StudentName == "Bill").FirstOrDefault();

// Use LINQ to find student whose StudentID is 5
Student student5 = studentArray.Where(s => s.StudentID == 5).FirstOrDefault();
```

TAL TECH

7. JA 8. NÄDALA ÜLESANNETE PÕHI GIT-S

https://gitlab.cs.ttu.ee/tarvo.treier/2023-itb2203-practice7

TAL TECH