# INFOSÜSTEEMIDE ARENDAMINE III - HAJUSRAKENDUSED

Loeng 2 – Veebiteenused, REST

Tarvo Treier
Tarkvarateaduse instituut

09.09.2024

## ARENDUSVAHENDID

- **Visual Studio Code**
- **.NET 8**
- **Postman**

Keskkonna paigaldamise juhend moodles.

- **NB! Palun paigaldage arendusvahendid enda arvutisse juba harjutustunniks!**

TAL TECH

## TÄNA KAVAS

- Sissejuhatus
- Liides
- API
- Veebiteenus
- REST
- JSON

TAL TECH

## SISSEJUHATUS: INTEGREERIMISE VÕIMALUSED

Tooge näiteid, kuidas on võimalik kahte rakendust või nende komponente omavahel suhtlema/andmeid vahetama panna.

TAL TECH

## LIIDES: NÄITED

- View-d, c# interface, ..
- Palun tooge liidese näiteid, mis pole otseselt seotud programmeerimisega
- Punase noole näide

TAL TECH

# API (APPLICATION PROGRAMMING INTERFACE)

TAL TECH

## VEEBITEENUSED: SISSEJUHATUS

- Maailmas on palju programmeerimiskeeli, milles saab rakendusi kirjutada.
- Vahel on vaja need rakendused omavahel rääkima panna.
- Siinkohal võivad osutuda heaks valikuks veebiteenused.

TAL TECH

## VEEBITEENUS…

…tähendab programmide omavahelist suhtlemist ja andmevahetust üle hariliku veebi.

TAL TECH

Viide: T.Tammet, Veebiteenuste võlu ja valu

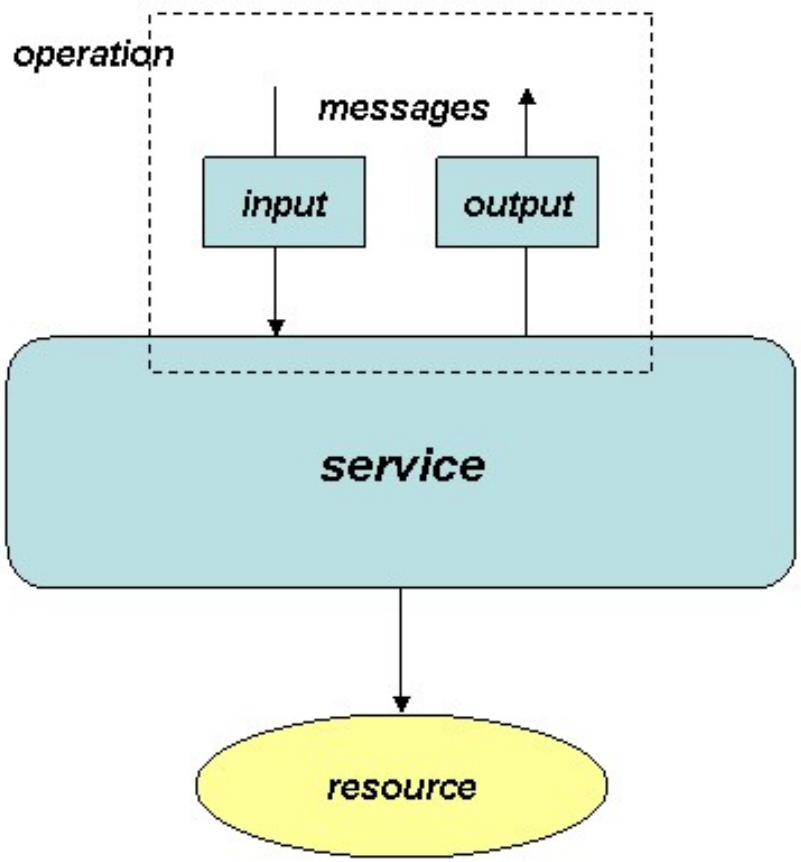
## VEEBITEENUS: DEFINITSIOON

- Veebiteenus on üle veebi (http) välja kutsutav(ad) / käivitatav(ad) meetod(id).
- Veebiteenust kutsutakse välja mingis kindlas formaadis sõnumiga ja vastus saadakse samuti sõnumina.
- Sarnaselt tavaliste funktsioonidega saab ka veebiteenuse väljakutsel määrata sisendparameetreid.

Viide: http://www.saasbook.info/

TAL TECH

# VEEBITEENUS: OPERATSIOONID JA SÕNUMID




TAL TECH

- Viide : http://msdn.microsoft.com/en-us/library/ms996486.aspx

## VEEBITEENUS: EELISED

- Erinevate platvormide rakenduste koostöö võimaldamine
- Teksti põhised ja avatud standardid on arendajale arusaadavad (**ISO8583** näide)
- Annavad võimaluse erinevate ettevõtete erinevas kohas asuvaid rakendusi ja teenuseid integreerida üheks uueks teenuseks
- Veebiteenuste taaskasutamise võimalus
- Erinevad kasutajaliidesed ühel kesksüsteemil
- Kliendi ja teenusepakkuja sõltumatu arendus
- Varjatud realisatsioon

TAL TECH

## VEEBITEENUS: TUVASTAMISE MEETODID

- Alt ülesse
- Ülevalt alla

## VEEBITEENUS: TÜÜBID

- Olemiteenus (entity)
  - Esindab ühte või mitut äriolemit. CRUD operatsioonid.
  - Näiteks CustomerAccount võib vajada juba teiste osapooltega suhtlemist ja pole enam lihtne olemiteenus.
- Funktsionaalne teenus
  - Tehnoloogiale orienteeritud teenus (mitte ärile).
  - Abiteenused, mida teised saavad kasutada(logimine, emaili saatmine...)
- Protsessiteenus
  - Esindab teenust, kus on terve seeria omavahel seotud ülesandeid (teenuseid).

TAL TECH

Viide: http://www.saasbook.info/

## VEEBITEENUS: MODELLEERIMINE

- Üldistamine
  - Klient on inimene ja töötaja on inimene
- Dekomponeerimine
  - Mida väiksemad tükid, seda suurem on tn taaskasutada
    - Näiteks: Kui ühel protsessil on vaja kliendi andmeid koos krediidikontrolliga ja teisel ilma, siis tuleks eraldi teha teenus kliendi andmete küsimiseks ja krediidikontrolliks.

TAL TECH

# REST (REPRESENTATIONAL STATE TRANSFER)

- REST is *an architecture style* for designing networked applications.

- The idea is that, rather than using complex mechanisms such as CORBA, RPC or SOAP to connect between machines, simple HTTP is used to make calls between machines.

- Viide: http://rest.elkstein.org/

TAL TECH

## REST: SPETSIFIKATSIOON

- Pole spetsifikatsiooni
- On ainult parimad praktikad ja soovitused

TAL TECH

## REST: TEKKELUGU

- REST-i defineeris 2000 aastal oma doktoritöös Roy T. Fielding.
- Roy T. Fielding on HTTP ja URI standardite kaasautor.

TAL TECH

## REST VS WEB

- REST doesn’t build on the principles of the Web—the Web was built based on RESTful principles. They just weren’t so named until a few years later.
- The idea of REST is essentially a reverse-engineering of how the Web works. HTTP itself, and URIs themselves, are written with REST principles.

TAL TECH

Viide: http://www.saasbook.info/

## REST: RESURSS

- Resources are the key abstractions in REST.
- They are the remote accessible objects of the application (a Web site, an HTML page, an XML document, a Web service, a physical device, *etc.*).
- A resource is a unit of identification.
- Everything that might be accessed or be manipulated remotely could be a resource.
  - http://soacookbook.com/customers
  - http://soacookbook.com/customers/1234
  - http://soacookbook.com/orders/456/customer

TAL TECH

Viide: http://www.saasbook.info/

## WEB DESIGN: AXIOM 0

- **Axiom 0: Universality 1**
  - Any resource anywhere can be given a URI
- **Axiom 0a: Universality 2**
  - Any resource of significance should be given a URI.

TAL TECH

- Viide: https://www.w3.org/DesignIssues/Axioms.html (Tim Berners-Lee)

## REST: PÕHIMÕTTED

- REST services are stateless
  - no cookies; Cache-ability is important too, especially for GETs.
- REST services have a uniform interface
  - Interface is provided by the standard HTTP methods (PUT, GET,POST, DELETE).
- Resources are manipulated through representations
  - The components in the system exchange data that represents the resource.
    - JSON
    - XML
    - XHTML
    - JPEG image

Viide: http://www.saasbook.info/

TAL TECH

## REST: URI NÄIDE (QUERYING A PHONEBOOK)

- Hea näide:
  - **http://www.acme.com/phonebook/UserDetails/12345**
- Halb näide:
  - http://www.acme.com/phonebook/getUserDetails?id=12345
- Veel halvem näide☺
  - http://www.acme.com/phonebook/user12345.xml

TAL TECH

Viide: http://rest.elkstein.org/

## REST: SOOVITUS 1

- GET access requests should never cause a state change. Anything that changes the server state should be a POST request (or other HTTP verbs, such as DELETE)

TAL TECH

Viide: http://rest.elkstein.org/

## REST: SOOVITUS 2

- Queries should not return an overload of data.
- If needed, provide a paging mechanism. For example, a "product list" GET request should return the first n products (e.g., the first 10), with next/prev links.

TAL TECH

Viide: http://rest.elkstein.org/

## REST: SOOVITUS 3

- Even though the REST response can be anything, make sure it's well documented, and do not change the output format lightly (since it will break existing clients).
- Remember, even if the output is human-readable, your clients aren't human users.
- If the output is in XML, make sure you document it with a schema.

TAL TECH

Viide: http://rest.elkstein.org/

## REST: SOOVITUS 4

- Rather than letting clients construct URLs for additional actions, include the actual URLs with REST responses. For example, a "product list" request could return an ID per product, and the specification says that you should use http://www.acme.com/product/PRODUCT_ID to get additional details. That's bad design. Rather, the response should include the actual URL with each item: http://www.acme.com/product/001263, etc.
- Yes, this means that the output is larger. But it also means that you can easily direct clients to new URLs as needed, without requiring a change in client code.

Viide: http://rest.elkstein.org/

TAL TECH

## REST: LIGIPÄÄSU PIIRAMINE

- http/s autentimise kasutamine
- access_token
  - teenusepakkuja lahendus (kasutaja küsib teenusepakkujalt access tokeni)
    - JSON Web Token (JWT)
  - kolmanda osapoole kaudu ( kasutades OAuth, OpenID vms lahendust)

TAL TECH

## JSON (JAVASCRIPT OBJECT NOTATION)

- Nimi-väärtus paaride kollektsioon

```
{„nimi“:“Juhan“
, „vanus“: 21
}
```

TAL TECH

# JSON: NÄITED

```
{"firstName":"John", "lastName":"Doe"}
```

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

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

https://www.w3schools.com/whatis/whatis_json.asp

## JSON: VALIDEERIMINE/KONVERTEERIMINE

- https://jsonformatter.org/
- https://jsonformatter.curiousconcept.com/

TAL TECH

## URI MALLID RESSURSSIDELE SUUNAMISEKS

- /api/[ressurss]/{id}/[alamressurss]

- Näiteks: /api/people/{id}/cars
- Valikuline väärtus {id?}
- Vaikimisi väärtus{id = 0}
- piirang: kohustuslik int tüüpi väärtus {id:int}

TAL TECH

# MEETODITE SISENDPARAMEETRID JA NENDE VALIDEERIMINE

- Meetodi sisendparameetritena kasutatakse päringu andmeid
  - https://learn.microsoft.com/en-us/aspnet/core/mvc/models/model-binding
- Vaikimisi otsitakse sobilikke parameetreid järgmises järjekorras (saab muuta):
  1. Form fields
  2. Request body
  3. Route data
  4. Query string parameters
  5. Uploaded files
- Kui soovitakse kasutada päringu sisuks olevat JSON-it, siis tuleb kasutada vastava päringu juures [FromBody]
- Piiranguid seatakse kasutades Data Annotationit
  - https://learn.microsoft.com/en-us/aspnet/core/tutorials/first-mvc-app/validation

TAL TECH

## VASTUSSÕNUMI STAATUSE KOODID

- 200 – ok, kõik läks hästi
  - 201 – lisamine õnnestus ja uue ressurssi URI
  - 204 – muutmine õnnestus
- 400 – päringu/kliendi viga (bad syntax)
  - 401 – pole õigusi selle ressurssi nägemiseks
  - 404 – ressurssi ei leitud
- 500 – serveri probleem. Sisemine viga.

TAL TECH

Viide: https://www.restapitutorial.com/httpstatuscodes.html

# REST TEENUSE REALISEERIMINE: DEMO

- Juhendid on moodle.taltech.ee lehel üldiste juhendite all
  - REST-i realiseerimise juhendid

Viide: https://learn.microsoft.com/en-us/aspnet/core/tutorials/first-web-api?tabs=visual-studio-code

- https://docs.microsoft.com/en-us/azure/architecture/best-practices/api-design

- https://restfulapi.net/resource-naming/