

PROJEKAT 1 - IoT Smart-building

Cilj projekta 1 je kreiranje tri servisa, u različitim tehnologijama, koji implementiraju sledeće API-je: REST, gRPC i GraphQL.

Izabrani skup podataka je Smart-building-system sa sledećeg link <https://www.kaggle.com/datasets/ranakrc/smart-building-system>

SmartBuilding_REST

SmartBuilding_REST je prvi servis koji implementira REST API. Odabrana tehnologija za REST servis je Node.js

Neophodni koraci:

```
> cd SmartBuilding_REST
> npm install
> node index.js
```

Kako bi proverili metode koristimo Postman za testiranje. Kreiranjem novog HTTP Request, odabirom odgovarajućeg metoda (GET/POST/PUT/DELETE) i unosom rute do metode u kontroleru (iz */SmartBuilding_RET/index.js*) proveravamo implementirane metode.

SmartBuilding_gRPC

SmartBuilding_gRPC je drugi servis koji implementira gRPC. Odabrana tehnologija u kojoj je implementiran gRPC servis je .NET Core.

Odgovarajući proto fajl za gRPC servis se nalazi na sledećoj putanji
/SmartBuilding_gRPC/Protos/room.proto

Neophodni koraci:

```
> dotnet build
```

```
> dotnet run
```

Nakon što je servis pokrenut moguće je testirati ga u okviru Postmana. Kreiramo nov gRPC Request, unosimo URL servera i dodajemo definiciju za učitavanje servisa i metoda. Kao opciju za definiciju za učitavanje servisa i metoda biramo *Import a .proto file* i dodajemo odgovarajući proto fajl sa putanje koja je navedena u ovom uputstvu.

SmartBuilding_GraphQL

SmartBuilding_GraphQL je poslednji servis u okviru ovog projekta i implementira GraphQL. Odabrana tehnologija u kojoj je implementiran navedeni servis je .NET Core tako da je postupak pokretanja sličan kao i kod prethodno navedenog servisa.

Neophodni koraci:

```
> dotnet build
```

```
> dotnet run
```

Nakon što je servis pokrenut moguće je testirati ga u okviru Postmana. Kreiramo nov GraphQL Request, unosimo URL servera (sa sufiksom / *graphql*) i unosimo željeni upit. Odgovor dobijamo u Body delu u JSON formatu.

Format upita je sledeći:

```
query {  
  sensorValues(order: {notedData:<uređenje>},  
Where: {uslov[ugnježdeni uslovi...]*}){  
    parametar, [listaParametra]*  
  }  
}
```

e.g.

```
query {  
  sensorValues(order: {notedData:ASC},  
Where: {and:{temperature:{gte:25}, humidity:{gte:10}}}{  
    id,  
    temperature,  
    humidity  
  }  
}
```