

<https://github.com/prokoks121/VacationService11.git>

Branch: Master

# Data Import Service

PORT: 8123

Svaki End point ima key autentifikaciju, ono sto je potrebno podesiti za svaki poziv jeste

http://localhost:8123/upload/employee

POST	http://localhost:8123/upload/employee					
Params	Authorization	Headers (10)	Body	Pre-request Script	Tests	Settings
Headers 9 hidden						
	KEY	VALUE				
<input checked="" type="checkbox"/>	api_key	123123				
	Key	Value				

Unutar hedera proslediti kljuc "api\_key" uz vrednost kljuca "123123"

## 1.Unos zaposlenih

POST: /upload/employee

<http://localhost:8123/upload/employee>

Potrebno je proslediti csv fajl sa zaposlenima

http://localhost:8123/upload/employee

POST	http://localhost:8123/upload/employee					
Params	Authorization	Headers (10)	Body	Pre-request Script	Tests	Settings
none form-data x-www-form-urlencoded raw binary GraphQL						
	KEY	VALUE				
<input checked="" type="checkbox"/>	file	File	employee_profiles.csv			
		Value				

This file isn't in your working directory. Teammates you share this request with won't be able to use this file. To make collaboration easier you can setup your working directory in Settings.

Unutar body-a treba proslediti fajl sa sadrzajem zaposlenih pod kljucem "file"

## 2.Unos koriscenog odmora

POST: /upload/used-vacation

<http://localhost:8123/upload/used-vacation>

Isto kao I za Unos zaposlenih samo sto prosledjujemo fajl sa sadrzajem koriscenog odmora

## 3.Unos broja dana godisnjeg odmora

POST: /upload/vacation

<http://localhost:8123/upload/vacation>

Isto kao I za Unos zaposlenih samo sto prosledjujemo fajl sa sadrzajem koriscenog odmora

# Data Search Service

PORT: 8223

Svaki End point ima basic autentifikaciju, tako da svaki zaposleni koji je unet u bazu podataka preko dataImportService/upload/employee moze da se prijavi I koristi end point

http://localhost:8223/api/employee/vacation

GET	http://localhost:8223/api/employee/vacation
-----	---------------------------------------------

Params Authorization Headers (8) Body Pre-request Script Tests Settings

Type Basic Auth

The authorization header will be automatically generated when you send the request. [Learn more about authorization](#)

Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, [Learn more about variables](#)

Username user2@rbl.rs

Password Abcl@#\$

☒ Show Password

Korisnik moze da vidi samo svoje podatke

# 1. Prikaz ukupnog broja slobodnih dana, iskoriscenih dana i preostalih slobodnih dana

GET: /api/employee/vacation

<http://localhost:8223/api/employee/vacation>

```
{
  "total_used_vacation": 53,
  "total_vacation": 63,
  "total_free_vacation": 10
}
```

Takodje mozemo da prosledimo paramtar (nije obavezan) "type" sa jednim od sledecih 3 vrednosti

- "total" – prikazuje samo `total_vacation`

- "totalUsed" – prikazuje samo `total_used_vacation`

- "totalFree" – prikazuje samo `total_free_vacation`

## 2. Prikaz koriscenog odmora

GET: /api/employee/vacation/used

<http://localhost:8223/api/employee/vacation/used>

<http://localhost:8223/api/employee/vacation/used?dateStart=2019-01-01&dateEnd=2019-12-30>

Obavezni parametric:

"dateStart" – od kog datuma se vrši pretraga

"dateEnd" – do kog datuma se vrši pretraga

Obavezan format za oba parametra je "yyyy-MM-dd" tj 2019-12-30

```
[
  {
    "id": 100,
    "employeeEmail": "user1@rbt.rs",
    "vacationStart": "2019-08-30",
    "vacationEnd": "2019-09-11"
  }
]
```

```

    },
    {
      "id": 101,
      "employeeEmail": "user1@rbt.rs",
      "vacationStart": "2019-10-24",
      "vacationEnd": "2019-10-24"
    }
  ]

```

### 3. Dodavanje koriscenog odmora

POST: /api/employee/vacation/used

<http://localhost:8223/api/employee/vacation/used>

potrebno je unutar body-a proslediti json u format

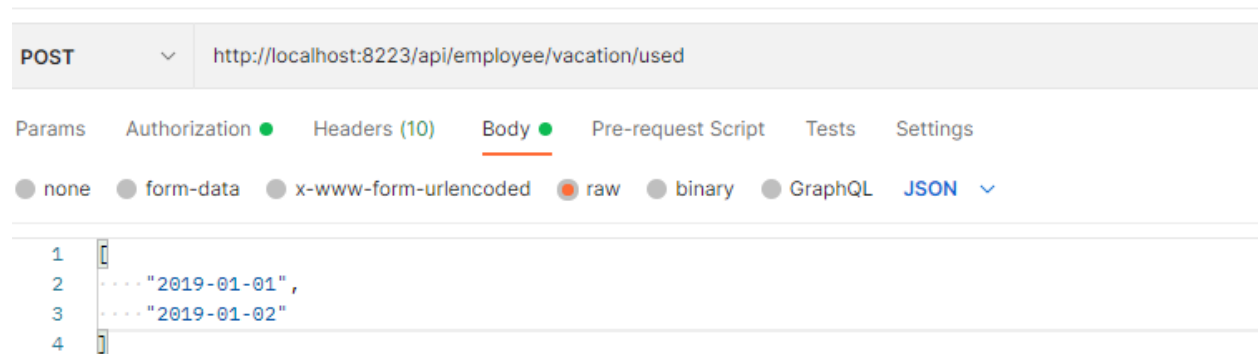
```

[
  "2019-01-01",
  "2019-01-02"
]

```

Oba parametra su String u format "yyyy-MM-dd"

<http://localhost:8223/api/employee/vacation/used>



Izlaz:

```

{
  "response": {
    "status": "CREATED",
    "message": "Successfully added used vacation"
  },
  "used_vacation": {
    "id": 789,

```

```
        "employeeEmail": "user1@rbt.rs",
        "vacationStart": "2019-01-01",
        "vacationEnd": "2019-01-02"
    }
}
```

# DOCKER

<https://github.com/prokoks121/VacationService11.git>

Branch: Docker

Projekat sa izmenjenim properties fajlovima, dockerfile I docker-compose.yml se nalazi na istom gitu ali pod Docker

Dockerfile za Data Import Service se nalazi unutar svog projekta

Dockerfile za Data Search Service se nalazi unutar sbog projekta

FROM openjdk:8

COPY /target/DataSearchService-0.0.1-SNAPSHOT.jar DataSearchService-0.0.1-SNAPSHOT.jar

EXPOSE 8223

ENTRYPOINT ["java", "-jar", "/DataSearchService-0.0.1-SNAPSHOT.jar"]

Docker-compose.yml se nalazi unutar projekta Data import Service

version: '3.1'

services:

data-import-service:

container\_name: data-import

image: data-import

build: ./

ports:

- "8123:8123"

depends\_on:

- dbpostgresql

app:

container\_name: app-springboot-postgresql

image: app-springboot-postgresql

build: ../DataSearchService

ports:

- "8223:8223"

depends\_on:

- dbpostgresql

dbpostgresql:

image: postgres

ports:

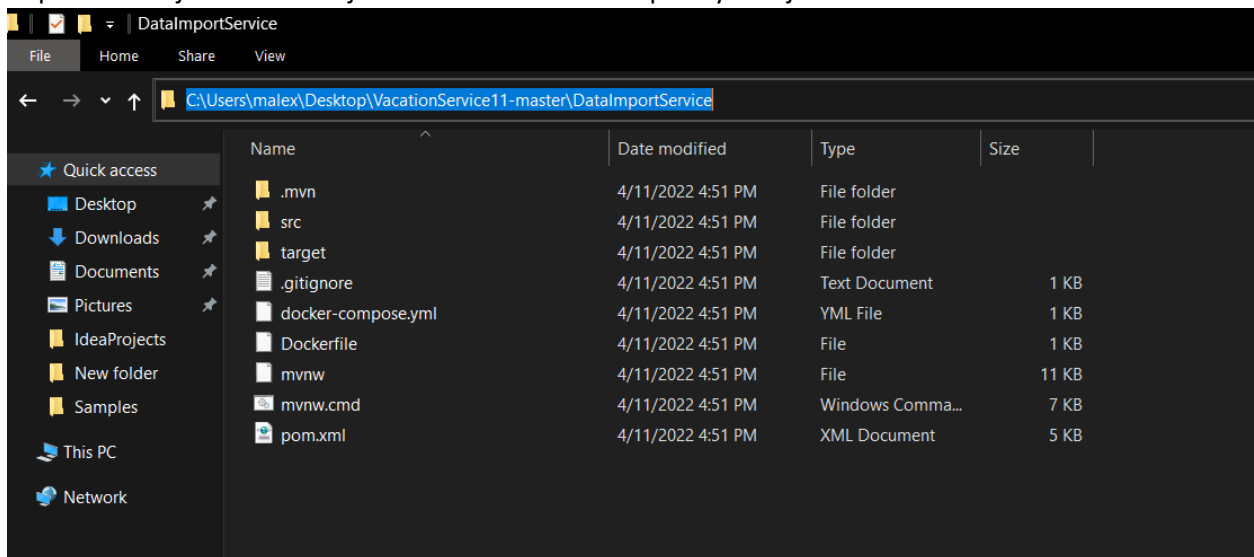
- "5432:5432"

environment:

- POSTGRES\_PASSWORD=password
- POSTGRES\_USER=postgres
- POSTGRES\_DB=vacation

## POKRETANJE PROJEKTA NA DOCKER

1. Pokrenuti Docker na racunaru
2. Preuzeti projekat sa GitHub (<https://github.com/prokoks121/VacationService11.git>)  
(Download as Zip)
3. Preuzeti fajl otpakovati na poznatu lokaciju (Desktop)
4. (Windows) pokrenuti PowerShell kao administrator
  - 4.1. Unutar widnws search kucati PowerShell
  - 4.2. Desni klik na PowerShell
  - 4.3. Run as administrator I zatim kliknite na Yes
5. Kopirati lokaciju foldera u kojem se nalazi docker-compose.yml fajl



Primer:

`C:\Users\malex\Desktop\VacationService11-master\DataImportService`

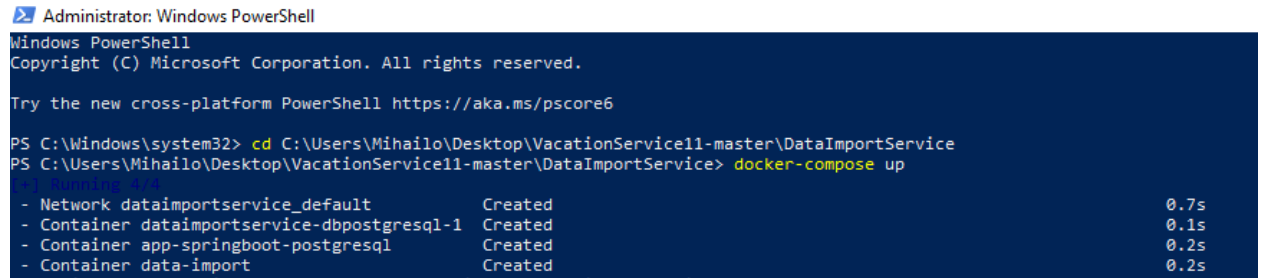
6. Unutar PowerShell kucati komandu koja ce vas prebaciti u direktorijum u kome se nalazi docker-compose.yml fajl

➤ `cd C:\Users\malex\Desktop\VacationService11-master\DataImportService`

tj. cd %VASA LOKACIJA FAJLA%

7. Unutar PowerShell kucati komandu koja ce pokrenuti docker-compose.yml fajl I kreirati nas kontenjer

➤ **docker-compose up**



```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Windows\system32> cd C:\Users\Mihailo\Desktop\VacationService11-master\DataImportService
PS C:\Users\Mihailo\Desktop\VacationService11-master\DataImportService> docker-compose up
[+] Running 4/4
 - Network dataimportservice_default      Created                                0.7s
 - Container dataimportservice-dbpostgresql-1 Created                        0.1s
 - Container app-springboot-postgresql    Created                              0.2s
 - Container data-import                  Created                              0.2s
```

Nakon nekoliko sekundi ce se sve pokrenuti I mozete pristupiti API

#### NAPOMENA

Data-import I data-search rade dok je pokrenut powershell, nakog gasenja powershell, sa njim se gase I ova dva kontenjera