

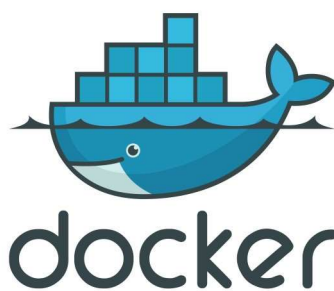
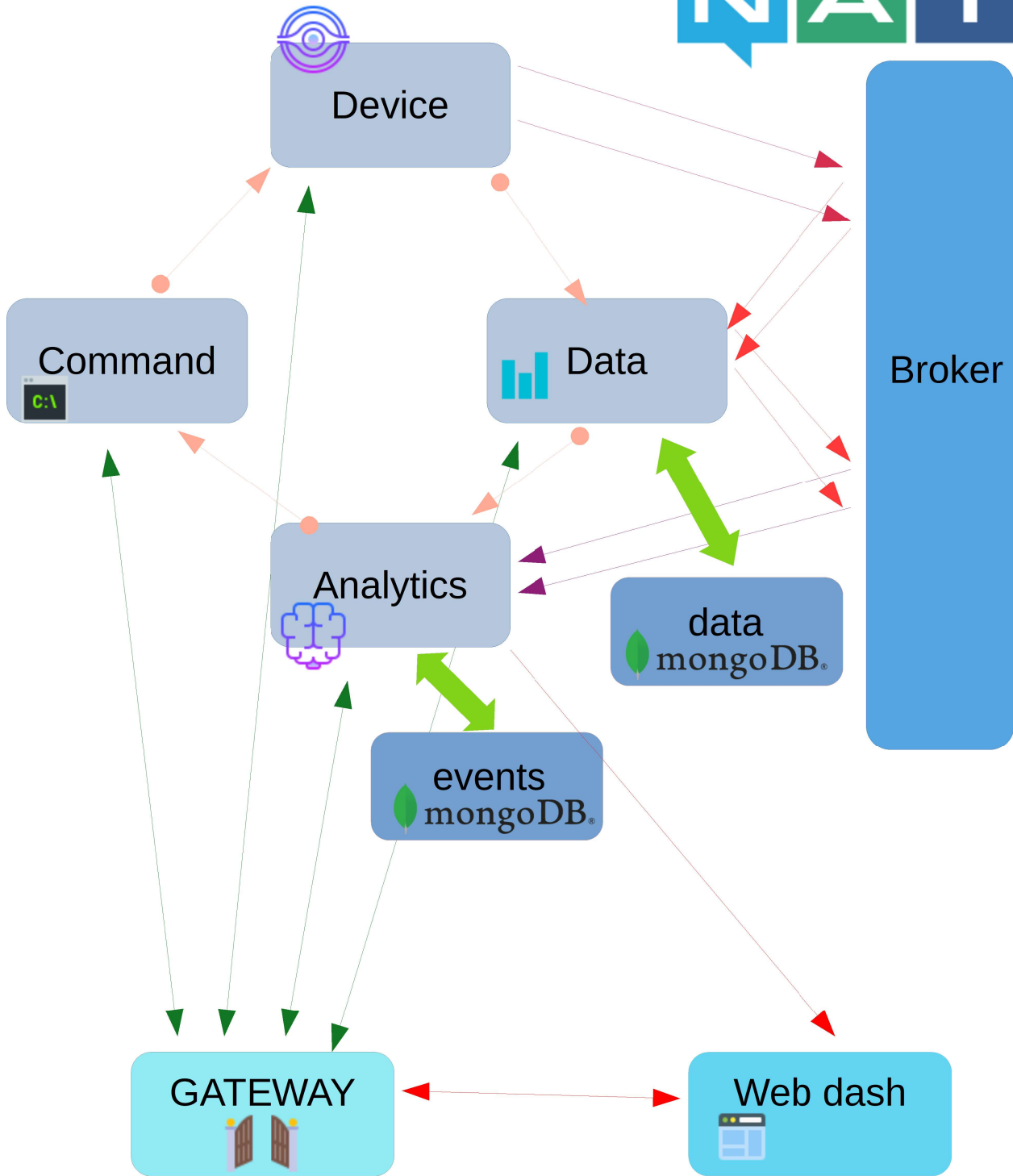
# Servisno orijentisane arhitekture

Student: Marija Stanković 16350

## Opis aplikacije

Aplikacija ilustruje implementiranje mikroservisne arhitekture koja se sastoji od *device*, *data*, *analytics*, *command*, *gateway* i *WEB dashboard* mikroservisa. Aplikacija nalazi primenu u sistemu gradskih bicikala koje se mogu iznajmiti na nekom od parkinga za bicikle a kasnije vratiti na neki drugi parking. Deo sistema kojim se aplikacija bavi je praćenje stanja na parkiralištima: koliko ima bicikli, praznih mesta i da li je negde došlo do kvara.

## Arhitektura



## Device mikroservis

Ovaj mikroservis onda prima naredbe od Command mikroservisa na osnovu kojih upravlja senzorima. Informacije o naredbama šalju se POST i PUT zahtevima: POST na */turnoff* isključuje stanicu, PUT na */full* govori stanici da ne prihvata nove bicikle, PUT na */empty* kaže da stanica ne može da izdaje nove bicikle. Testiranje preko *gateway* mikroservisa:

PUT http://localhost:4004/station/full

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

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

```
1 {
2   "stationId": 3
3 }
```

Body Cookies Headers (7) Test Results

Pretty Raw Preview Visualize HTML

1 ok

PUT http://localhost:4004/station/empty

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

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

```
1 {
2   "stationId": 3
3 }
```

Body Cookies Headers (7) Test Results

Pretty Raw Preview Visualize HTML

1 ok

Slanjem GET zahteva (bez dodatne putanje) dobija se lista senzora i njihovih statusa. Testiranje preko *gateway* mikroservisa:

GET
http://localhost:4004/devices

+
...

Untitled Request

GET

http://localhost:4004/devices

Params

Authorization

Headers (11)

Body

Pre-request Script

Tests

Settings

Query Params

KEY	VALUE
<input type="checkbox"/> event	"taken"
Key	Value

Body

Cookies

Headers (7)

Test Results

Pretty

Raw

Preview

Visualize

HTML

```

1 [{"id":4,"state":"Station empty"}, {"id":14,"state":"Station full"}, {"id":15,"state":"Station full"}, {"id":20,"state":"In
2 Service"}, {"id":21,"state":"In Service"}, {"id":22,"state":"In Service"}, {"id":23,"state":"In
3 Service"}, {"id":24,"state":"In Service"}, {"id":25,"state":"In Service"}, {"id":26,"state":"In
4 Service"}, {"id":27,"state":"In Service"}, {"id":28,"state":"In Service"}, {"id":29,"state":"In
5 Service"}, {"id":30,"state":"In Service"}, {"id":31,"state":"In Service"}, {"id":32,"state":"In
6 Service"}, {"id":33,"state":"In Service"}, {"id":34,"state":"In Service"}, {"id":35,"state":"In
7 Service"}, {"id":36,"state":"In Service"}, {"id":37,"state":"In Service"}, {"id":38,"state":"In
8 Service"}, {"id":39,"state":"In Service"}, {"id":40,"state":"In Service"}, {"id":41,"state":"In
9 Service"}, {"id":42,"state":"In Service"}, {"id":43,"state":"In Service"}, {"id":44,"state":"In
10 Service"}, {"id":45,"state":"In Service"}, {"id":46,"state":"In Service"}, {"id":47,"state":"In
11 Service"}, {"id":48,"state":"In Service"}, {"id":49,"state":"In Service"}, {"id":50,"state":"In
12 Service"}, {"id":51,"state":"In Service"}, {"id":52,"state":"In Service"}, {"id":53,"state":"In
13 Service"}, {"id":54,"state":"In Service"}, {"id":55,"state":"In Service"}, {"id":56,"state":"In
14 Service"}, {"id":57,"state":"In Service"}, {"id":58,"state":"In Service"}, {"id":59,"state":"In
15 Service"}, {"id":60,"state":"In Service"}, {"id":61,"state":"In Service"}, {"id":62,"state":"In
16 Service"}, {"id":66,"state":"In Service"}, {"id":67,"state":"In Service"}, {"id":68,"state":"In
17 Service"}, {"id":69,"state":"In Service"}, {"id":71,"state":"In Service"}, {"id":72,"state":"In
18 Service"}, {"id":73,"state":"In Service"}, {"id":74,"state":"In Service"}, {"id":75,"state":"In
19 Service"}, {"id":76,"state":"In Service"}, {"id":77,"state":"In Service"}, {"id":80,"state":"In
20 Service"}, {"id":81,"state":"In Service"}, {"id":84,"state":"In Service"}, {"id":85,"state":"In
21 Service"}, {"id":86,"state":"In Service"}, {"id":87,"state":"In Service"}, {"id":88,"state":"In
22 Service"}, {"id":89,"state":"In Service"}, {"id":90,"state":"In Service"}, {"id":91,"state":"In
23 Service"}, {"id":92,"state":"In Service"}, {"id":93,"state":"Station empty"}, {"id":94,"state":"In
24 Service"}, {"id":95,"state":"In Service"}, {"id":96,"state":"In Service"}, {"id":97,"state":"In
25 Service"}, {"id":98,"state":"In Service"}, {"id":99,"state":"In Service"}, {"id":100,"state":"In
26 Service"}, {"id":101,"state":"In Service"}, {"id":102,"state":"In Service"}, {"id":103,"state":"In
27 Service"}, {"id":106,"state":"In Service"}, {"id":107,"state":"In Service"}, {"id":108,"state":"Station
28 full"}, {"id":109,"state":"In Service"}, {"id":110,"state":"In Service"}, {"id":111,"state":"In
29 Service"}, {"id":112,"state":"In Service"}, {"id":113,"state":"In Service"}, {"id":114,"state":"In
30 Service"}, {"id":115,"state":"In Service"}, {"id":116,"state":"In Service"}, {"id":117,"state":"In
31 Service"}, {"id":118,"state":"In Service"}, {"id":119,"state":"In Service"}, {"id":120,"state":"In
32 Service"}, {"id":121,"state":"In Service"}, {"id":122,"state":"In Service"}, {"id":123,"state":"In

```

Find and Replace

Cancel

## Data mikroservis

Ovaj mikroservis je subscribe-ovan na **bike.taken** i **bike.returned**. Kada primi poruku na nekom od ovih kanala, poruku upisuje u bazu i dalje objavljuje preko kanala **bike.num.up** i **bike.num.down**. Poruke koje primi preko **PUT /fixed** i **PUT /broken** takođe upisuje u bazu i šalje Analytics mikroservisu na **ANALYTICS\_URL/fixed** tj. **ANALYTICS\_URL/brokendock**.

Ovaj mikroservis ima REST API GET za pretraživanje podataka u bazi: **/raw** i **/rawinrange**. Parametri za pretraživanje podataka su opcioni: *stationId*, *event* i u prvom slučaju *time*, u drugom slučaju *time1* i *time2* za pretraživanje informacija za određeni vremenski period. Testiranje preko *gateway* mikroservisa:

[illegible][illegible]

Ovaj mikroservis je subscribe-ovan na **bike.num.up** i **bike.num.down**. Poruke može da primi i preko PUT/fixed i PUT/brokendock. Kada primi poruku, proverava stanje stanice u bazi, update-uje ga i proverava da li je došlo do nekog ključnog događaja:

Ovaj mikroservis je subscribe-ovan na **bike.num.up** i **bike.num.down**. Poruke može da primi i preko PUT/fixed i PUT/brokendock. Kada primi poruku, proverava stanje stanice u bazi, update-uje ga i proverava da li je došlo do nekog ključnog događaja:

```

if (station.docks_available == station.docks - 1)
{
  console.log("stanica prazna " + station.stationId);
  request.put(process.env.COMMAND_URL + "empty", {
    json: {
      stationId: station.stationId
    }
  }, (err, res, body) => {
    if (err) {
      console.log(err);
      return;
    }
    console.log(res.statusCode);
    console.log(body);
  });
}
let res = await this.adapter.updateById(station._id, { $inc: { docks_available: 1}});

```

Mogući događaji su: stanica nema više bicikli (prazna), stanica nema više mesta za parking (puna), stanica u kvaru, stanica radi ili da je došlo do greške. U skladu sa detektovanim događajem šalje se POST ili PUT zahtev Command mikroservisu.

Analytic mikroservis ima GET API */events* preko koga se dobijaju update-ovi stanja sistema.

## Command mikroservis

REST API za primanje parametara naredbi: POST */error*, POST */brokenstation* i PUT */full*, PUT */empty*. Svaka od ovih komandi očekuje stationId kao parametar. Naredbe koje izdaje, a koje prima Device mikroservis su za isključivanje stanice POST *DEVICE\_URL/turnoff*, stanica je puna PUT *DEVICE\_URL/full* i stanica je prazna PUT *DEVICE\_URL/empty*. Testirane preko *gateway* mikroservisa:

POST

http://localhost:4004/command/error

Params

Authorization

Headers (9)

Body

Pre-request Script

Tests

Settings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

1

2

3

"stationId":3

Body

Cookies

Headers (7)

Test Results

Pretty

Raw

Preview

Visualize

HTML

1 [2,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,80,81,82,83,84,85,86,87,88,89,90,91,92,93,94,95,96,97,98,99,100,101,102,103,104,105,140,141,142,143,144,145,146,147,148,149,150,151,152,153,154,155,156,157,158,159,160,195,196,197,198,199,200,201,202,203,204,205,206,207,208,209,210,211,212,213,214,215,250,251,252,253,254,255,256,257,258,259,260,261,262,263,264,265,266,267,268,269,270,305,306,307,308,309,310,311,312,313,314,315,316,317,318,319,320,321,322,323,324,325,360,361,362,363,364,365,366,367,368,369,370,371,372,373,374,375,376,377,378,379,380,415,416,417,418,419,420,421,422,423,424,425,426,427,428,429,430,431,432,433,434,435,470,471,472,473,474,475,476,477,478,479,480,481,482,483,484,485,486,487,488,489,490,525,526,527,528,529,530,531,532,533,534,535,536,537,538,539,540,541,542,543,544,545,580,581,582,583,584,585,586,587,588,589,590,591,592,593,594,595,596,597,598,599]

POST

http://localhost:4004/command/brokenstation

Params

Authorization

Headers (9)

Body

Pre-request Script

Tests

Settings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

1

2

3

"stationId":2

Body

Cookies

Headers (7)

Test Results

Pretty

Raw

Preview

Visualize

HTML

1 [4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,81,82,83,84,85,86,87,88,89,90,91,92,93,94,95,96,97,98,99,100,101,102,103,1,141,142,143,144,145,146,147,148,149,150,151,152,153,154,155,156,157,158,15,196,197,198,199,200,201,202,203,204,205,206,207,208,209,210,211,212,213,21,251,252,253,254,255,256,257,258,259,260,261,262,263,264,265,266,267,268,26,306,307,308,309,310,311,312,313,314,315,316,317,318,319,320,321,322,323,32,361,362,363,364,365,366,367,368,369,370,371,372,373,374,375,376,377,378,37,416,417,418,419,420,421,422,423,424,425,426,427,428,429,430,431,432,433,43,471,472,473,474,475,476,477,478,479,480,481,482,483,484,485,486,487,488,48,526,527,528,529,530,531,532,533,534,535,536,537,538,539,540,541,542,543,54,581,582,583,584,585,586,587,588,589,590,591,592,593,594,595,596,597,598,59




PUT http://localhost:4004/command/fullstation

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

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

```
1 {
2   "stationId":4
3 }
```

Body Cookies Headers (7) Test Results

Pretty Raw Preview Visualize HTML 

1 ok

PUT http://localhost:4004/command/emptystation

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

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

```
1 {
2   "stationId":4
3 }
```

Body Cookies Headers (7) Test Results

Pretty Raw Preview Visualize HTML 

1 ok

Slanjem GET zahteva na /cmd dobija se lista i objašnjenje naredbi:

```
GET http://localhost:4004/commands

Params Authorization Headers (9) Body ● Pre-request Script Tests Settings
● none ● form-data ● x-www-form-urlencoded ● raw ● binary ● GraphQL JSON ▼

Body Cookies Headers (7) Test Results
Pretty Raw Preview Visualize JSON ↺

2  {
3      "req type": "POST",
4      "path": "/error",
5      "params": [
6          "stationId",
7          "errorCode"
8      ],
9      "paramTypes": [
10         "string",
11         "string"
12     ],
13     "description": "sensor reports that a bike is taken/returned from a station t
14 },
15 {
16     "req type": "POST",
17     "path": "/notfixed",
18     "params": [
19         "stationId"
20     ],
21     "paramTypes": [
22         "string"
23     ],
24     "description": "sensor reports that a station is fixed but it is not"
25 },
26 {
27     "req type": "POST",
28     "path": "/brokenstation",
29     "params": [
30         "stationId"
31     ],
32     "paramTypes": [
33         "string"
```

## Gateway mikroservis

Ovaj mikroservis nudi pristup svim ostalim mikroservisima:

```

methods: {
  initRoutes(app) {
    app.get("/devices", this.getDevices);
    app.get("/data/raw", this.getRawData);
    app.get("/data/indaterange", this.getRawDataInRange);
    app.get("/commands", this.getCommands);

    app.post("/station/turnoff", this.postTurnOffStation);
    app.post("/device/notFixed", this.postNotFixed);
    app.post("/command/error", this.postError);
    app.post("/command/notFixed", this.postCommandNotFixed);
    app.post("/command/brokenstation", this.postBrokenStation);

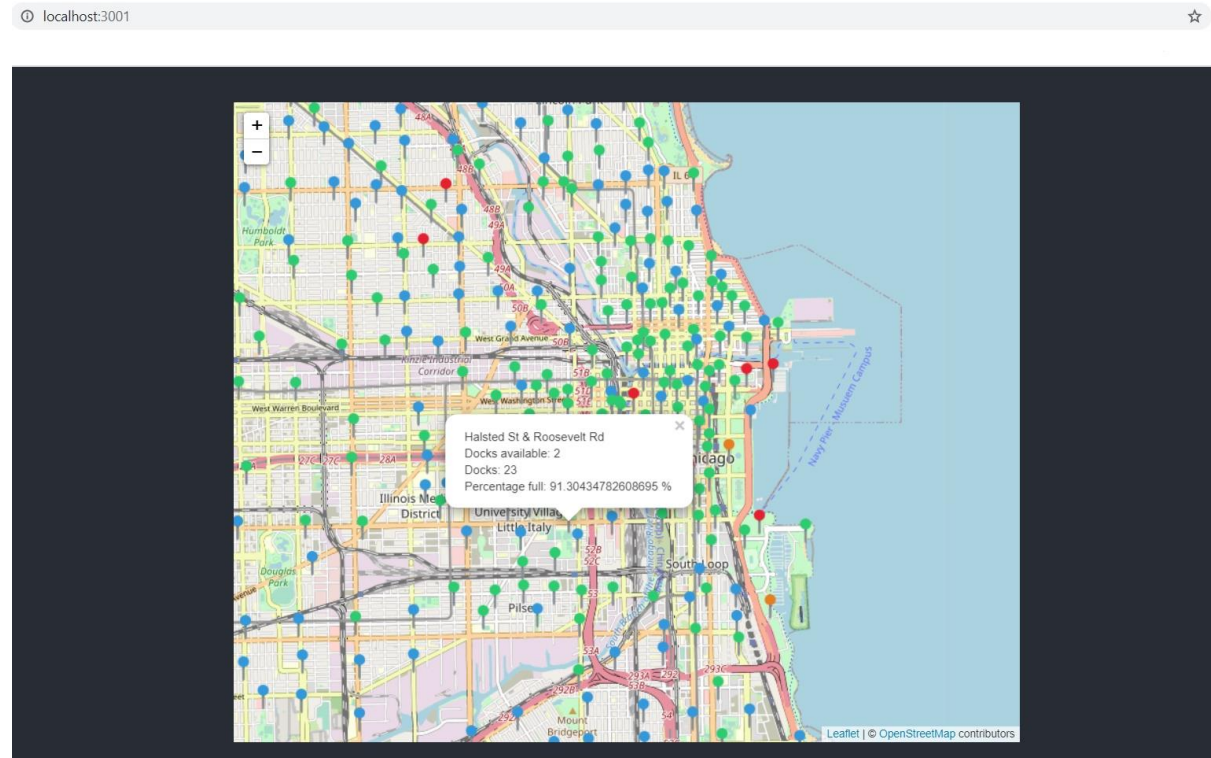
    app.put("/station/full", this.putFullStation);
    app.put("/station/empty", this.putEmptyStation);
    app.put("/data/fixedstation", this.putFixedData);
    app.put("/data/brokendock", this.putDataBrokenDock);
    app.put("/analytics/fixed", this.putAnalyticsFixed);
    app.put("/analytics/brokendock", this.putAnalyticsBrokenDock);
    app.put("/command/fullstation", this.putFullStationCommand);
    app.put("/command/emptystation", this.putEmptyStationCommand);
  },
}

```

Njegova uloga je da samo rutira zahteve do određenog mikroservisa i vrati odgovore. Ovaj mikroservis može da se koristi za testiranje.

## Web dashboard

Web dashboard na slikovit, čoveku razumljiviji način prikazuje stanje sistema:



Kada Analytics mikroservis detektuje događaj, šalje notifikaciju Web dashboard-u koji prikazuje novo stanje, pa je prikaz uvek validan.