

Používateľská príručka

Tím PARKety, č. 15

Vedúci projektu: Ing. Ivan Srba, PhD.

Predmet: Tímový projekt I

Ročník: 2017/2018

Vypracoval(i): Jakub Mičo, Marek Karas

Babinec Peter, Bc.	Hoang Martin, Bc.
Hučko Jakub, Bc.	Karas Marek, Bc.
Lehotský Miroslav, Bc.	Mičo Jakub, Bc.
Púčať Samuel, Bc.	Vnenčák Stanislav, Bc.

11. mája 2018

Obsah

1	Manipulácia s parkoviskami	1
1.1	Parkovisko	1
1.2	Parkovací box	4
1.3	Senzor	6
2	Manipulácia s parking providermi	7
3	Manipulácia s právami používateľov	7
4	Konfigurácia serverovej časti	8
5	Konfigurácia prezentačnej časti	9

Zoznam ukážok

1	Príklad volania pre vytvorenie parkoviska	1
2	Príklad metadata objektu	1
3	Príklad volania pre vytvorenie parkovacieho boxu	4
4	Príklad volania pre vymazanie parkovacieho boxu	6
5	Príklad volania pre vytvorenie senzora	6
6	Príklad volania pre vymazanie senzora	7
7	Príklad volania pre priradenie role používateľovi	8
8	Konfiguračný súbor development.json	8
9	Ukážka časti konfiguračného súboru uu5-environment.json	9
10	Príklad LSI objektu	10
11	Príklad Access Manager konfiguračného objektu	10

1 Manipulácia s parkoviskami

Nasledujúca sekcia opisuje vytvorenie základných databázových entít, menovite parkingLot, parkingBox a sensor. Všetky volania musia zodpovedať patričným validačným schémam, ktoré sú umiestnené v súbore *fiit_smartparking_main-server/app/validation_types/parking-lot-types.js*.

1.1 Parkovisko

Vytvorenia parkoviska

Zadefinovať parkovisko je možné pomocou POST requestu na adresu API endpointu s názvom *createParkingLot*, ktorý zodpovedá validačnej schéme *createParkingLotDtoInType*. Volanie je zložené z dvoch objektov *metadata* a *image*, ako ilustruje nasledujúci kód:

```
metadata: "Retazec_znakov",  
image: "binary"
```

Ukážka 1: Príklad volania pre vytvorenie parkoviska

Kde objekt metadata musí zodpovedať validačnej schéme *createParkingLot-MetadataType* a formát je nasledovný:

```
{  
  "name" : "FEI, Slovenska Technicka Univerzita",  
  "address" : "Litovska 14, 841 04 Karlova Ves, Slovensko",  
  "location" : {  
    "type" : "Point",  
    "coordinates" : [  
      48.1538234007923,  
      17.0741921865082  
    ]  
  },  
  "zoom" : 20.0,  
  "metadata" : {  
    "monitored" : true,  
    "charged" : true,  
    "handicapedBoxes" : true,  
    "electricCharger" : true,  
    "reserved" : true  
  },  
}
```

```

"parkingBoxes" : [
  {
    "name" : "A00",
    "occupance" : true,
    "sensors" : [
      {
        "sensorId" : "00001D",
        "lastChange" : ISODate("2018-05-06T15:08:38.962+0000"),
        "location" : {
          "lat" : 48.153907,
          "lon" : 17.074159
        },
        "value" : {
          "occupance" : true,
          "battery" : NumberInt(100),
          "signalRssi" : NumberInt(-115),
          "signalSnr" : NumberInt(-6)
        },
        "state" : "active",
        "stats" : {
          "totalParkingTime" : NumberInt(0),
          "totalVehicleCount" : NumberInt(1),
        }
      }
    ],
    "metadata" : {
      "handicapedBox" : true,
      "electricCharger" : false,
      "monitored" : false,
      "charged" : false,
      "reserved" : true
    },
    "parkingPolicy" : {
      "monday" : {
        "freeStayingTime" : NumberInt(5400000),
        "hourlyTariff" : 2.0,
        "chargedFrom" : NumberInt(8),
        "chargedTo" : NumberInt(17)
      },
      "tuesday" : {
        "freeStayingTime" : NumberInt(5400000),
        "hourlyTariff" : 2.0,
        "chargedFrom" : NumberInt(8),
        "chargedTo" : NumberInt(17)
      },
      "wednesday" : {
        "freeStayingTime" : NumberInt(5400000),
        "hourlyTariff" : 2.0,
        "chargedFrom" : NumberInt(8),
        "chargedTo" : NumberInt(17)
      }
    }
  }
]

```

```

    },
    "thursday" : {
        "freeStayingTime" : NumberInt(5400000),
        "hourlyTariff" : 2.0,
        "chargedFrom" : NumberInt(8),
        "chargedTo" : NumberInt(17)
    },
    "friday" : {
        "freeStayingTime" : NumberInt(5400000),
        "hourlyTariff" : 2.0,
        "chargedFrom" : NumberInt(8),
        "chargedTo" : NumberInt(17)
    },
    "saturday" : {
        "freeStayingTime" : NumberInt(0),
        "hourlyTariff" : 0.0,
        "chargedFrom" : NumberInt(8),
        "chargedTo" : NumberInt(17)
    },
    "sunday" : {
        "freeStayingTime" : NumberInt(0),
        "hourlyTariff" : 0.0,
        "chargedFrom" : NumberInt(8),
        "chargedTo" : NumberInt(17)
    }
}
},
"boundingBox" : {
    "topRight" : {
        "lat" : 48.154035,
        "lon" : 17.074454
    },
    "bottomLeft" : {
        "lat" : 48.15364,
        "lon" : 17.07396
    }
},
"parkingPolicy" : {...},
"parkingProviderUuid" : "21-7117-1"
}

```

Ukážka 2: Príklad metadata objektu

Objekt *parkingPolicy* na úrovni parkoviska zodpovedá rovnakého formátu ako *parkingPolicy* na úrovni parkovacieho boxu.

Objekt *image* je binary súbor obrázku pôdorysu parkoviska.

1.2 Parkovací box

Pridanie parkovacieho boxu

Pridať parkovací box do parkoviska je možné pomocou POST HTTP request-u na API endpoint-u */addParkingBox*. Vstupné argumenty musia zodpovedať validačnej schéme *addParkingBoxDtoInType*. Príklad volania pre vytvorenie parkoviska teda vyzerá nasledovne:

```
{
  "parkingLotId": ObjectId("5a13231151e4951b940ec8ab"),
  "parkingBox": {
    "occupance" : true,
    "sensors" : [
      {
        "sensorId" : "00001D",
        "lastChange" : ISODate("2018-05-06T15:08:38.962+0000"),
        "location" : {
          "lat" : 48.153907,
          "lon" : 17.074159
        },
        "value" : {
          "occupance" : true,
          "battery" : NumberInt(100),
          "signalRssi" : NumberInt(-115),
          "signalSnr" : NumberInt(-6)
        },
        "state" : "active",
        "stats" : {
          "totalParkingTime" : NumberInt(0),
          "totalVehicleCount" : NumberInt(1),
        }
      }
    ],
    "metadata" : {
      "handicapedBox" : true,
      "electricCharger" : false,
      "monitored" : false,
      "charged" : false,
      "reserved" : true
    },
    "parkingPolicy" : {
      "monday" : {
        "freeStayingTime" : NumberInt(5400000),
        "hourlyTariff" : 2.0,
        "chargedFrom" : NumberInt(8),
        "chargedTo" : NumberInt(17)
      }
    }
  }
}
```

```

    },
    "tuesday" : {
        "freeStayingTime" : NumberInt(5400000),
        "hourlyTariff" : 2.0,
        "chargedFrom" : NumberInt(8),
        "chargedTo" : NumberInt(17)
    },
    "wednesday" : {
        "freeStayingTime" : NumberInt(5400000),
        "hourlyTariff" : 2.0,
        "chargedFrom" : NumberInt(8),
        "chargedTo" : NumberInt(17)
    },
    "thursday" : {
        "freeStayingTime" : NumberInt(5400000),
        "hourlyTariff" : 2.0,
        "chargedFrom" : NumberInt(8),
        "chargedTo" : NumberInt(17)
    },
    "friday" : {
        "freeStayingTime" : NumberInt(5400000),
        "hourlyTariff" : 2.0,
        "chargedFrom" : NumberInt(8),
        "chargedTo" : NumberInt(17)
    },
    "saturday" : {
        "freeStayingTime" : NumberInt(0),
        "hourlyTariff" : 0.0,
        "chargedFrom" : NumberInt(8),
        "chargedTo" : NumberInt(17)
    },
    "sunday" : {
        "freeStayingTime" : NumberInt(0),
        "hourlyTariff" : 0.0,
        "chargedFrom" : NumberInt(8),
        "chargedTo" : NumberInt(17)
    }
},
"name" : "A00"
}
}

```

Ukážka 3: Príklad volania pre vytvorenie parkovacieho boxu

Vymazanie parkovacieho boxu

Vymazať parkovací box a všetky informácie vo vnútri je možné pomocou POST HTTP request-u na API endpoint `/removeParkingBox`. Ten vyžaduje dva povinné vstupné argumenty `parkingLotId` a `parkingBoxId`.

```
{
  "parkingLotId": ObjectId("5a13231151e4951b940ec8ab"),
  "parkingBoxId": ObjectId("5a2486afe4237a6db2c101e2"),
}
```

Ukážka 4: Príklad volania pre vymazanie parkovacieho boxu

1.3 Senzor

Pridanie senzora do parkovacieho boxu

Pridať senzor do parkovacieho boxu je možné pomocou POST HTTP request-u na endpoint `/addParkingLotSensor`. Toto volanie musí zodpovedať schéme s názvom `addParkingLotSensorDtoInType`, príklad volania vyzerá nasledovne:

```
{
  "parkingLotId": ObjectId("5a13231151e4951b940ec8ab"),
  "parkingBoxId": ObjectId("5a2486afe4237a6db2c101e2"),
  "sensor" : {
    "sensorId" : "00001M",
    "lastChange" : ISODate("2018-05-06T15:08:38.968+0000"),
    "location" : {
      "lat" : 48.153861,
      "lon" : 17.07416
    },
  },
  "value" : {
    "occupance" : false,
    "battery" : NumberInt(100),
    "signalRssi" : NumberInt(-115),
    "signalSnr" : NumberInt(-6)
  },
  "state" : "active",
  "stats" : {
    "totalParkingTime" : NumberInt(0),
    "totalVehicleCount" : NumberInt(0),
  },
}
```

```
}
```

Ukážka 5: Príklad volania pre vytvorenie senzora

Vymazanie senzora z parkovacieho boxu

Senzor je možné vymazať pomocou HTTP requestu na API endpoint `/remove-ParkingLotSensor`, toto volanie musí zodpovedať validačnej schéme s názvom `removeParkingLotSensorDtoInType`. Volanie musí obsahovať 3 povinné kľúče.

```
{
  "parkingLotId": ObjectId("5a13231151e4951b940ec8ab"),
  "parkingBoxId": ObjectId("5a2486afe4237a6db2c101e2"),
  "sensorId": "00001M"
}
```

Ukážka 6: Príklad volania pre vymazanie senzora

2 Manipulácia s parking providermi

3 Manipulácia s právami používateľov

Priradenie práv používateľovi

System pozná nasledovné role:

- *Authorities* - osoby zodpovedné za platformu
- *PrivilegedParkingProvider* - privilegovaný prevádzkovateľ parkoviska
- *ParkingProvider* - prevádzkovateľ parkoviska
- *Inspector* - kontrolné orgány
- *RegisteredUser* - registrovaný používateľ
- *SysOwner* - vlastník systému
- *Public* - široká verejnosť

Priradiť rolu používateľovi je možné pomocou HTTP requestu na endpoint `/sys/createPermission`, ktorého formát obsahuje názov role a `uuIdentity`. Volanie vyzerá nasledovne:

```
{
  "profileCode": "Authorities",
  "uuIdentityList": "20-7420-1"
}
```

Ukážka 7: Príklad volania pre priradenie role používateľovi

4 Konfigurácia serverovej časti

Štandardný konfiguračný súbor pre serverovú časť od Unicorn Systems bol doplnený o parameter `mqtt`, ktorý obsahuje pole `subscription`. Štandardne má každý z tenantov zadaný vlastný objekt pre `subscription`. Každý objekt v poli `subscriptions` obsahuje nasledovné parametre:

- `url` - jednoznačný identifikátor pre pripojenie k sprostredkovateľovi zariadení
- `apiKey` - kľúč pre autorizáciu
- `topic` - topic pre odoberanie MQTT správ
- `parkingProviderAwid` - AWID pre providera

```
{
  "uuSubAppDataStoreMap": {
    "primary": "mongodb://ParkingAdmin:team15fiit@147.175.149.211:8080/»
      » smartparking",
    "binary": "mongodb://ParkingAdmin:team15fiit@147.175.149.211:8080/»
      » smartparkingBinary?authSource=smartparking"
  },
  "privilegedUserMap": {
    "uuSubAppInstanceSysOwner": "20-7377-1"
  },
  "tid": "00000000000000000000000000000000",
  "asid": "00000000000000000000000000000001",
  "log_level": "DEBUG",
  "UuApp.AppClient.Client.log_level": "DEBUG",
}
```

```

"UuApp.AppServer": "DEBUG",
"mqtt": {
  "subscriptions": [
    {
      "url": "mqtt://fut.m2m.orange.com:1883",
      "apiKey": "e25ab350fd044dffbfad3b56e702326f",
      "topic": "fifo/parkingsensor1",
      "parkingProviderAwid": "3b45e8465263439d898828e00dbcb5d8"
    }
  ]
}
}
}

```

Ukážka 8: Konfiguračný súbor *development.json*

Okrem toho konfiguračný súbor obsahuje reťazce pripojenia vo formáte URI pre pripojenie do databázy MongoDB umiestnenej na školskom serveri (*primary* a *binary*).

5 Konfigurácia prezentačnej časti

Základná konfigurácia prezentačnej časti sa nachádza v súbore:

`uu5-environment.json`

a obsahuje zoznam podporovaných jazykov a reťazec *googleMapsApiKey*, ktorý je potrebný na prístup k funkcionalite systému. Podporované jazyky majú prispôsobiteľnú ikonu a zobrazovaný text. Nižšie je uvedený príklad konfiguračného súboru:

```

"languageList": {
  "sk": {
    "flag": "https://cdn.plus4u.net/uu-uu5g04/1.2.0/assets/flags/sk.png",
    "language": "Slovensky"
  },
  .....
},
"googleMapsApiKey": "api_key_string"

```

Ukážka 9: Ukážka časti konfiguračného súboru *uu5-environment.json*

Napriec systémom existuje viacero lokálnych konfiguračných súborov, ktoré používajú jednotlivé komponenty prezentačnej časti. Tieto konfiguračné súbory obsahujú *LSI* objekty, ktoré definujú reťazcové konštanty v rôznych

jazykoch. Tie slúžia na jednoduchú *internacionalizáciu* celého systému. Príklad takého objektu môžeme vidieť nižšie.

```
Cfg.LSILABEL_VEHICLE = {
  sk: "vozidlo",
  en: "vehicle"
};
```

Ukážka 10: Príklad LSI objektu

Ďalšou zložkou konfiguračných súborov sú objekty pre manažment prístupu. Tie konfigurujú prítomnosť jednotlivých komponentov na základe aktuálne prihláseného používateľa. Obsahujú *boolean* objekty *current* a *history*, ktoré definujú či sú komponenty prítomné v aktuálnom a historickom zobrazení parkoviska. Objekt *componentAccessProfileList*, obsahuje jednotlivé roly pre ktoré sa komponent zobrazí. Zoznam dostupných rolí je 3. Nižšie je uvedený príklad takéhoto objektu pre *label*.

```
Cfg.USERVISUALISATION_PL_LEFTPANEL = {
  label: {
    current: true,
    history: true,
    componentAccessProfileList: [
      "Public", "RegisteredUser",
      "Inspector",
      "ParkingProvider", "PrivilegedParkingProvider",
      "Authorities", "SysOwner"
    ],
  },
  ....
}
```

Ukážka 11: Príklad Access Manager konfiguračného objektu

V konfigurácii sú aj nasledovné konštanty:

```
Cfg.MAP_RELOAD_INTERVAL = 2000;
Cfg.ALLOWED_PARKING_TIME_MS = 7200000;
Cfg.CRITICAL_BATTERY_VALUE = 20;
Cfg.CRITICAL_RSSI_VALUE = -125;
Cfg.CRITICAL_SNR_VALUE = 2;

Cfg.SENSOR_SIZE = 1.5;
```

```
Cfg.SENSOR_SIZE_OCCUPIED = 1;  
Cfg.SENSOR_ZOOM_STREET_VIEW = 21;
```

Konštanta *MAP_RELOAD_INTERVAL* definuje čas v milisekundách kedy sa obnovujú údaje zobrazované na mape. Konšanta *ALLOWED_PARKING_TIME_MS* definuje štandardný čas povoleného parkovania zadarmo. Konštanta *CRITICAL_BATTERY_VALUE* definuje štandardnú hodnotu pri ktorej systém upozorňuje na kritický stav batérie. Podobne konštanty *CRITICAL_RSSI_VALUE* a *CRITICAL_SNR_VALUE* definujú kritické hodnoty signálu jednotlivých senzorov, na základe ktorých systém vytvára upozornenia.

Pomocou *SENSOR_SIZE* a *SENSOR_SIZE_OCCUPIED* je definovaná štandardná veľkosť vizualizácie voľných a obsadených senzorov. Konštanta *SENSOR_ZOOM_STREET_VIEW* definuje limit vzdalovania sa na mape, pri ktorom je používateľ presmerovaný z obrazovky s prehľadom konkrétneho parkoviska na obrazovku zoznamu parkovísk.

zoznam config suborov:

```
uu5-environment.json  
bricks/_config.js  
  _config-role-access.js  
  _config-map.js  
  _config-notifications.js  
  _config-parking-history.js  
  _config-parking-lot-map-widgets.js  
  _config-parking-lot-sensor-infowindow.js  
  _config-parking-lot-details-panel.js  
  _config-search-bar.js  
core/_config.js
```