**Title**: Establishments service

**Project description** : In the modern world, it is difficult to find only one place where you can see detailed information about different establishments. This web service allows you to find description for all types of establishments. If these establishments do not exist, the web service provides an opportunity to create your own type and interface of the establishment. It is also important that a person can update the rating of an institution with their voice using a unique token that the web service provides when the users authorize.  
  
**Scenarios** :  
**USE CASE 1**  
  
When a user wants to put a review on one of the establishments  
  
@JWTTokenNeeded

@POST

@Path("/update/{id}/{rating}")

Required arguments:  
int id  
int rating

Take a token (api/auth/login)

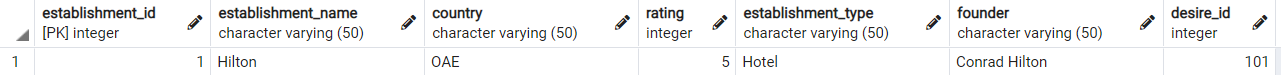
Put the token

Result Before:



Send (/api/establishments/update/1/5)

Result after:



**USE CASE 2**  
  
Client wants to see All establishments

@JWTTokenNeeded

@GET  
  
Required arguments:

Take a token (api/establishments/auth/login)

Put the token(api/establishments)

GET THE ALL RESULT FROM TABLE

[

    {

        "country": "Kazakhstan",

        "desire\_id": 103,

        "founder": "Unknown",

        "id": 0,

        "name": "Degirmen",

        "rating": 2,

        "type": "Cafe"

    },

    {

        "country": "Kazakhstan",

        "desire\_id": 102,

        "founder": "Unknown",

        "id": 0,

        "name": "Winox First",

        "rating": 4,

        "type": "FitnessCenter"

    },

    {

        "country": "Kazakhstan",

        "desire\_id": 102,

        "founder": "Jean Pierre",

        "id": 0,

        "name": "CrepeCafe",

        "rating": 3,

        "type": "Cafe"

    },

    {

        "country": "OAE",

        "desire\_id": 102,

        "founder": "Unknown",

        "id": 0,

        "name": "Fitness First",

        "rating": 10,

        "type": "FitnessCenter"

    },

    {

        "country": "Kazakhstan",

        "desire\_id": 102,

        "founder": "Unknown",

        "id": 0,

        "name": "Invictus Astana",

        "rating": 8,

        "type": "FitnessCenter"

    },

    {

        "country": "Kazakhstan",

        "desire\_id": 102,

        "founder": "Unknown",

        "id": 0,

        "name": "Fitness Palace",

        "rating": 10,

        "type": "FitnessCenter"

    },

    {

        "country": "OAE",

        "desire\_id": 101,

        "founder": "Conrad Hilton",

        "id": 0,

        "name": "Hilton",

        "rating": 5,

        "type": "Hotel"

    }

]

**USE CASE 3**  
  
Client wants to establishment by type of establishment

@JWTTokenNeeded

@POST

@Path("/{type}")

@Consumes(MediaType.APPLICATION\_JSON)

@Produces(MediaType.APPLICATION\_JSON)

Required arguments

establishment\_type

Take a token (api/establishments/auth/login)

Put the token(api/establishments)

Send (api/establishments/Hotel)

[

    {

        "country": "OAE",

        "desire\_id": 101,

        "founder": "Conrad Hilton",

        "id": 0,

        "name": "Hilton",

        "rating": 5,

        "type": "Hotel"

    }

]

**USE CASE 4**  
  
Client wants to find establishment by name

@JWTTokenNeeded

@GET

@Path("/{name}")

@Consumes(MediaType.APPLICATION\_JSON)

@Produces(MediaType.APPLICATION\_JSON)

Required arguments:

establishment\_type

Take a token (api/establishments/auth/login)

Put the token(api/establishments)

Send (api/establishments/Hotel)

**USE CASE 5**  
  
Client wants to find establishment by id

@JWTTokenNeeded

@GET

@Path("select/{id}")

Required arguments:

establishment\_id

Take a token (api/establishments/auth/login)

Put the token(api/establishments)

Send (api/establishments/1) and will give establishment by id

**USE CASE 6**  
  
Client wants to create establishment

@POST

@Path("/create/establishment")

Required arguments:

Establishment establishment

Take a token (api/establishments/auth/login)

Put the token(api/establishments)

Create new establishment

{

"id":6,

"name":"Fitness Palace",

"country":"Kazakhstan",

"rating" : 5,

"type" : "FitnessCenter",

"founder" : "Unknown",

"desire\_id" : "102"

}

Send (api/establishments/Hotel)

**USE CASE 7**  
  
Client wants to delete establishment by id

@DELETE

@Path("/{id}")

@Produces(MediaType.APPLICATION\_JSON)  
  
  
Required arguments:

Int id

Choose DELETE

Send (api/establishments/1) and will be deleted

**USE CASE 8**  
  
Client wants to sort establishment by rating

@JWTTokenNeeded

@GET

@Path("/sort/rating")

Required arguments:

Int rating

Take a token (api/establishments/auth/login)

Put the token(api/establishments)

Send (api/establishments/sort/5) and will be sorted by id from highest to lowest

**USE CASE 8**  
  
Client wants to update establishment by id

@POST

@Path("/update")

Required arguments:

Establishment establishment

{

"name":"Fitness Palace",

"country":"Kazakhstan",

"rating" : 5,

"type" : "FitnessCenter",

"founder" : "Unknown",

"desire\_id" : "102",

"id":6

}

Send (api/establishments/update) and will be updated by id

**USE CASE 9**  
  
Client wants to update type of establishment

@POST

@Path("/desire")

Required arguments:

Desire desire

{

"name":"Hotel",

"desire\_description" : "Lorem Lorem Lorem Lorem Lorem ",

"id":102

}

Send (api/establishments/desire) and will be updated by id

**USE CASE 10**  
  
Client wants to create new type of establishment

@POST

@Path("/create/desire")

Required arguments:

Desire desire

{

"id":102,

"name":"Hotel",

"desire\_description" : "Lorem Lorem Lorem Lorem Lorem "

}

Send (api/establishments/create/desire) and will be created

**USE CASE 11**  
  
Client wants to create new update establishment country by founder and id

@POST

@Path("/update/country")

Required arguments:

Establishment establishment

{

"country":"Kazakhstan",

"founder" : "Conrad Hilton",

"id":1,

}

Send (api/establishments/update/country) and will be updated

**USE CASE 12**  
  
Client wants to delete desire by id

@DELETE

@Path("/delete/desire/{id}")

@Produces(MediaType.APPLICATION\_JSON)

Required arguments:

desire\_id

Select DELETE in Postman

Write a path @Path("/delete/desire/{id}")

Send and will be deleted

**USE CASE 13**  
  
Client wants to update founder

@POST

@Path("/update/founder")

Required arguments:

Establisment establishment

{

"founder" : "Erasyl"

"id":5

}

Send @Path("/update/founder") and it will be updated to “Erasyl”