**MP1 Services**

The task was to implement the ‘services’ module defined in ETSI GS MEC 003.

Link do dokumentacji : <https://www.etsi.org/deliver/etsi_gs/MEC/001_099/011/01.01.01_60/gs_mec011v010101p.pdf>

Link do swagger UI: [https://forge.etsi.org/swagger/ui/?url=https://forge.etsi.org/gitlab/mec/gs011-app-enablement-api/raw/master/Mp1.yaml#/subscriptions](https://forge.etsi.org/swagger/ui/?url=https://forge.etsi.org/gitlab/mec/gs011-app-enablement-api/raw/master/Mp1.yaml)

API root : [http://<IP-OF-A-SERVER>:3000/mp1/v1/](http://%3Cip-of-a-server%3E:3000/mp1/v1/)

**Possible Queries:**

**GET –**/services



Query used to present all services. This method retrieves information about a list of meService resources. This method is typically used in "service availability query" procedure. There are three possible parameters.

|  |  |
| --- | --- |
| ser\_instance\_id | A mobile edge application instance may use multiple ser\_instance\_ids as an input parameter to query the availability of a list of mobile edge service instances. |
| ser\_name | A mobile edge application instance may use multiple ser\_names as an input parameter to query the availability of a list of mobile edge service instances. |
| ser\_category\_id | A mobile edge application instance may use ser\_category\_id as an input parameter to query the availability of a list of mobile edge service instances in a serCategory. |

Example Value of Query:



**Response Parameters**

|  |  |
| --- | --- |
| serInstanceId | Identifier of the service instance assigned by the MEPM / mobile edge platform. |
| serName | The name of the service. This is how the service producing mobile edge application identifies the service instance it produces. |
| serCategory | *This type represents the category reference* |
| version | Service version |
| state | Service State |
| transportInfo | *This type represents the general information of a mobile edge service.* |
| serializer | The enumeration SerializerTypes represents types of serializers |

**POST** -/services

This method is used to create a meService resource. This method is typically used in "service availability update and new service registration procedure.

**Parameters** **–** No parameters

**Request Body Example:**



**New Service Registration:**



**GET** -/services/{serviceId}

This method retrieves information about a meService resource. This method is typically used in "service availability query" procedure. It returns information about specific service, defined by the service ID defined in query.

**PUT** -/services/{serviceId}

This method updates the information about a specific meService resource. As specified in ETSI GS MEC 009 [5], the PUT HTTP method has "replace" semantics. With restriction about the IDs, which cannot be updated.

**API MODEL**



**Implementation**



The model I analogic to subscription module defined earlier. Here I am only using one new MongoDB query.

**Service.findOneAndUpdate({"\_id" : req.params.serviceId},{**

**serName : req.body.serName,**

**serCategory : req.body.serCategory,**

**version : req.body.version,**

**state : req.body.state,**

**transportInfo : req.body.transportInfo,**

**serializer : req.body.serializer}, {new: true})**

This query find element in data base by the ID and update listed information.

**Testing**

**GET**



**GET** with parameter



**GET** specific service



**POST** new service



**PUT** existing service



**MongoDB**

I use mongoDB. Which can be run typeing 'mongo' in cmd.