RedisInterface API Documentation

The purpose of the RedisInterface API is to allow the clients of the application to retrieve, insert and update data from/into the Redis storage. More precisely, the API is sending the request from the client to the Redis system and sending the Redis system response back to the client.

The endpoint classes that handle this functionality through HTTP Methods are the controller classes. When a request is made to manipulate the data, such as a request to retrieve data from the database, an HttpGet request will be executed and will return the data to the client.

The RedisInterface API allows querying the system for a particular data model, which is stored in JSON format, and querying for the relationships between the models, which will be stored in the GRAPH database.

To setup the environment please visit this link: <https://github.com/5G-ERA/middleware> , and follow the instructions from the Readme.md file. Once you have the environment ready, please select the RedisInterface project as your start-up project in Visual Studio (as can be seen in the image below) and run the application.

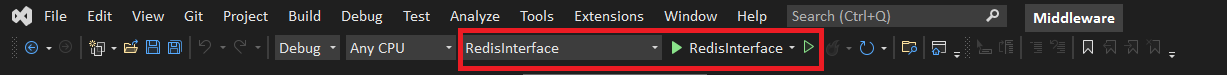


Figure 1 Visual Studio start-up project

Once the application is running, your browser will open the Swagger UI (as per the below image) where you will see the API endpoints which are implemented for the RedisInterface API.

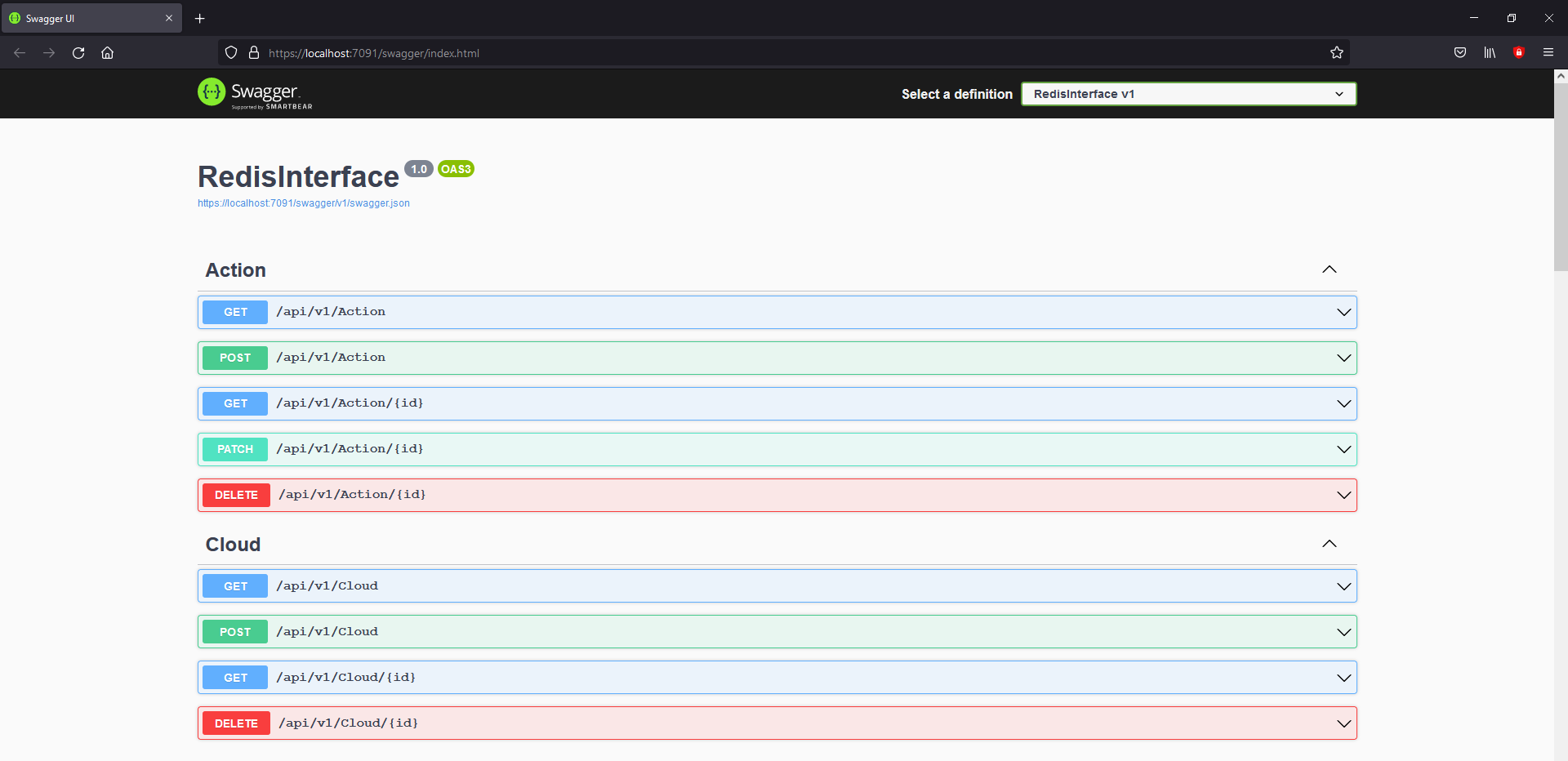


Figure 2 RedisInterface API endpoints

The CRUD operations which are implemented in the Action, Cloud, ContainerImage, Edge, Instance, Policy, Robot and Task controllers are described in the table below.

|  |  |  |
| --- | --- | --- |
| **Method** | **Request URL** | **Use Case** |
| GET | api/v1/Action | Get all the Action entities. |
| POST | api/v1/Action | Add a new Action entity. |
| GET | api/v1/Action/{id} | Get an Action entity by id. |
| PATCH | api/v1/Action/{id} | Partially update an existing Action entity. |
| DELETE | api/v1/Action/{id} | Delete an Action entity for the given id. |
| GET | api/v1/Cloud | Get all the Cloud entities. |
| POST | api/v1/Cloud | Add a new Cloud entity. |
| GET | api/v1/Cloud/{id} | Get a Cloud entity by id. |
| DELETE | api/v1/Cloud/{id} | Delete a Cloud entity for the given id. |
| GET | api/v1/ContainerImage | Get all the ContainerImage entities. |
| POST | api/v1/ContainerImage | Add a new ContainerImage entity. |
| GET | api/v1/ContainerImage/{id} | Get a ContainerImage entity by id. |
| PATCH | api/v1/ContainerImage/{id} | Partially update an existing ContainerImage entity. |
| DELETE | api/v1/ContainerImage/{id} | Delete a ContainerImage entity for the given id. |
| GET | api/v1/Edge | Get all the Edge entities. |
| POST | api/v1/Edge | Add a new Edge entity. |
| GET | api/v1/Edge/{id} | Get an Edge entity by id. |
| DELETE | api/v1/Edge/{id} | Delete an Edge entity for the given id. |
| GET | api/v1/Instance | Get all the Instance entities. |
| POST | api/v1/Instance | Add a new Instance entity. |
| GET | api/v1/Instance/{id} | Get an Instance entity by id. |
| PATCH | api/v1/Instance/{id} | Partially update an existing Instance entity. |
| DELETE | api/v1/Instance/{id} | Delete an Instance entity for the given id. |
| GET | api/v1/Policy/{id} | Get a Policy entity by id. |
| GET | api/v1/Policy | Get all the Policy entities. |
| GET | api/v1/Policy/current | Get active policies. |
| GET | api/v1/Robot | Get all the Robot entities. |
| POST | api/v1/Robot | Add a new Robot entity. |
| GET | api/v1/Robot/{id} | Get a Robot entity by id. |
| DELETE | api/v1/Robot/{id} | Delete a Robot entity for the given id. |
| GET | api/v1/Task | Get all the Task entities. |
| POST | api/v1/Task | Add a new Task entity. |
| GET | api/v1/Task/{id} | Get a Task entity by id. |
| DELETE | api/v1/Task/{id} | Delete a Task entity for the given id. |

Table 1 RedisInterface CRUD operations

To execute one of the above-mentioned methods, just click on the desired method, and then click on the “Try it out” button from the upper right corner (highlighted in the below image). Before clicking on the “Try it out” button, the Responses section shows example values will be returned when executing this method.

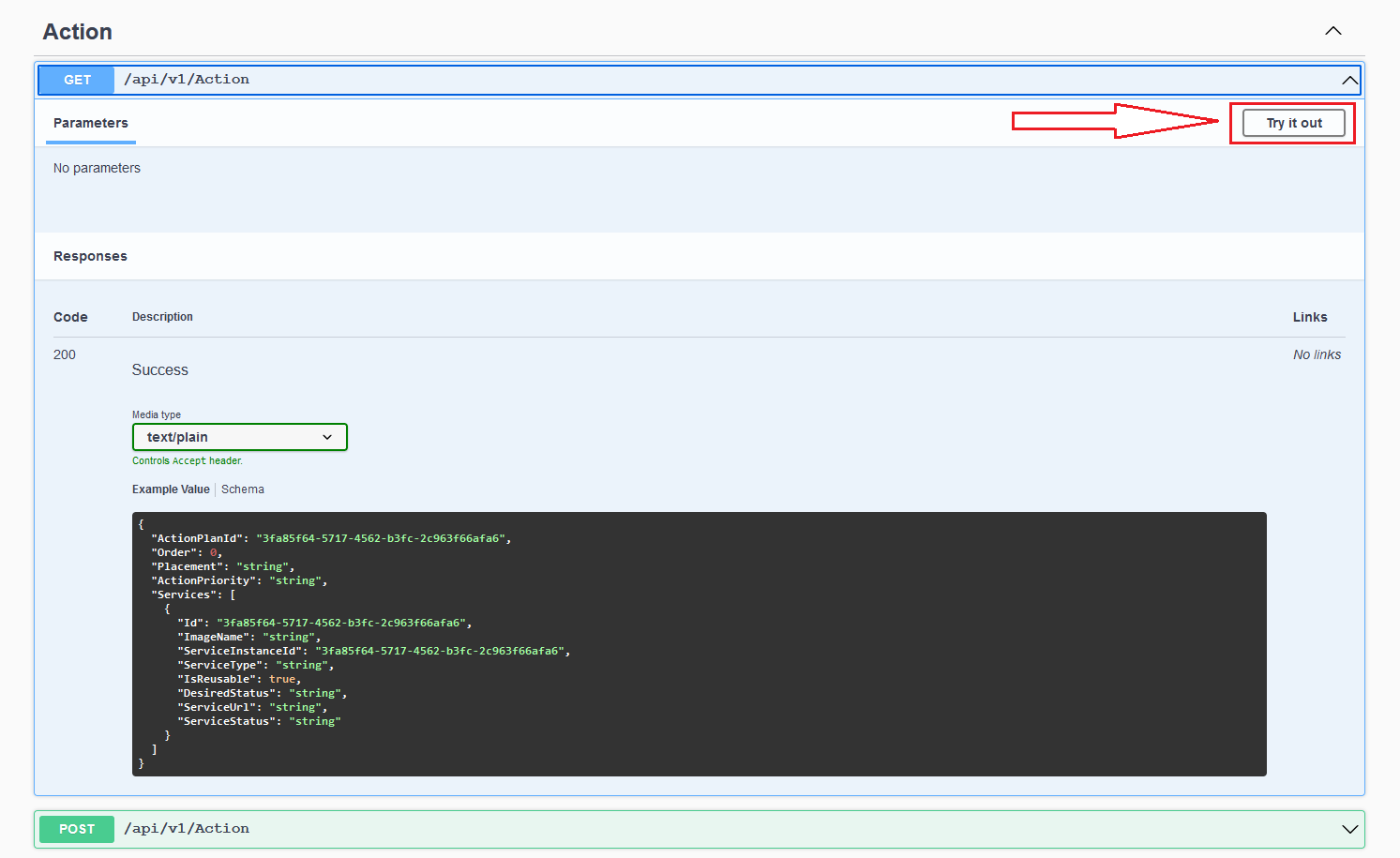


Figure 3 Action GET all method

After clicking on the “Try it out” button, click on the “Execute” button, as in the image below.

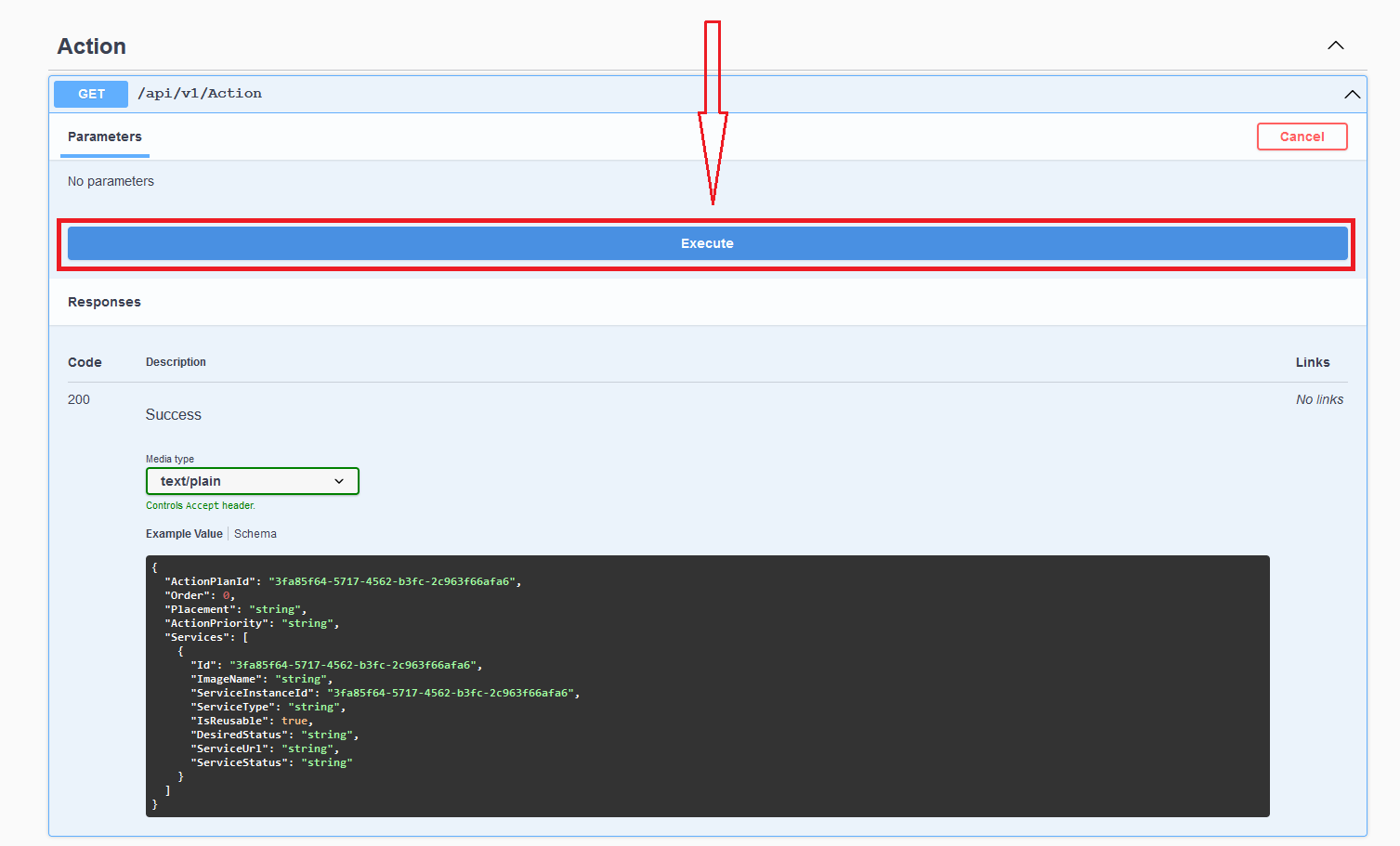


Figure 4 Action GET all method execute button

The image below describes the result of the execution. The first highlighted sections from the image shows the Request URL. The URL contains the hostname of our RedisInterface API (i.e., <https://localhost:7091>) and the designated endpoint, which is “/api/v1/Action". The second highlighted section represents the actual values that were retrieved from the Redis storage.

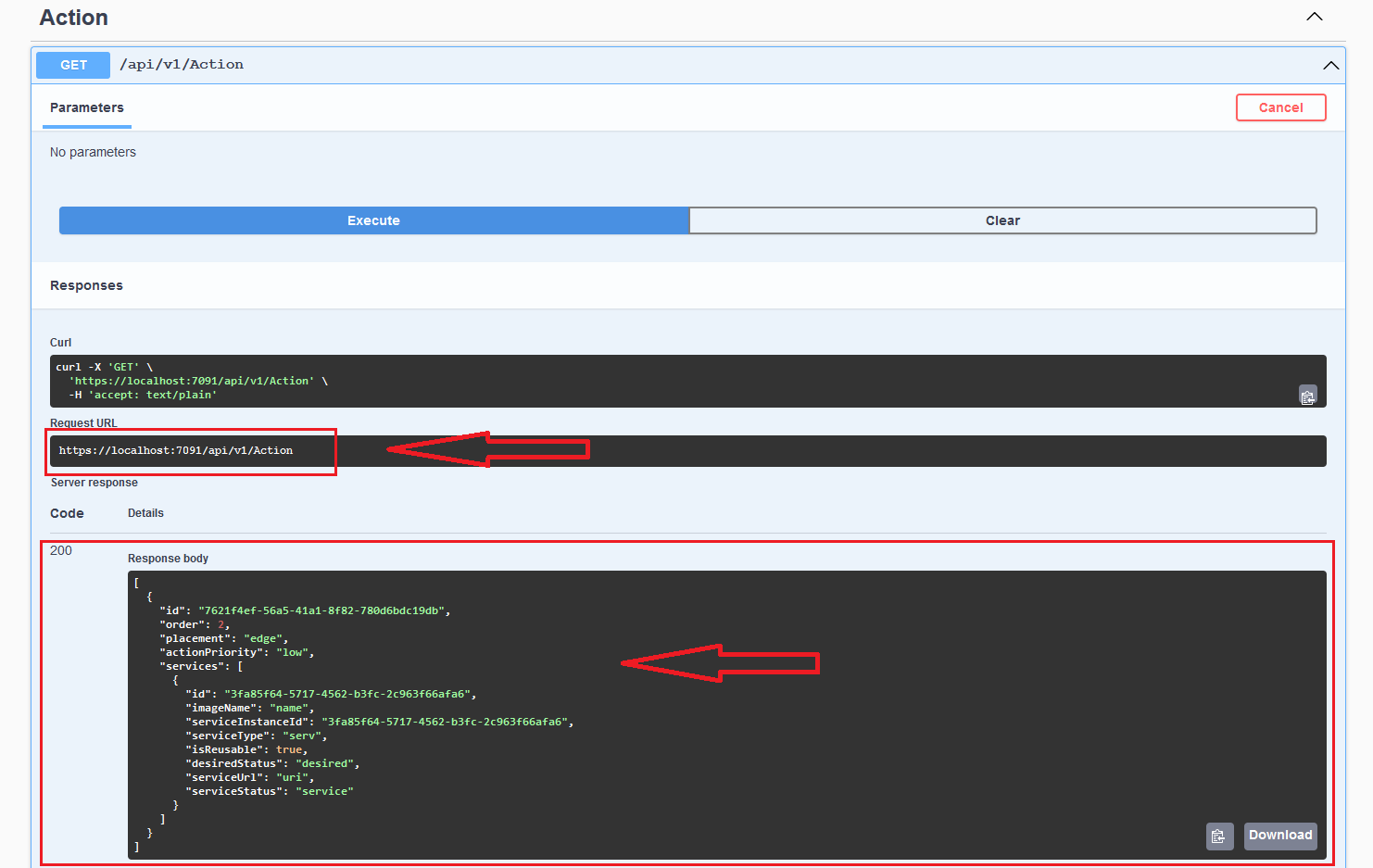


Figure 5 Action GET all method response values