InteliQ API Documentation

1. {baseURL}/admin/healthcheck

It supports the GET method and confirms the end-to-end connectivity between the user and the database.

We use Postman to perform the request

GET	∨	http://localhost:9103/intelliq_api/admin/healthcheck

On successful request returns a JSON object and the response has a status code of 200 :

```
1    {
2        "status": "OK",
3        "dbconnection": "Everything is okey"
4    }
```

2. {baseURL}/admin/questionnaire_upd

It supports the POST method and enables the admin to upload a questionnaire in the database.
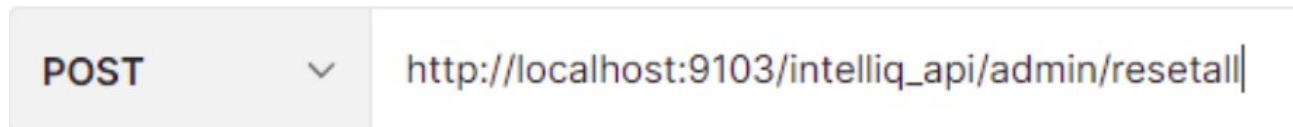
We use Postman. We go to "Form Data" and we fill out the blanks:
- **key**: "file"
- **value**: the actual file we want to upload

3. {baseURL}/admin/resetall

It supports the POST method and enables the admin to erase the entire database.

We use Postman to perform the request.

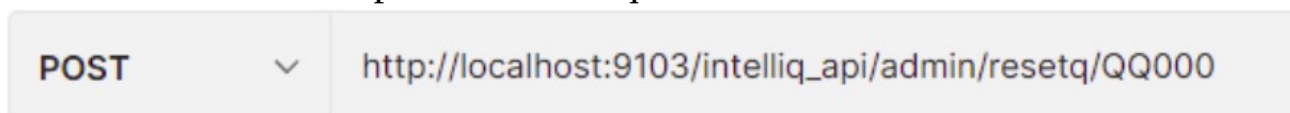| POST	⌄	http://localhost:9103/intelliq_api/admin/resetall

On successful request returns a JSON object and the response has a status code of 200 :

```
1    {
2        "status": "OK"
3    }
```

4. {baseURL}/admin/resetq/:questionnaireID

It supports the POST method and enables the admin to delete the questionnaire with id {questionnaireID}.
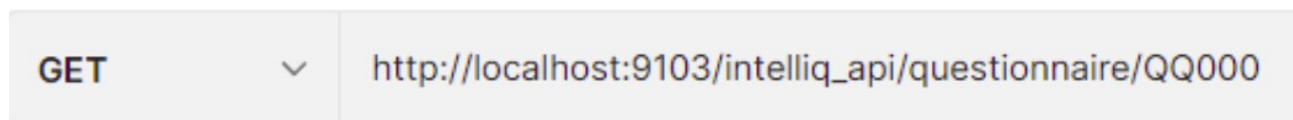
We use Postman to perform the request.

POST	⌄	http://localhost:9103/intelliq_api/admin/resetq/QQ000

On successful request returns a JSON object and the response has a status code of 200 :

```
1  {
2      "status": "OK"
3  }
```

a. {baseURL}/questionnaire/:questionnaireID

It supports the GET method and returns the questionnaire with id {questionnaireID}.

We use Postman to perform the request.

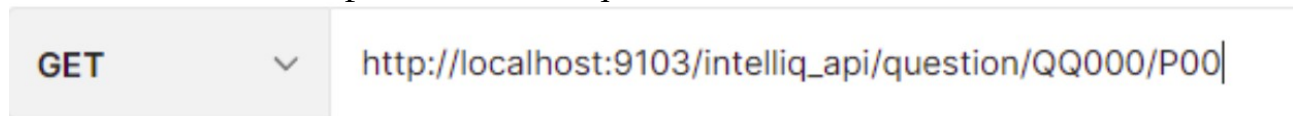| GET	∨	http://localhost:9103/intelliq_api/questionnaire/QQ000

On successful request returns a JSON object and the response has a status code of 200 :

```
1  {
2      "questionnaireID": "QQ000",
3      "questionnaireTitle": "My first research questionnaire",
4      "keywords": [
5          "footbal",
6          "islands",
7          "timezone"
8      ],
9      "questions": [
10         {
11             "qID": "P00",
12             "qtext": "Ποιο είναι το mail σας;",
13             "required": false,
14             "type": "profile"
15         },
16         {
17             "qID": "P01",
18             "qtext": "Ποια είναι η ηλικία σας;",
19             "required": true,
20             "type": "profile"
21         }
22     ]
23  }
```

b. {baseURL}/question/:questionnaireID/:questionID

It supports the GET method and returns information about the question with id {questionID} from the questionnaire with id {questionnaireID}.

We use Postman to perform the request.

GET	⌄	http://localhost:9103/intelliq_api/question/QQ000/P00

On successful request returns a JSON object and the response has a status code of 200 :

```
1   {
2       "questionnaireID": "QQ000",
3       "qID": "P00",
4       "qtext": "Ποιο είναι το mail σας;",
5       "required": false,
6       "type": "profile",
7       "options": [
8           {
9               "optID": "P00TXT",
10              "opttxt": "<open string>",
11              "nextqID": "P01",
12              "_id": "63ee3ecaa2c1d20d1a7fc1b7"
13          }
14      ]
15  }
```

c. {baseURL}/doanswer/:questionnaireID/:questionID/:session/:optionID

It supports the POST method and enables the user to create an answer with id {optionID} for the question with id {questionID} in the questionnaire with id {questionnaireID} inside the session with id {session}.

We use Postman to perform the request.

POST	∨	http://localhost:9103/intelliq_api/doanswer/QQ000/P01/S002/P01A2

On successful request returns a JSON object and the response has a status code of 200 :

```
1  {
2      "status": "success"
3  }
```

d. {baseURL}/getsessionanswers/:questionnaireID/:session

It supports the GET method and returns all the answers of the questionnaire with id {questionnaireID} in the session with id {session}.

We use Postman to perform the request.

GET	∨	http://localhost:9103/intelliq_api/getsessionanswers/QQ000/S001

On successful request returns a JSON object and the response has a status code of 200 :

```
1    {
2        "questionnaireID": "QQ000",
3        "session": "S001",
4        "answers": [
5            {
6                "qID": "P01",
7                "ans": "P01A1"
8            }
9        ]
10   }
```

e. {baseURL}/getquestionanswers/:questionnaireID/:questionID

It supports the GET method and returns all the answers of question with id {questionID} in the questionnaire with id {questionnaireID}.

We use Postman to perform the request.

| GET | ∨ | http://localhost:9103/intelliq_api/getquestionanswers/QQ000/P01 |

On successful request returns a JSON object and the response has a status

code of 200 :

```
1    {
2        "questionnaireID": "QQ000",
3        "questionID": "P01",
4        "answers": [
5            {
6                "session": "S001",
7                "ans": "P01A1"
8            }
9        ]
10   }
```

In case of errors, all of the above request may return the following three status codes each corresponding to one specific error:

- Status code **400**: Bad request
-
- Status code **402**: Empty request
-
- Status code **500**: Any other error