QuestionIT

ReST Service Definition

In this document you will find a description of all the Endpoints as defined in backend of the QustionIT tool. For the POST and PUT commands a description is given of the JSON objects that have to sent. Before getting into detail about the Endpoints, first some information about the Enums that are used. This important because when you sent a JSON object that requires an Enum, you can choose to send a number or a specific string.

Enums

EnumExams

```
  0  :  "NONE
  1  :  "OCA"
  2  :  "OCP"
  3  :  "_70_480"
```

EnumLanguages

```
  0  :  "NONE"
  1  :  "JAVA"
  2  :  "HTML_CSS_JS"
```

QuestionEndpoint

GET all Questions
Command: GET
Path: `'api/questions'`
Return:
- 200 *(ok)* + JSON([QuestionsModelBasic]),
- 404 *(Not Found)*

Note:
- For the Creator only the id and name are returned

GET one Question
Command: GET
Path: `'api/questions/{id}/{format}'`
Parameter: `{format}`: must be '`exam`', '`examreview`' or '`basic`'
Return:
- 200 *(ok)* + JSON(QuestionModelBasic)
- 400 *(Bad Request)* if format not specified correctly
- 404 *(Not Found)* if id does not exist

Note:

- For the Creator only the id and name are returned
- When format equals exam, only fields below are returned:

```
      "id" : number,
      "name" : "string",
      "question" : "string",
      "possibleAnswers" : ["string", "string"],
      "typeOfQuestion" : "string",
      "enabled" : boolean
```

- When format equals examreview, only fields below are returned: These are:

```
      "id" : number,
      "name" : "string",
      "question" : "string",
      "possibleAnswers" : ["string", "string"],
      "typeOfQuestion" : "string",
      "enabled" : boolean,
      "correctAnswers": { "id": number, "answers": [boolean, boolean]},
      "explanationAnswer": "string"
```

Create a new Question
Command: POST
Path: `'api/questions/creator/{instructor_id}/correct-answers/{answerlist_id}'`
Return:
- 200 *(ok)* + id
- 404 *(Not Found)* if answerlist or instructor do not exist
- 500 *(Internal Server Error)*

Notes:
- Exclude 'creator', 'correctAnswers' and 'givenAnswers' from JSON
- If an existing 'id' is included than the question is made obsolete and a new one is created
- If non-existing 'id' is included it is ignored
- For the new question:
  - obsolete is set to false,
  - enabled is set to true and
  - creationDateTime is set to current time

```
      {
        "name": "string",
        "programmingLanguage": EnumLanguages,
        "forExam": EnumExams,
        "question": "string",
        "possibleAnswers": [
          "string",
          "string"
        ],
        "explantionAnswer": "string",
        "typeOfQuestion": "string"
      }
```

Update/Add AnswerList for/to existing Question
Command: POST
Path: `'api/question/{id}/{answerlist_id}/{field}'`
Return:
- 200 *(ok)*
- 400 *(Bad Request)* if field not specified correctly
- 404 *(Not Found)* if id or answerlist_id do not exist

Parameter: `{field}`: must be '`correct`' or '`given`'

Notes:
- No JSON object needs to be included
- If the field equals correct, the AnswerList is set for correctAnswers
- If the field equals given, the AnserList is add to givenAnswerLists

Delete an existing Question
Command: DELETE
Return:
- 200 *(ok)*
- 404 *(Not Found)*
Path: `'api/questions/{id}'`

AnswerListEndpoint

GET one AnswerList
Command: GET
Path: `'api/questions/{id}'`
Return: 200 *(ok)* + JSON(AnswerList) or 404 *(Not Found)*
Note:
- For the Creator only the id and name are returned

Create a new AnswerList
Command: POST
Path: `'api/questions/answer-list'`
Return: 200 *(ok)* + id or 500 *(Internal Server Error)*
Notes:
- In the database, a list always consists of 10 answers. If the posted list containts less than 10 answers, the remainder are set to false
- If an existing 'id' is included than the AnswerList is updated
- If non-existing 'id' is included a new AnswerList is created

```
{
  "answers": [
    boolean,
    boolean,
    boolean,
    boolean,
    boolean,
    boolean,
    boolean,
    boolean,
    boolean,
    boolean
  ]
}
```

<<<< DOCUMENT BELOW NEEDS TO BE UPDATED >>>>

QuestionListEndpoint

GET all QuestionLists

Command: GET

Path: `'api/questionlists'`

GET one QuestionList

Command: GET

Path: `'api/questionlists/{id}'`

Create new or change one QuestionList

Command: POST

Path: `'api/questionlists'`

Notes:

- Exclude 'creator' and 'questions' from JSON
- If an existing 'id' is included than the data will be overwritten
- If non-existing 'id' is included it is ignored

```
{
  "name": "string",
  "programmingLanguage": EnumLanguages,
  "forExam": EnumExams,
  "creationDateTime": number,
  "examTimeInMinutes": number
}
```

Add new Question to existing QuestionList

Command: POST

Path: `'api/questionlists/{id}/question'`

Notes:

- Exclude 'creator', 'correctAnswers' and 'givenAnswers' from JSON
- If an existing 'id' is included than the data will be overwritten
- If non-existing 'id' is included, there is error ("500 Internal Server Error")

```
{
  "name": "string",
  "programmingLanguage": EnumLanguages,
  "forExam": EnumExams,
  "creationDateTime": number,
  "question": "string",
  "explantionAnswer": "string",
  "typeOfQuestion": "string"
}
```

Add existing Question to existing QuestionList

Command: POST

Path: `'api/questionlists/{id}/question/{question_id}'`

Note:

- No JSON object needs to be included
- If both the Question and the QuestionList exist, the Question is attached

Add existing Instructor to existing QuestionList

Command: POST

Path: `'api/questionlists/{id}/instructor/{instructor_id}'`

Note:

- No JSON object needs to be included
- If both the Instructor and the QuestionList exist, the Instructor is attached