

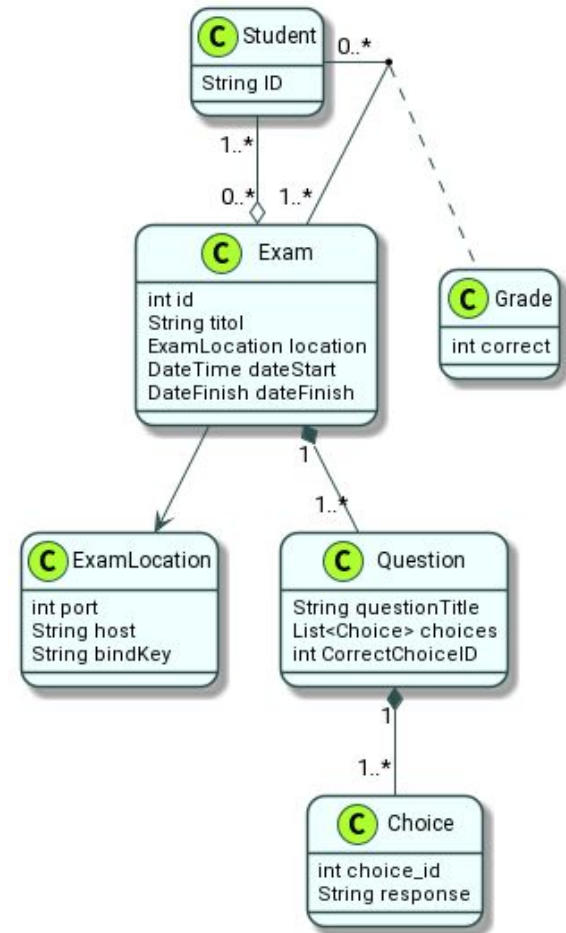
WS Project

Oriol Alàs Cercós & Marta Albets Mitjaneta

CLASSES AND UML

Classes:

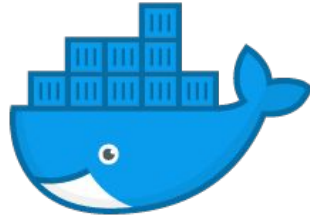
- Student
- Exam
- ExamLocation
- Question
- Choice
- Grade



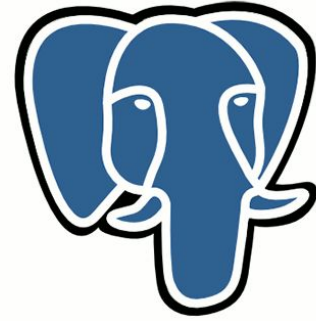
TECHNOLOGIES

django

django
REST
framework



docker



PostgreSQL

ENDPOINTS

Endpoints	Operations	Resulting codes	Representations
/api/exam/	GET	HTTP 200 OK	Output: <pre>[{"id": 1, "title": "first exam","description": "first exam description", "date_start": "2021-01-04T01:36:00Z", "date_finish": "2021-01-05T11:40:00Z", "location": {"port": 998, "host": "localhost", "bind_key": "string1"}, "questions": [{"title": "first question", "choices": [{"choice_id": 1, "response": "response one"}]}, "correct_choice": 1}, {"title": "second question", "choices": [{"choice_id": 1, "response": "response one"}]}, "correct_choice": 1}], "students": [{"studentID": "423"}]}</pre> <pre>{ "id": 2, "title": "second exam","description": "second exam description", "date_start": "2021-01-04T01:36:00Z", "date_finish": "2021-01-05T11:40:00Z", "location": {"port": 998, "host": "localhost", "bind_key": "string1"}, "questions": [{"title": "first question", "choices": [{"choice_id": 1, "response": "response one"}]}, "correct_choice": 1}, {"title": "second question", "choices": [{"choice_id": 1, "response": "response one"}]}, "correct_choice": 1}], "students": [{"studentID": "423"}]}</pre>
	GET	HTTP 200 OK	Input: ?search=second_exam_description Output: <pre>{ "id": 2, "title": "second exam","description": "second exam description", "date_start": "2021-01-04T01:36:00Z", "date_finish": "2021-01-05T11:40:00Z", "location": {"port": 998, "host": "localhost", "bind_key": "string1"}, "questions": [{"title": "first question", "choices": [{"choice_id": 1, "response": "response one"}]}, "correct_choice": 1}, {"title": "second question", "choices": [{"choice_id": 1, "response": "response one"}]}, "correct_choice": 1}], "students": [{"studentID": "423"}]}</pre>
	POST	HTTP 201 Created	Input: <pre>{ "title": "third exam","description": "third exam description", "date_start": "2021-01-04T01:36:00Z", "date_finish": "2021-01-05T11:40:00Z", "location": {"port": 998, "host": "localhost", "bind_key": "string1"}, "questions": [{"title": "first question", "choices": [{"choice_id": 1, "response": "response one"}]}, "correct_choice": 1}, {"title": "second question", "choices": [{"choice_id": 1, "response": "response one"}]}, "correct_choice": 1}], "students": [{"studentID": "423"}]}</pre> Output: <pre>{ "title": "third exam","description": "third exam description", "date_start": "2021-01-04T01:36:00Z", "date_finish": "2021-01-05T11:40:00Z", "location": {"port": 998, "host": "localhost", "bind_key": "string1"}, "questions": [{"title": "first question", "choices": [{"choice_id": 1, "response": "response one"}]}, "correct_choice": 1}, {"title": "second question", "choices": [{"choice_id": 1, "response": "response one"}]}, "correct_choice": 1}], "students": [{"studentID": "423"}]}</pre>

/api/exam/3/	GET	HTTP 200 OK	Input: {"id": 3} Output: {"title": "third exam", "description": "third exam description", "date_start": "2021-01-04T01:36:00Z", "date_finish": "2021-01-05T11:40:00Z", "location": {"port": 998, "host": "localhost", "bind_key": "string1"}, "questions": [{"title": "first question", "choices": [{"choice_id": 1, "response": "response one"}], "correct_choice": 1}, {"title": "second question", "choices": [{"choice_id": 1, "response": "response one"}], "correct_choice": 1}], "students": [{"studentID": "423"}]}
/api/exam/3/	DELETE	HTTP 204 No content	No content(undefined)
/api/exam/2/desc	GET	HTTP 200 OK	Output: {"description": "second exam description"}
	PUT	HTTP 200 OK	Input: {"description": "exam two description"} Output: {"description": "exam two description"}
/api/exam/1/423	GET	HTTP 200 OK	Output: {"port": 998, "host": "localhost", "bind_key": "string1"}
/api/exam/1/grades/	GET	HTTP 200 OK	Output: [{"student": {"studentID": "423"}, "correct": 0}]
/api/exam/1/grades/423/	GET	HTTP 200 OK	Output: {"student": {"studentID": "423"}, "correct": 0}
	PUT	HTTP 200 OK	Input: {"correct": 1} Output: {"student": {"studentID": "423"}, "correct": 1}
/api/location/	GET	HTTP 200 OK	Output: {"port": 998, "host": "localhost", "bind_key": "string1"}
	POST	HTTP 201 Created	Input: {"port": 998, "host": "localhost", "bind_key": "string2"} Output: {"port": 998, "host": "localhost", "bind_key": "string2"}
/api/location/1/	GET	HTTP 200 OK	Input: {"id": 1}

			Output: { "port": 998, "host": "localhost", "bind_key": "string1" }
/api/question/	GET	HTTP 200 OK	Output: [{"title": "first question", "choices": [{"choice_id": 1, "response": "response one"}], "correct_choice": 1}, {"title": "second question", "choices": [{"choice_id": 1, "response": "response one"}], "correct_choice": 1}, {"title": "first question", "choices": [{"choice_id": 1, "response": "response one"}], "correct_choice": 1}, {"title": "second question", "choices": [{"choice_id": 1, "response": "response one"}], "correct_choice": 1}]
	POST	HTTP 201 Created	Input: {"title": "new question", "choices": [{"choice_id": 1, "response": "first response"}, {"choice_id": 2, "response": "second response"}], "correct_choice": 1} Output: {"title": "new question", "choices": [{"choice_id": 1, "response": "first response"}, {"choice_id": 2, "response": "second response"}], "correct_choice": 1}
/api/question/3	GET	HTTP 200 OK	Output: { "title": "first question", "choices": [{"choice_id": 1, "response": "response one"}], "correct_choice": 1 }
	DELETE	HTTP 204 No content	No content(undefined)

/api/student/	GET	HTTP 200 OK	Output: [{"studentID": "78099079A"}, {"studentID": "12345632A"}, {"studentID": "78103041D"}, {"studentID": "78320584C"}, {"studentID": "48930SFD2"}]
/api/student/78099079A/	GET	HTTP 200 OK	Output: { "studentID": "78099079A" }
/api/student/78099079A/grades	GET	HTTP 200 OK	Output: [{"student": {"studentID": "78099079A"}, "correct": 0, "exam_id": 1}, {"student": {"studentID": "78099079A"}, "correct": 6, "exam_id": 2}]

SCREENSHOTS

List all the exams

GET /api/exam/

HTTP 200 OK

Allow: GET, POST, HEAD, OPTIONS

Content-Type: application/json

Vary: Accept

```
[
  {
    "id": 7,
    "title": "My First Exam",
    "description": "Distributed Computing exam description",
    "date_start": "2021-01-04T01:36:00Z",
    "date_finish": "2021-01-05T11:40:00Z",
    "location": {
      "port": 998,
      "host": "localhost",
      "bind_key": "string1"
    },
    "questions": [
      {
        "title": "Question 1",
        "choices": [
          {
            "choice_id": 1,
            "response": "response1"
          }
        ],
        "correct_choice": 1
      }
    ],
    "students": [
      {
        "studentID": "423"
      }
    ]
  },
  {
    "id": 8,
    "title": "My Second Exam",
    "description": "Distributed Computing2 description",
    "date_start": "2021-01-04T01:36:00Z",
    "date_finish": "2021-01-05T11:40:00Z",
    "location": {
      "port": 998,
      "host": "localhost",
      "bind_key": "string1"
    },
    "questions": [
```

```
    "title": "Question 1",
    "choices": [
      {
        "choice_id": 1,
        "response": "response1"
      }
    ],
    "correct_choice": 1
  },
  {
    "title": "Question 2",
    "choices": [
      {
        "choice_id": 1,
        "response": "response1"
      },
      {
        "choice_id": 2,
        "response": "response2"
      }
    ],
    "correct_choice": 2
  }
],
"students": [
  {
    "studentID": "423"
  },
  {
    "studentID": "123"
  }
]
```

The outcome we obtain when we list all the exams is the attributes, id, title, description, date start, date finish, location, the list of questions with their respective choices and the list of students, of every one of them.

Search an exam by its id

```
GET /api/exam/7/
```

```
HTTP 200 OK
```

```
Allow: GET, DELETE, HEAD, OPTIONS
```

```
Content-Type: application/json
```

```
Vary: Accept
```

```
{
  "id": 7,
  "title": "My First Exam",
  "description": "Distributed Computing exam description",
  "date_start": "2021-01-04T01:36:00Z",
  "date_finish": "2021-01-05T11:40:00Z",
  "location": {
    "port": 998,
    "host": "localhost",
    "bind_key": "string1"
  },
  "questions": [
    {
      "title": "Question 1",
      "choices": [
        {
          "choice_id": 1,
          "response": "response1"
        }
      ],
      "correct_choice": 1
    }
  ],
  "students": [
    {
      "studentID": "423"
    }
  ]
}
```

Searching an exam by its id we obtain all the parameters that take part of that specific exam.

Search an exam by its description

Searching a description ("Distributed Computing2 description" in this case) the API found the exam with id=9, and the outcome is all the information of this particular exam.

```
GET /api/exam/?search=Distributed%20Computing2%20description
```

```
HTTP 200 OK
```

```
Allow: GET, POST, HEAD, OPTIONS
```

```
Content-Type: application/json
```

```
Vary: Accept
```

```
[
  {
    "id": 8,
    "title": "My Second Exam",
    "description": "Distributed Computing2 description",
    "date_start": "2021-01-04T01:30:00Z",
    "date_finish": "2021-01-05T11:40:00Z",
    "location": {
      "port": 098,
      "host": "localhost",
      "bind_key": "string1"
    },
    "questions": [
      {
        "title": "Question 1",
        "choices": [
          {
            "choice_id": 1,
            "response": "response1"
          }
        ],
        "correct_choice": 1
      },
      {
        "title": "Question 2",
        "choices": [
          {
            "choice_id": 1,
            "response": "response1"
          },
          {
            "choice_id": 2,
            "response": "response2"
          }
        ],
        "correct_choice": 2
      }
    ],
    "students": [
      {
        "studentID": "423"
      },
      {
        "studentID": "123"
      }
    ]
  }
]
```

Create an exam

Writing all the exam required information, title, description, date_start, date_finish, location, list of questions, list of choices for every question and a list of allowed students; the API creates a new exam and the outcome showed is all these previous information and also the respectively id of that exam.

```
POST /api/exam/

HTTP 201 Created
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

{
  "id": 10,
  "title": "My Third Exam",
  "description": "exam description",
  "date_start": "2021-01-04T01:30:00Z",
  "date_finish": "2021-01-05T11:40:00Z",
  "location": {
    "port": 998,
    "host": "localhost",
    "bind_key": "string1"
  },
  "questions": [
    {
      "title": "Question 1",
      "choices": [
        {
          "choice_id": 1,
          "response": "response1"
        }
      ],
      "correct_choice": 1
    },
    {
      "title": "Question 2",
      "choices": [
        {
          "choice_id": 1,
          "response": "response1"
        },
        {
          "choice_id": 2,
          "response": "response2"
        },
        {
          "choice_id": 3,
          "response": "response3"
        }
      ],
      "correct_choice": 2
    }
  ],
  "students": [
    {
      "studentID": "423"
    },
    {
      "studentID": "999"
    }
  ]
}
```

Delete an exam

```
DELETE /api/exam/10/
```

```
HTTP 204 No Content
```

```
Allow: GET, DELETE, HEAD, OPTIONS
```

```
Content-Type: application/json
```

```
Vary: Accept
```

Specifying an id of an existing exam and deleting it implies that the API shows a message of deletion (no content) with any outcome, because that exam does not exist anymore.

Show an exam description

```
GET /api/exam/8/desc/
```

```
HTTP 200 OK
```

```
Allow: GET, PUT, PATCH, HEAD, OPTIONS
```

```
Content-Type: application/json
```

```
Vary: Accept
```

```
{  
  "description": "Distributed Computing2 description"  
}
```

If we specify `/desc/` at the end of a particular exam we will receive the description of that exam as the outcome.

Upload an exam description

```
PUT /api/exam/8/desc/
```

HTTP 200 OK

Allow: GET, PUT, PATCH, HEAD, OPTIONS

Content-Type: application/json

Vary: Accept

```
{
  "description": "Distributed Computing2 description changed"
}
```

If we change the description of an existing exam we obtained the new description as the outcome.

Search an student by its id on a exam

```
GET /api/exam/8/123/
```

HTTP 200 OK

Allow: GET, HEAD, OPTIONS

Content-Type: application/json

Vary: Accept

```
{
  "port": 998,
  "host": "localhost",
  "bind_key": "string1"
}
```

When we search if a student is allowed to do a specific exam or not, the outcome we receive is the location of the exam.

List all the grades of an exam

```
GET /api/exam/8/grades/
```

```
HTTP 200 OK
Allow: GET, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept
```

```
[
  {
    "student": {
      "studentID": "423"
    },
    "correct": 0
  },
  {
    "student": {
      "studentID": "123"
    },
    "correct": 0
  }
]
```

We can list all the students grades of an exam, viewing as the outcome a list where we can see for every student its id and its respectively grade.

Show the exam grade for a specific student

```
GET /api/exam/8/grades/423/
```

```
HTTP 200 OK
Allow: GET, PUT, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept
```

```
{
  "student": {
    "studentID": "423"
  },
  "correct": 0
}
```

If we want to see only the grade of a particular student, the outcome we obtain is only the id of this student and its grade.

Store grades

```
PUT /api/exam/8/grades/423/
```

```
HTTP 200 OK
```

```
Allow: GET, PUT, HEAD, OPTIONS
```

```
Content-Type: application/json
```

```
Vary: Accept
```

```
{  
  "student": {  
    "studentID": "423"  
  },  
  "correct": 2  
}
```

Specifying an exam and an allowed student to that exam we are able to change its grade and the API outcome is the student id with the new grade.

List all locations

The outcome when we list all the locations is a list with all the information of each location (port, host and bind_key).

```
GET /api/location/
```

```
HTTP 200 OK
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept
```

```
[
    {
        "port": 998,
        "host": "localhost",
        "bind_key": "string1"
    },
    {
        "port": 998,
        "host": "localhost",
        "bind_key": "string1"
    },
    {
        "port": 998,
        "host": "localhost",
        "bind_key": "string1"
    },
    {
        "port": 998,
        "host": "localhost",
        "bind_key": "string1"
    }
]
```

```
{
    "port": 998,
    "host": "localhost",
    "bind_key": "string1"
},
{
    "port": 998,
    "host": "localhost",
    "bind_key": "string1"
},
{
    "port": 998,
    "host": "localhost",
    "bind_key": "string1"
},
{
    "port": 998,
    "host": "localhost",
    "bind_key": "string1"
},
{
    "port": 998,
    "host": "localhost",
    "bind_key": "string1"
}
```

Create a location

```
POST /api/location/
```

```
HTTP 201 Created
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept
```

```
{
  "port": 998,
  "host": "localhost",
  "bind_key": "string1"
}
```

When we create a new location the outcome we receive is the information of all the parameters of this new location, which are the port, the host and the bind_key.

Search a location by its id

```
GET /api/location/1/
```

```
HTTP 200 OK
Allow: GET, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept
```

```
{
  "port": 998,
  "host": "localhost",
  "bind_key": "string1"
}
```

The outcome shows us the information of the location (port, host and bind_key) of the location with the id we gave as an input.

List all the questions

The outcome when we list all the questions is a list with all the information of each question (title, the list of choices and the correct choice).

GET /api/question/

HTTP 200 OK
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

```
[
  {
    "title": "Question1",
    "choices": [
      {
        "choice_id": 1,
        "response": "response1"
      }
    ],
    "correct_choice": 1
  },
  {
    "title": "Question 1",
    "choices": [
      {
        "choice_id": 1,
        "response": "response1"
      }
    ],
    "correct_choice": 1
  },
  {
    "title": "Question 1",
    "choices": [
      {
        "choice_id": 1,
        "response": "response1"
      }
    ],
    "correct_choice": 1
  },
  {
    "title": "Question 2",
    "choices": [
      {
        "choice_id": 1,
        "response": "response1"
      },
      {
        "choice_id": 2,
        "response": "response2"
      }
    ],
    "correct_choice": 2
  },
  {
    "title": "Question 1",
    "choices": [

```

```

    {
      "choice_id": 1,
      "response": "response1"
    }
  ],
  "correct_choice": 1
},
{
  "title": "Question 2",
  "choices": [
    {
      "choice_id": 1,
      "response": "response1"
    },
    {
      "choice_id": 2,
      "response": "response2"
    }
  ],
  "correct_choice": 2
},
{
  "title": "Question 1",
  "choices": [
    {
      "choice_id": 1,
      "response": "response1"
    }
  ],
  "correct_choice": 1
},
{
  "title": "Question 2",
  "choices": [
    {
      "choice_id": 1,
      "response": "response1"
    },
    {
      "choice_id": 2,
      "response": "response2"
    },
    {
      "choice_id": 3,
      "response": "response3"
    }
  ],
  "correct_choice": 2
}
]
```

Create a question

POST /api/question/

HTTP 201 Created

Allow: GET, POST, HEAD, OPTIONS

Content-Type: application/json

Vary: Accept

```
{
  "title": "New question",
  "choices": [
    {
      "choice_id": 1,
      "response": "response1"
    },
    {
      "choice_id": 2,
      "response": "response2"
    },
    {
      "choice_id": 3,
      "response": "response3"
    }
  ],
  "correct_choice": 2
}
```

Search a question by its id

GET /api/question/9/

HTTP 200 OK

Allow: GET, DELETE, HEAD, OPTIONS

Content-Type: application/json

Vary: Accept

```
{
  "title": "Question1",
  "choices": [
    {
      "choice_id": 1,
      "response": "response1"
    }
  ],
  "correct_choice": 1
}
```

The outcome shows us the parameters information of the question with the id we gave as an input.

When we create a question the outcome we receive in exchange is the information of the question we just created.

Delete a question

```
DELETE /api/question/3/
```

```
HTTP 204 No Content  
Allow: GET, DELETE, HEAD, OPTIONS  
Content-Type: application/json  
Vary: Accept
```

Specifying an id of an existing question and deleting it the API shows us a message of deletion with any outcome, because that question does not exist anymore.

Listing all the students

```
GET /api/student/

HTTP 200 OK
Allow: GET, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

[
  {
    "studentID": "423"
  },
  {
    "studentID": "123"
  },
  {
    "studentID": "999"
  },
  {
    "studentID": "STUDENTID1"
  }
]
```

Detail of a student

```
GET /api/student/423/

HTTP 200 OK
Allow: GET, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

{
  "studentID": "423"
}
```

List all the grades from a student

```
GET /api/student/423/grades/

HTTP 200 OK
Allow: GET, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

[
  {
    "student": {
      "studentID": "423"
    },
    "correct": 0,
    "exam_id": 7
  },
  {
    "student": {
      "studentID": "423"
    },
    "correct": 2,
    "exam_id": 8
  }
]
```

The response is a list of grade objects composed by the student object, the number of correct answers and the id of the exam.

GITHUB REPOS & TIME OF WORK

- The API can be found in <https://github.com/Oriolac/my-exams-api> and the RMI project in <https://github.com/Marta99/multiple-choice-exam>
- The estimated hour of work we've dedicated to this project are 100 hours.

RMI Integration

- Use of `java.net.HttpClient` for WS Connection.
- Package for the integration and adapt the `HttpClient` to our features in a higher level.