

Documentation

Teams

GET /teams

Returns a list with all teams

Authorization token required: **yes**

Admin required: **no**

Response body

```
[
  {
    "team_name": "String",
    "team_captain": "Participant object",
    "team_members": ["Participant object", "Participant object",
"Participant object"],
    "is_full": "boolean",
    "is_approved": "boolean"
  }
]
```

POST /teams

Creates a team if it does not exist yet and then returns the created team

Authorization token required: **yes**

Admin required: **no**

Request body

```
{
  "team_name": "String"
}
```

Response body

```
{
  "team_name": "String",
  "team_captain": "Participant object",
  "team_members": ["Participant object", "Participant
```

```
object", "Participant object"],
  "is_full": "boolean",
  "is_approved": "boolean"
}
```

GET /teams/{team_name}

Returns the team with the given team name

Authorization token required: **yes**

Admin required: **no (I think so? please double check this)**

Response body

```
{
  "team_name": "String",
  "team_captain": "Participant object",
  "team_members": ["Participant object", "Participant
object", "Participant object"],
  "is_full": "boolean",
  "is_approved": "boolean"
}
```

POST /teams/{invite_code}/join

The logged in user joins a team, if successful the server should respond with the team object

Authorization token required: **yes**

Admin required: **no**

Response body

```
{
  "team_name": "String",
  "team_captain": "Participant object",
  "team_members": ["Participant object", "Participant
object", "Participant object"],
  "is_full": "boolean",
  "is_approved": "boolean"
}
```

PATCH /teams/{team_name}/approve

An organizer can approve the team

Authorization token required: **yes**

Admin required: **yes**

User management

POST /login

The user supplies a student id and password, the server should return with a JWT

Authorization token required: **no**

Admin required: **no**

Request body

```
{
  "sid": "Integer",
  "password": "String"
}
```

Response body

```
{
  "JWTToken": "String"
}
```

POST /register

The user supplies a student id, name, phone number and password, the server should return with a status code

Authorization token required: **no**

Admin required: **no**

Request body

```
{
  "s_num": "Integer",
  "name": "String",
  "phone_num": "String",
  "password": "String"
}
```

GET /user

The server should respond with the Person object of the logged in client

Authorization token required: **yes**

Admin required: **no**

Responde body

```
{
  "s_numb": "Integer",
  "name": "String",
  "phone_numb": "String"
}
```

Game

GET /leaderboard

Returns the current leaderboard

Authorization token required: **yes**

Admin required: **no** - *unless the leaderboard is deactivated, then only admins can see it*

TODO: what does the leaderboard look like? what data do we want to display

GET /leaderboard/{team_name}

Returns the statistiscs of the given team

Authorization token required: **yes**

Admin required: **no** - *unless the leaderboard is deactivated, then only admins and the team itself can see it*

Response body

```
{
  team_name: "string",
  score: "Number",
  rank: "Number",
}
```

`PATCH /leaderboard/availability`
Activate or deactivate the leaderboard

Authorization token required: `yes`

Admin required: `yes`

Request body
``json

```
{
```

```
    "is_active": "boolean"  
  }
```

Submissions

GET /submissions

Returns a list of all submissions

Available filters:

- `team_name`: returns the submissions of the given team
- `ungraded`: returns the submissions that have not been graded yet
- `graded`: returns the submissions that have been graded

Authorization token required: **yes**

Admin required: **no** - *but teams can only see their own submissions, admins all*

Response body

```
[  
  {  
    "team_name": "String",  
    "problem_id": "int",  
    "submission": "varchar(255)",  
    "grading_description": "String",  
    "score": "int",  
    "used_hint": "boolean"  
  }  
]
```

POST /submissions/{problem_id}

The team submits a solution to a problem

Authorization token required: **yes**

Admin required: **no**

Request body

File (picture)

GET /challenges

Returns a list of all challenges

Authorization token required: **yes**

Admin required: **no**

Response body

```
[
  {
    "problem_name": "String",
    "problem_id": "int",
    "location_id": "int",
    "score": "int",          //NOTE: Mostly fixed accross all
    challenges.
    "description": "string"
  }
]
```

GET /challenges/{problem_id}

Returns the challenge with the given problem id

Authorization token required: **yes**

Admin required: **no**

Response body

```
{
  "problem_name": "String",
  "problem_id": "int",
  "location_id": "int",
  "score": "int",          //NOTE: Mostly fixed accross all
  challenges.
  "description": "string"
}
```

POST /challenges

Creates a new challenge

Authorization token required: **yes**

Admin required: **yes**

Request body

```
{
  "problem_name": "String",
  "location_id": "int",
```

```
    "score": "int",           //NOTE: Mostly fixed accross all
    challenges.
    "description": "string"
  }
```

Response body

```
{
  "problem_id": "int"
}
```

PATCH /challenges/{problem_id}

Updates the challenge with the given problem id

Request body

These attributes are all optional, only overwrite in the database the ones that are present in the request body

```
{
  "problem_name": "String",
  "location_id": "int",
  "score": "int",           //NOTE: Mostly fixed accross all
  challenges.
  "description": "string"
}
```

GET /locations

Returns a list of all locations

Authorization token required: **yes**

Admin required: **no**

Response body

```
[
  {
    "location_id": "int",
    "location_name" : "String",
    "challenges": ["Challenge object"]
  }
]
```

GET /location/{location_id}

Returns the location with the given location id

Authorization token required: **yes**

Admin required: **no**

Response body

```
{
  "location_id": "int",
  "location_name" : "String",
  "challenges": ["Challenge object"]
}
```

POST /locations

Creates a new location

Authorization token required: **yes**

Admin required: **yes**

Request body

```
{
  "location_id": "int",
  "location_name" : "String",
  "challenges": ["Challenge object"]
}
```

Response body

```
{
  "location_id": "int"
}
```

PATCH /locations/{location}

Updates the location with the given location id

Request body

These attributes are all optional, only overwrite in the database the ones that are present in the request body

```
{  
  "location_id": "int",  
  "location_name" : "String",  
  "challenges": ["Challenge object"]  
}
```