

Api

GET	/score/GetAll
POST	/score/AddScore
GET	/score/GetScoreById/{id}
DELETE	/score/DeleteScoreById/{id}
PUT	/score/UpdateScoreById/{id}
GET	/score/GetScoreByName/{name}
GET	/score/GetGradeAScoreByModule/{module}
GET	/score/GetAverageGradeOfEachModule

GetAll

Get all scores ranked by `createdAt` in descending order.

Success:

Return `200 OK` with the list of scores.

AddScore

Add a score to database.

Fail:

1. If `mark` is not between 0 and 100
 - a. Return BadRequest with message **Score must be between 0 and 100**
2. If there is already an existing one
 - a. Return BadRequest with message **Already exists!**

Success:

Return `200 OK` with the added score being returned message.

GetScoreById

Get a score by its id.

Fail:

1. If there is not such a score
 - a. Return NotFound with message **Score with id {id} not found**

Success:

Return `200 OK` with the score being returned message.

DeleteScoreById

Delete a score by its id.

Fail:

1. If there is not such a score
 - a. Return NotFound with message **Score with id {id} not found**

Success:

Return `200 OK` with message **{id} has been removed!**

UpdateScoreById

Update a score by its id.

Fail:

1. If there is not such a score
 - a. Return NotFound with message **Score with id {id} not found**

Success:

Return `204 No Content`.

GetScoreByName

Get all scores by student's name (non-case sensitive).

Success:

Return `200 OK` with all scores belonging to that student.

GetGradeAScoreByModule

Get all scores of a module with marks at least 80.

Fail:

1. The module name being passed is an empty string or strings with only white spaces.
 - a. Return BadRequest with message **Module is required**
2. If there is no such a module:
 - a. Return NotFound with message **No such a module - {module} !**

Success:

Return `200 OK` with all scores that contain only `mark` and `name`. Sort the data by `mark` in descending order.

Example of Return Message:

```
[  
  {  
    "name": "Alex",  
    "mark": 87.5  
  },  
  {  
    "name": "John",  
    "mark": 81  
  }  
]
```

GetAverageScoreOfEachModule

Calculate the average score of each module.

Success:

Return `200 OK` with average score of each module. Sort the data by `mark` in descending order.

Example of Return Message:

```
[  
  {  
    "module": "mad",  
    "mark": 79.783333  
  },  
  {  
    "module": "edp",  
    "mark": 70.666667  
  }  
]
```