

Student Competitions API — Test Cases (Short Template)

Filters

1. Filter participations by year (less)

Шаги:

- Send POST request to `/api/participations/search`

Тело запроса:

```
{  
  "filters": [  
    {  
      "field": "year",  
      "operation": "less",  
      "value": 2023  
    }  
  ]  
}
```

Ожидаемый результат:

- HTTP 200 OK
- Получаем записи, где `year < 2023`
- Пример тела ответа:

```
{  
  "id": "a5ad4f26-0527-406a-919e-9d8c7031ee56",  
  "student": "http://localhost:8080/api/students/52010de1-e416-41a8-8e57-54299f781ce2",  
  "competition": "http://localhost:8080/api/competitions/b55102d6-1749-49cc-9864-6244a36382",  
  "level": "http://localhost:8080/api/levels/REG",  
  "result": "http://localhost:8080/api/results/PRIIZE",  
  "year": 2022,  
  "supervisor": "Доц. Петрова Е.В.",  
  "points": null,  
  "description": "Призер регионального CTF",  
  "isBlocked": true  
}
```

2. Filter participations by year (less or equals)

Шаги:

- Send POST request to `/api/participations/search`

Тело запроса:

```
{  
  "filters": [  
    {  
      "field": "year",  
      "operation": "less or equals",  
      "value": 2022  
    }  
  ]  
}
```

Ожидаемый результат:

- HTTP 200 OK
- Получаем записи, где `year <= 2022`
- Пример тела ответа:

```
{  
  "id": "b82dac31-cd6a-445e-ae29-69085bbfe069",  
  "student": "http://localhost:8080/api/students/ed27c4f1-891f-4715-b3f6-5e17c5fa2ed2",  
  "competition": "http://localhost:8080/api/competitions/949b1328-2f66-4d56-b5f8-b99e2aa3d1",  
  "level": "http://localhost:8080/api/levels/UNI",  
  "result": "http://localhost:8080/api/results/DIP",  
  "year": 2022,  
  "supervisor": "Проф. Иванова М.П.",  
  "points": null,  
  "description": "Диплом за перспективное исследование",  
  "isBlocked": true  
},
```

3. Filter participations by year (between)

Шаги:

- Send POST request to `/api/participations/search`

Тело запроса:

```
{  
  "filters": [  
    {  
      "field": "year",  
      "operation": "between",  
      "value": [2021, 2022]  
    }  
  ]  
}
```

```
        "value": 2024,  
        "value2": 2030  
    }  
]  
}
```

Ожидаемый результат:

- HTTP 200 OK
- Получаем записи, где `2024 <= year <= 2030`
- Пример тела ответа:

```
{  
    "id": "919ff010-db7c-4290-b4d7-58f165745c5f",  
    "student": "http://localhost:8080/api/students/12f002fb-e957-4922-9863-0133b883c340",  
    "competition": "http://localhost:8080/api/competitions/693fb3e5-b5ff-4111-bf3d-7697009d65",  
    "level": "http://localhost:8080/api/levels/RUS",  
    "result": "http://localhost:8080/api/results/WIN",  
    "year": 2024,  
    "supervisor": "Доц. Петров К.К.",  
    "points": null,  
    "description": "Победитель заключительного этапа",  
    "isBlocked": false  
},
```

4. Filter participations by year (greater)

Шаги:

- Send POST request to `/api/participations/search`

Тело запроса:

```
{  
    "filters": [  
        {  
            "field": "year",  
            "operation": "greater",  
            "value": 2023  
        }  
    ]  
}
```

Ожидаемый результат:

- HTTP 200 OK
- Получаем записи, где `year > 2023`
- Пример тела ответа:

```
{  
    "id": "919ff010-db7c-4290-b4d7-58f165745c5f",  
    "student": "http://localhost:8080/api/students/12f002fb-e957-4922-9863-0133b883c340",  
    "competition": "http://localhost:8080/api/competitions/693fb3e5-b5ff-4111-bf3d-7697009d65",  
    "level": "http://localhost:8080/api/levels/RUS",  
    "result": "http://localhost:8080/api/results/WIN",  
    "year": 2024,  
    "supervisor": "Доц. Петров К.К.",  
    "points": null,  
    "description": "Победитель заключительного этапа",  
    "isBlocked": false  
},
```

5. Filter participations by year (equals)

Шаги:

- Send POST request to /api/participations/search

Тело запроса:

```
{  
    "filters": [  
        {  
            "field": "year",  
            "operation": "equals",  
            "value": 2023  
        }  
    ]  
}
```

Ожидаемый результат:

- HTTP 200 OK
- Получаем записи, где year = 2023

6. Filter participations by supervisor (starts with)

Шаги:

- Send POST request to /api/participations/search

Тело запроса:

```
{  
    "filters": [  
        {  
            "field": "supervisor",  
            "operation": "startsWith",  
            "value": "Доц."  
        }  
    ]  
}
```

```
        "operation": "starts with",
        "value": "Д"
    }
]
}
```

Ожидаемый результат:

- HTTP 200 OK
- Поле Supervisor начинается с символа "Д"

7. Filter participations by supervisor (ends with)

Шаги:

- Send POST request to /api/participations/search

Тело запроса:

```
{
  "filters": [
    {
      "field": "supervisor",
      "operation": "ends with",
      "value": "."
    }
  ]
}
```

Ожидаемый результат:

- HTTP 200 OK
- Поле Supervisor оканчивается на символ "."

Students

8. Create student

Шаги:

- Send POST request to /api/students

Тело запроса:

```
{
  "name": "aleksander",
  "surname": "slelin",
```

```
"middleName": "vladimirovich",
"cardNumber": 1036576,
"faculty": "Faculty of Mathematics",
"department": null,
"studyGroup": "PMI-11MO",
"email": "a.slelin.work@mail.ru",
"phone": null
}
```

Ожидаемый результат:

- HTTP 201 Created
- Пустое тело ответа
- В переменную `studentID` заносится url с созданным студентом

9. Get student

Шаги:

- Send GET request to `/api/students/{studentId}`

Ожидаемый результат:

- HTTP 200 OK
- Тело ответа содержит информацию о студенте с заданным `id`

```
{
  "id": "9ab96b9d-bd00-4cf5-a0ee-a7bc8fac5ba3",
  "name": "aleksander",
  "surname": "slelin",
  "middleName": "vladimirovich",
  "cardNumber": 1036576,
  "faculty": "Математический факультет",
  "department": null,
  "studyGroup": "PMI-11MO",
  "email": "a.slelin.work@mail.ru",
  "phone": null
}
```

10. Delete student

Шаги:

- Send DELETE request to `/api/students/{studentId}`

Ожидаемый результат:

- HTTP 204 No Content

- Удаление студента с заданным `id`
-

Competitions

11. Create competition

Шаги:

- Send POST request to `/api/competitions`

Тело запроса:

```
{  
  "name": "Bear&Mouse",  
  "organizer": "NPO Krissta",  
  "type": "olympiad"  
}
```

Ожидаемый результат:

- HTTP 201 Created
 - В переменную `competitionID` заносится информация о созданном соревновании
-

12. Get competition

Шаги:

- Send GET request to `/api/competitions/{competitionId}`

Ожидаемый результат:

- HTTP 200 OK
- Тело ответа содержит информацию о соревновании с заданным `id`

```
{  
  "id": "13cb3327-ef09-4170-927b-785a4927acbb",  
  "name": "Bear&Mouse",  
  "organizer": "NPO Krissta",  
  "type": "Олимпиада"  
}
```

13. Delete competition

Шаги:

- Send DELETE request to /api/competitions/{competitionId}

Ожидаемый результат:

- HTTP 204 No Content
- Удаление соревнования с заданным id

Levels

14. Get levels

Шаги:

- Send GET request to /api/levels

Ожидаемый результат:

- HTTP 200 OK
- Тело ответа содержит список уровней

```
{  
    "content": [  
        {  
            "code": "UNI",  
            "name": "Внутривузовский уровень",  
            "order": 1  
        },  
        {  
            "code": "REG",  
            "name": "Региональный уровень",  
            "order": 2  
        },  
        {  
            "code": "RUS",  
            "name": "Всероссийский уровень",  
            "order": 3  
        },  
        {  
            "code": "INT",  
            "name": "Международный уровень",  
            "order": 4  
        }  
    "page": {  
        "number": 0,  
        "size": 10,  
        "sorts": [],  
        "totalElements": 4,  
        "totalPages": 1,  
        "first": true,  
        "last": true  
    }  
}
```

```
        "last": true,  
        "empty": false  
    }  
}
```

Results

15. Get results

Шаги:

- Send GET request to /api/results

Ожидаемый результат:

- HTTP 200 OK
- Тело ответа содержит список возможных результатов

```
{  
    "content": [  
        {  
            "code": "WIN",  
            "name": "Победитель",  
            "order": 1  
        },  
        {  
            "code": "PRIZE",  
            "name": "Призер",  
            "order": 2  
        },  
        {  
            "code": "PART",  
            "name": "Участник",  
            "order": 3  
        },  
        {  
            "code": "DIQ",  
            "name": "Дисквалифицирован",  
            "order": 4  
        },  
        {  
            "code": "DIP",  
            "name": "Дипломант",  
            "order": 5  
        }  
    "page": {  
        "number": 0,  
        "size": 10,  
        "sorts": []  
    }  
}
```

```
        "totalElements": 5,  
        "totalPages": 1,  
        "first": true,  
        "last": true,  
        "empty": false  
    }  
}
```

Participations

16. Get all participations

Шаги:

- Send GET request to `/api/participations?page=5`

Ожидаемый результат:

- HTTP 200 OK
- Пример тела ответа

```
{  
    "id": "e640bdfd-e1b2-480f-8f1d-71afa5b4a57f",  
    "student": "http://localhost:8080/api/students/1e39cfb2-c09c-4363-927b-90e3e79d9321",  
    "competition": "http://localhost:8080/api/competitions/bcef57df-ebb8-4a90-ac4e-4fea703c6f",  
    "level": "http://localhost:8080/api/levels/REG",  
    "result": "http://localhost:8080/api/results/PART",  
    "year": 2024,  
    "supervisor": null,  
    "points": null,  
    "description": "Участник регионального этапа",  
    "isBlocked": false  
}
```

17. Create participation

Шаги:

- Send POST request to `/api/participations`

Тело запроса:

```
{  
    "student": "{studentId}",  
    "competition": "{competitionId}",  
    "level": "REG",  
    "result": "PART",  
    "year": 2024,  
    "supervisor": null,  
    "points": null,  
    "description": "Участник регионального этапа",  
    "isBlocked": false  
}
```

```
"year": 2030,  
"supervisor": "Проф. Смирнов А.А.",  
"points": 0,  
"description": "Описание"  
}
```

Ожидаемый результат:

- HTTP 201 Created
- Создание участия студента в соревновании

18. Get participation

Шаги:

- Send GET request to /api/participations/{participationId}

Ожидаемый результат:

- HTTP 200 OK
- Тело ответа содержит информацию о конкретном участии студента в соревновании

```
{  
    "id": "e94c5514-34af-4cf7-aa77-069088ac7cce",  
    "student": "http://localhost:8080/api/students/405a05ad-0da1-44f3-bcb0-30a438a29217",  
    "competition": "http://localhost:8080/api/competitions/4b6268ef-cef8-4a7d-acbe-ed864ab2b9",  
    "level": "http://localhost:8080/api/levels/REG",  
    "result": "http://localhost:8080/api/results/PART",  
    "year": 2032,  
    "supervisor": "Проф. Смирнов А.А.",  
    "points": 0,  
    "description": "Описание",  
    "isBlocked": false  
}
```

19. Update participation

Шаги:

- Send PATCH request to /api/participations/{participationId}

Тело запроса:

```
{  
    "points": 1,  
    "description": "Описание 2"  
}
```

Ожидаемый результат:

- HTTP 200 OK
- Частичное обновление информации об участии студента в соревновании
- Пример тела ответа

```
{  
    "id": "243d4cc9-d60e-4746-acee-9a9c86efb8bb",  
    "student": "http://localhost:8080/api/students/8849efa3-72f3-480a-8f09-32fa4a3dd1b6",  
    "competition": "http://localhost:8080/api/competitions/e582899c-c6f7-4830-b462-5169faf390",  
    "level": "http://localhost:8080/api/levels/REG",  
    "result": "http://localhost:8080/api/results/PART",  
    "year": 2030,  
    "supervisor": "Проф. Смирнов А.А.",  
    "points": 1,  
    "description": "Описание 2",  
    "isBlocked": false  
}
```

20. Delete participation

Шаги:

- Send DELETE request to /api/participations/{participationId}

Ожидаемый результат:

- HTTP 204 No Content
- Удаление участия студента в соревновании

Negative cases

21. Get student

Шаги:

- Send GET request to /api/students/{studentID}

Ожидаемый результат:

- HTTP 404 Not Found
- Пример тела ответа, если запрашиваемой информации нет

```
{  
    "path": "/api/students/8849efa3-72f3-480a-8f09-32fa4a3dd1b6",  
    "method": "GET",  
    "httpStatus": 404,
```

```
"debugMessage": "Entity not found.",  
"message": "Entity \"Student\" not found by field = \"id\" with value = \"8849efa3-72f3-4  
"exception": "EntityNotFoundException",  
"causeException": null,  
"details": {  
    "value": "8849efa3-72f3-480a-8f09-32fa4a3dd1b6",  
    "field": "id",  
    "class": "Student"  
},  
"timeStamp": "2025-12-24T07:39:44.3593443"  
}
```

22. Create student

- Send POST request to /api/students
- Поле cardNumber из тела запроса уже должно существовать в БД
- Пример тела запроса

```
{  
    "name": "aleksander",  
    "surname": "slelin",  
    "middleName": "vladimirovich",  
    "cardNumber": 1036576,  
    "faculty": "Faculty of Mathematics",  
    "department": null,  
    "studyGroup": "PMI-11MO",  
    "email": "a.slelin.work@mail.ru",  
    "phone": null  
}
```

Ожидаемый результат:

- HTTP 400 Bad Request
- Ожидаемое тело ответа

```
{  
    "path": "/api/students",  
    "method": "POST",  
    "httpStatus": 400,  
    "debugMessage": "Entity's field is not unique.",  
    "message": "Field \"cardNumber\" from entity \"Student\" with value \"1036576\" is not un  
"exception": "NotUniqueFieldException",  
    "causeException": null,  
    "details": {  
        "value": 1036576,  
        "field": "cardNumber",  
        "class": "Student"  
},  
    "timeStamp": "2025-12-24T07:45:42.9310817"  
}
```

