



Public

Run in Postman

ENVIRONMENT No Environment ▾

LAYOUT Single Column ▾

LANGUAGE cURL ▾



## SAVS FINAL INTEGRATION TEST SUITE

## Introduction

**GET** Clear database**POST** Signup a student**POST** Signup a student | Missing Informati...**POST** Signup a student | Student already e...**GET** Get Student Information | Unregiste...**GET** Get Student Information**POST** Signup a student for club member**POST** Signup a student for club member 2**POST** Signup a student for club head**POST** Signup a student for club head 2**POST** Create a new club | Unauthorized**POST** Create a new club | Missing Informati...**POST** Create a new club**POST** Create a new club | Duplicate Club ...**POST** Create a new club 2**PUT** Add a member to club 1**PUT** Add a member to club 2**PUT** Add a member 2 to club 2**DEL** Remove a member form club 1**PUT** Add a member back to club**POST** Propose Event Wrong Club ID**POST** Propose Event 1**POST** Propose Event 2

# SAVS Final Integration Test suite

This is a complete test suite for testing all endpoints for SAVS app. Before running this please run the authentication suite, so that all the authentication tokens are set.

To run this collection -

- click on the collection name to view the home page.
- click on the Run button at top right corner
- all the tests will be run one by one.

Note: the tests need to be run in the specified order. Note: If you are seeing unauthorized anywhere you shouldn't, try running the authentication again. It is probably because the tokens were not populated.

If you wish to experiment more and get yourself added as a student here!!

- Go back to Authentication suite and follow steps to authenticate yourself.
- Duplicate the test to register as a student.
- Replace the details in request body with your details.
- We are good to go! experiment and expect results that are limited to student.
- If you wish to become a head of a club or member.
- Create a new club by duplicating the test in similar way with yourself as head or someone else who is registered as a student.
- get yourself added as a member,

## GET Clear database

```
curl --location -g --request GET '{{hHost}}/database/clear_database' \
```

We are clearing database here, so that we can run a full suite of tests from scratch.

### Example Request

### Clear database

```
curl --location -g --request GET '{{hHost}}/database/clear_database' \
```

## POST Signup a student

```
curl --location -g --request POST '{{hHost}}/student' \
```

Here we test that a student can sign up on the app. The student we sign up here is a sample student (Relevant for tests ahead)

### HEADERS

Authorization {{token\_student}}

### BODY raw

```
{  
  "name": "TestStudent",  
  "email_id": "teststudentsavs@gmail.com",  
  "college": "Fu Foundation",  
  "department": "Computer Science"  
}
```

### Example Request

### Signup a student

```
curl --location -g --request POST '{{hHost}}/student' \
```

```
--header 'Authorization: {{token_student}}' \
```

```
--data-raw '{  
  "name": "TestStudent",  
  "email_id": "teststudentsavs@gmail.com",  
  "college": "Fu Foundation",  
  "department": "Computer Science"  
}'
```

## POST Signup a student | Missing Information

```
curl --location -g --request POST '{{hHost}}/student'
```

Here we test that if the student can sign up on the app with missing information.

### HEADERS

Authorization {{token\_student}}

### BODY raw

```
{
  "name": "TestStudent",
  "college": "Fu Foundation",
  "department": "Computer Science"
}
```

Example Request

Signup a student | Missing Information

```
curl --location -g --request POST '{{hHost}}/student' \
--header 'Authorization: {{token_student}}' \
--data-raw '{
  "name": "TestStudent",
  "college": "Fu Foundation",
  "department": "Computer Science"
}'
```

## POST Signup a student | Student already exists

```
curl --location -g --request POST '{{hHost}}/student'
```

Here we test that a student can sign up on the app more than once.

### HEADERS

Authorization {{token\_student}}

### BODY raw

```
{
  "name": "TestStudent",
  "email_id": "teststudentsavs@gmail.com",
  "college": "Fu Foundation",
  "department": "Computer Science"
}
```

Example Request

Signup a student | Student already exists

```
curl --location -g --request POST '{{hHost}}/student' \
--header 'Authorization: {{token_student}}' \
--data-raw '{
  "name": "TestStudent",
  "email_id": "teststudentsavs@gmail.com",
  "college": "Fu Foundation",
  "department": "Computer Science"
}'
```

## GET Get Student Information | Unregistered student

```
curl --location -g --request GET '{{hHost}}/student/test_unregistered_student@columbia.edu'
```

We test that the admin/club member/club head/student can view the details of the student who is unregistered.

### HEADERS

Authorization {{token\_student}}

Example Request

Get Student Information | Unregistered student

```
curl --location -g --request GET '{{hHost}}/student/test_unregistered_student@columbia.edu' \
--header 'Authorization: {{token_student}}'
```

## GET Get Student Information

```
 {{hHost}}/student/teststudentsavs@gmail.com
```

We test that the admin/club member/club head/student can view the details of the student.

### HEADERS

Authorization      {{token\_student}}

Example Request

Get Student Information

```
curl --location -g --request GET '{{hHost}}/student/teststudentsavs@gmail.com' \
--header 'Authorization: {{token_student}}'
```

## POST Signup a student for club member

```
 {{hHost}}/student
```

Here we sign up a student that acts as club member later in the tests.

### HEADERS

Authorization      {{token\_student}}

### BODY raw

```
{
  "name": "TestClubMember",
  "email_id": "testclubmembersavs@gmail.com",
  "college": "Fu Foundation",
  "department": "Computer Science"
}
```

Example Request

Signup a student for club member

```
curl --location -g --request POST '{{hHost}}/student' \
--header 'Authorization: {{token_student}}' \
--data-raw '{
  "name": "TestClubMember",
  "email_id": "testclubmembersavs@gmail.com",
  "college": "Fu Foundation",
  "department": "Computer Science"
}'
```

## POST Signup a student for club member 2

```
 {{hHost}}/student
```

Here we sign up another student that acts as club member later in the tests.

### HEADERS

Authorization      {{token\_student}}

### BODY raw

```
{
  "name": "TestClubMember2",
  "email_id": "testclubmember2savs@gmail.com",
  "college": "Fu Foundation",
  "department": "Computer Science"
}
```

Example Request

Signup a student for club member 2

```
curl --location -g --request POST '{{hHost}}/student' \
--header 'Authorization: {{token_student}}' \
--data-raw '{
  "name": "TestClubMember2",
  "email_id": "testclubmember2savs@gmail.com",
  "college": "Fu Foundation",
  "department": "Computer Science"
}'
```

```
"department": "Computer Science"  
}'
```

#### POST Signup a student for club head

```
curl --location --request POST '{{hHost}}/student'
```

Here we sign up a student that acts as club head for TestClub1 later in the tests.

#### HEADERS

Authorization {{token\_student}}

#### BODY raw

```
{  
  "name": "TestClubHead",  
  "email_id": "testclubheadsavs@gmail.com",  
  "college": "Fu Foundation",  
  "department": "Computer Science"  
}'
```

Example Request

Signup a student for club head

```
curl --location --request POST '{{hHost}}/student' \  
--header 'Authorization: {{token_student}}' \  
--data-raw '{  
  "name": "TestClubHead",  
  "email_id": "testclubheadsavs@gmail.com",  
  "college": "Fu Foundation",  
  "department": "Computer Science"  
}'
```

#### POST Signup a student for club head 2

```
curl --location --request POST '{{hHost}}/student'
```

Here we sign up a student that acts as club head for TestClub2 later in the tests.

#### HEADERS

Authorization {{token\_student}}

#### BODY raw

```
{  
  "name": "TestClubHead2",  
  "email_id": "testclubheadsavs2@gmail.com",  
  "college": "Fu Foundation",  
  "department": "Computer Science"  
}'
```

Example Request

Signup a student for club head 2

```
curl --location --request POST '{{hHost}}/student' \  
--header 'Authorization: {{token_student}}' \  
--data-raw '{  
  "name": "TestClubHead2",  
  "email_id": "testclubheadsavs2@gmail.com",  
  "college": "Fu Foundation",  
  "department": "Computer Science"  
}'
```

#### POST Create a new club | Unauthorized

```
curl --location --request POST '{{hHost}}/student/club'
```

Here we test creation of a new club Club1 by a student who is not registered.

#### HEADERS

Authorization {{token\_invalid}}

#### BODY raw

```
{
  "club": {
    "name": "Test Club 1",
    "head": "test_club_head@columbia.edu",
    "category": "Test Category 1",
    "description": "Test Club Description 1"
  }
}
```

Example Request Create a new club | Unauthorized

```
curl --location -g --request POST '{{hHost}}/student/club' \
--header 'Authorization: {{token_invalid}}' \
--data-raw '{
  "club": {
    "name": "Test Club 1",
    "head": "test_club_head@columbia.edu",
    "category": "Test Category 1",
    "description": "Test Club Description 1"
  }
}'
```

[View More](#)

#### POST Create a new club | Missing Information

[{{hHost}}/student/club](#)

Here we test creation of a new club Club1 by a student without complete information.

#### HEADERS

Authorization {{token\_student}}

#### BODY raw

```
{
  "club": {
    "head": "testclubheadsav@gmail.com",
    "category": "Test Category 1",
    "description": "Test Club Description 1"
  }
}
```

Example Request Create a new club | Missing Information

```
curl --location -g --request POST '{{hHost}}/student/club' \
--header 'Authorization: {{token_student}}' \
--data-raw '{
  "club": {
    "head": "testclubheadsav@gmail.com",
    "category": "Test Category 1",
    "description": "Test Club Description 1"
  }
}'
```

[View More](#)

#### POST Create a new club

[{{hHost}}/student/club](#)

Here we test creation of a new club Club1 by a student. Email id is used to check if the student exists and is eligible to be head of a club.

#### HEADERS

Authorization {{token\_student}}

#### BODY raw

```
{
  "club": {
    "name": "Test Club 1",
    "head": "testclubheadsav@gmail.com",
    "category": "Test Category 1",
    "description": "Test Club Description 1"
  }
}
```

Example Request Create a new club

```

curl --location -g --request POST '{{hHost}}/student/club' \
--header 'Authorization: {{token_student}}' \
--data-raw '{
  "club": {
    "name": "Test Club 1",
    "head": "testclubheadsav2@gmail.com",
    "category": "Test Category 1",
    "description": "Test Club Description 1"
  }
}'

```

[View More](#)

#### POST Create a new club | Duplicate Club Name

[{{hHost}}/student/club](#)

Here we test creation of a new club with duplicate name by a student.

#### HEADERS

Authorization      {{token\_student}}

#### BODY raw

```
{
  "club": {
    "name": "Test Club 1",
    "head": "testclubheadsav2@gmail.com",
    "category": "Test Category 1",
    "description": "Test Club Description 1"
  }
}
```

Example Request

Create a new club | Duplicate Club Name

```

curl --location -g --request POST '{{hHost}}/student/club' \
--header 'Authorization: {{token_student}}' \
--data-raw '{
  "club": {
    "name": "Test Club 1",
    "head": "testclubheadsav2@gmail.com",
    "category": "Test Category 1",
    "description": "Test Club Description 1"
  }
}'

```

[View More](#)

#### POST Create a new club 2

[{{hHost}}/student/club](#)

Here we test creation of a new club Club2 by a student. Email id is used to check if the student exists and is eligible to be head of the club. This will be used in tests later.

#### HEADERS

Authorization      {{token\_student}}

#### BODY raw

```
{
  "club": {
    "name": "Test Club 2",
    "head": "testclubheadsav2@gmail.com",
    "category": "Test Category 2",
    "description": "Test Club Description 2"
  }
}
```

Example Request

Create a new club 2

```

curl --location -g --request POST '{{hHost}}/student/club' \
--header 'Authorization: {{token_student}}' \
--data-raw '{
  "club": {
    "name": "Test Club 2",
    "head": "testclubheadsav2@gmail.com",
    "category": "Test Category 2",
    "description": "Test Club Description 2"
  }
}'

```

[View More](#)

#### PUT Add a member to club 1

```
 {{hHost}}/member/1
```

Here we test that the club head is able to add a member to their club .

#### HEADERS

Authorization {{token\_club\_head}}

#### BODY raw

```
{  
    "student_email_id": "testclubmembersavs@gmail.com"  
}
```

Example Request

Add a member to club 1

```
curl --location -g --request PUT '{{hHost}}/member/1' \  
--header 'Authorization: {{token_club_head}}' \  
--data-raw '{  
    "student_email_id": "testclubmembersavs@gmail.com"  
}'
```

### PUT Add a member to club 2

```
 {{hHost}}/member/2
```

Here we test that the club head is able to add a member to their club .

#### HEADERS

Authorization {{token\_club\_head\_2}}

#### BODY raw

```
{  
    "student_email_id": "testclubmembersavs@gmail.com"  
}
```

Example Request

Add a member to club 2

```
curl --location -g --request PUT '{{hHost}}/member/2' \  
--header 'Authorization: {{token_club_head_2}}' \  
--data-raw '{  
    "student_email_id": "testclubmembersavs@gmail.com"  
}'
```

### PUT Add a member 2 to club 2

```
 {{hHost}}/member/2
```

Here we test that the club head is able to add a member to their club .

#### HEADERS

Authorization {{token\_club\_head\_2}}

#### BODY raw

```
{  
    "student_email_id": "testclubmember2savs@gmail.com"  
}
```

Example Request

Add a member 2 to club 2

```
curl --location -g --request PUT '{{hHost}}/member/2' \  
--header 'Authorization: {{token_club_head_2}}' \  
--data-raw '{  
    "student_email_id": "testclubmember2savs@gmail.com"  
}'
```

### DEL Remove a member from club 1

```
 {{hHost}}/member/1
```

Here we test that the club head is able to remove a member from their club.

#### HEADERS

Authorization {{token\_club\_head}}

#### BODY raw

```
{  
    "student_email_id": "testclubmembersavs@gmail.com"  
}
```

Example Request

Remove a member from club 1

```
curl --location -g --request DELETE '{{hHost}}/member/1' \  
--header 'Authorization: {{token_club_head}}' \  
--data-raw '{  
    "student_email_id": "testclubmembersavs@gmail.com"  
}'
```

### PUT Add a member back to club

```
 {{hHost}}/member/1
```

Here we add back the member we removed.

#### HEADERS

Authorization {{token\_club\_head}}

#### BODY raw

```
{  
    "student_email_id": "testclubmembersavs@gmail.com"  
}
```

Example Request

Add a member back to club

```
curl --location -g --request PUT '{{hHost}}/member/1' \  
--header 'Authorization: {{token_club_head}}' \  
--data-raw '{  
    "student_email_id": "testclubmembersavs@gmail.com"  
}'
```

### POST Propose Event Wrong Club ID

```
 {{hHost}}/events
```

Here we test if a club member can propose an event for a club they are not member of.

#### HEADERS

Authorization {{token\_club\_head\_2}}

#### BODY raw

```
{  
    "event": {  
        "name": "Bug Testing 2",  
        "club_id": 1,  
        "start_timestamp": "2022-12-03 09:30:00",  
        "end_timestamp": "2022-12-05 00:00:00",  
        "location": "SF",  
        "max_registration": 250,  
        "description": "Bug Testing ~~~~~",  
        "fee": 15,  
        "category": "Academic"  
    }  
}
```

[View More](#)

Example Request

Propose Event Wrong Club ID

```

curl --location -g --request POST '{{hHost}}/events' \
--header 'Authorization: {{token_club_head_2}}' \
--data-raw '{
  "event": {
    "name": "Bug Testing 2",
    "club_id": 1,
    "start_timestamp": "2022-12-03 09:30:00",
    "end_timestamp": "2022-12-05 00:00:00",
    "location": "SF",
    "max_registration": 250
}

```

[View More](#)

### POST Propose Event 1

curl {{hHost}}/events

Here we test that a club member is able to propose an event.

#### HEADERS

Authorization {{token\_club\_member}}

#### BODY raw

```

{
  "event": {
    "name": "TestEvent1",
    "club_id": 1,
    "start_timestamp": "2022-12-03 09:30:00",
    "end_timestamp": "2022-12-05 00:00:00",
    "location": "New York City",
    "max_registration": 125,
    "description": "Winter Hackathon 2021",
    "fee": 10,
    "category": "Academic"
}

```

[View More](#)

Example Request

Propose Event 1

```

curl --location -g --request POST '{{hHost}}/events' \
--header 'Authorization: {{token_club_member}}' \
--data-raw '{
  "event": {
    "name": "TestEvent1",
    "club_id": 1,
    "start_timestamp": "2022-12-03 09:30:00",
    "end_timestamp": "2022-12-05 00:00:00",
    "location": "New York City",
    "max_registration": 125
}

```

[View More](#)

### POST Propose Event 2

curl {{hHost}}/events

Here we test that a club member is able to propose another event. This will be used for testing later.

#### HEADERS

Authorization {{token\_club\_member}}

#### BODY raw

```

{
  "event": {
    "name": "TestEvent2",
    "club_id": 1,
    "start_timestamp": "2020-12-03 09:30:00",
    "end_timestamp": "2020-12-05 00:00:00",
    "location": "New York City",
    "max_registration": 125,
    "description": "Winter Hackathon 2021",
    "fee": 10,
    "category": "Academic"
}

```

[View More](#)

Example Request

Propose Event 2

```

curl --location -g --request POST '{{hHost}}/events' \
--header 'Authorization: {{token_club_member}}' \
--data-raw '{
  "event": {
    "name": "TestEvent2",
    "club_id": 1,
    "start_timestamp": "2020-12-03 09:30:00",
    "end_timestamp": "2020-12-05 00:00:00",
    "location": "New York City",
    "max_registration": 125
}

```

[View More](#)

### POST Propose Event 3 by student

```
 {{hHost}}/events
```

Here we test that a club member is able to propose another event. This will be used for testing later.

#### HEADERS

Authorization {{token\_student}}

#### BODY raw

```
{  
  "event": {  
    "name": "TestEvent3",  
    "club_id": 1,  
    "start_timestamp": "2020-12-03 09:30:00",  
    "end_timestamp": "2020-12-05 00:00:00",  
    "location": "New York City",  
    "max_registration": 125,  
    "description": "Winter Hackathon 2021",  
    "fee": 10, View More  
    "category": "Academic"
```

Example Request

Propose Event 3 by student

```
curl --location -g --request POST '{{hHost}}/events' \  
--header 'Authorization: {{token_student}}' \  
--data-raw '{  
  "event": {  
    "name": "TestEvent3",  
    "club_id": 1,  
    "start_timestamp": "2020-12-03 09:30:00",  
    "end_timestamp": "2020-12-05 00:00:00",  
    "location": "New York City", View More  
    "max_registration": 125,
```

### POST Propose Event 3 | Low capacity

```
 {{hHost}}/events
```

Here we test that a club member is able to propose an event(note the capacity is 1 ) This is used for testing later.

#### HEADERS

Authorization {{token\_club\_member}}

#### BODY raw

```
{  
  "event": {  
    "name": "TestEvent3",  
    "club_id": 1,  
    "start_timestamp": "2022-12-03 09:30:00",  
    "end_timestamp": "2022-12-05 00:00:00",  
    "location": "New York City",  
    "max_registration": 1,  
    "description": "Winter Hackathon 2021",  
    "fee": 10, View More  
    "category": "Academic"
```

Example Request

Propose Event 3 | Low capacity

```
curl --location -g --request POST '{{hHost}}/events' \  
--header 'Authorization: {{token_club_member}}' \  
--data-raw '{  
  "event": {  
    "name": "TestEvent3",  
    "club_id": 1,  
    "start_timestamp": "2022-12-03 09:30:00",  
    "end_timestamp": "2022-12-05 00:00:00",  
    "location": "New York City", View More  
    "max_registration": 1,
```

### POST Propose Event 4

```
 {{hHost}}/events
```

Here we test that a club member 2 is able to propose another event for club 2. This will be used for testing later.

#### HEADERS

Authorization {{token\_club\_member\_2}}

#### BODY raw

```
{  
  "event": {  
    "name": "TestEvent4",  
    "club_id": 2,  
    "start_timestamp": "2020-12-03 09:30:00",  
    "end_timestamp": "2020-12-05 00:00:00",  
    "location": "New York City",  
    "max_registration": 125,  
    "description": "Winter Hackathon 2021",  
    "fee": 10,  
    "category": "Academic"}  
}
```

[View More](#)

#### Example Request

#### Propose Event 4

```
curl --location -g --request POST '{{hHost}}/events' \  
--header 'Authorization: {{token_club_member_2}}' \  
--data-raw '{  
  "event": {  
    "name": "TestEvent4",  
    "club_id": 2,  
    "start_timestamp": "2020-12-03 09:30:00",  
    "end_timestamp": "2020-12-05 00:00:00",  
    "location": "New York City",  
    "max_registration": 125,  
    "description": "Winter Hackathon 2021",  
    "fee": 10,  
    "category": "Academic"}  
}'
```

[View More](#)

#### DEL Delete Event 4 unsuccessful

curl {{hHost}}/events/4

Here we test that a club member cannot delete an event once created.

#### HEADERS

Authorization {{token\_club\_member}}

#### Example Request

#### Delete Event 4 unsuccessful

```
curl --location -g --request DELETE '{{hHost}}/events/4' \  
--header 'Authorization: {{token_club_member}}' \  
--data-raw ''
```

#### DEL Delete Event 4

curl {{hHost}}/events/4

Here we test that a club head is able to delete the event

#### HEADERS

Authorization {{token\_club\_head}}

#### Example Request

#### Delete Event 4

```
curl --location -g --request DELETE '{{hHost}}/events/4' \  
--header 'Authorization: {{token_club_head}}' \  
--data-raw ''
```

#### GET View proposed events

curl {{hHost}}/events?created\_by=2

Here the club member can view the events they have proposed so far. Query parameter filters the results for particular club member.

**HEADERS**

Authorization {{token\_club\_member}}

**PARAMS**

created\_by 2

Example Request

View proposed events

```
curl --location -g --request GET '{{hHost}}/events?created_by=2' \
--header 'Authorization: {{token_club_member}}'
```

**POST Register for event1**

```
{{hHost}}/student/register_event/1
```

Here we test whether a student is able to register for event1.

**HEADERS**

Authorization {{token\_student}}

Example Request

Register for event1

```
curl --location -g --request POST '{{hHost}}/student/register_event/1' \
--header 'Authorization: {{token_student}}' \
--data-raw ''
```

**POST Register for event1 | Club Member**

```
{{hHost}}/student/register_event/1
```

Here we test whether a club member is able to register for event1.

**HEADERS**

Authorization {{token\_club\_member}}

Example Request

Register for event1 | Club Member

```
curl --location -g --request POST '{{hHost}}/student/register_event/1' \
--header 'Authorization: {{token_club_member}}' \
--data-raw ''
```

**POST Register for event3 | Student cannot register**

```
{{hHost}}/student/register_event/3
```

Here we test whether a student is able to register for event where only members can register.

**HEADERS**

Authorization {{token\_student}}

Example Request

Register for event3 | Student cannot register

```
curl --location -g --request POST '{{hHost}}/student/register_event/3' \
--header 'Authorization: {{token_student}}' \
--data-raw ''
```

**POST Register for event3 | Club Member success**

```
{{hHost}}/student/register_event/3
```

Here we test whether a club member is able to register for event3, restricted to members.

#### HEADERS

Authorization {{token\_club\_member}}

Example Request

Register for event3 | Club Member success

```
curl --location -g --request POST '{{hHost}}/student/register_event/3' \
--header 'Authorization: {{token_club_member}}' \
--data-raw ''
```

#### POST Register for event3 | Already exists

curl {{hHost}}/student/register\_event/3

Here we test what happens when a student tries to register for an event they are already registered for.

#### HEADERS

Authorization {{token\_club\_member}}

Example Request

Register for event3 | Already exists

```
curl --location -g --request POST '{{hHost}}/student/register_event/3' \
--header 'Authorization: {{token_club_member}}' \
--data-raw ''
```

#### POST Register for event3 | Max registration

curl {{hHost}}/student/register\_event/3

Here we test whether a student is unable to register for event once capacity is full.

#### HEADERS

Authorization {{token\_club\_head}}

Example Request

Register for event3 | Max registration

```
curl --location -g --request POST '{{hHost}}/student/register_event/3' \
--header 'Authorization: {{token_club_head}}' \
--data-raw ''
```

#### POST Register for event2 | cannot register for event in past

curl {{hHost}}/student/register\_event/2

Here we test whether the student is able to register for event in past.

#### HEADERS

Authorization {{token\_student}}

Example Request

Register for event2 | cannot register for event in past

```
curl --location -g --request POST '{{hHost}}/student/register_event/2' \
--header 'Authorization: {{token_student}}' \
--data-raw ''
```

#### GET View all upcoming events

```
 {{hHost}}/student/get_upcoming_events
```

Here the student can view the events they have registered to and are upcoming i.e. the past events will not show up here.

#### HEADERS

Authorization	{{token_student}}
---------------	-------------------

Example Request

View all upcoming events

```
curl --location -g --request GET '{{hHost}}/student/get_upcoming_events' \
--header 'Authorization: {{token_student}}' \
--data-raw ''
```

### GET View all upcoming events | No permission

```
 {{hHost}}/student/get_upcoming_events
```

Here we see what happens when a unregistered student can view the events they have upcoming.

#### HEADERS

Authorization	{{token_invalid}}
---------------	-------------------

Example Request

View all upcoming events | No permission

```
curl --location -g --request GET '{{hHost}}/student/get_upcoming_events' \
--header 'Authorization: {{token_invalid}}' \
--data-raw ''
```

### GET View registered events

```
 {{hHost}}/student/get_registered_events
```

Here the student can view all the events they have registered to, irrespective of the status.

#### HEADERS

Authorization	{{token_student}}
---------------	-------------------

Example Request

View registered events

```
curl --location -g --request GET '{{hHost}}/student/get_registered_events' \
--header 'Authorization: {{token_student}}' \
--data-raw ''
```

### GET View registered events | Unauthorized person

```
 {{hHost}}/student/get_registered_events
```

Here we see what happens when an unregistered student tries to view all the events they have registered to.

#### HEADERS

Authorization	{{token_invalid}}
---------------	-------------------

Example Request

View registered events | Unauthorized person

```
curl --location -g --request GET '{{hHost}}/student/get_registered_events' \
--header 'Authorization: {{token_invalid}}' \
--data-raw ''
```

### POST Withdraw event | Unauthorized person

```
curl --location -g --request POST '{{hHost}}/student/withdraw_event'
```

Here we see what happens if an unregistered student can withdraw from event.

#### HEADERS

Authorization {{token\_invalid}}

#### BODY raw

```
{  
    "eventId": 1  
}
```

Example Request

Withdraw event | Unauthorized person

```
curl --location -g --request POST '{{hHost}}/student/withdraw_event' \  
--header 'Authorization: {{token_invalid}}' \  
--data-raw '{  
    "eventId": 1  
}'
```

### POST Withdraw event | Student not registered in the event

```
curl --location -g --request POST '{{hHost}}/student/withdraw_event'
```

Here we see what happens if a student tries to withdraw from an event they are not registered for.

#### HEADERS

Authorization {{token\_student}}

#### BODY raw

```
{  
    "eventId": 5  
}
```

Example Request

Withdraw event | Student not registered in the event

```
curl --location -g --request POST '{{hHost}}/student/withdraw_event' \  
--header 'Authorization: {{token_student}}' \  
--data-raw '{  
    "eventId": 5  
}'
```

### POST Withdraw event | Valid Case

```
curl --location -g --request POST '{{hHost}}/student/withdraw_event'
```

Here the student can withdraw from an event they registered to.

#### HEADERS

Authorization {{token\_student}}

#### BODY raw

```
{  
    "eventId": 1  
}
```

Example Request

Withdraw event | Valid Case

```
curl --location -g --request POST '{{hHost}}/student/withdraw_event' \  
--header 'Authorization: {{token_student}}' \  
--data-raw '{  
    "eventId": 1  
}'
```

## GET View withdrew events

```
curl --location -g --request GET '{{hHost}}/student/get_registered_events'
```

Here we a student can see what events they have withdrawn from.

### HEADERS

Authorization {{token\_student}}

Example Request

View withdrew events

```
curl --location -g --request GET '{{hHost}}/student/get_registered_events' \
--header 'Authorization: {{token_student}}' \
--data-binary ''
```

## GET See all the clubs and role in the club for club member | Unauthorized

```
curl --location -g --request GET '{{hHost}}/student/get_roles'
```

Here a student can check what their role is in various clubs. Here we see what happens if an unregistered student does so.

### HEADERS

Authorization {{token\_invalid}}

Example Request

See all the clubs and role in the club for club member | Unauthorized

```
curl --location -g --request GET '{{hHost}}/student/get_roles' \
--header 'Authorization: {{token_invalid}}' \
--data-binary ''
```

## GET See all the clubs and role in the club for club member

```
curl --location -g --request GET '{{hHost}}/student/get_roles'
```

Here a student can check what their role is in various clubs. The example used here is of a club member.

### HEADERS

Authorization {{token\_club\_member}}

Example Request

See all the clubs and role in the club for club member

```
curl --location -g --request GET '{{hHost}}/student/get_roles' \
--header 'Authorization: {{token_club_member}}' \
--data-binary ''
```

## GET See all the clubs and role in the club for club head

```
curl --location -g --request GET '{{hHost}}/student/get_roles'
```

Here a student can check what their role is in various clubs. The example used here is of a club head.

### HEADERS

Authorization {{token\_club\_head}}

Example Request

See all the clubs and role in the club for club head

```
curl --location -g --request GET '{{hHost}}/student/get_roles' \
--header 'Authorization: {{token_club_head}}' \
--data-raw ''
```

## PUT Replace the head

```
{{hHost}}/clubs/2
```

Here we test if the club head can replace themselves with the next head.

### HEADERS

Authorization {{token\_club\_head\_2}}

### BODY raw

```
{
  "new_head": "testclubheadsav@gmail.com"
}
```

Example Request

Replace the head

```
curl --location -g --request PUT '{{hHost}}/clubs/2' \
--header 'Authorization: {{token_club_head_2}}' \
--data-raw '{
  "new_head": "testclubheadsav@gmail.com"
}'
```

## GET Get all clubs

```
{{hHost}}/clubs
```

Here we test if a student can view all the clubs in the college.

### HEADERS

Authorization {{token\_student}}

Example Request

Get all clubs

```
curl --location -g --request GET '{{hHost}}/clubs' \
--header 'Authorization: {{token_student}}' \
--data-raw ''
```

## PUT Edit club happy case

```
{{hHost}}/clubs/2
```

Here we test if the club head can edit the club.

### HEADERS

Authorization {{token\_club\_head}}

### BODY raw

```
{
  "club": {
    "description": "Test Club Description edit"
  }
}
```

Example Request

Edit club happy case

```
curl --location -g --request PUT '{{hHost}}/clubs/2' \
--header 'Authorization: {{token_club_head}}' \
--data-raw '{
  "club": {
    "description": "Test Club Description edit"
  }
}'
```

```
}
```

## GET Get Club by id

```
{{{hHost}}}/clubs/2
```

here we test if a student can view details for a particular club given the id. Notice the description for club2 has been modified in last test.

### HEADERS

Authorization {{token\_student}}

Example Request

Get Club by id

```
curl --location -g --request GET '{{hHost}}/clubs/2' \
--header 'Authorization: {{token_student}}' \
--data-raw ''
```

## DEL Delete club happy case

```
{{{hHost}}}/clubs/2
```

here we test whether a club head can delete the club.

### HEADERS

Authorization {{token\_club\_head}}

Example Request

Delete club happy case

```
curl --location -g --request DELETE '{{hHost}}/clubs/2' \
--header 'Authorization: {{token_club_head}}' \
--data-raw ''
```

## PUT Edit Event past event

```
{{{hHost}}}/events/2
```

Here we test what happens if the club member can edit an event from past.

### HEADERS

Authorization {{token\_club\_member}}

### BODY raw

```
{
  "event": {
    "name": "TestEvent2",
    "club_id": 1,
    "start_timestamp": "2021-12-03 09:30:00",
    "end_timestamp": "021-12-05 00:00:00",
    "location": "New York City",
    "max_registration": 305,
    "description": "Test Change ~~~~~~",
    "fee": 10,
    "category": "Academic"
  }
}
```

[View More](#)

Example Request

Edit Event past event

```
curl --location -g --request PUT '{{hHost}}/events/2' \
--header 'Authorization: {{token_club_member}}' \
--data-raw '{
  "event": {
    "name": "TestEvent2",
    "club_id": 1,
    "start_timestamp": "2021-12-03 09:30:00",
    "end_timestamp": "021-12-05 00:00:00",
    "location": "New York City",
    "max_registration": 305,
    "description": "Test Change ~~~~~~",
    "fee": 10,
    "category": "Academic"
  }
}'
```

[View More](#)

## PUT Edit Event 2

```
 {{(hHost)}/events/1}
```

Here we test if the club member can edit an event for their club.

### HEADERS

Authorization {{token\_club\_member}}

### BODY raw

```
{  
  "event": {  
    "name": "TestEvent1",  
    "club_id": 1,  
    "start_timestamp": "2022-12-03 09:30:00",  
    "end_timestamp": "2022-12-05 00:00:00",  
    "location": "New York City",  
    "max_registration": 125,  
    "description": "Test Change Description",  
    "fee": 10, View More  
    "category": "Academic",
```

Example Request

Edit Event 2

```
curl --location -g --request PUT '{{(hHost)}/events/1' \  
--header 'Authorization: {{token_club_member}}' \  
--data-raw '{  
  "event": {  
    "name": "TestEvent1",  
    "club_id": 1,  
    "start_timestamp": "2022-12-03 09:30:00",  
    "end_timestamp": "2022-12-05 00:00:00",  
    "location": "New York City", View More  
    "max_registration": 125,
```

## PUT Edit Event Missing Authorization

```
 {{(hHost)}/events/4}
```

Here we see what happens if someone tries to edit the event with missing authorization/ without logging in.

### HEADERS

Authorization {{token\_invalid}}

### BODY raw

```
{  
  "event": {  
    "name": "Bug Testing Check",  
    "club_id": 2  
  }  
}
```

Example Request

Edit Event Missing Authorization

```
curl --location -g --request PUT '{{(hHost)}/events/4' \  
--data-raw '{  
  "event": {  
    "name": "Bug Testing Check",  
    "club_id": 2  
  }  
}'
```

## PUT Edit Event Permission Error

```
 {{(hHost)}/events/3}
```

Here we see what happens if a member from another club tries to edit event.

### HEADERS

Authorization {{token\_club\_member\_2}}

### BODY raw

```
{
  "event": {
    "name": "ACM-ICPC Practice",
    "club_id": 1
  }
}
```

Example Request Edit Event Permission Error

```
curl --location -g --request PUT '{{{hHost}}}/events/3' \
--header 'Authorization: {{{token_club_member_2}}}' \
--data-raw '{
  "event": {
    "name": "ACM-ICPC Practice",
    "club_id": 1
  }
}'
```

#### GET View event

`{{{hHost}}}/events/1`

here we test if anyone can view the event details given the event id.

#### HEADERS

Authorization      `{{{token_student}}}`

Example Request View event

```
curl --location -g --request GET '{{{hHost}}}/events/1' \
--header 'Authorization: {{{token_student}}}'
```

#### PUT Edit club Invalid request

`{{{hHost}}}/clubs/1`

This is an error test for editing the club. If anyone other than club head tries to edit the club, we throw error that it is an Invalid request.

#### HEADERS

Authorization      `{{{token_club_member}}}`

#### BODY raw

```
{
  "club": {
    "description": "this is a test change"
  }
}
```

Example Request Edit club Invalid request

```
curl --location -g --request PUT '{{{hHost}}}/clubs/1' \
--header 'Authorization: {{{token_club_member}}}' \
--data-raw '{
  "club": {
    "description": "this is a test change"
  }
}'
```

#### DEL Delete club Invalid Request

`{{{hHost}}}/clubs/1`

This is an error test for deleting the club. If anyone other than club head tries to delete the club, we throw error that it is an Invalid request.

#### HEADERS

Authorization {{token\_club\_member}}

Example Request

Delete club Invalid Request

```
curl --location -g --request DELETE '{{hHost}}/clubs/1' \
--header 'Authorization: {{token_club_member}}' \
--data-raw ''
```

#### PUT Add a member bad request

curl {{hHost}}/member/1

This is an error test for adding member to the club. If anyone other than club head tries to add member to the club, we throw error that it is an Invalid request.

#### HEADERS

Authorization {{token\_club\_member}}

#### BODY raw

```
{
  "student_email_id": "testemail1@columbia.edu"
}
```

Example Request

Add a member bad request

```
curl --location -g --request PUT '{{hHost}}/member/1' \
--header 'Authorization: {{token_club_member}}' \
--data-raw '{
  "student_email_id": "testemail1@columbia.edu"
}'
```

#### PUT Event Approve attempt by club member

curl {{hHost}}/events/1

Here we test if the club member can approve an event for their club.

#### HEADERS

Authorization {{token\_club\_member}}

#### BODY raw

```
{
  "status": "Approved"
}
```

Example Request

Event Approve attempt by club member

```
curl --location -g --request PUT '{{hHost}}/events/1' \
--header 'Authorization: {{token_club_member}}' \
--data-raw '{
  "status": "Approved"
}'
```

#### PUT Event Approve by club head

curl {{hHost}}/events/1

Here we test if the club head can approve an event.

#### HEADERS

Authorization {{token\_club\_head}}

#### BODY raw

```
{
  "status": "Approved"
}
```

```
}
```

Example Request      Event Approve by club head

```
curl --location -g --request PUT '{{hHost}}/events/1' \
--header 'Authorization: {{token_club_head}}' \
--data-raw '{
    "status": "Approved"
}'
```

#### GET View event after approving

```
{{hHost}}/events/1
```

here we view the event after it was approved by club head.

#### HEADERS

Authorization      {{token\_student}}

Example Request      View event after approving

```
curl --location -g --request GET '{{hHost}}/events/1' \
--header 'Authorization: {{token_student}}'
```

#### PUT Event Rejected by club head

```
{{hHost}}/events/3
```

Here we test if the club head can reject an event.

#### HEADERS

Authorization      {{token\_club\_head}}

#### BODY raw

```
{
    "status": "Rejected"
}
```

Example Request      Event Rejected by club head

```
curl --location -g --request PUT '{{hHost}}/events/3' \
--header 'Authorization: {{token_club_head}}' \
--data-raw '{
    "status": "Rejected"
}'
```