

Auth

Name & Description	HTTP Method	Data Types	Exceptions
auth/login	post	Body Parameters: { uid } Return type if no error: 200	400 Error when any of: Missing params 403 Error when any of:
auth/logout	post	Body Parameters: {uid} Return type if no error: 200	400 Error when any of: Missing params 403 Error when any of:
auth/register	post	Body Parameters: {uid, f_name, l_name, email_addr, account_type) Return type if no error: 200	400 Error when any of: Missing params 403 Error when any of:
auth/delete	delete	Query Parameters: (uid) Return type if no error: 200	400 Error when any of: Missing params 403 Error when any of:

User

Name & Description	HTTP Method	Data Types	Exceptions
user/edit_profile	put	Body Parameters: REQUIRES: uid OPTIONAL: f_name, l_name, email_addr, address, phone_number, dob, gender, education, research_interest, Skills, availability Return type if no error:	400 Error when any of: Missing params 403 Error when any of:

		200	
user/get_profile	get	Query Parameters: Return type if no error: 200	400 Error when any of: Missing params 403 Error when any of:

Conference

Name & Description	HTTP Method	Data Types	Exceptions
conference/create	post	Body Parameters: Name, address Return type if no error: 200	400 Error when any of: Missing params 403 Error when any of:
conference/join	post	Body Parameters: cid Return type if no error:	400 Error when any of: Missing params 403 Error when any of:
conference/invite	POST	Body Parameters: Email_addr, f_name, l_name, account_type, cid, dob, gender	400 Error when any of: Missing params 403 Error when any of:
conference/list Provide an array of all conferences that the authorised user is part of.	get	Query Parameters: Return type if no error:	400 Error when any of: Missing params 403 Error when any of:
conference/list_all Provide an array of all conferences that volunteer can search and request to join	GET	Parameters: Return type if no error:	400 Error when any of: Missing params 403 Error when any of:
conference/approve Organizer can approve the join request from volunteer	POST	Parameters: Cid, uid Return type if no error:	400 Error when any of: 403 Error when any of:

conference/details Show the details of a conference	GET	Parameters: Cid Return type if no error:	400 Error when any of: Missing params 403 Error when any of:
conference/nearby	GET	Parameters: Uid, radius Return type if no error: [conference]	400 Error when any of: Cannot find uid 403 Error when any of: Invalid addresses

Schedule

Name & Description	HTTP Method	Data Types	Exceptions
schedule/event_add Add an event (talk/workshop/social event) to the conference schedule.	POST	Parameters: Cid, event_name, date (DATE), time_start (DATETIME), time_end (DATETIME), description, authors Return type if no error: 200	400 Error when any of: Missing params 403 Error when any of: Conflicting times
schedule/event_remove Remove an event from the conference schedule	DELETE	Parameters: Cid, eid Return type if no error: 200	400 Error when any of: Missing params 403 Error when any of: Unable to find event
schedule/event_get_all Get an ordered list of all events in the conference	GET	Parameters: cid Return type if no error: d	400 Error when any of: Missing params 403 Error when any of: Unable to find conference
schedule/event_get Get a single event in the conference	GET	Parameters: Cid, eid Return type if no	400 Error when any of: Missing params 403 Error when any

schedule		error: {eid, event_name, date, time_start, time_end, description, authors}	of: Unable to find conference or event
schedule/event_author_add Add an author to the event	POST	Parameters: Cid, eid, author Return type if no error: 200	400 Error when any of: Missing params 403 Error when any of: Unable to find conference or event
schedule/event_author_remove Remove an author to the event	DELETE	Parameters: Cid, eid, author Return type if no error: 200	400 Error when any of: Missing params 403 Error when any of: Unable to find conference or event

Forum (new)

Name & Description	HTTP Method	Data Types	Exceptions
forum/make_post	POST	Body Parameters: { cid, fid, post_title, post_body } Return type if no error: { pid }	400 Error when any of: - cid does not refer to a valid conference - fid does not refer to a valid forum in the conference - title or text is empty 403 Error when any of: - for announcement, authorised user is not the organiser of the conference - for question, authorised user is not the member of the conference
forum/respond_post	POST	Body Parameters: { cid, fid, pid, body } Return type if no error: { mid }	400 Error when any of: - cid does not refer to a valid conference - fid does not refer to a valid forum in the conference - pid does not refer to a valid posts within a conference that the authorised user has joined - text is empty 403 Error when any of:

			<ul style="list-style-type: none"> - authorised user is not the members of the conference
forum/edit_post	PUT	Body Parameters: { cid, fid, pid, post_body } Return type if no error: {}	400 Error when any of: <ul style="list-style-type: none"> - cid does not refer to a valid conference - fid does not refer to a valid forum in the conference - pid does not refer to a valid post within a conference that the authorised user has joined - text is empty 403 Error when any of: <ul style="list-style-type: none"> - for announcement, authorised user is not the organiser of the conference or the post is not made by them - for question, authorised user is not the member of the conference or the post is not made by them
forum/remove_post	DELETE	Query Parameters: (cid, fid, pid) Return type if no error: {}	400 Error when any of: <ul style="list-style-type: none"> - cid does not refer to a valid conference - fid does not refer to a valid forum in the conference - pid does not refer to a valid post within a conference that the authorised user has joined 403 Error when any of: <ul style="list-style-type: none"> - for announcement, authorised user is not the organiser of the conference or the post is not made by them - for question, authorised user is not the member of the conference or the post is not made by them
forum/get_all_posts	GET	Query Parameters: (cid, fid) Return type if no error: { posts }	400 Error when any of: <ul style="list-style-type: none"> - cid does not refer to a valid conference - fid does not refer to a valid forum 403 Error when any of: <ul style="list-style-type: none"> - authorised user is not the members of the conference
forum/get_all_messages	GET	Query Parameters: (cid, fid, pid) Return type if no error: { messages }	400 Error when any of: <ul style="list-style-type: none"> - cid does not refer to a valid conference - fid does not refer to a valid forum in the conference - pid does not refer to a valid post within a conference that the authorised user has joined 403 Error when any of: <ul style="list-style-type: none"> - authorised user is not the members of the conference

forum/edit_message	PUT	Body Parameters: { cid, fid, pid, mid, body } Return type if no error: {}	400 Error when any of: <ul style="list-style-type: none"> - cid does not refer to a valid conference - fid does not refer to a valid forum in the conference - pid does not refer to a valid post within a - mid does not refer to a valid message within a conference that the authorised user has joined - text is empty 403 Error when any of: <ul style="list-style-type: none"> - authorised user is not the members of the conference or the message is not sent by them
forum/remove_message	DELETE	Query Parameters: (cid, fid, pid, mid) Return type if no error: {}	400 Error when any of: <ul style="list-style-type: none"> - cid does not refer to a valid conference - fid does not refer to a valid forum in the conference - pid does not refer to a valid post within a - mid does not refer to a valid message within a conference that the authorised user has joined 403 Error when any of: <ul style="list-style-type: none"> - authorised user is not the members of the conference or the message is not sent by them
forum/get_all_forums	GET	Query Parameters: (cid) Return type if no error: { forums }	400 Error when any of: <ul style="list-style-type: none"> - cid does not refer to a valid conference 403 Error when any of: <ul style="list-style-type: none"> - authorised user is not the members of the conference

task

Name & Description	HTTP Method	Data Types	Exceptions
--------------------	-------------	------------	------------

task/create_task Organiser only	POST	Body Parameters: { cid, task_name, task_description, date_time_start, date_time_end } Return type if no error: { tid }	400 Error when any of: - cid does not refer to a valid conference - task_description or task_name is empty - not valid start or end time - end_time is earlier than start time 403 Error when any of: - authorised user is not the organiser of the conference
task/remove_task Organiser only	DELETE	Query Parameters: (cid, tid) Return type if no error: {}	400 Error when any of: - cid does not refer to a valid conference - tid does not refer to a valid task in the conference 403 Error when any of: - authorised user is not the organiser of the conference
task/assign_task Organiser only	POST	Body Parameters: { cid, tid, uid } Return type if no error: {}	400 Error when any of: - cid does not refer to a valid conference - tid does not refer to a valid task in the conference - uid does not refer to a valid member in the conference - uid has already been assigned or signed up for the task 403 Error when any of: - authorised user is not the organiser of the conference

task/unassign_task Organiser only	DELETE	Query Parameters: (cid, tid, uid) Return type if no error: { }	400 Error when any of: <ul style="list-style-type: none"> - cid does not refer to a valid conference - tid does not refer to a valid task in the conference - uid does not refer to a valid member in the conference - uid has not been assigned or signed up for the task 403 Error when any of: <ul style="list-style-type: none"> - authorised user is not the organiser of the conference
task/get_all_tasks	GET	Query Parameters: (cid) Return type if no error: { tasks }	400 Error when any of: <ul style="list-style-type: none"> - cid does not refer to a valid conference 403 Error when any of: <ul style="list-style-type: none"> - authorised user is not the member of the conference
task/sign_up_task	POST	Body Parameters: { cid, tid } Return type if no error: { }	400 Error when any of: <ul style="list-style-type: none"> - cid does not refer to a valid conference - tid does not refer to a valid task in the conference - authorised user has already been assigned or signed up for the task 403 Error when any of: <ul style="list-style-type: none"> - authorised user is not the member of the conference

task/unsign_up_task	DELETE	Query Parameters: (cid, tid) Return type if no error: {}	400 Error when any of: <ul style="list-style-type: none"> - cid does not refer to a valid conference - tid does not refer to a valid task in the conference - authorised user has not been assigned or signed up for the task 403 Error when any of: <ul style="list-style-type: none"> - authorised user is not the member of the conference
task/get_all_user_tasks Get all the tasks related to that user in the conference	GET	Query Parameters: (cid) Return type if no error: { tasks }	400 Error when any of: <ul style="list-style-type: none"> - cid does not refer to a valid conference 403 Error when any of: <ul style="list-style-type: none"> - authorised user is not the member of the conference
task/get_all_assigned_users Get all the assigned users in a task	GET	Query Parameters: (cid, tid) Return type if no error: { users_infos }	400 Error when any of: <ul style="list-style-type: none"> - cid does not refer to a valid conference - tid does not refer to a valid task in the conference 403 Error when any of: <ul style="list-style-type: none"> - authorised user is not the member of the conference

task/get_all_unassigned_users Get all the unassigned users in a task	GET	Query Parameters: (cid, tid) Return type if no error: { users_infos }	400 Error when any of: - cid does not refer to a valid conference - tid does not refer to a valid task in the conference 403 Error when any of: - authorised user is not the member of the conference
--	-----	--	---

Attendance

Name & Description	HTTP Method	Data Types	Exceptions
attendance/log_attendance 	POST	Body Parameters: { cid, description, date, hours } Return type if no error: { aid }	400 Error when any of: - cid does not refer to a valid conference - not a valid date - description is empty - hours is not greater than 0 403 Error when any of: - authorised user is not the member of the conference - authorised user is the organiser of the conference
attendance/get_all_users For organiser, get all managers and volunteers. For managers, get volunteers only.	GET	Query Parameters: (cid) Return type if no error: { users_infos }	400 Error when any of: - cid does not refer to a valid conference 403 Error when any of: - authorised user is not the organiser or manager of the conference
attendance/get_all_user_attendances Get all the logged attendances of an user	GET	Query Parameters: (cid, uid) Return type if no error: { attendances }	400 Error when any of: - cid does not refer to a valid conference - uid does not refer to a valid member in the conference 403 Error when any of: - authorised user is not the members of the conference - if the authorised user is not the organiser or manager, uid does not refer the user
attendance/approve_attendance Status: approved, pending, declined	POST	Body Parameters: { cid, aid } Return type if no error: {}	400 Error when any of: - cid does not refer to a valid conference - aid does not refer to a valid attendance in the conference - the attendance is already approved or declined

			403 Error when any of: <ul style="list-style-type: none"> - authorised user is not the organiser of the conference
attendance/delete_attendance Status: approved, pending, declined	DELETE	Query Parameters: (cid, aid) Return type if no error: {}	400 Error when any of: <ul style="list-style-type: none"> - cid does not refer to a valid conference - aid does not refer to a valid attendance in the conference - the attendance is already approved or declined 403 Error when any of: <ul style="list-style-type: none"> - authorised user is not the organiser of the conference
attendance/fill_in_feedback	POST	Body Parameters: { cid, uid, feedback } Return type if no error: { fbid }	400 Error when any of: <ul style="list-style-type: none"> - cid does not refer to a valid conference - uid does not refer to a valid member in the conference - feedback is empty - uid refers to the user 403 Error when any of: <ul style="list-style-type: none"> - When the user is attempting to give feedback to user account type they are not privileged to do
attendance/get_user_feedback	GET	Query Parameters: (cid, uid) Return type if no error: [{ feedbacks }]	400 Error when any of: <ul style="list-style-type: none"> - cid does not refer to a valid conference - uid does not refer to a valid member in the conference 403 Error when any of: <ul style="list-style-type: none"> - authorised user is not the members of the conference - if the authorised user is not the organiser or manager, uid does not refer the user
attendance/get_feedback	GET	Query Parameters: (uid, fid) Return type if no error: { feedback }	400 Error when any of: <ul style="list-style-type: none"> - fid does not exist - uid does not exist
attendance/rate_volunteer	POST	Body Parameters: { cid, uid, score } Return type if no error: {}	400 Error when any of: <ul style="list-style-type: none"> - cid does not refer to a valid conference - uid does not refer to a valid member in the conference - score is not between 1 to 5 - uid refers to the user

			<p>- the user has already been rated by the authorised user</p> <p>403 Error when any of:</p> <ul style="list-style-type: none"> - authorised user is not the organiser or manager of the conference
<p>attendance/get_averaged_rating</p> <p>Return 0 if the user has not been rated by anyone</p>	GET	<p>Query Parameters: (uid)</p> <p>Return type if no error: { averaged_score }</p>	<p>400 Error when any of:</p> <ul style="list-style-type: none"> - uid does not refer to a valid member in the conference <p>403 Error when any of:</p> <ul style="list-style-type: none"> - authorised user is the organiser of the conference - uid does not refer to the user
<p>attendance/get_rating</p> <p>Return 0 if the user has not been rated by the authorised user</p>	GET	<p>Query Parameters: (cid, uid)</p> <p>Return type if no error: { score }</p>	<p>400 Error when any of:</p> <ul style="list-style-type: none"> - cid does not refer to a valid conference - uid does not refer to a valid member in the conference <p>403 Error when any of:</p> <ul style="list-style-type: none"> - authorised user is not the organiser or manager of the conference

Datastore Structure and Functions

(python variables declaration: all lower cases with _)

General Functions

Function	Use
db_reset() need to add deletes for subcollections	<u>WILL RESET WHOLE DATABASE, USE WITH CAUTION</u> To run: <ul style="list-style-type: none">- Remove '..' in db_conn.py credentials.certificate file path and replace with single '.'- Use 'py .\backend\reset_db.py reset' in command line

User Functions

Function	Use
General	
db_user_add(uid, f_name, l_name, email_addr, acc_type)	Adds a user to database
db_user_first_name_edit(uid, f_name)	Updates first name
db_user_last_name_edit(uid, l_name)	Updates last name
db_user_account_type_edit(uid, new_acc_type)	Updates account type
db_user_conf_joined_add(uid, cid)	Adds conference id to user profile
db_user_email_addr_edit(uid, email_addr)	Updates email address
db_user_address_edit(uid, place_id)	Updates address
db_user_phone_number_edit(uid, phone_num)	Updates phone number
db_user_dob_edit(uid, dob)	Updates dob
db_user_gender_edit(uid, gender)	Updates gender
db_user_education_edit(uid, education)	Updates education
db_user_skills_edit(uid, skills)	Updates skills
db_user_research_interests_edit(uid,	Updates research interests

res_int)	
db_user_availability_edit(uid, availability)	Updates availability
db_user_delete(uid)	Deletes user according to uid NEED TO DELETE RATING/FEEDBACK
db_user_get(uid)	Returns user dictionary if exists, else None
db_user_account_type_get(uid)	Gets user account type
Rating	
db_user_rating_add(uid, rater_id, rating)	Adds rating to user, should be int between 0-5
db_user_rating_edit(uid, rater_id, new_rating)	
db_user_rating_delete(uid, rater_id)	
db_user_rating_get(uid, rater_id)	Returns rating rater_id gave
db_user_rating_get_rating(uid)	
Feedback	
db_user_feedback_add(uid, fid, sent_by, cid, feedback)	Adds feedback to user
db_user_feedback_edit(uid, fid, new_feedback)	Edits feedback
db_user_feedback_delete(uid, fid)	Deletes feedback
db_user_feedback_get(uid, fid)	Gets feedback according to fid
db_user_feedback_sent_by_get(uid, sent_by)	Gets feedback according to sent_by
db_user_feedback_cid_get(uid, cid)	Gets feedback according to cid
db_user_feedback_get_all(uid)	Gets all feedback for user

Conference Functions

Function	Use
Conferences	
db_conf_add(cid, uid, conf_name, place_id)	Adds new conference
db_conf_name_edit(cid, new_name)	Updates conference name
db_conf_address_edit(cid, place_id)	Updates conference address

db_conf_volunteers_request_join_add(cid, uids)	Adds uid volunteer conference join request - Uids is array
db_conf_volunteers_add(cid, uid)	Adds volunteer to conference properly - Will remove volunteer uid from request array if in there
db_conf_date_time_start_edit(cid, new_date_time_start)	Updates conference date time start
db_conf_date_time_end_edit(cid, new_date_time_end)	Updates conference date time end
db_conf_delete(cid)	Deletes conference ADD DELETE ATTENDANCE, DELETE FEEDBACK
db_conf_get(cid)	Gets conference dictionary according to cid
db_conf_joined_conf_get(uid)	Gets joined conferences names and cids according to uid
db_conf_get_all()	Gets all conferences names and cids
db_conf_get_organiser_uid(cid)	Gets organiser uid of conference
db_conf_is_member(cid, uid)	Returns true if uid is member in conference
Forum	
db_conf_forum_add(cid, fid, forum_title, forum_type)	Adds a new forum to conference
db_conf_forum_title_edit(cid, fid, new_title)	Edits title of forum
db_conf_forum_post_add(cid, fid, pid, sent_by, post_title, post_body)	Adds a post to forum
db_conf_forum_post_title_edit(cid, fid, pid, new_title)	Edits forum post title
db_conf_forum_post_body_edit(cid, fid, pid, new_body)	Edits body of forum post
db_conf_forum_post_delete(cid, fid, pid)	Deletes forum post
db_conf_forum_delete(cid, fid)	Deletes entire forum
db_conf_forum_get(cid, fid)	Gets forum name, fid and forum type
db_conf_forum_get_all(cid)	Gets all forum names and fid
db_conf_forum_post_get(cid, fid, pid)	Gets forum post dictionary
db_conf_forum_post_get_all(cid, fid)	Gets all forum post dictionary in forum

db_conf_forum_post_response_add(cid, fid, pid, mid, sent_by, body)	Adds response to forum post
db_conf_forum_post_response_edit(cid, fid, pid, mid, new_body)	Edits response to forum post
db_conf_forum_post_response_delete(cid, fid, pid, mid)	Deletes forum response
db_conf_forum_post_response_get(cid, fid, pid, mid)	Gets forum post response dictionary
db_conf_forum_post_response_get_all(cid, fid, pid)	Gets all forum post response dictionary
db_conf_is_forum_post_made_by_user(cid, fid, pid, uid)	Returns true if forum post is made by user
db_conf_is_forum_post_response_made_by_user(cid, fid, pid, mid, uid)	Returns true if forum post response is made by user
Tasks	
db_conf_task_add(cid, tid, task_name, task_desc, date_time_start, date_time_end)	Adds a new task
db_conf_task_name_edit(cid, tid, new_task_name)	Edits task name
db_conf_task_description_edit(cid, tid, new_task_desc)	Edits task description
db_conf_task_assignees_add(cid, tid, assignees)	Adds assignee to task - Assignees is array
db_conf_task_assignees_remove(cid, tid, assignees)	Removes assignee from task - Assignees is array
db_conf_task_status_edit(cid, tid, new_status)	Edits task status
db_conf_task_date_time_start_edit(cid, tid, new_date_time_start)	Edits conference task date time start
db_conf_task_date_time_end_edit(cid, tid, new_date_time_end)	Edits conference task date time end
db_conf_task_assignees_remove(cid, tid, assignee)	Removes assignee from task
db_conf_task_delete(cid, tid)	Deletes task
db_conf_task_get(cid, tid)	Gets task dictionary according to tid
db_conf_task_get_all(cid)	Gets all task dictionary in conference

db_conf_task_assigned_get(cid, uid)	Gets assigned tasks in conference according to uid sorted by date_time_start
db_conf_task_assigned_users_get(cid, tid)	Returns user info for each assigned user
db_conf_task_unassigned_users_get(cid, tid)	Returns user info for each unassigned user
db_conf_task_is_user_assigned(cid, tid, uid)	Returns true if user is assigned to task
Events	
db_conf_event_add(cid, eid, name, description, date_time_start, date_time_end)	Adds a new event
db_conf_event_name_edit(cid, eid, new_name)	Edits event name
db_conf_event_description_edit(cid, eid, new_description)	Edits event description
db_conf_event_authors_add(cid, eid, authors)	Adds event authors
db_conf_event_authors_remove(cid, eid, authors)	Removes event authors
db_conf_event_date_time_start_edit(cid, eid, new_date_time_start)	Edits event start time
db_conf_event_date_time_end_edit(cid, eid, new_date_time_end)	Edits event end time
db_conf_event_delete(cid, eid)	Deletes event
db_conf_event_get(cid, eid)	Gets event dictionary
db_conf_event_get_all(cid)	Gets all events in conference
Attendance	
db_conf_attendance_add(cid, aid, volunteer, date, description, hours_worked)	Adds attendance object to conference - Status starts at pending
db_conf_attendance_status_edit(cid, aid, new_status)	Edits status of attendance
db_conf_attendance_date_edit(cid, aid, new_date)	Edits date of attendance
db_conf_attendance_description_edit(cid, aid, new_desc)	Edits description of attendance
db_conf_attendance_hours_worked_edit(ci	Edits hours worked of attendance

d, aid, new_hours_worked)	
db_conf_attendance_delete(cid, aid)	Deletes attendance record
db_conf_attendance_get(cid, aid)	Gets attendance record according to aid
db_conf_attendance_uid_get(cid, uid)	Gets all attendance records according to uid
db_conf_attendance_date_get(cid, date)	Gets all attendance records according to date
db_conf_attendance_status_get(cid, status)	Gets all attendance records according to status
db_conf_attendance_get_all(cid)	Gets all attendance records for cid

SPRINT 3

- Attendance
- Calendar export
 - <https://learnpython.com/blog/working-with-icalendar-with-python/>
 - <https://pypi.org/project/ics/>

```

data = {
    conferences[];
    users[];
}

conference = {
    cid: STRING;
    name: STRING;
    place_id: STRING;
    organiser: user.uid;
    date_time_start: DATETIME;
    date_time_end: DATETIME;
    tasks: task[];
    Volunteers_request_join: uid[]
    volunteers: uid[];
    Forums: forum[];
    Events: event[];
    Attendance: attendance[];
}

Attendance = {
    Aid = STRING;

```

```

    volunteer: UID;
    Date: DATETIME;
    Description: STRING;
    status: STRING (pending/declined/approved);
    Hours_worked: NUM;
}

Event = {
    Eid: STRING;
    Name: STRING;
    Description: STRING;
    Authors: STRING[]
    Date_time_start: DATETIME;
    Date_time_end: DATETIME;
}

task = {
    tid: STRING;
    task_name: STRING;
    task_description: STRING;
    assignees = uid[];
    status = STRING/ENUM;
    date_time_start = DATETIME;
    date_time_end = DATETIME;
}

forum = {
    fid: STRING;
    forum_title: STRING;
    forum_type: NUM;
    Posts: post[]
}

post = {
    pid: STRING;
    post_title: STRING;
    post_body: STRING;
    sent_by: uid;
    date_time: DATETIME;
    responses: message[];
}

message = {

```

```

    mid: STRING;
    sent_by: uid;
    body: STRING;
    date_time: DATETIME;
}

user = {
    uid: STRING;
    first_name: STRING;
    last_name: STRING;
    Account_type: NUM corresponds to account;
    Joined_conferences = cid[];
    place_id: STRING;
    phone_number: NUM,
    dob: DATETIME;
    gender: STRING;
    education: STRING;
    skills: STRING[];
    research_interest: STRING[];
    Availability: STRING[];
    Ratings: rating[];
    Feedback: feedback[];
}

Feedback = {
    Fbid: STRING
    Sent_by: uid
    Message: STRING
}

Rating = {
    rating = NUM;
    Rater_id = uid;
}

```