

##Auth API

- description: signs in as the given user
- request: `POST /signin/`
 - content-type: `application/json`
 - body: object
 - username: (string) the username of the given user
 - password: (string) the password of the given user
- response: 200
 - content-type: `application/json`
 - body: username of the signed in user
- response: 404
 - content-type: `application/json`
 - body: No user with this email
- response: 400
 - content-type: `application/json`
 - body: Error: [error message]

```
```\n$ curl -X POST\n  -H "Content-Type: application/json"\n  -d '{"username":"test","password":"test"}'\n  http://localhost:5000/signin/\n```\n
```

- description: sign up as a new user and adds it to the database
- request: `POST /signup/`
  - content-type: `application/json`
  - body: object
    - username: (string) the username of the user
    - password: (string) the password of the user
- response: 200
  - content-type: `application/json`
  - body: object
    - username: (string) the username of the user
    - password: (string) the salted password of the user
    - salt: (string) the salt for the password
    - name: (string) the real name of the user
    - interests: ([string]) the list of interests the user has
    - friends: ([ObjectId]) the list of user object IDs that are friends with the user
    - friend\_requests: ([ObjectId]) the list of users who are requesting to be friends with the user

- invitedEvents: ([ObjectId]) the list of events that the user is invited to
  - attendingEvents: ([ObjectId]) the list of events that the user is planning to attend
  - history: ([ObjectId]) the list of events that the user has attended in the past
- response: 409
    - content-type: `application/json`
    - body: User with this username already exists
  - response: 500
    - content-type: `application/json`
    - body: [error message]

```

$ curl -X POST
 -H "Content-Type: application/json"
 -d '{"username":"test","password":"test"}'
 http://localhost:5000/signup/

```

- description: logs out of the current user
- request: `GET /logout/`
- response: 200
  - content-type: `application/json`
  - body: Logged out

```

$ curl -X GET
 http://localhost:5000/logout/

```

- description: verifies that a user is currently logged in
- request: `GET /verify/`
- response: 200
  - content-type: `application/json`
  - body: object
    - isValid: (boolean) determines if there is a logged in user

```

$ curl -X GET
 http://localhost:5000/verify/

```

## ##Chat API

- description: creates a user to be registered into chatkit's servers
- request: `POST /chat/createUser/``
  - content-type: `application/json``
  - body: object
    - username: (string) the username of the logged in user
- response: 200
  - content-type: `application/json``
  - body: created new user for Chatkit
- response: 500
  - content-type: `application/json``
  - body: error message

```

```
$ curl -X POST
  -H "Content-Type: application/json"
  -d '{"username":"test"}'
  http://localhost:5000/chat/createUser/
```

```

- description: Authorizes the user to use/join chatkit
- request: `POST /chat/auth/:user_id``
- response: 200
  - content-type: `application/json``
  - body: chatkit response to the authentication request

```

```
$ curl -X POST
  -H "Content-Type: application/json"
  -d '{"user_id":"test"}'
  http://localhost:5000/chat/auth/:user_id/
```

```

- description: authenticates user with Google in order to gain access to Google Calendar
- request: `GET /authenticateGoogleUser/``
  - content-type: `application/json``
- response: 200
  - content-type: `application/json``
  - body: Google Calendar OAuth URL

## ##Consts

- description: retrieves the list of all user interests tags
- request: `GET /consts/interests/`
- response: 200
  - content-type: `application/json`
  - body: list of all user interests values

...

```
$ curl -X GET
http://localhost:5000/consts/interests/
```

...

## ##Event API

- Description: get all events in the database
- request: `GET /events/`
- response: 200
  - content-type: `application/json`
  - body: list of objects
    - public: (boolean) determines if the event is public or private
    - title: (string) the title of the event
    - username: (string) the host user of the event
    - description: (string) the description of the event
    - date: (Date) the date the event is held
    - location: ([Number]) the latitude and longitude coordinates of the event's

location

- roomId: (string) the chatkit room ID for the event
  - invited: ([ObjectId]) the IDs of the users who are invited to the event
  - attending: ([ObjectId]) the IDs of the users who are attending the event
  - tags: ([string]) the tags associated with the event
- response: 400
    - content-type: `application/json`
    - body: Error: [error message]

...

```
$ curl -X GET
http://localhost:5000/events/
```

...

- description: adds an event to the database
- request: `POST /events/`
  - content-type: `application/json`

- body: object
  - public: (boolean) determines if the event is public or private
  - title: (string) the title of the event
  - username: (string) the host user of the event
  - description: (string) the description of the event
  - date: (Date) the date the event is held
  - location: ([Number]) the latitude and longitude coordinates of the event's

location

- roomId: (string) the chatkit room ID for the event
- invited: ([ObjectId]) the IDs of the users who are invited to the event
- attending: ([ObjectId]) the IDs of the users who are attending the event
- tags: ([string]) the tags associated with the event

-response: 200

- content-type: `application/json`
- body: [ID of the new event]

-response: 400

- content-type: `application/json`
- body: Error: [error message]

...

```
$ curl -X POST
 -H "Content-Type: application/json"
 -d
'{"public":"true","title":"test","username":"test","description":"test","date":"2019-01-01","invited":[],"attending":[],"location":[0,0],"roomId":"test","tags":[]}'
 http://localhost:5000/events/
...
```

-description: retrieves all the public events from the database

- request: `GET /events/public/`

-response: 200

- content-type: `application/json`
- body: list of objects
  - public: (boolean) determines if the event is public or private
  - title: (string) the title of the event
  - username: (string) the host user of the event
  - description: (string) the description of the event
  - date: (Date) the date the event is held
  - location: ([Number]) the latitude and longitude coordinates of the event's

location

- roomId: (string) the chatkit room ID for the event
- invited: ([ObjectId]) the IDs of the users who are invited to the event

- attending: ([ObjectId]) the IDs of the users who are attending the event
- tags: ([string]) the tags associated with the event

...

```
$ curl -X GET
 http://localhost:5000/events/public/
```

...

-description: Retrieves the event given the event ID

-request: `GET /events/:id/`

-response: 200

- content-type: `application/json`
- body: object

- public: (boolean) determines if the event is public or private
- title: (string) the title of the event
- username: (string) the host user of the event
- description: (string) the description of the event
- date: (Date) the date the event is held
- location: ([Number]) the latitude and longitude coordinates of the event's

location

- roomId: (string) the chatkit room ID for the event
- invited: ([ObjectId]) the IDs of the users who are invited to the event
- attending: ([ObjectId]) the IDs of the users who are attending the event
- tags: ([string]) the tags associated with the event

- response: 400

- content-type: `application/json`
- body: Error: [error message]

...

```
$ curl -X GET
 http://localhost:5000/events/1/
```

...

-description: deletes an event from the database

- request: `DELETE /events/:id/`

- response: 200

- content-type: `application/json`
- body: [ID of the deleted event]

- response: 400

- content-type: `application/json`
- body: Error: [error message]

...

```
$ curl -X DELETE
 http://localhost:5000/events/1/
```

...

-description: updates an event given the ID and values to update

- request: `PUT /events/:id/`

- content-type: `application/json`

- body: object

- desc: add a picture to the database and put the url address in the list of pics[] for an event

- request: POST /events/pics/add/:id

- Content type: 'application JSON'

- response: 200

- Content type: application/json

- Body: picture added

- response: 400

- Content type: application/json

- Body: Error: [error message]

```
$ curl -X POST
```

```
 http://localhost:5000/events/pics/add/:id
```

- desc: get a list of picture url addresses for an event

- request: GET /events/pics/get/:id

- Content type: 'application JSON'

- Body: object

- target: (string) user id to have the friend id added to their list

- response: 200

- Content type: application/json

- Body: object

- pic: [string] url addresses for the event's pictures

- response: 400

- Content type: application/json

- Body: Error: [error message]

\$ curl -X GET

http://localhost:5000/events/pics/get/:id

- desc: get a url addresses for an event's gif

- request: GET /events/pics/gif/:id

- Content type: 'application JSON'

- Body: object

- target: (string) user id to have the friend id added to their list

- response: 200

- Content type: application/json

- Body: object

- pic: (String) url address for the event's gif

- response: 400

- Content type: application/json

- Body: Error: [error message]

\$ curl -X GET

http://localhost:5000/events/pics/gif/:id

- desc: remove a picture from the database and the url address from the list of pics[] for an event

- request: DELETE /events/pics/delete

- Content type: 'application JSON'

- response: 200

- Content type: application/json

- Body: picture deleted



- response: 400
  - Content type: application/json
  - Body: Error: [error message]

```
$ curl -X POST
http://localhost:5000/events/pics/delete/:id
```

## **##Friends API**

- desc: add current user id to the list of friend requests of another user

- request: POST /friends/requests/add/:id
  - Content type: 'application JSON'
  - Body: object
    - req: (string) user id of current user

- response: 200
  - Content type: application/json
  - Body: request sent

- response: 400
  - Content type: application/json
  - Body: Error: [error message]

```
$ curl -X POST
http://localhost:5000/friends/requests/add/:id
```

- desc: add current user id to the list of friends of target user

- request: POST /friends/friends/add/:id
  - Content type: 'application JSON'
  - Body: object
    - target: (string) user id to have the friend id added to their list

- response: 200

- Content type: application/json
- Body: request sent

- response: 400

- Content type: application/json
- Body: Error: [error message]

\$ curl -X POST

http://localhost:5000/friends/requests/add/:id

- desc: remove a user id from the list of friend requests of the current user

- request: DELETE /friends/requests/delete/:id

- Content type: 'application JSON'

- Body: object

- req: (string) user id of current user

- response: 200

- Content type: application/json

- Body: request deleted

- response: 400

- Content type: application/json

- Body: can't delete: [error message]

\$ curl -X POST

http://localhost:5000/friends/requests/delete/:id

- desc: remove a user id from the list of friends of the current user

- request: DELETE /friends/friends/add/:id

- Content type: 'application JSON'

- Body: object

- target: (string) user id of current user

- response: 200

- Content type: application/json

- Body: friend deleted

- response: 400

- Content type: application/json

- Body: can't delete: [error message]

\$ curl -X POST

<http://localhost:5000/friends/requests/delete/:id>

- desc: edit an event
- request: 'PUT /events/:id'
  - body:
    - event object
- response: 200
  - the updated event object
- response: 401
  - The user does not have permission to update the event (not attending/not invited/not the creator)

## ##Users API

- desc: attend or unattend an event
- request: 'PATCH /users/:id/attendingEvents'
  - body:
    - op: (string) add or remove
    - value: (ObjectId) the event id
- Response: 200
  - content-type: `application/json`
  - body: the updated user
- Response: 400
  - error: The user is already attending

-description: Gets all the users from the database

-request: `GET /users/`

-response: 200

- content-type: `application/json`
- body: list of objects
  - username: (string) the username of the user
  - password: (string) the salted password of the user
  - salt: (string) the salt for the password
  - name: (string) the real name of the user
  - interests: ([string]) the list of interests the user has
  - friends: ([ObjectId]) the list of user object IDs that are friends with the user
  - friend\_requests: ([ObjectId]) the list of users who are requesting to be friends with the user
  - invitedEvents: ([ObjectId]) the list of events that the user is invited to

- attendingEvents: ([ObjectId]) the list of events that the user is planning to attend
- createdEvents: ([ObjectId]) the list of events that the user created
- history: ([ObjectId]) the list of events that the user has attended in the past
- location: ([Number]) the latitude and longitude of the user's home
- gcAuthToken: (string) the authentication token for the user's google calendar

- response: 400

- content-type: `application/json`
- body: Error: [error message]

-response: 403

...

```
$ curl -X GET
http://localhost:5000/users/
```

...

-description: Gets all the currently logged in user

-request: `GET /users/currentUser/`

-response: 200

- content-type: `application/json`
- body: Object
  - username: (string) the username of the user
  - password: (string) the salted password of the user
  - salt: (string) the salt for the password
  - name: (string) the real name of the user
  - interests: ([string]) the list of interests the user has
  - friends: ([ObjectId]) the list of user object IDs that are friends with the user
  - friend\_requests: ([ObjectId]) the list of users who are requesting to be friends

with the user

- invitedEvents: ([ObjectId]) the list of events that the user is invited to
- attendingEvents: ([ObjectId]) the list of events that the user is planning to attend
- createdEvents: ([ObjectId]) the list of events that the user created
- history: ([ObjectId]) the list of events that the user has attended in the past
- location: ([Number]) the latitude and longitude of the user's home
- gcAuthToken: (string) the authentication token for the user's google calendar

- response: 400

- content-type: `application/json`
- body: Error: [error message]

-response: 403

...

```
$ curl -X GET
http://localhost:5000/users/currentUser/
```

...

- description: Retrieves a user with the given ID
- request: `GET /users/:id/`
- response: 200
  - content-type: `application/json`
  - body: object
    - username: (string) the username of the user
    - password: (string) the salted password of the user
    - salt: (string) the salt for the password
    - name: (string) the real name of the user
    - interests: ([string]) the list of interests the user has
    - friends: ([ObjectId]) the list of user object IDs that are friends with the user
    - friend\_requests: ([ObjectId]) the list of users who are requesting to be friends

with the user

- invitedEvents: ([ObjectId]) the list of events that the user is invited to
- attendingEvents: ([ObjectId]) the list of events that the user is planning to attend
- createdEvents: ([ObjectId]) the list of events that the user created
- history: ([ObjectId]) the list of events that the user has attended in the past
- location: ([Number]) the latitude and longitude of the user's home
- gcAuthToken: (string) the authentication token for the user's google calendar
- response: 400
  - content-type: `application/json`
  - body: Error: [error message]
- response: 404
  - content-type: `application/json`
  - body: User not found

...

```
$ curl -X GET
http://localhost:5000/users/test/
```

...

- description: Updates a user with the given ID
- request: `PATCH /users/:id/`
  - content-type: `application/json`
  - body: object
    - firstname: (string) first name of the user
    - lastname: (string) last name of the user
    - location: ([Number]) the latitude and longitude for the home of the user
    - address: (string) the address of the user
- response: 200

- content-type: `application/json`
- body: object
  - username: (string) the username of the user
  - password: (string) the salted password of the user
  - salt: (string) the salt for the password
  - name: (string) the real name of the user
  - interests: ([string]) the list of interests the user has
  - friends: ([ObjectId]) the list of user object IDs that are friends with the user
  - friend\_requests: ([ObjectId]) the list of users who are requesting to be friends

with the user

- invitedEvents: ([ObjectId]) the list of events that the user is invited to
- attendingEvents: ([ObjectId]) the list of events that the user is planning to attend
- createdEvents: ([ObjectId]) the list of events that the user created
- history: ([ObjectId]) the list of events that the user has attended in the past
- location: ([Number]) the latitude and longitude of the user's home
- gcAuthToken: (string) the authentication token for the user's google calendar
- response: 400
- response: 500

```

$ curl -X PATCH
 -H "Content-Type: application/json"
 -d
 '{"firstname":"test","lastname":"test","location":[0,0],"address":"test"}'
 http://localhost:5000/users/1/

```

- description: Updates a users set of interests
- request: `PUT /users/:id/interests/`
  - content-type: `application/json`
  - body: [string] array of interests to be added to the user
- response: 200
  - content-type: `application/json`
  - body: object
    - username: (string) the username of the user
    - password: (string) the salted password of the user
    - salt: (string) the salt for the password
    - name: (string) the real name of the user
    - interests: ([string]) the list of interests the user has
    - friends: ([ObjectId]) the list of user object IDs that are friends with the user
    - friend\_requests: ([ObjectId]) the list of users who are requesting to be friends

with the user

- invitedEvents: ([ObjectId]) the list of events that the user is invited to
  - attendingEvents: ([ObjectId]) the list of events that the user is planning to attend
  - createdEvents: ([ObjectId]) the list of events that the user created
  - history: ([ObjectId]) the list of events that the user has attended in the past
  - location: ([Number]) the latitude and longitude of the user's home
  - gcAuthToken: (string) the authentication token for the user's google calendar
- response: 400
- content-type: `application/json`
  - body: Invalid interest [invalid interest name] specified
- response: 500

```

$ curl -X PUT
 -H "Content-Type: application/json"
 -d '{"value":["Music"]}'
 http://localhost:5000/users/1/interests/

```

- description: Retrieves a user with the given username
- request: `GET /users/:username/`
- response: 200
- content-type: `application/json`
  - body: object
    - username: (string) the username of the user
    - password: (string) the salted password of the user
    - salt: (string) the salt for the password
    - name: (string) the real name of the user
    - interests: ([string]) the list of interests the user has
    - friends: ([ObjectId]) the list of user object IDs that are friends with the user
    - friend\_requests: ([ObjectId]) the list of users who are requesting to be friends with the user
    - invitedEvents: ([ObjectId]) the list of events that the user is invited to
    - attendingEvents: ([ObjectId]) the list of events that the user is planning to attend
    - createdEvents: ([ObjectId]) the list of events that the user created
    - history: ([ObjectId]) the list of events that the user has attended in the past
    - location: ([Number]) the latitude and longitude of the user's home
    - gcAuthToken: (string) the authentication token for the user's google calendar
- response: 400
- content-type: `application/json`
  - body: Error: [error message]
- response: 404
- content-type: `application/json`



- body: User not found

...

```
$ curl -X GET
http://localhost:5000/users/test/
```

...

- description: Retrieves the currently logged in user's friend requests
- request: `GET /users/currentUser/requests/`
- response: 200
  - content-type: `application/json`
  - body: [ObjectID] the list of users who are requesting to be friends with the logged in user
- response: 400
  - content-type: `application/json`
  - body: Error: [error message]
- response: 404
  - content-type: `application/json`
  - body: User not found

...

```
$ curl -X GET
http://localhost:5000/users/currentUser/requests/
```

...

- description: Retrieves the currently logged in user's list of friends
- request: `GET /users/currentUser/friends/`
- response: 200
  - content-type: `application/json`
  - body: [ObjectID] the list of users who are friends with the logged in user
- response: 400
  - content-type: `application/json`
  - body: Error: [error message]
- response: 404
  - content-type: `application/json`
  - body: User not found

...

```
$ curl -X GET
http://localhost:5000/users/currentUser/friends/
```

...