

Notes:

Only notable response codes are mentioned (e.g. 500 for server error, or 401 for invalid authentication is not mentioned)

User operations:

Login (POST)

Route	/api/user/login
Request Body	{ username: string, password: string}
Response Body	{isAdmin: bool}
Response codes	200 - successful login 400 - invalid login or no user found

Check session (GET)

Route	/api/user/check-session
Response Body	{isAdmin: bool}
Response codes	200 - successful login 400 - invalid login or no user found

Signup (POST)

Route	/api/user/create
Request Body	{ username: string, password: string, email: string}
Response Body	The user object created
Response codes	200 - new user created 403 - bad formatted request or non-unique username -no user created

Note: the name of the user will be set to the username string

Get profile info of user with specific ID (GET)

Route	/api/user/profile/:id
Response Body	user object with profile info - matches user schema
Response codes	200 - return profile info 404 - not logged in or user does not exist

Get profile info of user currently logged in (GET)

Route	/api/user
Response Body	user object with profile info - matches user schema
Response codes	200 - return profile info 404 - not logged in or user does not exist

Edit profile (PUT)

Route	/api/user/modify
Request Body	{name: string, email: string, userName: string, password: string, bio: string, profileImageIndex: number}
Response Body	user object with profile info - matches user schema
Response codes	200 - return profile info 404 - not logged in or user does not exist

Get reported users (GET)

Route	/api/user/report
Request Body	
Response Body	User list - matches user schema
Response codes	200 - return reported users 404 - not logged in or user does not exist

Report User (POST)

Route	/api/user/report
Request Body	{userID: string}
Response codes	200 - updated user info 404 - not logged in or user does not exist

Unreport User (PUT) (admin only)

Route	/api/user/unreport
Request Body	{userID: string}
Response Body	
Response codes	200 - updated user info 404 - not logged in or user does not exist

Delete User (DELETE)

Route	/api/user/unreport
Request Body	{userID: string}
Response Body	
Response codes	200 - updated user info 404 - not logged in or user does not exist

Posting operations:

Note: posting objects returned have some additional info about the users associated with the post added on for example name, profileImageIndex,

Search (POST)

Does not return posts that the current user is a creator of or reported posts

Route	/api/postings/search
Request Body	{ tags: list } - optionally can be empty to return all posts
Response Body	list[Posting]
Response codes	200 - return list of postings, may be empty if no tags match

Report (PUT)

Route	/api/postings/report
Request Body	{ posting_id : string }
Response Body	
Response codes	200 - Updated Posting 404 - Not Found (posting)

Un-Report Post (PUT)

(admin only)

Route	/api/postings/unreport
Request Body	{ posting_id : string }
Response Body	

Response codes	200 - Updated Posting 404 - Not Found (posting)
----------------	--

Apply (POST)

Route	/api/postings/apply
Request Body	{ posting_id : string }
Response codes	200 - Updated Posting 404 - Not Found (either user or posting)

Create post (POST)

Route	/api/postings
Request Body	{title: str, tags: string[], endDate:str, description:str, capacity: int}
Response codes	200 - Applicant Removed 404 - Not Found

Accept applicant (PUT) (only creator)

Route	/api/postings/accept
Request Body	{ posting_id : string, user_id : string }
Response Body	
Response codes	200 - Applicant Moved to Members 404 - Not Found

Deny applicant (PUT) (only creator)

Route	/api/postings/decline
Request Body	{ posting_id : string, user_id : string }
Response Body	
Response codes	200 - Applicant Moved to Members 404 - Not Found

Delete post (DELETE) (only creator or admin)

Route	/api/postings
Request Body	{ posting_id : integer }
Response codes	200 - Posting Deleted 404 - Not Found

Delete Application (DELETE) (only creator or admin)

Route	/api/postings/apply
Request Body	{ posting_id : integer }
Response codes	200 - ApplicationDeleted 404 - Not Found

Get Post from postID (GET)

Route	/api/postings/get-by-id/:pid
URL parameters?	Posting id
Response Body	Posting object
Response codes	200 - Posting Returned 404 - Not Found

Get posts the logged in user has created (GET) done, connected

Route	/api/postings/created
Response Body	list[Posting]
Response codes	200 - Posting Returned 404 - Not Found

Get posts the currently logged in user is a member of (not creator) (GET) connected to manage page (not to profile yet)

Route	/api/postings/member
Response Body	list[Posting]
Response codes	200 - Posting Returned 404 - Not Found

Get reported posts (GET)

Route	/api/postings/report
Response Body	list[Posting]
Response codes	200 - Posting Returned 404 - Not Found

Get posts that the logged in user has a PENDING application for (GET)
done, connected

Route	/api/postings/pending
Response Body	{ postings : list[Posting] }
Response codes	200 - Posting Returned 404 - Not Found

Get posts that the logged in user has a DENIED application for (GET)

Route	/api/postings/denied
Response Body	{ postings : list[Posting] }
Response codes	200 - Posting Returned 404 - Not Found

Add comment to post (POST)
(creator, member or admin only)

Route	/api/postings/denied
Request Body	{ content: str, postID: str }
Response codes	200 - Created comment 404 - Posting Not Found

Edit Post (PUT)

Route	/api/postings
Request Body	{posting, postID}
Response codes	200 - Successfully Edited post 404 - Not Found