

Events API

Action	Request format	Response format	Possible errors
Get all events, optionally sorted by posted date, optionally starting from some time.	GET /api/events ?sortBy= <i>sortBy</i> &start= <i>start</i> &tags= <i>tags</i> &genlocs= <i>genlocs</i> &rsvp= <i>rsvp</i> &following= <i>boolean</i> <i>sortBy</i> : 'posted' or 'start' <i>start</i> : milliseconds <i>tags</i> : comma-separated <i>genlocs</i> : comma-separated <i>rsvp</i> : boolean <i>following</i> : boolean	Array of arrays of event objects which start on the same days. <pre>[[{ name: ... }, ...], ...]</pre>	DatabaseError NotFoundError
Create a new event	POST /api/groups/:groupId/events <pre>{ name: String, group: String ID description: String start: Milliseconds end: Milliseconds locationGeneral: String locationDescription: String tags: Stringified string array }</pre>	<pre>{ eventId: ... }</pre>	DatabaseError Unauthorized
Set event image	POST /api/groups/:groupId/events/:eventId/image <pre>{ image: imageData }</pre>	<pre>{ imageURL: ... }</pre>	NotFoundError DatabaseError Unauthorized SiteAdminRestrictionError
Edit event info	POST /api/groups/:groupId/events/:eventId <pre>{ name: String, description: String }</pre>	<pre>{ }</pre>	NotFoundError DatabaseError Unauthorized

	start: Milliseconds end: Milliseconds locationGeneral: String locationDescription: String tags: Stringified string array } <i>All body fields optional</i>		
Delete event	DELETE /api/groups/:groupId/events/:eventId	{}	NotFoundError DatabaseError Unauthorized
RSVP to event	POST /api/events/id/:eventId/rsvps/:userId	{}	NotFoundError DatabaseError Unauthorized SiteAdminRestrictionError
Un-RSVP to event	DELETE /api/events/id/:eventId/rsvps/:userId	{}	NotFoundError DatabaseError Unauthorized SiteAdminRestrictionError
Get RSVPs for event	GET /api/events/id/:eventId/rsvps	[{ userId: ... visible: ... }] If not visible and userId is not logged in user, userId is undefined.	NotFoundError DatabaseError

Groups API

Action	Request format	Response format	Possible errors
Get all approved groups.	GET /api/groups	Array of approved group objects [_id: ..., name: ..., description: ..., imageUrl: ..., admins: [{ }]], ...]	DatabaseError
Create an approved group. Action can only be performed by the Site Admin.	POST /api/groups { name: ..., description: ..., imageUrl: ..., creator }	Approved Group object { _id: ..., name: ..., description: ..., imageUrl: ..., admins: [{ }] }	SiteAdminAuth or DatabaseError
Get a specific group by its group id.	GET /api/groups/:id	Approved Group object specified by the input id. { _id: ..., name: ..., description: ..., imageUrl: ..., admins: [{ }] }	DatabaseError
Delete a group by its group id.	DELETE /api/groups/:id	Empty JSON {}	DatabaseError AdminAuthError
Put description changes	PUT /api/groups/:id/description { description: ... }	Approved Group object with updated description { _id: ..., name: ..., description: ..., imageUrl: ..., admins: [{ }] }	DatabaseError AdminAuthError
Put a new User as a Group admin. Requires the User's email to be sent.	PUT /api/groups/:id/admin { email: ... }	Approved Group object with updated admin array { _id: ..., name: ..., description: ..., imageUrl: ...,	DatabaseError AdminAuthError

		admins: [{ }] }	
Delete a User from being a Group admin. Requires the User's id to be sent.	DELETE /api/groups/:id/admin { personId: ... }	Approved Group object with updated admin array { _id: ..., name: ..., description: ..., imageURL: ..., admins: [{ }] }	DatabaseError AdminAuthError
Post image URL as a Group's image.	POST /api/groups/:id/image	JSON Object of the imageURL { imageURL: imageURL }	DatabaseError AdminAuthError SiteAdminRestrictionError

Users API

Action	Request format	Response format	Possible errors
Login	GET /users/login	redirects to homepage	No User Error Account Not Activated Error DatabaseError
Create a new user	POST /users	redirects to verification email route users/submit/:mail	User Already Exists
Send Verification Email	GET users/submit/:mail	redirect to homepage	Mailgun server error
Logs a user out	POST users/logout	redirects to homepage and destroys session	
Gets the verify account page	GET users/:userId/verify	Sends the user to the verify account page	
Attempts to verify account associated with userId	POST users/:userId/verify	Redirects to homepage	No User Error Database Error
Sends password to user	POST users/sendPassword	Redirects to homepage	No User Error Database Error Mailgun Server Error
Deletes user account	POST users/delete	Redirects to homepage	No User Error Database Error

Pending Groups API

Action	Request format	Response format	Possible errors
Get all pending groups	GET <code>/api/pending_groups</code>	Array of pending group objects [{ _id: ..., name: ..., description: ..., creator: { _id: ..., name: ..., email: ... } }, ...]	SiteAdminAuthError DatabaseError
Create a pending group	POST <code>/api/pending_groups</code>	Object with pending group id of newly created pending group { pendingGroupId: ... }	UserAuthError SiteAdminNotAllowedError DatabaseError
Delete a pending group	DELETE <code>/api/pending_groups/:pendingGroupId</code>	Empty JSON { }	SiteAdminAuthError DatabaseError