

Nicole Carr

SWDV 691

Service Layers Design

November 7th, 2021

Teeny Tiny Explorers API

Get Started with Teeny Tiny Explorers

The API document is broken up into three main sections. The first section provides a brief overview of the endpoints in Teeny Tiny Explorers. The second section is a reference section and includes detailed information about the endpoints with examples of requests and provides successful and unsuccessful request messages.

The last section is the API diagram, a very high-level diagram to demonstrate which endpoints belong to which pages within Teeny Tiny Explorers.

Overview

Users

Manage account settings and profile (MVP)

- **GET user/login**
 - Logs the user into the system
- **GET user/logout**
 - Logs the user out of the system
- **POST /signup**
 - Creates new user in system

Search and get local groups(MVP)

- **GET /localgroup**
 - Find local groups by state or county

Search and get events (Stretch Feature)

- **GET users/events**
 - find events by county or state

Become a host (MVP)

- POST /becomeahost
 - Creates host inquiry

Hosts

Manage members on member list (MVP)

- POST hosts/member
 - Host can add member information
- DELETE hosts/member
 - Host can delete member
- PUT hosts/member
 - Host updates member
- GET hosts/member
 - Host searches for member by name

Search, manage, and get local groups (MVP)

- POST hosts/localgroup
 - Host creates local group
- DELETE hosts/localgroup
 - Host deletes local group
- PUT hosts/localgroup
 - Host updates local group
- GET hosts/localgroups
 - Host searches local group by county or state

Search, manage, and get events (Stretch Feature)

- POST hosts/events
 - Host creates event
- DELETE hosts/events
 - Host deletes event
- PUT hosts/events
 - Host updates event
- GET hosts/events
 - Host searches for events by county or state

Reference

Users

Manage account settings and profile(MVP)

The manage account settings and profile allows user to login and logout of the system.

GET user/login

Description:

Logs the user into the system on the login page

Resource URL:

<https://teenytinyexplorers.com/user/login>

Parameters:

Name	Required	Type	Description	Example
Username	Required	String	The name of the user that is logging into the account	nicolecarr
Password	Required	String	Password that the user created	123plumAYx

Scenario:

A user logs into the system with their username and password. The successful message shows when the login is correct.

The unsuccessful would generate if the username or password that are entered are not correct. The user would need to enter the information again, and try to login.

Example Request:

```
Curl -X 'GET' \  'http://teenytinyexplorerres.com/user/login?
username=nicolecarr&password=123plumAYx'
```

```
-H 'accept: application/json'
```

Example Response:

Successful:

200 successful operation

Successful Response body:

```
{
  "code": 200,
  "type": "unknown",
```

```
    "message": "logged in user session: 6543"  
  }
```

Unsuccessful:

400 invalid username/password

GET user/logout

Description:

Logs the user out of the user account.

Resource URL:

<https://teenytinyexplorers.com/user/logout>

Parameters:

Name	Required	Type	Description	Example
No parameters				

Scenario:

The user needs to log out of the system, and because there are no parameters needed, nothing is listed. Logging out requires the user to click the logout button.

Example Request:

```
Curl -X 'GET' \  
'https://teenytinyexplorers.com/user/logout\  
-H 'accept: application/json'
```

Example Response:

Successful:

200 successful operation

Response body:

```
{  
    "code": 200,
```

```
    "type": "unknown",  
    "message": "ok"  
}
```

POST /signup

Description:

Creates a user with provided input

Resource URL:

<https://teenytinyexplorers.com/signup>

Parameters:

Name	Required	Type	Description	Example
User information	Required	Array	List of user information	[{ "id": 0, "username": "nicolecarr", "firstName": "nicki", "lastName": "carr", "email": "nc13@gmail.com", "password": "123abc", "phone": "3214569876" }]

Scenario:

A new user comes to the site and needs to create a login. The information that is needed includes their chosen username, first name, last name, email, password, and phone number.

The unsuccessful scenario would be if the user did not enter all fields.

Example Request:

```
Curl -X 'POST' \  
'https://teenytinyexplorers.com/signup'
```

```
-H 'accept: application/json\  
-H 'Content-Type: application/json\  
-d [  
{  
  "id": 0,  
  "username": "nicolecarr",  
  "firstName": "nicki",  
  "lastName": "carr",  
  "email": "nc13@gmail.com",  
  "password": "123abc",  
  "phone": "3124569876"  
}  
]
```

Example Response:

Successful:

200 successful operation

Response body:

```
{  
  "code": "200",  
  "type": "unknown",  
  "message": "ok"  
}
```

unsuccessful:

500 Error: Response Status is 500

Response body:

```
{  
  "code": "500",
```

```
"type": "unknown",  
"message": "something is incorrect"  
}
```

Search and get local groups(MVP)

Users will utilize the search and get local groups to search the information on the “find a local group” page on the Teeny Tiny Explorers web application.

GET /localgroup

Description:

Finds local group by county or state

Resource URL:

<https://teenytinyexplorers.com/localgroup>

Parameters:

Name	Required	Type	Description	Example
State	Optional	String	The state the user lives in will be entered to find a state near them	Michigan
County	Optional	String	The county the user lives in will be entered to find a state near them	Monroe

Scenario:

A user in Monroe Michigan is wanting to find a group near them. The user goes to the table to find a local group and looks for Michigan first. They see that there is a group in Monroe and does not need to search further.

The unsuccessful scenario would be if the user did not find any results for a group near them. They entered Michigan and there are no results at all.

Example Request:

```
Curl -X 'GET'\  
'https://teenytinyexplorers.com/localgroup/michigan'\  
-H 'accept: application/json'
```

Example Response:**Successful:**

200 successful operation

Response body:

```
{  
  "code": "200",  
  "type": "unknown",  
  "message": "ok"  
}
```

Unsuccessful:

404 Error: responses status is 404

Response body:

```
{  
  "code": 1,  
  "type": "error"  
  "message": "Not found"  
}
```

Search and get events(Stretch Feature)

Users will utilize the search and get events to search the information on the “events” page on the Teeny Tiny Explorers web application.

GET user/events

Description:

Finds events by county or state

Resource URL:

<https://teenytinyexplorers.com/user/events>

Parameters:

Name	Required	Type	Description	Example
State	Optional	String	The state the user lives in will be entered to find a state near them	Ohio
County	Optional	String	The county the user lives in will be entered to find a state near them	Adams

Scenario:

A user in Ohio is wanting to find an event in Adams County, in Ohio. They enter Adams in the search fields, and they find a result.

An unsuccessful message would occur if the member entered the information and there were no search results.

Example Request:

```
Curl -X 'GET'\  
'https://teenytinyexplorers.com/user/events/adams'\  
-H 'accept: application/json'
```

Example Response:**Successful:**

200 successful operation

Response body:**Response body:**

```
{  
  "code": "200",
```

```
"type": "unknown",  
"message": "ok"  
}
```

Unsuccessful:

404 Error: responses status is 404

Response body:

```
{  
"code": 1,  
"type": "error"  
"message": "Not found"  
}
```

Become a Host(MVP)

Users on the web application can enter their information on the “become a host” form. The collected information is then used by hosts and administrators to reach out to the applicant via phone.

POST /becomeahost

Description:

User is able to submit a form to become a host for Teeny Tiny Explorers

Resource URL:

<https://teenytinyexplorers.com/becomeahost>

Parameters:

Name	Required	Type	Description	Example
User information	Required	Array	List of user information	[{ "firstName": "Jenna", "lastName": "Smith",

				<pre> "email": "jsmith@aim.com", "phone": "9897865432", "address": "123 oak", "state": "Michigan", "county": "Wayne", "zip": "48192" }] </pre>
--	--	--	--	---

Scenario:

A user wants to become a host for a local group. They enter their information in the form and submit. The information is successfully submitted.

An unsuccessful scenario would be the user did not enter all of the required fields and they will receive an error message.

Example Request:

```

Curl -X 'POST'\
'https://teentyexplorer.com/becomehost'\
-H 'accept: application/json'\
-H 'Content-Type: application/json'\
-d '{
  "firstName": "Jenna",
  "lastName": "Smith",
  "email": "jsmith@aim.com",
  "phone": "9897865432",
  "address": "123 oak",
  "state": "Michigan",
  "county": "Wayne",

```

```
"zip": "48192"  
}
```

Example Response:

Successful:

200 successful operation

Response body:

Response body:

```
{  
  "code": "200",  
  "type": "unknown",  
  "message": "submitted"  
}
```

Unsuccessful:

404 Error: responses status is 404

Response body:

```
{  
  "code": "500",  
  "type": "unknown",  
  "message": "Missing requirements"  
}
```

Hosts

Manage members on member list(MVP)

Hosts serve as admins and can create new members, edit, or delete.

POST hosts/member

Description:

Host can add member information in the event they need to manually add a new member

Resource URL:

<https://teenytinyexplorers.com/hosts/member>

Parameters:

Name	Required	Type	Description	Example
User information	Required	Array	List of user information	[{ "id": 45, "username": "jcarr", "firstName": "Josh", "lastName": "Car r", "email": "jcarr@a im.com", "password": "can 33", "phone": "4536755656" }]

Scenario:

A host is needing to manually enter a new member to they system.
The host enters all of the needed information.

An unsuccessful attempt would result if there is missing information
in the necessary fields.

Example Request:

Curl -X 'POST' \

```
'https://teenytinyexplorers.com/hosts/member'\n-H 'accept: application/json'\n-H 'Content-Type: application/json'\n-d [\n{\n  "id": 45,\n  "username": "jcarr",\n  "firstName": "josh",\n  "lastName": "carr",\n  "email": "jcarr@aim.com",\n  "password": "can33",\n  "phone": "4536755656"\n}\n]
```

Example Response:

Successful:

200 successful operation

Response body:

```
{\n  "code": "200",\n  "type": "unknown",\n  "message": "ok"\n}
```

unsuccessful:

500 Error: Response Status is 500

Response body:

```
{
```

```
"code": "500",  
"type": "unknown",  
"message": "Missing requirements"  
}
```

DELETE hosts/member

Description:

Host can delete member information. In this example, a member is being deleted out of the system.

Resource URL:

<https://teenytinyexplorers.com/hosts/member>

Parameters:

Name	Required	Type	Description	Example
Member ID	Required	int	The idea of the member in the system this is autoincremented	45 This corresponds to member Josh Carr

Scenario:

A host that also functions as an admin is needing to remove a member from the database. They search for the member by their member ID and delete them.

An unsuccessful scenario such as that user not being in the system, would result in an error.

Example Request:

```
Curl -X 'DELETE'\  
'https://teenytinyexplorers.com/hosts/member/45'\  
-H 'accept: application/json'
```

-H 'api_key: 45'

Example Response:

Successful:

200 successful operation

Response body:

```
{  
  "code": "200",  
  "type": "unknown",  
  "message": "ok"  
}
```

unsuccessful:

404 Error: Response Status is 404

Response body:

```
{  
  "code": "404",  
  "type": "unknown",  
  "message": "Member not found"  
}
```

PUT hosts/member

Description:

Host can update member information. In this example a host is updating the last name of a member

Resource URL:

<https://teenytinyexplorers.com/hosts/member>

Parameters:

Name	Required	Type	Description	Example
ID	required	int	This is the specific member id that is original to each member in the system. No member has the same ID	45
firstName	required	string	The members first name in the system	Josh
lastName	required	string	The members last name in the system	Carr

The members last name is Carr but needs to be updated to Carn. The member is already stored with the below information:

```
{
  "id": 45,
  "username": "jcarr",
  "firstName": "josh",
  "lastName": "carr",
  "email": "jcarr@aim.com",
  "password": "can33",
  "phone": "4536755656"
}
```

Example Request:

```
Curl -X 'PUT'\
'https://teenytinyexplorers.com/hosts/member/update/Carn'\
-H 'accept: application/json'\
-H 'Content-Type: application/json'\
-d '{
  "id": 45,
  "username": "jcarr",
  "firstName": "josh",
```

```
"lastName":"carn",  
"email": "jcarr@aim.com",  
"password": "can33",  
"phone": "4536755656"  
}'
```

Example Response:

Successful:

200 successful operation

Response body:

```
{  
"code": "200",  
"type": "unknown",  
"message": "Updated!"  
}
```

unsuccessful:

500 Error: Response Status is 500

```
{  
"code": "505",  
"type": "unknown",  
"message": "Invalid entry"  
}
```

GET hosts/member

Description:

Host can search for member in table by name

Resource URL:

<https://teenytinyexplorers.com/hosts/member/find>

Parameters:

Name	Required	Type	Description	Example
Member First Name	Optional	String	Member first name	Nicole
Member Last Name	Optional	String	Member last name	Carr

Scenario:

A host needs to find a member and searches by first name and last name. The host can locate the member by using the search entries.

An error message will results if the entered search fields yield no results.

Example Request:

```
Curl -X 'GET'\  
'https://teenytinyexplorers.com/hosts/member/nicole'\  
-H 'accept: application/json'
```

Example Response:**Successful:**

200 successful operation

Response body:

```
{  
  "code": "200",  
  "type": "unknown",  
  "message": "Member found"  
}
```

Unsuccessful:

404 Error: responses status is 404

Response body:

```
{  
  "code": 1,  
  "type": "error"  
  "message": "Member not found"  
}
```

Search, manage, and get local groups(MVP)

Hosts serve as admins and can create new groups, edit, or delete.

POST hosts/localgroup**Description:**

Host can add group information

Resource URL:

<https://teenytinyexplorers.com/admin/localgroup>

Parameters:

Name	Required	Type	Description	Example
Group information	Required	Array	List of group information	[{ "id": 2, "state": "Michigan", "hostName": "Nicole Carr", "county": "Wayne", "email": "ncarr1@gmail.com" }]

Scenario:

An admin is needing to create a new group in the system. The admin creates the group and includes all of the necessary fields in the entry.

An error will result if all of the needed fields are not included in the entry when creating the new group

Example Request:

```
Curl -X 'POST' \  
'https://teenytinyexplorers.com/hosts/localgroup\'\  
-H 'accept: application/json\'\  
-H 'Content-Type: application/json\'\  
-d '[  
{  
  "id": 2,  
  "state": "Michigan",  
  "hostName": "Nicole Carr",  
  "county": "Wayne",  
  "email": "ncarr1@gmail.com",  
}  
'
```

Example Response:

Successful:

200 successful operation

Response body:

```
{  
  "code": "200",  
  "type": "unknown",  
  "message": "User created"  
}
```

unsuccessful:

500 Error: Response Status is 500

Response body:

```
{  
  "code": "500",  
  "type": "unknown",  
  "message": "please try again"  
}
```

DELETE hosts/localgroup**Description:**

Host can delete local group information, the group is able to be removed fully by utilizing the group ID.

Resource URL:

<https://teenytinyexplorers.com/hosts/localgroup>

Parameters:

Name	Required	Type	Description	Example
Group ID	Required	int	The group ID is being used to remove a group from the system.	2

Scenario:

An admin is needing to delete a group in the system. The admin goes to delete the group with the group ID 2. The group was successfully deleted.

An error will result if the group ID does not match.

Example Request:

Curl -X 'DELETE'

'https://teenytinyexplorers.com/hosts/localgroup/2'

-H 'accept: application/json'

-H 'api_key: 2'

Example Response:

Successful:

200 successful operation

Response body:

```
{  
  "code": "200",  
  "type": "unknown",  
  "message": "Group removed"  
}
```

unsuccessful:

404 Error: Response Status is 404

Response body:

```
{  
  "code": "404",  
  "type": "unknown",  
  "message": "Group not found"  
}
```

PUT hosts/localgroup

Description:

Host can update local group information and can update the ID, state, host name, county, and email of the host.

Resource URL:

<https://teenytinyexplorers.com/hosts/localgroup>

Parameters:

Name	Required	Type	Description	Example
Group ID	Required	int	Group ID that is specific to the group.	2
Host Name	Optional	string	The host name is the individual that runs the group	Nicole Carr
Email	Optional	string	The email is the email for the host of the group	Ncarr1@gmail.com
County	Optional	String	The county is the county in which the group takes place	Wayne
State	Optional	String	The state is the state in which the group takes place	Michigan

The group host is Nicole Carr and her email is ncarr1@gmail.com but needs to be updated. The group is already stored with the below information:

```
[
  {
    "id": 2,
    "state": "Michigan",
    "hostName": "Nicole Carr",
    "county": "Wayne",
    "email": "ncarr1@gmail.com",
  }
]
```

Example Request:

```
Curl -X 'PUT'\
'https://teenytinyexplorers.com/hosts/group/2'\
-H 'accept: application/json'\
-H 'Content-Type: application/json'\
-d '{
```



```
"id": 2,  
"state": "Michigan",  
"hostName": "Nicole Carr",  
"county": "Wayne",  
"email": "ncarr1@maryville.com",  
}
```

Example Response:

Successful:

200 successful operation

Response body:

```
{  
"code": "200",  
"type": "unknown",  
"message": "Updated!"  
}
```

unsuccessful:

500 Error: Response Status is 500

```
{  
"code": "505",  
"type": "unknown",  
"message": "Please try again"  
}
```

GET hosts/localgroup

Description:

Host can search for local group in table by name

Resource URL:

<https://teenytinyexplorers.com/hosts/localgroup>

Parameters:

Name	Required	Type	Description	Example
State	Optional	String	The state the user lives in will be entered to find a state near them	Michigan
County	Optional	String	The county the user lives in will be entered to find a state near them	Oakland

Scenario:

A user is searching for a local group and is looking by county. The user enters the county and finds a group in the county they are looking for.

An error will result if the county was entered or the state and there were no groups in that location.

Example Request:

```
Curl -X 'GET'\n'https://teenytinyexplorers.com/hosts/localgroup/oakland'\n-H 'accept: application/json'
```

Example Response:**Successful:**

200 successful operation

Response body:

```
{\n  "code": "200",\n  "type": "unknown",\n  "message": "ok"\n}
```

Unsuccessful:

404 Error: responses status is 404

Response body:

```
{  
  "code": 1,  
  "type": "error"  
  "message": "Group not found"  
}
```

Search, manage, and get events

Hosts serve as admins and can create new events, edit, or delete.

POST hosts/events**Description:**

Host can add group information

Resource URL:

<https://teenytinyexplorers.com/hosts/events>

Parameters:

Name	Required	Type	Description	Example
Event information	Required	Array	List of event information	[{ "id": 3, "state": "Ohio", "hostName": "Nicole Carr", "county": "Adam s", "email": "ncarr@ maryville.com", "phone": "73456 73214",

				<pre> "date": "12212021" }]</pre>
--	--	--	--	------------------------------------

Scenario:

An admin is needing to create a new event in the system. The admin creates the event and includes all of the necessary fields in the entry.

An error will result if all of the needed fields are not included in the entry when creating the new event

Example Request:

```

Curl -X 'POST' \
'https://teenytinyexplorers.com/hosts/events' \
-H 'accept: application/json' \
-H 'Content-Type: application/json' \
-d '[
  {
    "id": 3,
    "state": "Ohio",
    "hostName": "Nicole Carr",
    "county": "Adams",
    "email": "ncarr@maryville.com",
    "phone": "7345673214",
    "date": "12212021"
  }
]
```

Example Response:

Successful:

200 successful operation

Response body:

```
{
```

```
"code": "200",  
"type": "unknown",  
"message": "Event created"  
}
```

unsuccessful:

500 Error: Response Status is 500

Response body:

```
{  
"code": "500",  
"type": "unknown",  
"message": "please try again"  
}
```

DELETE hosts/events

Description:

Host can delete event information

Resource URL:

<https://teenytinyexplorers.com/hosts/events>

Parameters:

Name	Required	Type	Description	Example
Event ID	Required	int	The event ID is the unique identifier for the specific event	3

Scenario:

An admin needs to delete an event in the system. The admin goes to delete the event with the group ID 3. The event was successfully deleted.

An error will result if the event ID does not match.

Example Request:

```
Curl -X 'DELETE'\  
'https://teentyexplorers.com/hosts/events/3'\  
-H 'accept: application/json'\  
-H 'api_key: 3'
```

Example Response:**Successful:**

200 successful operation

Response body:

```
{  
  "code": "200",  
  "type": "unknown",  
  "message": "Group removed"  
}
```

unsuccessful:

404 Error: Response Status is 404

Response body:

```
{  
  "code": "404",  
  "type": "unknown",  
  "message": "Group not found"  
}
```

PUT hosts/events**Description:**

Host can update events information such as the state, host name, county, email, phone, and date of the event

Resource URL:

<https://teenytinyexplorers.com/hosts/events>

Parameters:

Name	Required	Type	Description	Example
Event ID	Required	int	The event ID is specific to one event	3
State	Required	string	The state is the state in which the event will be taking place this is needed for users to find events in their area	Ohio
County	Optional	string	The county is needed to narrow down search results if there are multiple events in a state happening	Adams
Email	Optional	string	The email is the email of the event host	Ncarr@maryville.com
Phone	Optional	string	The phone number is used for users to find the contact information for the event host	7345673214
Date	Optional	string	The date is used to help users identify upcoming events and can be used to search	12212021

The event date is 12212021 but needs to be updated to 11212021. The group is already stored with the below information:

```
[  
  {  
    "id": 3,  
    "state": "Ohio",  
    "hostName": "Nicole Carr",  
    "county": "Adams",  
    "email": "ncarr@maryville.com",  
    "phone": "7345673214",  
    "date": "12212021"  
  }  
]
```

]

Example Request:

```
Curl -X 'PUT'\  
'https://teenytinyexplorers.com/hosts/events/3'\  
-H 'accept: application/json'\  
-H 'Content-Type: application/json'\  
-d '{  
    "id": 3,  
    "state": "Ohio",  
    "hostName": "Nicole Carr",  
    "county": "Adams",  
    "email": "ncarr@maryville.com",  
    "phone": "7345673214",  
    "date": "11212021"  
}'
```

Example Response:

Successful:

200 successful operation

Response body:

```
{  
  "code": "200",  
  "type": "unknown",  
  "message": "Updated!"  
}
```

unsuccessful:

500 Error: Response Status is 500

```
{  
  "code": "505",
```



```
"type": "unknown",  
"message": "Please try again"  
}
```

GET hosts/events

Description:

Host can search for events in table by county and state

Resource URL:

<https://teenytinyexplorers.com/hosts/events>

Parameters:

Name	Required	Type	Description	Example
State	Optional	String	The state the event will be hosted in	Michigan
County	Optional	String	The county the event will be hosted in	Lenawee

Scenario:

A user is searching an event near them, and looks by county and state. The user finds a group near them after entering the search options.

An error will result if results do not appear after entering the fields from the user.

Example Request:

```
Curl -X 'GET'\  
'https://teenytinyexplorers.com/hosts/events/lenawee\  
-H 'accept: application/json'
```

Example Response:

Successful:

200 successful operation

Response body:

```
{  
  "code": "200",  
  "type": "unknown",  
  "message": "ok"  
}
```

Unsuccessful:

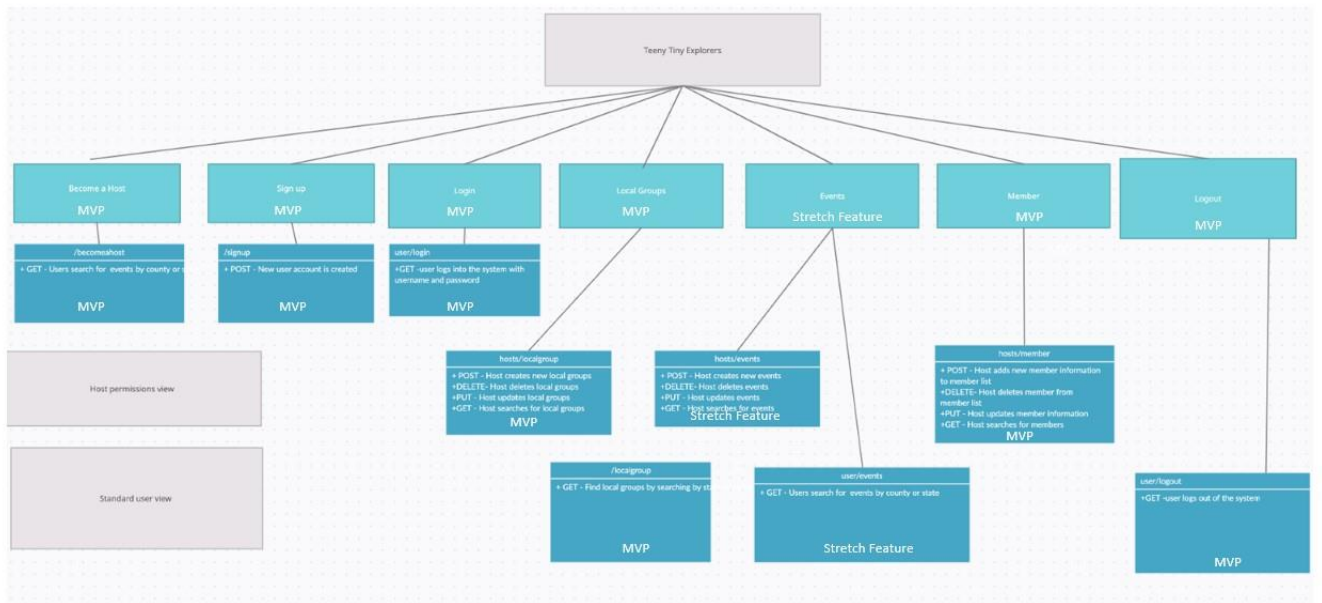
404 Error: responses status is 404

Response body:

```
{  
  "code": 1,  
  "type": "error"  
  "message": "Events not found"  
}
```

Diagram

1. Portrait View



2. Landscape View

