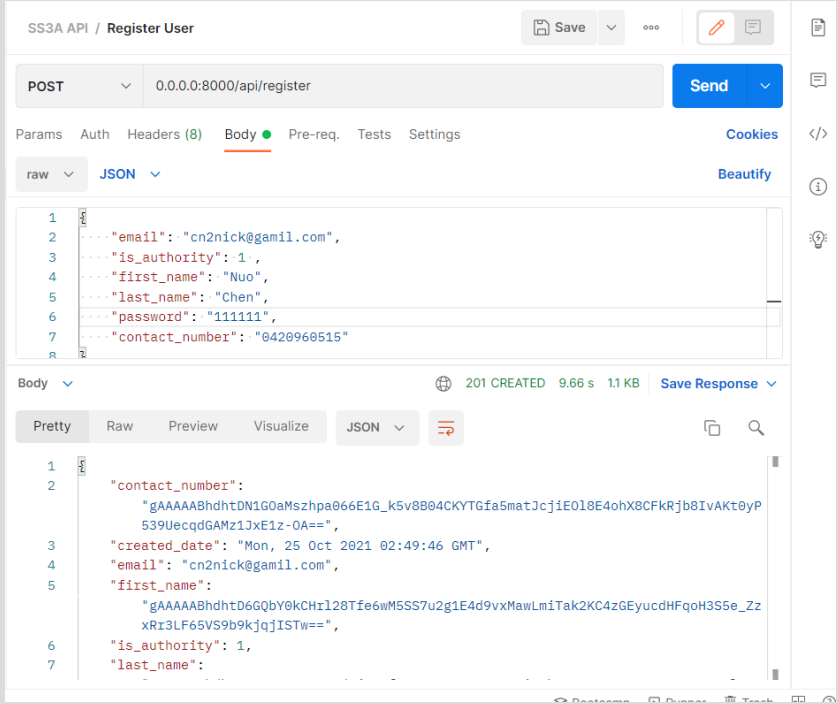


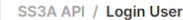
API Functions List

Last Updated: 25/10/21

Function	Purpose	Inputs	Outputs	ScreenShot Result - Postman
/register	Register new Users	User(**data) (email, first name, last name, password, contact number)	Confirm message, adds user to database	<p>Register successfully:</p>  <pre>1 { 2 "email": "cn2nick@gamil.com", 3 "is_authority": 1, 4 "first_name": "Nuo", 5 "last_name": "Chen", 6 "password": "111111", 7 "contact_number": "0420960515" 8 }</pre> <pre>1 { 2 "contact_number": 3 "gAAAAABhdhtDN1G0aMszhpa066E1G_k5v8B04CKYTGfa5matJcj1E018E4ohX8CFkRjb8IvAKt0yP 4 539UecqdGAMz1JxE1z-OA==", 5 "created_date": "Mon, 25 Oct 2021 02:49:46 GMT", 6 "email": "cn2nick@gamil.com", 7 "first_name": 8 "gAAAAABhdhtD6GQbY0kCH128Tfe6wM5SS7u2g1E4d9vxMawLmiTak2KC4zGEyucdHFqoH3S5e_Zz 9 xRi3LF65VS9b9kjIj1STw==", 10 "is_authority": 1, 11 "last_name": 12 "gAAAAABhdhtD6GQbY0kCH128Tfe6wM5SS7u2g1E4d9vxMawLmiTak2KC4zGEyucdHFqoH3S5e_Zz 13 xRi3LF65VS9b9kjIj1STw==" 14 }</pre>

Each email should be unique.


				<div> <div>SS3A API / Register User</div> <div> <div>Save</div> <div></div> <div></div> <div></div> </div> <div> <div></div> <div></div> </div> </div> <div> <div>POST</div> <div>0.0.0.0:8000/api/register</div> <div>Send</div> </div> <div> <div>Params</div> <div>Auth</div> <div>Headers (8)</div> <div>Body</div> <div>Pre-req.</div> <div>Tests</div> <div>Settings</div> <div>Cookies</div> </div> <div> <div>raw</div> <div>JSON</div> <div>Beautify</div> </div> <div> <pre> 1 { 2 "email": "cn2nick@gamil.com", 3 "is_authority": 1, 4 "first_name": "Nuo", 5 "last_name": "Chen", 6 "password": "111111", 7 "contact_number": "0420960515" </pre> </div> <div> <div>Body</div> <div>409 CONFLICT 1401 ms 493 B</div> <div>Save Response</div> </div> <div> <div>Pretty</div> <div>Raw</div> <div>Preview</div> <div>Visualize</div> <div>JSON</div> <div></div> </div> <div> <pre> 1 { 2 "message": "User with email cn2nick@gamil.com exists." 3 } </pre> </div>
/login	Login existing user	User(**data) (email, password)	user, token	Login successfully:

 Save

○○○

POST

0.0.0.0:8000/api/login

Send 

Params Auth Headers (8) **Body** ● Pre-req. Tests Settings

Cookies

raw \

JSON

Beautify

```
1 {
2   ... "email": "cn2nick@gmail.com",
3   ... "password": "111111"
4 }
```

Body 200 OK 933 ms 1.03 KB [Save Response](#)

Pretty

Raw

Preview

Visualize

JSON ▾


```
1  "token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJleHAiOiE2MzUxNjQ3Mjc5ZmlhdCI6MTYzNTE0TMyNywic3ViOiIjoiY2Y2YmlybmJja0BnbWpCb20ifQ.XaLhcw2jXjJP8pAV5YQvDnFVN220k2hq71ZrwoRS1w0",
2
3  "user": {
4    "contact_number": "0420960515",
5    "created_date": "Sat, 22 May 2021 12:13:09 GMT",
6    "email": "cn2nick@gmail.com",
7    "first_name": "Nuo",
8    "is_authority": 1,
9    "last_name": "Chen",
10   "location": "location",
11   "password":
12     "sha256$dztirt0M$A751f8429f0831b28b4d255c17d7a26acf9027b6b642e339b7056d75
13     d22005f"
```

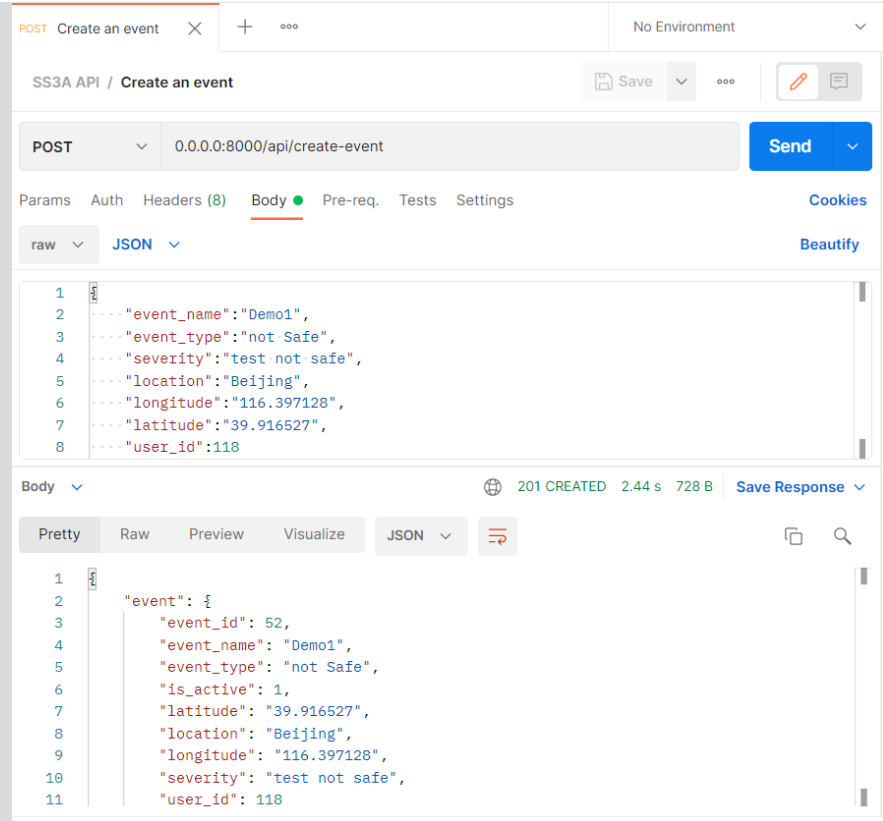
The user name and password do not match:

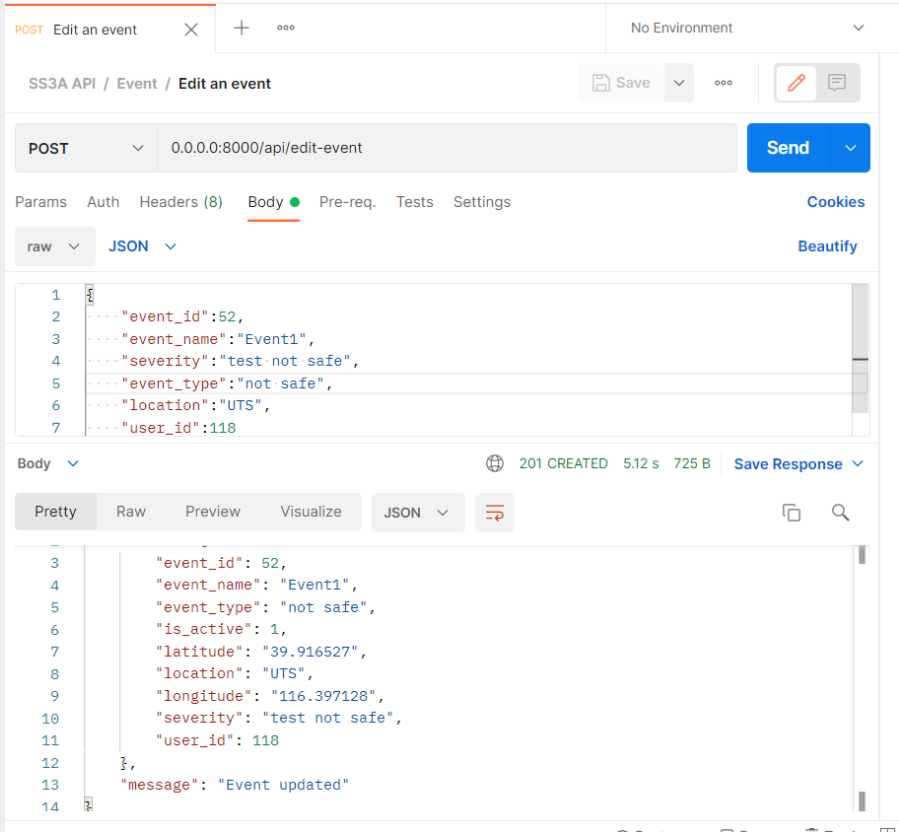
				<div><div>POST Create Chat</div><div>SS3A API / Create Chat</div><div>POST 0.0.0.0:8000/api/chat/create-chatroom</div><div>Params Auth Headers (8) Body Pre-req. Tests Settings</div><div>raw JSON</div><div><pre>1 { 2 ... "chatroom_info": { 3 ... "chatroom_name": "FRIENDS" 4 ... }, 5 ... "participants": [1,43] 6 }</pre></div><div>Body 201 CREATED 5.53 s 593 B Save Response</div><div>Pretty Raw Preview Visualize JSON</div><div><pre>1 { 2 "chatroom": { 3 "chatroom_id": 24, 4 "chatroom_name": "FRIENDS" 5 }, 6 "chatroom_participants": [7 1, 8 43 9], 10 "message": "Chatroom added" 11 }</pre></div></div>
/chat/save-message	Save a chat message	ChatMessages(**data) (chatroom_ID, user_id, message)	Add message to database in chatroom	Send successfully:

				<div><div>SS3A API / Save Chat Message</div><div><div>Save</div><div></div><div></div></div><div><div>Send</div></div><div><div>POST</div><div>0.0.0.0:8000/api/chat/save-message</div></div><div><div>Params</div><div>Auth</div><div>Headers (8)</div><div>Body</div><div>Pre-req.</div><div>Tests</div><div>Settings</div><div>Cookies</div><div>Beautify</div></div><div><div>raw</div><div>JSON</div></div><div><div>1</div><div>2</div><div>3</div><div>4</div><div>5</div></div><div><div>1</div><div>2</div><div>3</div><div>4</div><div>5</div></div><div><div>Body</div><div>201 CREATED</div><div>2.30 s</div><div>591 B</div><div>Save Response</div></div><div><div>Pretty</div><div>Raw</div><div>Preview</div><div>Visualize</div><div>JSON</div><div></div></div><div><div>1</div><div>2</div><div>3</div><div>4</div><div>5</div><div>6</div><div>7</div><div>8</div><div>9</div></div><div><div>1</div><div>2</div><div>3</div><div>4</div><div>5</div><div>6</div><div>7</div><div>8</div><div>9</div></div></div>
/chat/get-chatroom-list	Get chatrooms belonging to user	User_id	List of chatrooms user is part of	Get all the chatroom list:

				<div><div>SS3A API / Get Chat Room List</div><div><div>Save</div><div></div><div></div></div><div><div>POST</div><div>0.0.0.0:8000/api/chat/get-chatroom-list</div><div>Send</div></div><div><div>Params</div><div>Auth</div><div>Headers (8)</div><div>Body</div><div>Pre-req.</div><div>Tests</div><div>Settings</div><div>Cookies</div><div>Beautiful</div></div><div><div>raw</div><div>JSON</div></div><div><div>1</div><div>2</div><div>3</div><div>...</div><div>"user_id": 1</div></div><div><div>Body</div><div>200 OK</div><div>--</div><div>1.83 KB</div><div>Save Response</div></div><div><div>Pretty</div><div>Raw</div><div>Preview</div><div>Visualize</div><div>JSON</div></div><div><div>1</div><div>2</div><div>3</div><div>4</div><div>5</div><div>6</div><div>7</div><div>8</div><div>9</div><div>10</div><div>11</div><div>12</div><div>13</div><div>14</div><div>15</div><div>"chatroom_list": [</div><div>{</div><div>"chat_id": 1,</div><div>"chatroom_name": "Test_Room2"</div><div>},</div><div>{</div><div>"chat_id": 3,</div><div>"chatroom_name": "Jasmine Test"</div><div>},</div><div>{</div><div>"chat_id": 4,</div><div>"chatroom_name": "Jasmine Test"</div><div>},</div><div>]</div></div></div>
/chat/get-chatroom-messages	Get messages belonging to a chatroom	Chatroom_id	Return all messages in a chatroom	Get chatroom message:

				<div><div>POST Get Chat Room ...</div><div>SS3A API / Get Chat Room Messages</div><div>POST 0.0.0.0:8000/api/chat/get-chatroom-messages</div><div>Params Auth Headers (8) Body Pre-req. Tests Settings</div><div>raw JSON</div><div><pre>1 { 2 "chatroom_id": 24 3 }</pre></div><div>Body 200 OK 1402 ms 663 B Save Response</div><div>Pretty Raw Preview Visualize JSON</div><div><pre>1 { 2 "chatroom_messages": [3 { 4 "first_name": "Joe", 5 "last_name": "Blogs", 6 "message": "Hello! FRIENDS!", 7 "message_id": 46, 8 "timestamp": "Mon, 25 Oct 2021 03:04:30 GMT", 9 "user_id": 1 10 } 11] 12 }</pre></div></div>
/create-event	Register a new event	Event(**data) (event name, severity, event type, location, longitude, latitude, user id)	Saves event to database	Create successfully:

				 <p>The screenshot shows a REST client interface with a tab titled 'POST Create an event'. The URL is '0.0.0.0:8000/api/create-event'. The request body is a JSON object: <pre>{ "event_name": "Demo1", "event_type": "not Safe", "severity": "test not safe", "location": "Beijing", "longitude": "116.397128", "latitude": "39.916527", "user_id": 118 }</pre>. The response status is '201 CREATED' with a time of '2.44 s' and size of '728 B'. The response body is a JSON object: <pre>{ "event": { "event_id": 52, "event_name": "Demo1", "event_type": "not Safe", "is_active": 1, "latitude": "39.916527", "location": "Beijing", "longitude": "116.397128", "severity": "test not safe", "user_id": 118 } }</pre></p>
/edit-event	Edit existing event	Event_id, new (event_name, severity, event_type, location, user_id)	Updates event with new data	Edit successfully:

				
/delete-event	Delete the event	Event_id	Deletes the event from the database	Delete successfully:

POSTDelete an event

No Environment

SS3A API / Event / Delete an event

POST

0.0.0.0:8000/api/delete-event

Save

Send

ParamsAuthHeaders (8)BodyPre-req. Tests Settings

rawJSON

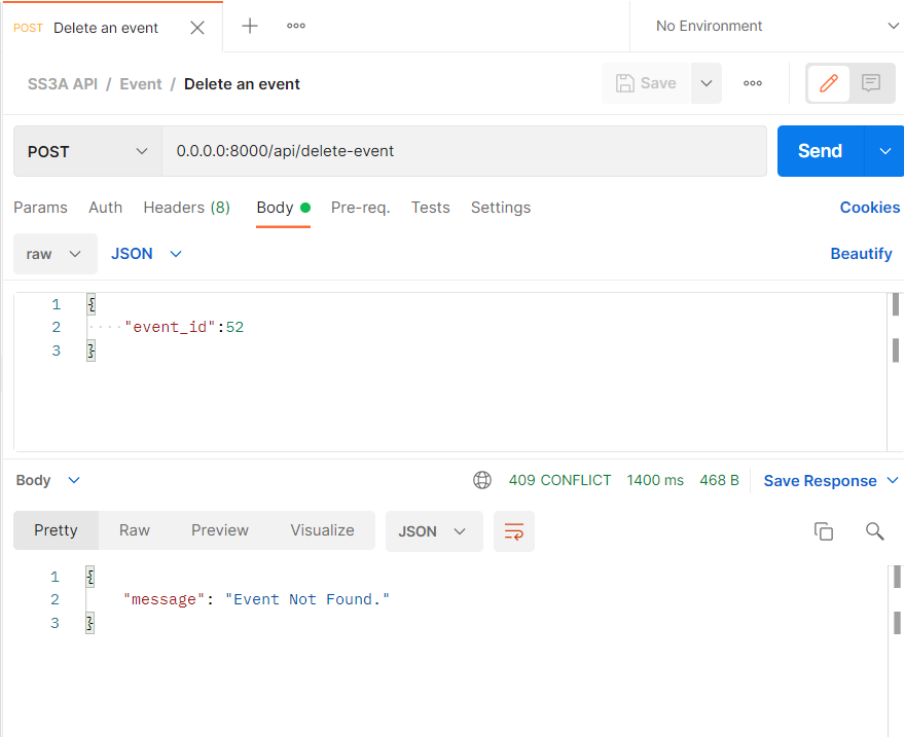
```
1 {
2   ... "event_id": 52
3 }
```

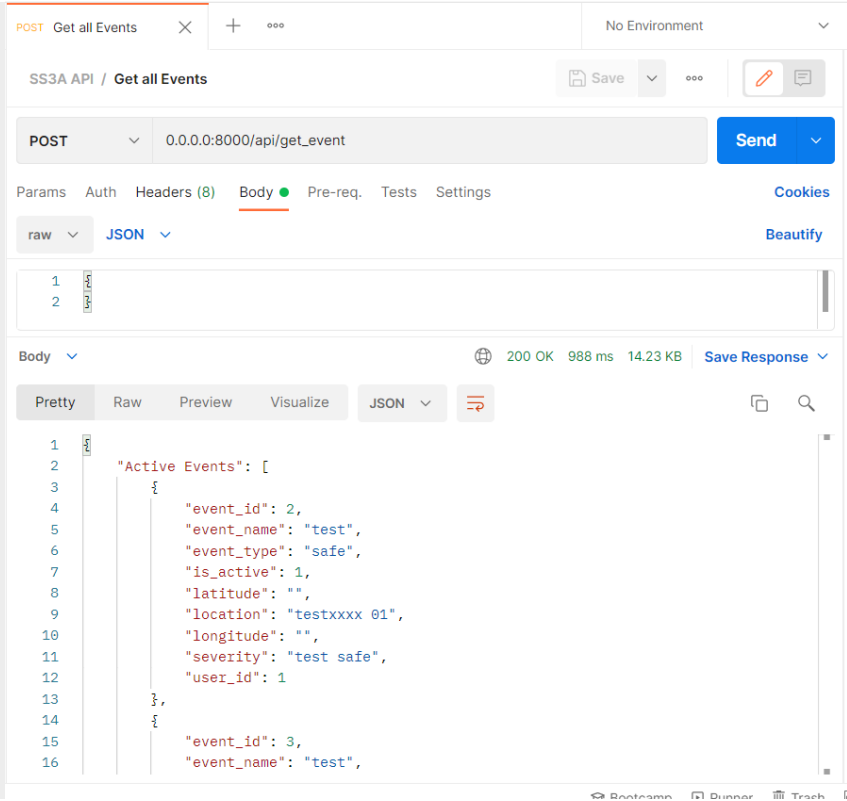
Body201 CREATED1880 ms463 BSave Response

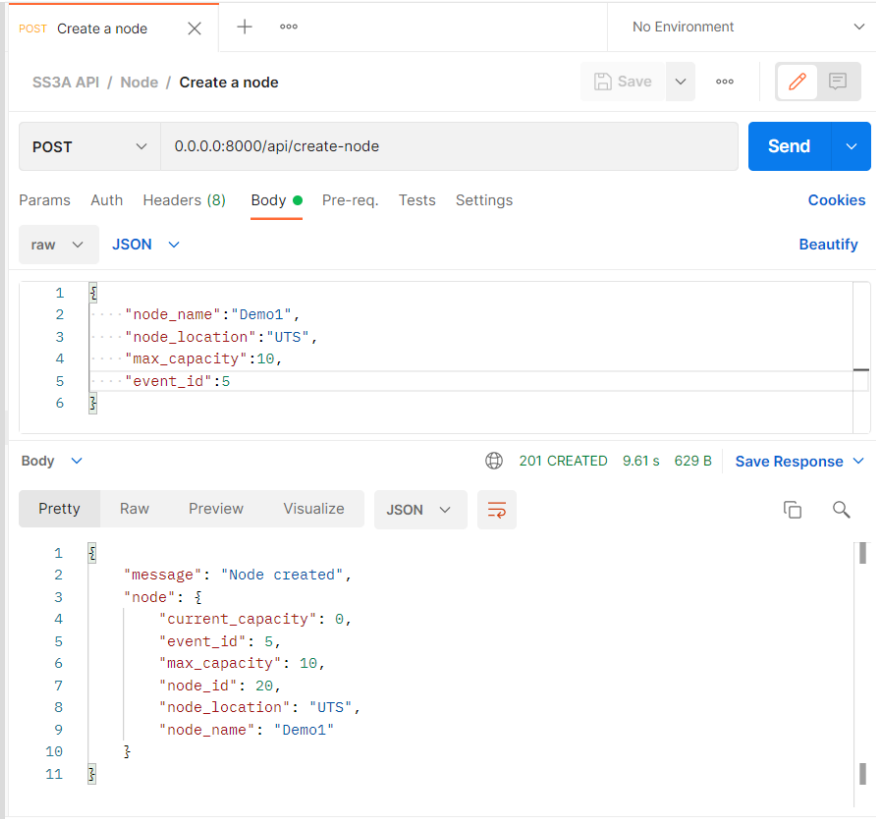
PrettyRawPreviewVisualizeJSON

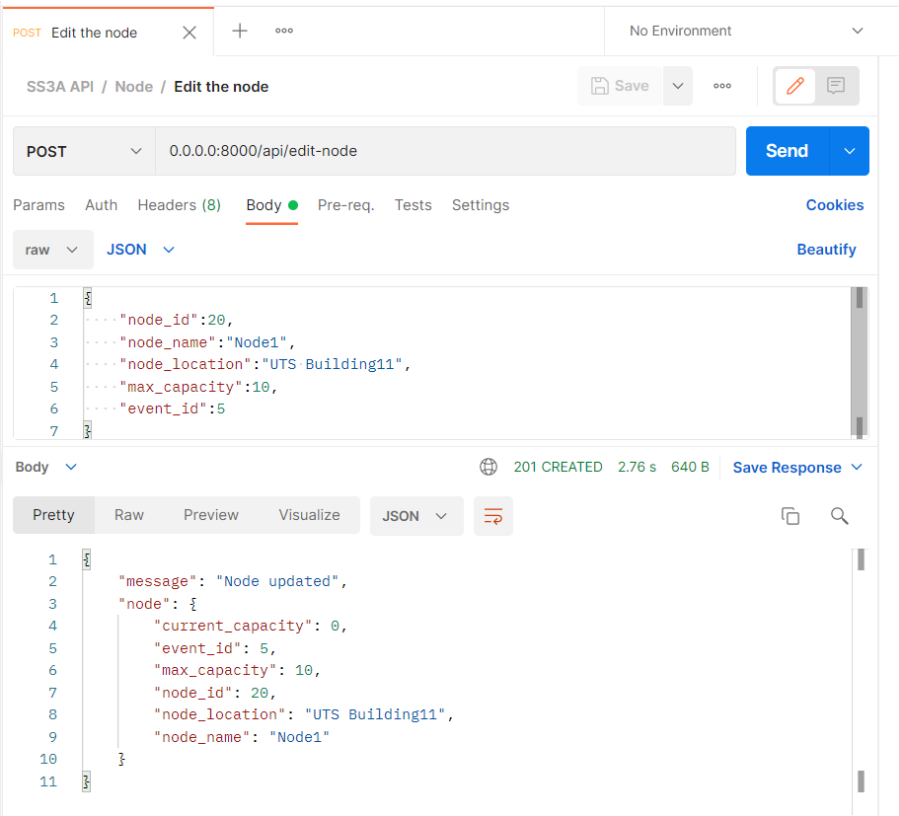
```
1 {
2   "message": "Event delete"
3 }
```

Error: Event not found

				
/get-event	Returns list of all active events	-	Returns all events where is_active attribute = 1	Get all the event list

				
/create-node	Register a new node	Node(**data) (node name, node location, max capacity, event id)	Adds new node to the database	Create successfully:

				 <p>POST Create a node</p> <p>SS3A API / Node / Create a node</p> <p>POST 0.0.0.0:8000/api/create-node</p> <p>Params Auth Headers (8) Body Pre-req. Tests Settings</p> <p>raw JSON</p> <pre> 1 { 2 "node_name": "Demo1", 3 "node_location": "UTS", 4 "max_capacity": 10, 5 "event_id": 5 6 }</pre> <p>Body 201 CREATED 9.61 s 629 B Save Response</p> <p>Pretty Raw Preview Visualize JSON</p> <pre> 1 { 2 "message": "Node created", 3 "node": { 4 "current_capacity": 0, 5 "event_id": 5, 6 "max_capacity": 10, 7 "node_id": 20, 8 "node_location": "UTS", 9 "node_name": "Demo1" 10 } 11 }</pre>
/edit-node	Edit an existing node	Node_id, new (node_name, node_location, max_capacity, event_id)	Updates attributes of node and resets the current_capacity to 0	Update successfully:

				
/delete-node	Delete an existing node	Node_id	Deletes the node from the database	Delete successfully:

POST Delete the node

SS3A API / Node / Delete the node

POST 0.0.0.0:8000/api/delete-node

Send

Params Auth Headers (8) Body Pre-req. Tests Settings Cookies Beautify

raw JSON

```
1 {
2   ... "node_id": 20
3 }
```

Body 201 CREATED 1838 ms 462 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "message": "Node delete"
3 }
```


/get_node

Returns a list
of all existing
nodes

-

Returns all
nodes as a list

POST Get the node list × + ... No Environment

SS3A API / Node / Get the node list Save ...

POST 0.0.0.0:8000/api/get_node Send

Params Auth Headers (8) Body Pre-req. Tests Settings Cookies

raw JSON Beautify

```
1 [
2   ...
3 ]
```

Body 200 OK 926 ms 1.15 KB Save Response

Pretty Raw Preview Visualize JSON

```
1 [
2   "Existing nodes": [
3     {
4       "current_capacity": 0,
5       "event_id": 2,
6       "max_capacity": 50,
7       "node_id": 10,
8       "node_location": "test not safe",
9       "node_name": "test001"
10    },
11    {
12       "current_capacity": 0,
13       "event_id": 4,
14       "max_capacity": 1,
15       "node_id": 13,
16       "node_location": "test not safe",
17       "node_name": "test002"
18    }
19  ]
20 }
```

/add-contacts
(Not use)

--
Takes string
with list of
contact
numbers like
"0437263746,
4837473272",
finds if user
exisits with
the phone
number
and adds
them as a
contact

--
User_id,
Phone_numbers

--

The screenshot displays a REST client interface for a POST request. The top bar shows the endpoint "POST Get Contacts List" and the environment "No Environment". The request is configured with the method "POST" and the URL "0.0.0.0:8000/api/add-contacts". The request body is in JSON format, containing the following data:

```
{  "user_id": "34",  "phone_numbers": "0437263746, 0420960515"}
```

The response status is "201 CREATED" with a response time of "470 ms" and a size of "507 B". The response body is displayed in a "Pretty" format, showing a message:

```
{ "message": " 2 contacts passed.0 added, 0 duplicates and 2 not found" }
```

/get-contacts

Returns all
contacts
associated
with the
user_id

User_id

Returns users
ContactList

POST Get Friends List × + ... No Environment

SS3A API / friendRequest / Get Friends List Save ...

POST 0.0.0.0:8000/api/get-contacts Send

Params Auth Headers (9) Body Pre-req. Tests Settings Cookies

raw JSON Beautify

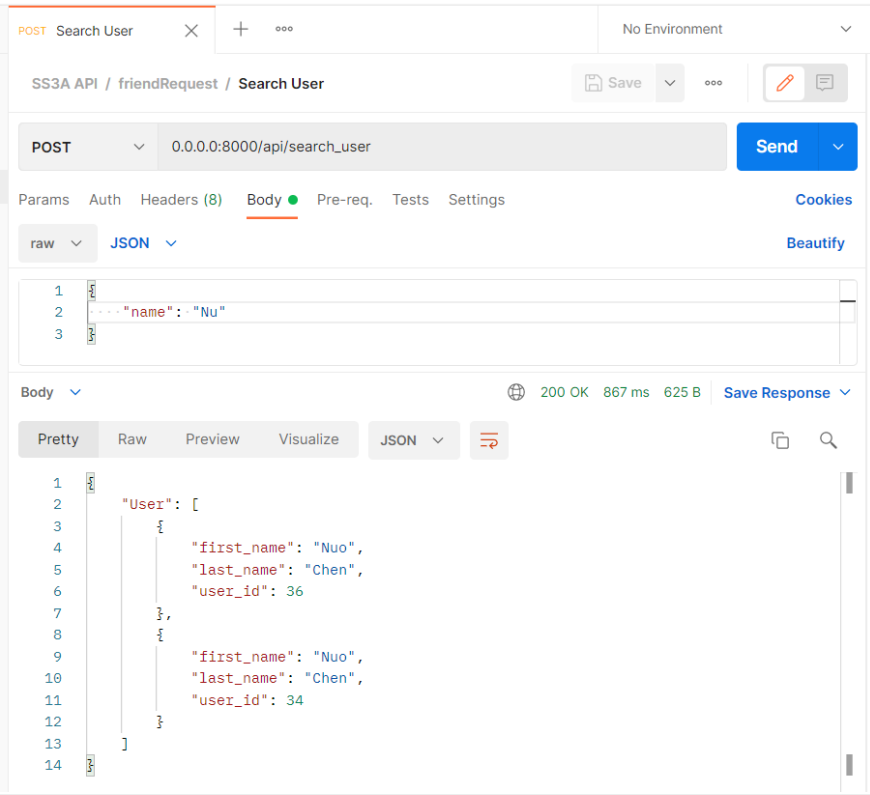
```
1
2
3
```

Body 200 OK 3.08 s 1023 B Save Response

Pretty Raw Preview Visualize JSON

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
```

```
"contact_list": [
  {
    "contact_number": "987654321",
    "email": "cn2@cn.com",
    "first_name": "Test2",
    "is_authority": 1,
    "last_name": "Nuo",
    "latitude": "",
    "location": "location",
    "longitude": "",
    "status": "null",
    "user_id": 77
  },
  {
    "contact_number": "444"
```

/search-user	Search specific user via id	User_name Can be Fuzzy search	User information	 <p>The screenshot shows a REST client interface for a POST request to the endpoint <code>0.0.0.0:8000/api/search_user</code>. The request body is a JSON object: <code>{ "name": "Nu" }</code>. The response is a 200 OK status with a response time of 867 ms and a body size of 625 B. The response body is a JSON array of two user objects:</p> <pre> { "User": [{ "first_name": "Nuo", "last_name": "Chen", "user_id": 36 }, { "first_name": "Nuo", "last_name": "Chen", "user_id": 34 }] } </pre>
/add_friends	Send friend request to user, adds request to the send/receive lists	User_id, request user_id, content	-	Send successfully

POST Add Friends

SS3A API / friendRequest / Add Friends

POST 0.0.0.0:8000/api/add_friends

Send

Params Auth Headers (9) Body Pre-req. Tests Settings

raw JSON

1
2
3
4
5

Body 201 CREATED 2.62 s 477 B Save Response

Pretty Raw Preview Visualize JSON

1
2
3

/get-friends

Returns all
friends
associated
with the user

User_id

Returns contacts
list of user

POST Get Friends List

SS3A API / friendRequest / Get Friends List

POST 0.0.0.0:8000/api/get_friends

Params Auth Headers (9) Body Pre-req. Tests Settings

raw JSON

```
1 {
2   "user_id": "76"
3 }
```

Body 200 OK 1723 ms 799 B Save Response

Pretty Raw Preview Visualize JSON

```
2   "contact_list": [
3     {
4       "contact_number": "111",
5       "email": "123456.com",
6       "first_name": "Test",
7       "id": 28,
8       "last_name": "Test",
9       "user_id": 35
10    },
11    {
12      "contact_number": "987654321",
13      "email": "cn2@cn.com",
14      "first_name": "Test2",
15      "id": 22,
16      "last_name": "Nuo",
17      "user_id": 77
18    }
19  ]
```

/get_receive_request

Returns list of
received
friend
requests

User_id

Returns pending
friend requests
received by the
user
Status: Applying,
Success, Fail

POST Get Received Re... x + ... No Environment

SS3A API / friendRequest / Get Received Request List Save ...

POST 0.0.0.0:8000/api/get_receive_request Send

Params Auth Headers (9) Body Pre-req. Tests Settings Cookies

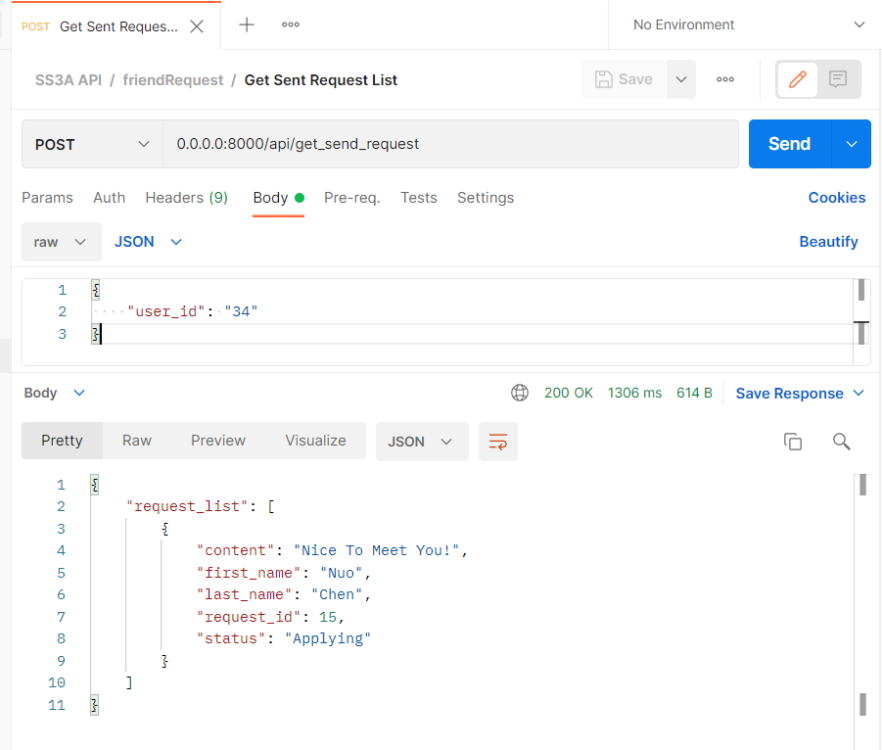
raw JSON Beautify

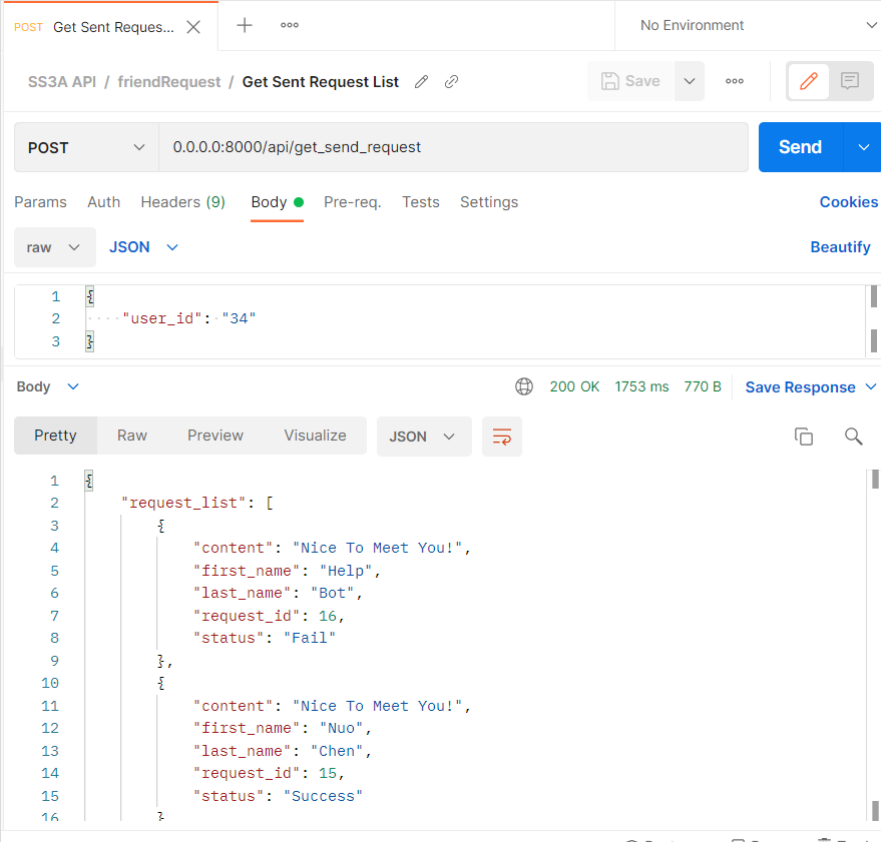
```
1 {
2   ... "user_id": "36"
3 }
```

Body 200 OK 1287 ms 614 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "request_list": [
3     {
4       "content": "Nice To Meet You!",
5       "first_name": "Nuo",
6       "last_name": "Chen",
7       "request_id": 15,
8       "status": "Applying"
9     }
10  ]
11 }
```

/get_send_request	Returns list of sent friend requests	User_id	Returns pending friend requests sent by the user Status: Applying, Success, Fail	
-------------------	--------------------------------------	---------	---	---

				
/approve_request	Accept/refuse the friend request	Request list id, Status: success/fail, reason	If success, adds user to the contacts. If fail, rejects request	Success: Accept the request

The screenshot shows the Swagger UI interface for a REST API. At the top, there's a header with "POST Approve Request" and a "No Environment" dropdown. Below this, the API path is "SS3A API / friendRequest / Approve Request". There are "Save" and "Send" buttons. The "Send" button is highlighted in blue. Below the path, there's a dropdown for "POST" and a text input for "0.0.0.0:8000/api/approve_request". The "Send" button is also present here. Below this, there are tabs for "Params", "Auth", "Headers (9)", "Body", "Pre-req", "Tests", and "Settings". The "Body" tab is selected and highlighted with an orange underline. Below the tabs, there's a "raw" dropdown and a "JSON" dropdown. The "JSON" dropdown is selected. The main area shows the request body in JSON format:

```
1 {
2   "request_list_id": "15",
3   "status": "Success",
4   "reason": "Cheers!"
5 }
```

 Below this, there's a "Body" dropdown and a "Save Response" button. The "Body" dropdown is selected. The main area shows the response body in JSON format:

```
1 {
2   "message": "Request approved Successfully!"
3 }
```

Success: Reject the request

POST Approve Request

SS3A API / friendRequest / Approve Request

POST 0.0.0.0:8000/api/approve_request

Send

Params Auth Headers (9) Body Pre-req. Tests Settings

raw JSON

```
1 {
2   "request_list_id": "16",
3   "status": "Fail",
4   "reason": "Sorry!"
5 }
```

Body 201 CREATED 1737 ms 481 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "message": "Request approved Successfully!"
3 }
```

/get_type_event

Return a list
of all active
events by
type

Event_type

Return a list of
all active events

POST Get events by type × + ... No Environment

SS3A API / Event / Get events by type Save ...

POST 0.0.0.0:8000/api/get_type_event Send

Params Auth Headers (8) Body Pre-req. Tests Settings Cookies

raw JSON Beautify

```
1 [
2   ... "event_type": "not Safe"
3 ]
```

Body 200 OK 952 ms 1.21 KB Save Response

Pretty Raw Preview Visualize JSON

```
1 "Active Events": [
2   {
3     "event_id": 5,
4     "event_name": "test101",
5     "event_type": "not safe",
6     "is_active": 1,
7     "latitude": "",
8     "location": "test01",
9     "longitude": "",
10    "severity": "test not safe",
11    "user_id": 1
12  }
```

/display_user_location

Return a list of all the user location details

Return a list of all the user location details

The screenshot shows a REST client interface with the following details:

- Method:** POST
- URL:** 0.0.0.0:8000/api/display_user_location
- Response Status:** 200 OK
- Response Time:** 8.37 s
- Response Size:** 15.39 KB
- Response Body (JSON):**

```
{
  "user location list": [
    {
      "first_name": "Joe",
      "last_name": "Blogs",
      "latitude": "",
      "location": "location",
      "longitude": "",
      "user_id": 1
    },
    {
      "first_name": "Joe",
      "last_name": "Blogs",
      "latitude": "",
      "location": "location",
      "longitude": "",
      "user_id": 1
    }
  ]
}
```

/check_danger

If user location enters into event radius, alert user

Longitude, latitude

Return true/false and warning message

In danger

POST Check danger

SS3A API / Check danger

POST 0.0.0.0:8000/api/check_danger

Params Auth Headers (8) Body Pre-req. Tests Settings

raw JSON

```
1  {
2    "longitude": "116.397128",
3    "latitude": "39.916527"
4  }
```

Body 200 OK 2.80 s 508 B Save Response

Pretty Raw Preview Visualize

{ "data": true, "message": "Warning! You're in a danger place, leave now!" }

Safe:

				<div><div>POST Check danger</div><div>SS3A API / Check danger</div><div>POST 0.0.0.0:8000/api/check_danger</div><div>Params Auth Headers (8) Body Pre-req. Tests Settings</div><div>raw JSON</div><div><div>1</div><div>2</div><div>3</div><div>4</div><div>....."longitude": "10.397128",</div><div>....."latitude": "39.916527"</div></div><div>Body</div><div>200 OK 2.81 s 464 B</div><div>Pretty Raw Preview Visualize</div><div>{ "data": false, "message": "" }</div></div>