Introduction
POST Signing into Sentry App
GET Authenticating
GET Getting Database Content
POST Updating Geolocation

Environment No Environment Layout Double Column Language cURL - cURL

Sentry Database and API

Comments 0

SENTRY DATABASE

We used MySQL to create our database.The database holds the information regarding sentinel users which includes users unique RFID number, Username, Password, Name, Geolocation, Heartrate and Authentication info. After creating our database in MySQL we hosted our database in AWS cloud. In order to connect our database with other subsystems we have created created Restful APIs.

The image of the database can be found in the following link:
https://github.com/SentrySentinel/sentrydb/blob/master/sentry-api/db_image.png

SENTRY RESTFUL API'S

Sentry Restful APIs are the APIs which are used for the secured transfer of informations between Sentry subsystems.

Mainly there are four functions for which we have created Restful APIs and they are as follows.

1. Signing Into Sentry App

2. Authenticating

3. Getting Database Content

4. Updating Geolocation

POST Signing into Sentry App

Comments 0

https://kix7tx694g.execute-api.us-east-1.amazonaws.com/dev/signIn/2494100525

Use (https://kix7tx694g.execute-api.us-east-1.amazonaws.com/dev/signIn/1001) to run this `POST` request.

This request allows Sentry user with RFID number 1001 to login into the Sentry App. To change the RFID number replace 1001 in the URL with the required RFID number in the database.

Headers

Content-Type	application/json

Body raw (application/json)

```
{
        "username":"Dilan",
        "password":"Dilan1"
}
```

Example Request — Sign into Sentry App

```
curl --location --request POST 'https://kix7tx694g.execute-api.us-east-1.amazonaws.com/dev/signIn/1001' \
--header 'Content-Type: application/json' \
--data-raw '{
        "username":"Dilan",
        "password":"Dilan1"
}'
```

Example Response — 200 OK

```
{
  "status": "login successful"
}
```

GET Authenticating

Comments 0

https://kix7tx694g.execute-api.us-east-1.amazonaws.com/dev/authenticate/1001

Use (https://kix7tx694g.execute-api.us-east-1.amazonaws.com/dev/authenticate/1001) to run this `GET` request.

Authentication helps to determine if user should be granted access or not to the secured area. 1001 at the end of URL is the primary key to the database; this can be replaced with any other valid RFID.

Example Request — Authentication

```
curl --location --request GET 'https://kix7tx694g.execute-api.us-east-1.amazonaws.com/dev/authenticate/1001'
```

Example Response — 200 OK

```
{
  "status": false
}
```

GET Getting Database Content

Comments 0

https://kix7tx694g.execute-api.us-east-1.amazonaws.com/dev/get_sentry_api

Use (https://kix7tx694g.execute-api.us-east-1.amazonaws.com/dev/get_sentry_api) to run this `GET` request.

Getting Database content function helps to get the content of the Sentry database.

Headers

Content-Type	application/json

Example Request — Getting Database content function

```
curl --location --request GET 'https://kix7tx694g.execute-api.us-east-1.amazonaws.com/dev/get_sentry_api' \
--header 'Content-Type: application/json'
```

Example Response — 200 OK

```
[
  {
    "RFID": 1001,
    "Username": "Dilan",
    "Password": "Dilan1",
    "Name": "Dilan",
    "Geolocation": "Coppell",
    "Heartrate": "1",
    "Authentication": "0"
  }
```

POST Updating Geolocation

Comments 0

https://kix7tx694g.execute-api.us-east-1.amazonaws.com/dev/updateGeolocation/1001

Use (https://kix7tx694g.execute-api.us-east-1.amazonaws.com/dev/updateGeolocation/1001) to run this `POST` request.

Updating geolocation function helps to update the geolocation of the Sentry's users. 1001 at the end of URL is the primary key to the database; this can be replaced with any other valid RFID.

Headers

Content-Type	application/json

Body raw (application/json)

```
{
    "geolocation": "30.28892, -97.73634"
}
```

Example Request — Updating geolcation function

```
curl --location --request POST 'https://kix7tx694g.execute-api.us-east-1.amazonaws.com/dev/updateGeolocation/1001' \
--header 'Content-Type: application/json' \
--data-raw '{
    "geolocation": "30.28892, -97.73634"
}'
```

Example Response — 200 OK

```
{
  "status": "geolocation updated"
}
```