**Policy Server**

### Policy Server(with Documentation) : [Link](https://secret-scrubland-31951.herokuapp.com/)

### Policy Server Rules:

1. **Limiting the number of devices of Non-Helper Node Owner Devices:**

This policy restricts the connection of the Non-Helper node Owner’s devices.

1. **Limiting the Bandwidth of Sharing:**

This policy basically sets the configuration for a helper node on how much it can share data and at what bandwidth.

1. **Priority Criteria:**

This policy ensures that the other two policies [1] and [2] apply only to non-helper node owner devices.

### Block Diagram:



### 

### Implementation:

Framework Used: Express/NodeJS

Database Used : MongoDB

Development Environment: Docker Compose

### DB Design:

#### Collections

* *User {*

*\_id:Object\_ID*

// \_id refers to the MongoDB ID

*helper\_user\_id:String,*

// helper\_user\_id refers to the unique ID common for all services.

*max\_limit\_data:Number,*

// max\_limit\_data refers to maximum data that non-helper node owner devices can .utilize via helper nodes

*data\_used:Number,*

// helper\_user\_id refers to the data\_usage

*allowed\_sharing:Boolean,*

*last\_renewed:String,*

// last\_renewed takes care on renewal plan of data\_usage

*renew\_span: String,*

*bandwidth\_set:Number,*

*connected\_devices:Number,*

*max\_limit\_connected\_devices:Number,*

*helperNodes:[*

*\_id:ObjectID*

*]*

// helperNodes stores all helper nodes of a helper user.

*created\_At:String,*

*updated\_At:String*

*}*

* *Helper\_Node {*

*\_id:Object\_ID,*

*helper\_node\_id:String,*

*user\_id:Object\_ID,*

*created\_At: String,*

*updated\_At:String*

*}*

* *Connection\_Tracker {*

*\_id:Object\_ID,*

*helper\_user\_id:String,*

*helper\_node\_id:String,*

*device\_serial\_number:String,.....device serial number*

*connected\_at:String,*

*disconnected\_at:String,*

*data\_used: Number,*

// data\_used refers to data used in a connected state.

*created\_At: String,*

*updated\_At:String*

*}*

### API Endpoints:

* For Policy Server:

1. **/api/v1/connection/connect/:**

Description: This endpoint allows devices to connect to helper node for sharing based on policy server restrictions.

Method: POST

Request Body: {

device\_serial\_number:String,

helper\_node\_id:String

}

Response: i or ii

(i) {

valid:false

}

(ii) {

valid:true,

data:{

bandwidthSet:Number,

}

}

1. **/api/v1/connection/disconnect/**

Description: This endpoint allows devices to disconnect from helper node owner.

Method: POST

Request Body: {

device\_serial\_number:String,

helper\_node\_id:String,

data\_used:Number

}

Response:i or ii

(i) {

valid:false

}

(ii) {

valid:true,

data: “Removed a Connection”

}

1. **/api/v1/user/ :**

Description: This endpoint allows to get all users(with query parameters)

Method: GET

Response : {

data: [

{

// User Fields as per Schema

}

],

count: Number

}

1. **/api/v1/user/ :**

Description: This endpoint allows creating a user.

Method: POST

Request Body: {

helper\_user\_id:String,

// other optional fields

}

Response: {

data: {

// User Fields as per Schema

}

}

1. **/api/v1/user/:helper\_user\_id :**

Description: This endpoint allows to get a user.

Method: GET

Response : {

data: {

//User Object

}

}

1. **/api/v1/user/:helper\_user\_id :**

Description: This endpoint allows to update a user.

Method: PATCH

Request Body: {

bandwidth\_set:Number,

max\_limit\_connected\_devices:Number,

renew\_span:String,

allowed\_sharing:Boolean,

max\_limit\_data:Number

}

Response : {

data: {

//User Object

}

}

1. **/api/v1/user/:helper\_user\_id :**

Description: This endpoint allows to delete a user.

Method: DELETE

Response : {

data: “Deleted a Helper User”

}

1. **/api/v1/helperNode/ :**

Description: This endpoint allows to create a helper node.

Method: POST

Request Body: {

helper\_node\_id:String,

helper\_user\_id: String

}

Response : {

data: {

// HelperNode Fields as per Schema

}

}

1. **/api/v1/helperNode/ :**

Description: This endpoint allows to get all helper nodes(with query parameters)

Method: GET

Response : {

data: [

{

// HelperNode Fields as per Schema

}

],

Count: Number

}

1. **/api/v1/helperNode/:helper\_node\_id :**

Description: This endpoint allows to get a helper node (populated data)

Method: GET

Response : {

data: {

// HelperNode Fields as per Schema

}

}

1. **/api/v1/helperNode/:helper\_node\_id :**

Description: This endpoint allows to update a helper node.

Method: PATCH

Request Body: {

status:Boolean,

}

Response: {

data: {

// HelperNode Fields as per Schema

}

}

1. **api/v1/helperNode/:helper\_node\_id :**

Description: This endpoint allows to delete a helper node.

Method: DELETE

Response : {

data: “Deleted a Helper Node”

}

1. **api/v1/tracker/:helper\_user\_id :**

Description: This endpoint allows to get connections for a helper\_user\_id(with query parameters)

Method: GET

Response : {

data: [

{

// ConnectionTracker Fields as per Schema

}

],

count:Number

}

### 

|  |  |  |
| --- | --- | --- |
| **Tasks** | **Assigned** | **Status** |
| API Application Setup | Keshav | **Done** |
| Writing Schemas | Keshav | **Done** |
| Get All Users | Ayushee | **Done** |
| Create User | Keshav | **Done** |
| Get User | Ayushee | **Done** |
| Update User | Keshav | **Done** |
| Delete User | Keshav | **Done** |
| Get All Helper Nodes | Ayushee | **Done** |
| Get Helper Node | Ayushee | **Done** |
| Create Helper Node | Ayushee | **Done** |
| Update Helper Node | Ayushee | **Done** |
| Delete Helper Node | Keshav | **Done** |
| Connect | Keshav | **Done** |
| Disconnect | Keshav | **Done** |
| Get Tracker | Ayushee | **Done** |
| API Documentation | Ayushee | **Done** |
| CLI Application Setup | Keshav | **Done** |
| Integrating API and Adding Methods | Keshav | **Done** |
| Improving CLI UI | Keshav | **Done** |