API Documentation

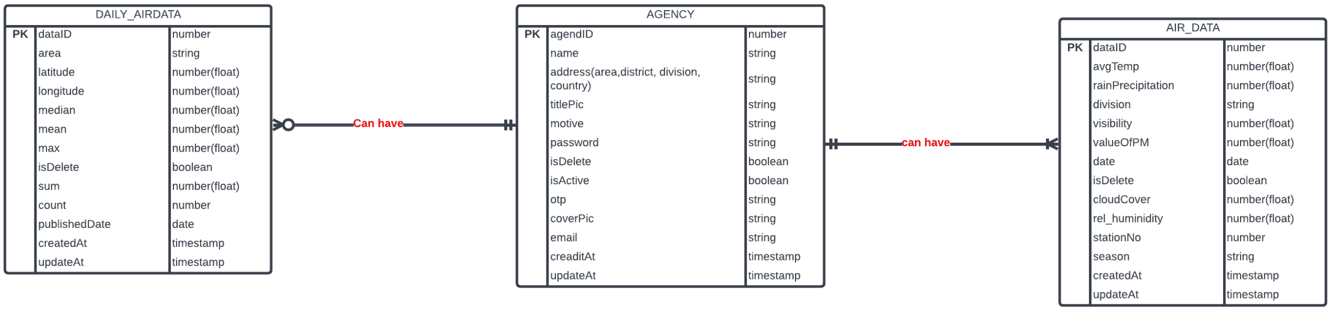
Technology Stack (Server Side)

Objective	Technology
Runtime	Node js
Server-side	Express js
Data Base	MySQL (with ORM TypeORM)
API Type	REST Full
Authentication	JWT
Email Sender	Node mailer

TABLE OF CONTENTS

SL	Topic Name	No of API	Page Number
01	ERD	0	4
02	Table Diagram	0	5
03	Dashboard	10	6-33
04	Global	7	34 – 47
05	Fetcher	14	48 – 95

ERD Diagram

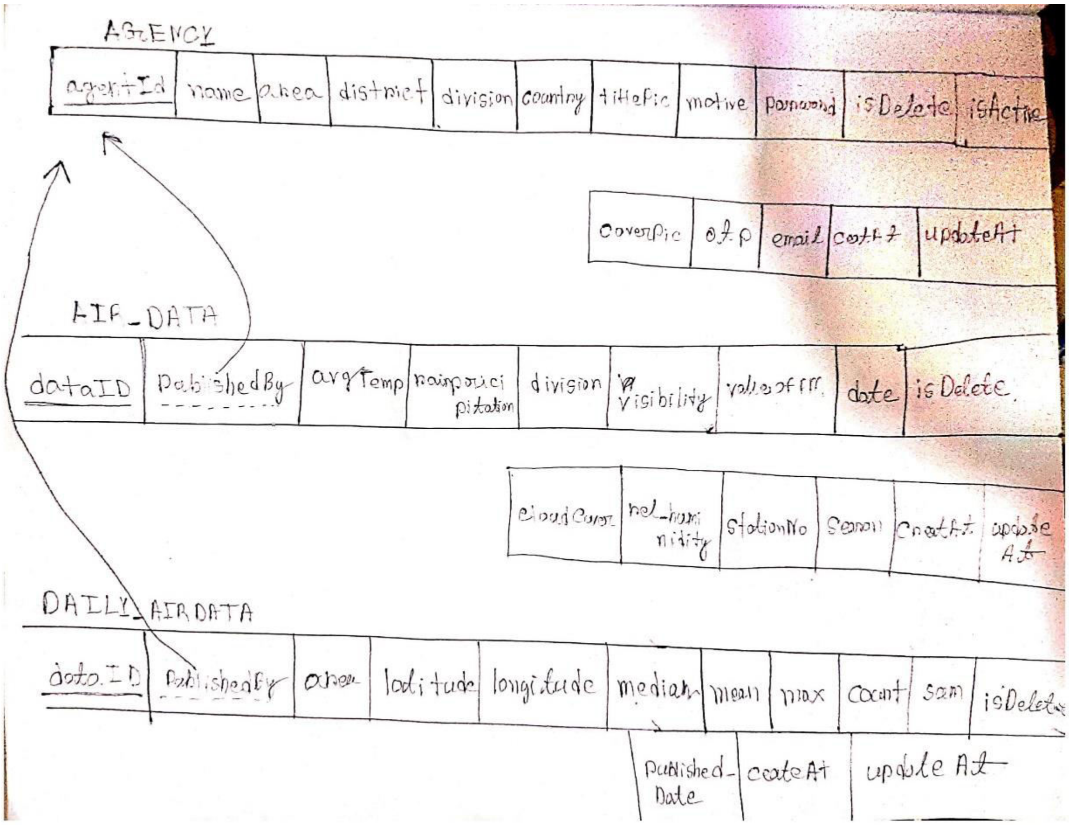


DAILY_AIRDATA		
PK	dataID	number
	area	string
	latitude	number(float)
	longitude	number(float)
	median	number(float)
	mean	number(float)
	max	number(float)
	isDelete	boolean
	sum	number(float)
	count	number
	publishedDate	date
	createdAt	timestamp
	updateAt	timestamp

AGENCY		
PK	agendID	number
	name	string
	address(area,district, division, country)	string
	titlePic	string
	motive	string
	password	string
	isDelete	boolean
	isActive	boolean
	otp	string
	coverPic	string
	email	string
	creaditAt	timestamp
	updateAt	timestamp

AIR_DATA		
PK	dataID	number
	avgTemp	number(float)
	rainPrecipitation	number(float)
	division	string
	visibility	number(float)
	valueOfPM	number(float)
	date	date
	isDelete	boolean
	cloudCover	number(float)
	rel_huminidity	number(float)
	stationNo	number
	season	string
	createdAt	timestamp
	updateAt	timestamp

Table Diagram



AGENCY

agentId	name	area	district	division	country	titlePic	motive	password	isDelete	isActive

coverPic	otp	email	creatAt	updateAt

AIR_DATA

dataID	PublishedBy	avgTemp	rainprecipitation	division	visibility	valueofPM	date	isDelete

cloudCover	rel_huminidity	stationNo	Season	CreatAt	update At

DAILY_AIRDATA

data.ID	PublishedBy	area	laditude	longitude	median	mean	max	count	sam	isDelete

Published-Date	ceateAt	update At

Dashboard API'S

1. Update logged in agency profile

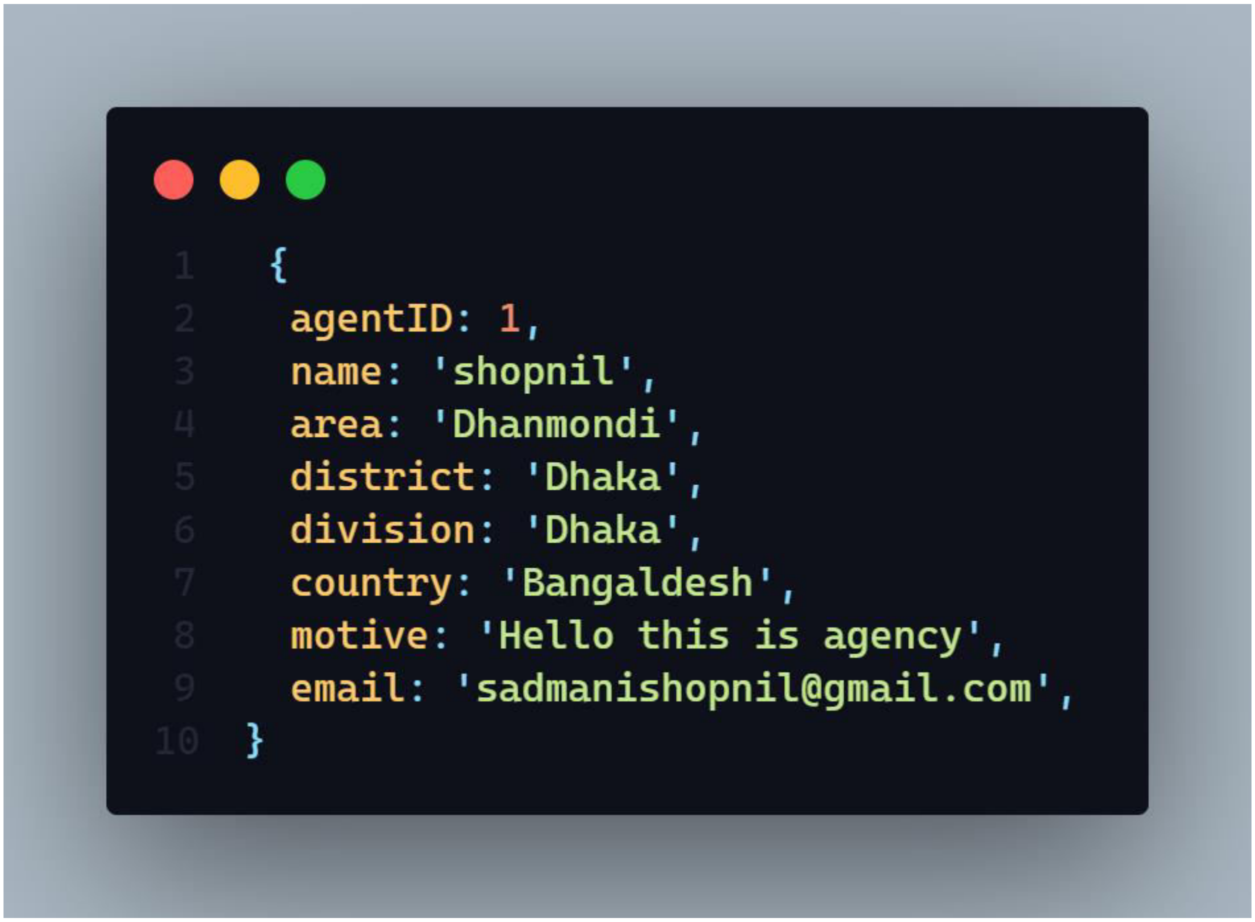
APl: baseUrl/ agency /profile/update

Type: REST

Method : PUT

Access: **: All**

Body :



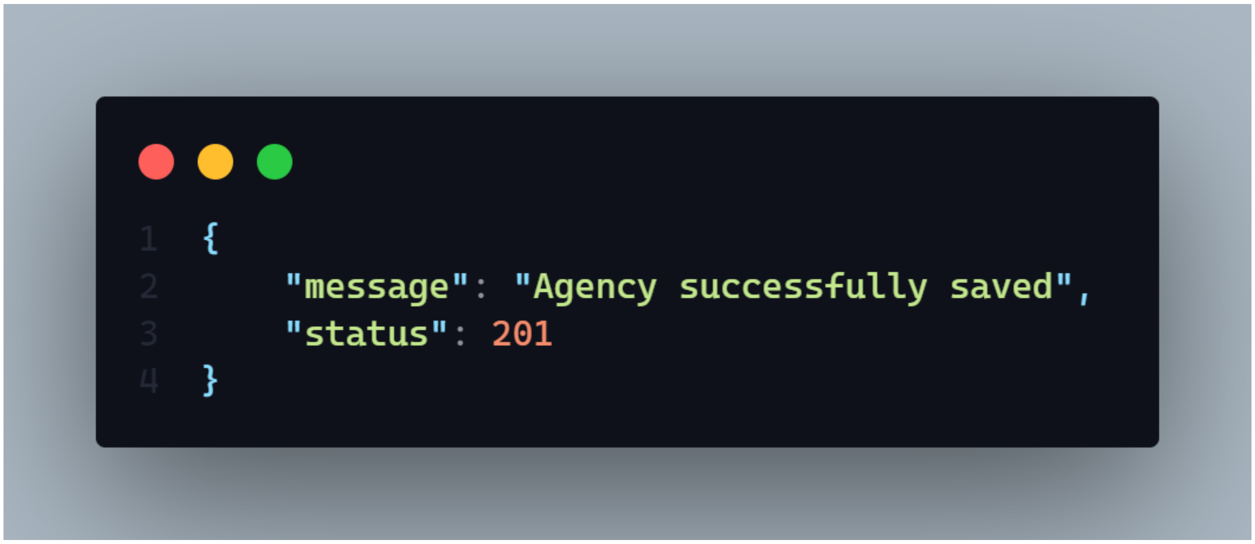
```
 {
  agentID: 1,
  name: 'shopnil',
  area: 'Dhanmondi',
  district: 'Dhaka',
  division: 'Dhaka',
  country: 'Bangaldesh',
  motive: 'Hello this is agency',
  email: 'sadmanishopnil@gmail.com',
}
```

isAuth: true

Params: N/A

Query: N/A

Response:



```
{
      "message": "Agency successfully saved",
      "status": 201
}
```

Here

- **message** will sent a response message. Every time it will come mandatory
- **status** will return a http status code here it will sent **406** and **201**

Description: If status code is

- **202 =>**
  - Agency has updated successfully
- **406 =>**
  - Agency update failed
  - Internal error!!
- **402 =>**
  - Body input validation error

2. Logged in user can update current password

a. **Data need to take from body**
   i. **New password**
   ii. **Confirm password**

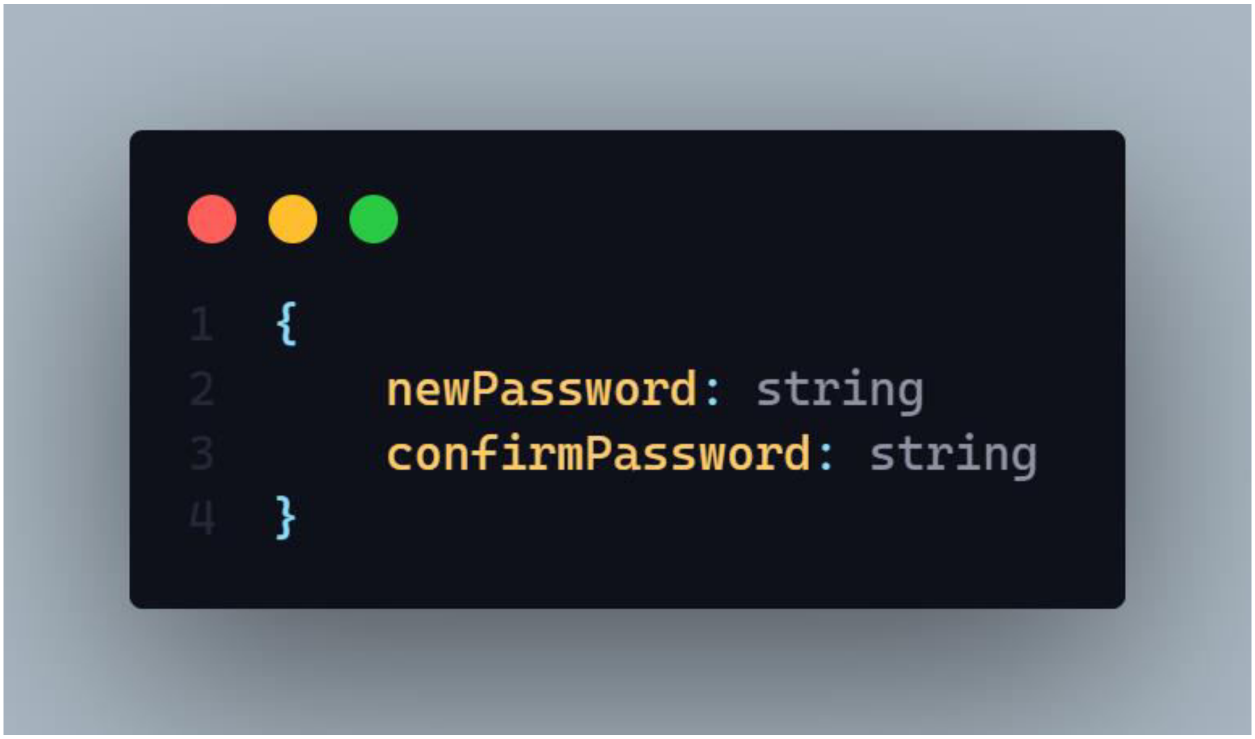
API: baseUrl/ agency/password/update

Type: REST

Method : PUT

Access: **: All**

Body :



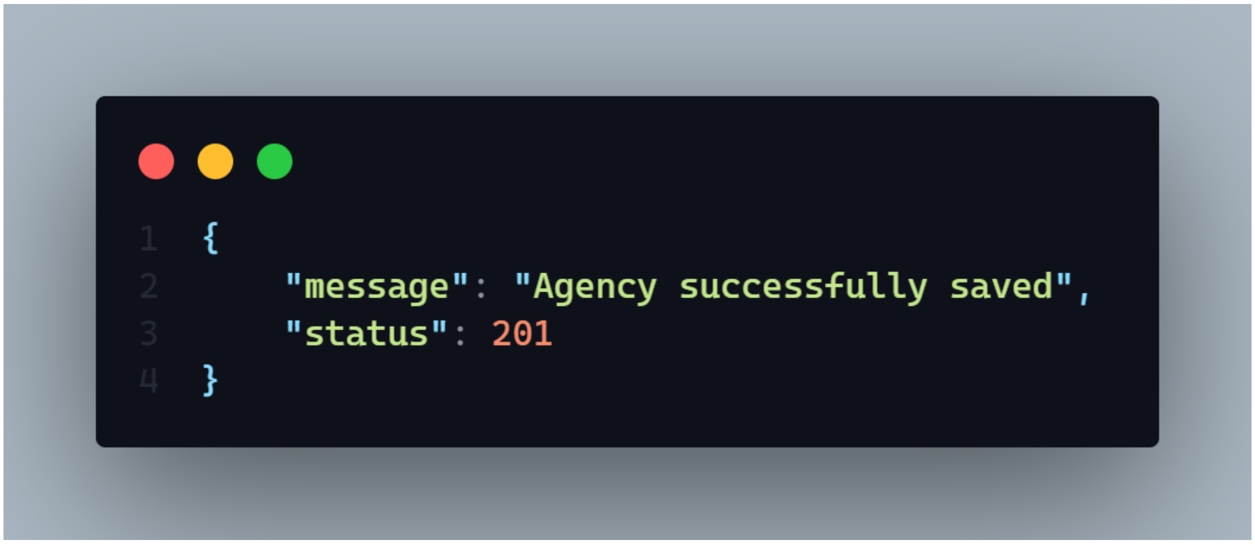
```
{
      newPassword: string
      confirmPassword: string
}
```

isAuth: true

Params: N/A

Query: N/A

Response:



```
{
      "message": "Agency successfully saved",
      "status": 201
}
```

Here

- **message** will sent a response message. Every time it will come mandatory
- **status** will return a http status code here it will sent **406** and **201**

Description: If status code is

- **202 =>**
  - Password has successfully updated
- **406 =>**
  - Password update failed
  - Internal error!!
- **402 =>**
  - Body input validation error

3. Logged in agency can change it’s title or cover picture

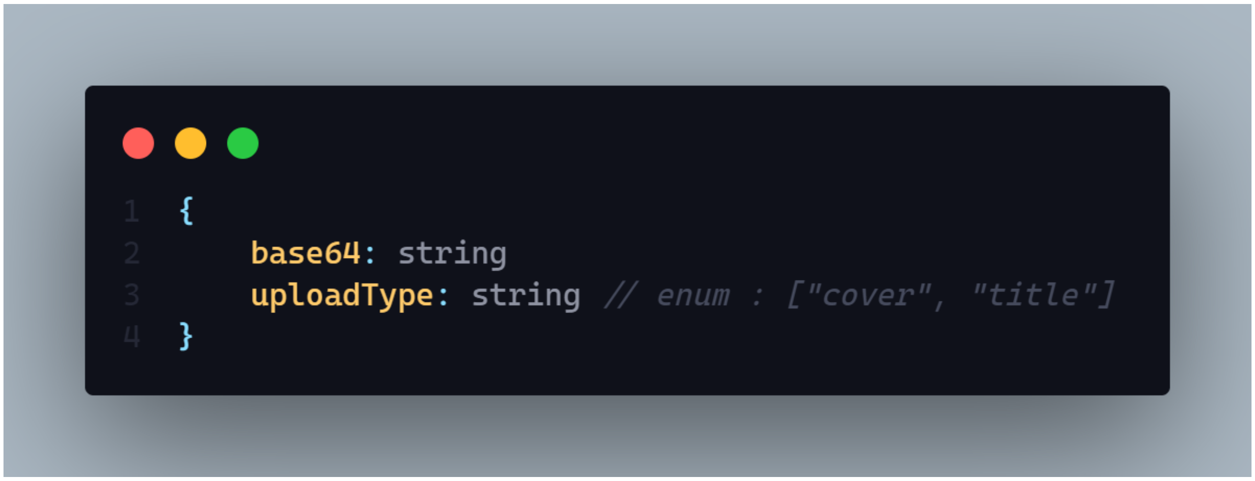
API: baseUrl/agency/profile/picture/update

Type: REST

Method : PUT

Access: **: All**

Body :



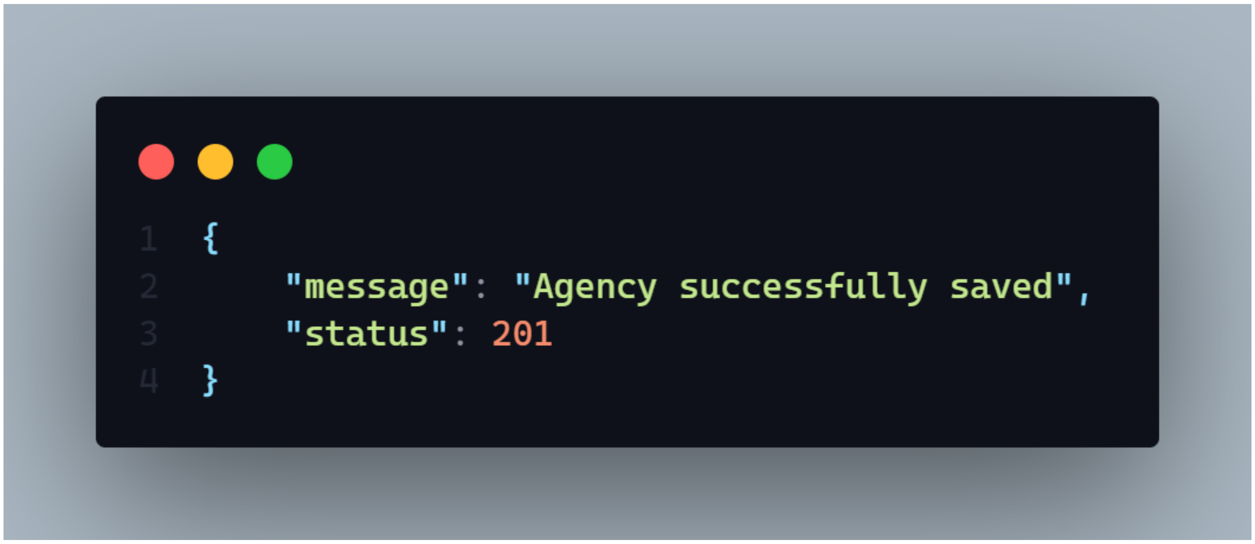
```
{
    base64: string
    uploadType: string // enum : ["cover", "title"]
}
```

isAuth: true

Params: N/A

Query: N/A

Response:



```
{
      "message": "Agency successfully saved",
      "status": 201
}
```

Here

- **message** will sent a response message. Every time it will come mandatory
- **status** will return a http status code here it will sent **406** and **201**

Description: If status code is

- **202 =>**
  - Cover or Title picture has been updated successfully
- **406 =>**
  - Data delete failed from public folder
  - Internal error!!
- **402 =>**
  - Body input validation error

4. Logged in user can input multiple air data

- **b. Input policy**
  - **i. by form (manual type)**
    1. **division**
    2. **date**
    3. **Value of Pm2.5**
    4. **Avg Temperature (Not)**
    5. **Rain Preception (Not)**
    6. **Wind speed (Not)**
    7. **Visibility (Not)**
    8. **Cloud cover (Not)**
    9. **Relative humidity (Not)**
    10. **Station**
    11. **Season**
  - **ii. by CSV file**

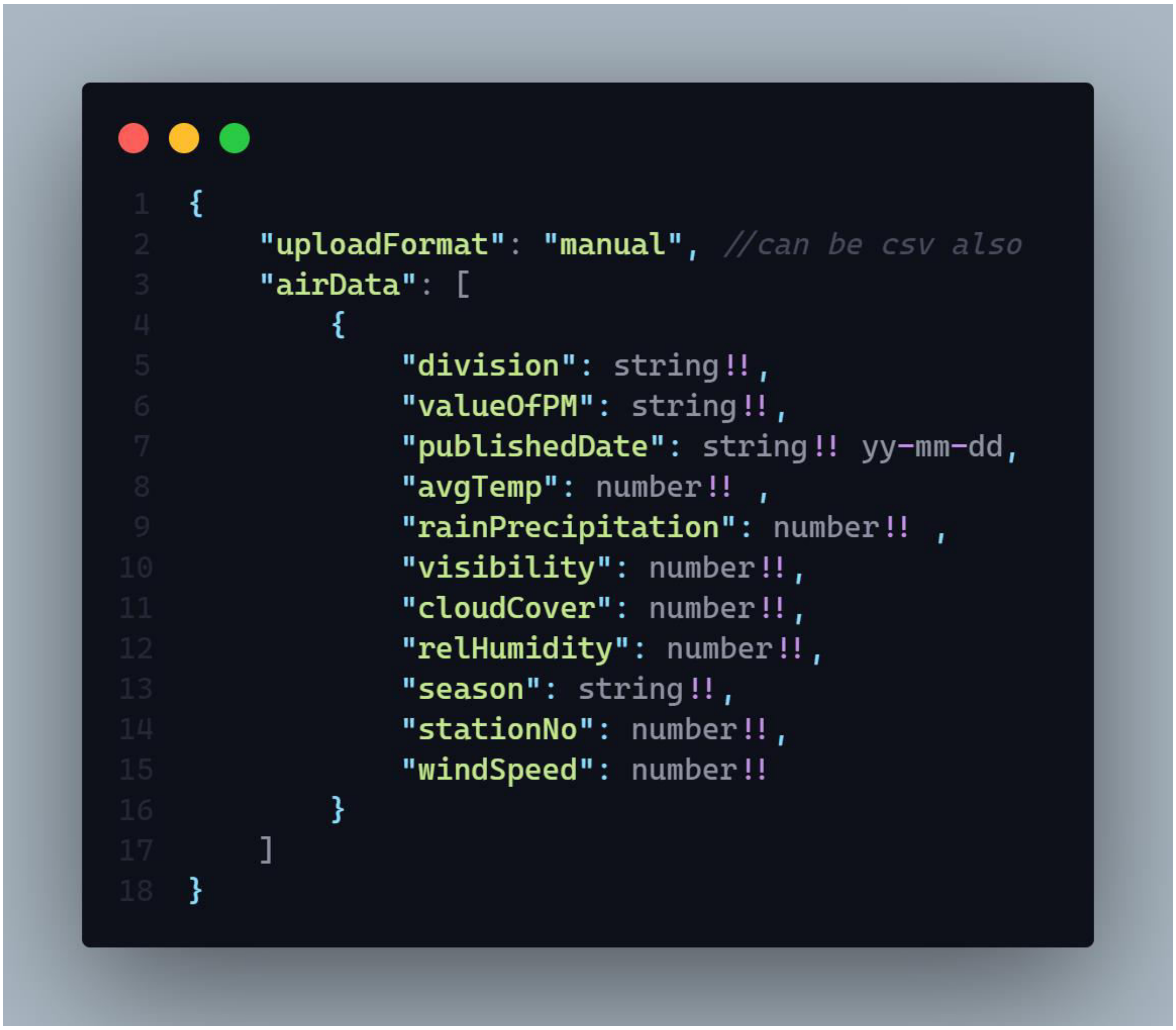
API: baseUrl/airData/create

Type: REST

Method : POST

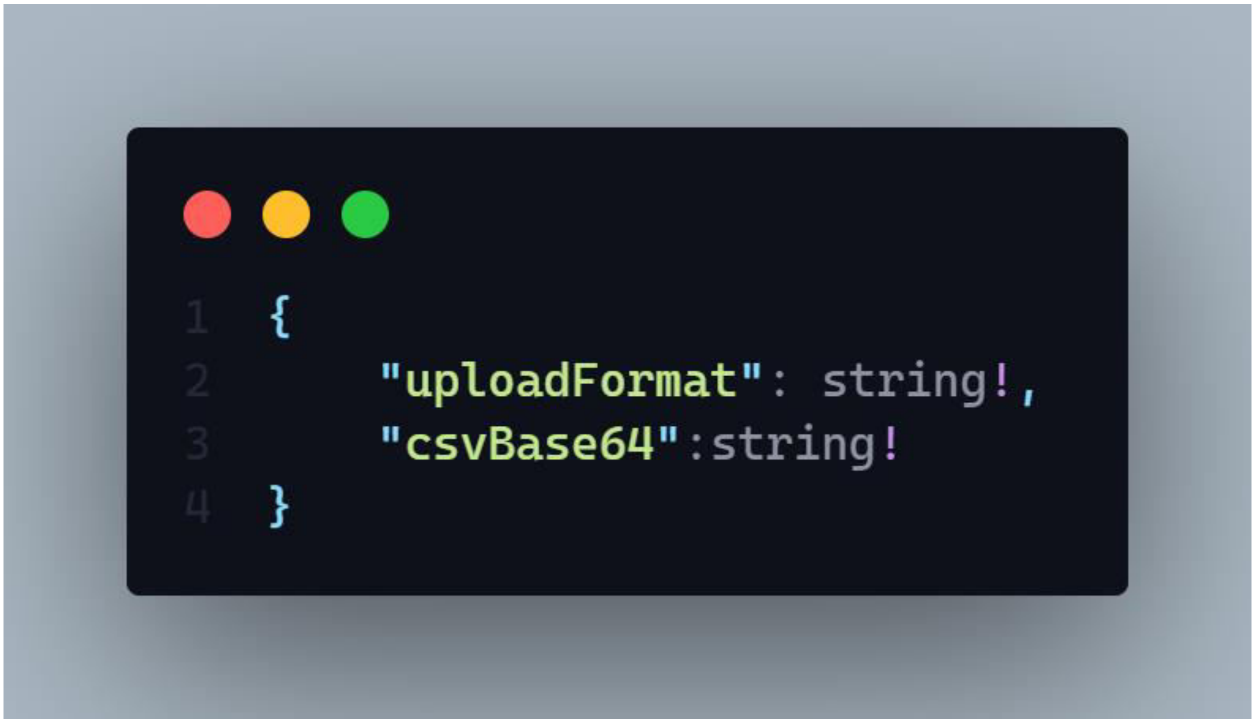
Access: **: All**

Body : For Manual Upload



```
{
    "uploadFormat": "manual", //can be csv also
    "airData": [
        {
            "division": string!!,
            "valueOfPM": string!!,
            "publishedDate": string!! yy-mm-dd,
            "avgTemp": number!! ,
            "rainPrecipitation": number!!  ,
            "visibility": number!!,
            "cloudCover": number!!,
            "relHumidity": number!!,
            "season": string!!,
            "stationNo": number!!,
            "windSpeed": number!!
        }
    ]
}
```

For CSV Upload:



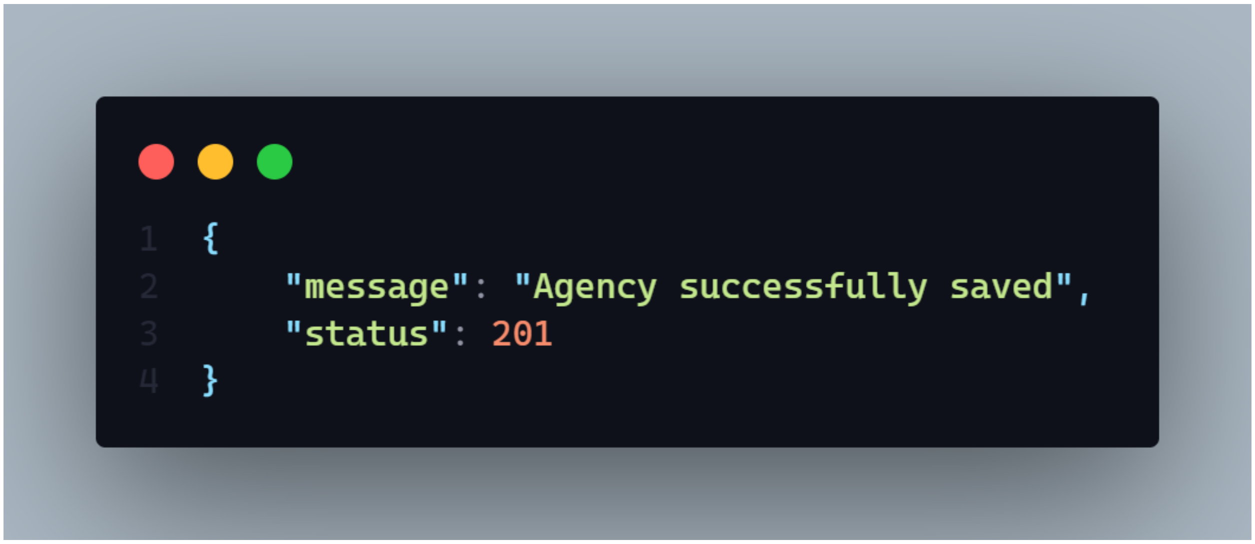
```
{
      "uploadFormat": string!,
      "csvBase64":string!
}
```

isAuth: true

Params: N/A

Query: N/A

Response:



```
{
      "message": "Agency successfully saved",
      "status": 201
}
```

Here

- **message** will sent a response message. Every time it will come mandatory
- **status** will return a http status code here it will sent **406** and **201**

Description: If status code is

- **201 =>**
    - New Air data collected!!!
- **406 =>**
    - Csv file delete failed from the server
    - CSV file parsing failed
    - CSV file upload failed to the server
    - New air data failed to collect please try again
    - Internal error!!
- **402 =>**
    - Body input validation error

5. Logged in user can see his provided input data

a. Sort by
   i. Latest (default)
   ii. A – Z by Division
   iii. Z – A by Division
b. Search by
   i. dataId
   ii. Division
   iii. pmValue
c. Pagination include
   i. Default 5 data will be show
d. Get Data
   1. division
   2. published date
   3. Value of Pm2.5
   4. Avg Temperature (Not)
   5. Rain Preception (Not)
   6. Wind speed (Not)
   7. Visibility (Not)
   8. Cloud cover (Not)
   9. Relative humidity (Not)
   10. Station
   11. Season

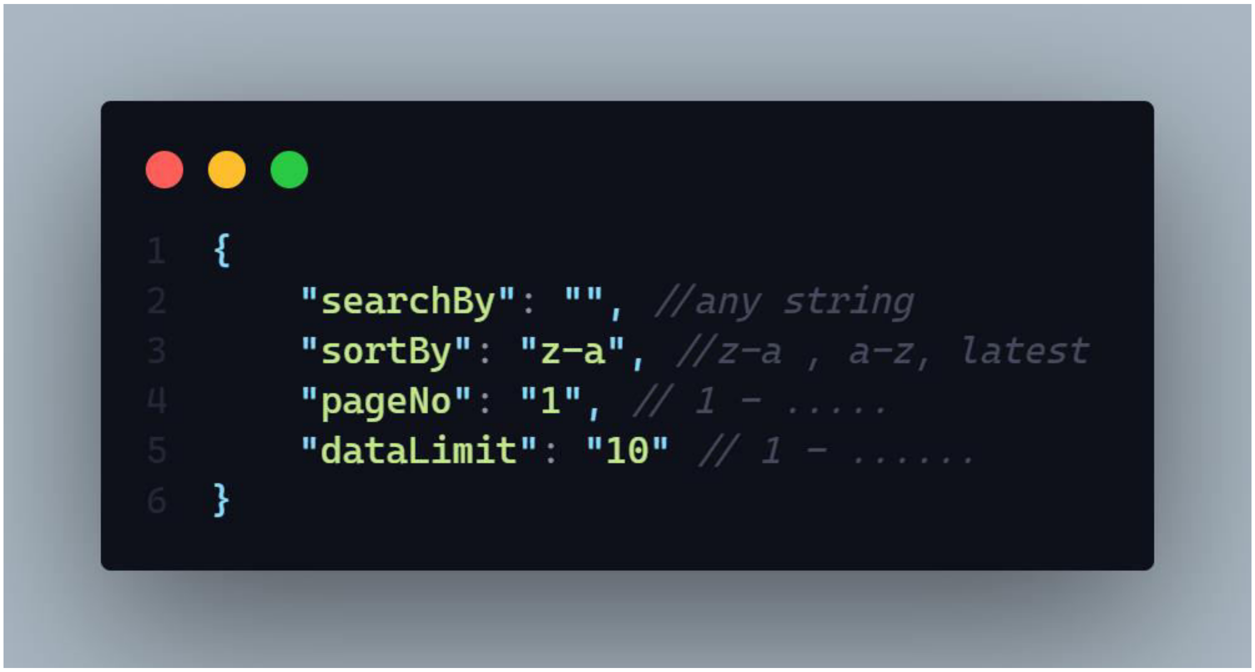
API: baseUrl/airData/showOwn/data

Type: REST

Method : POST

Access: **: All**

Body :



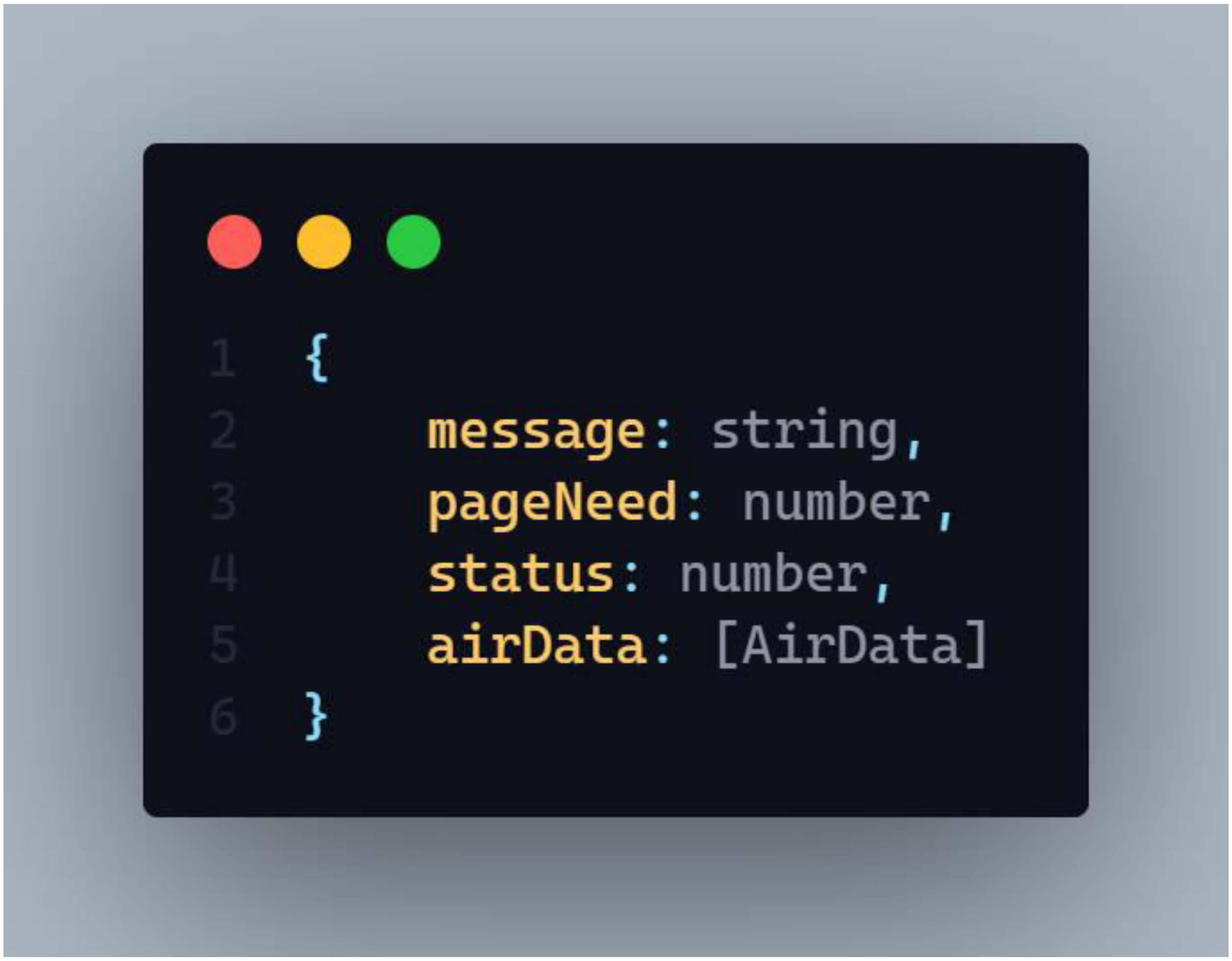
```
{
      "searchBy": "", //any string
      "sortBy": "z-a", //z-a , a-z, latest
      "pageNo": "1", // 1 - .....
      "dataLimit": "10" // 1 - ......
}
```

isAuth: true

Params: N/A

Query: N/A

Response:



```
{
        message: string,
        pageNeed: number,
        status: number,
        airData: [AirData]
}
```

Here

- **message** will sent a response message. Every time it will come mandatory
- **status** will return a http status code here it will sent **406** and **201**
- **pageNeed:** It will show how many page need to do this operation
- **airData:** here all match data will be show.

Description: If status code is

- **202 =>**
    - {numberOf} data found!!!
- **406 =>**
    - Internal error!!
- **404 =>**
    - No data has found!!!

6. Logged in user can update his provided input data by airDataId

API: baseUrl/airData/update /:id

Type: REST

Method : PUT

Access: : All

Body :

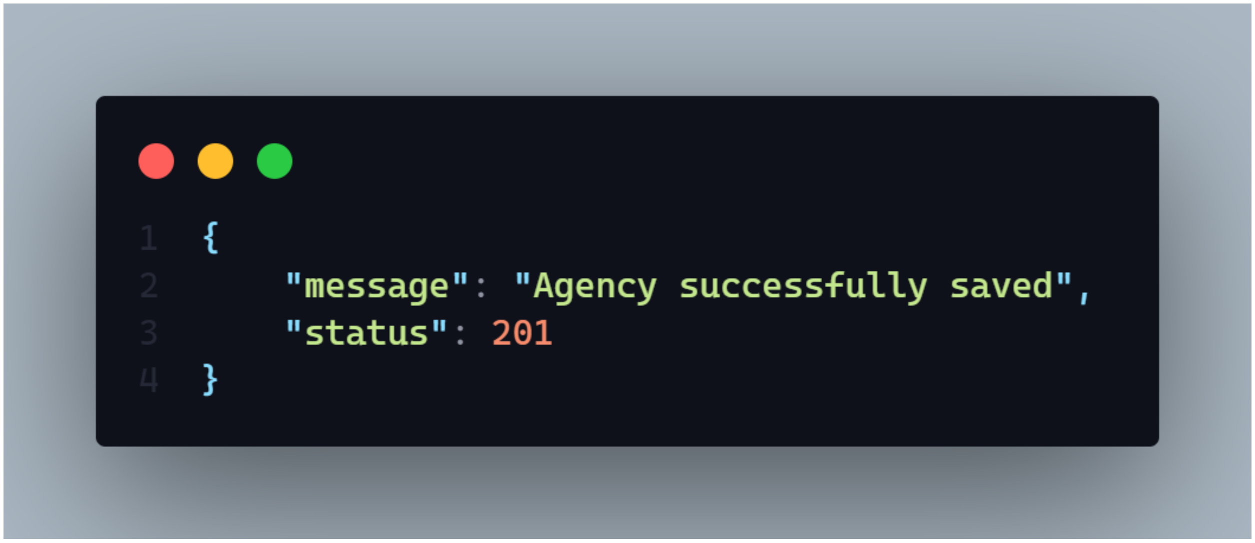
1. **division**
2. **Value of Pm2.5**
3. **Avg Temperature**
4. **Rain Preception**
5. **Wind speed**
6. **Visibility**
7. **Cloud cover**
8. **Relative humidity**
9. **Station**
10. **Season**

isAuth: true

Params: id: string

Query: N/A

Response:



```
{
      "message": "Agency successfully saved",
      "status": 201
}
```

Here

- **message** will sent a response message. Every time it will come mandatory
- **status** will return a http status code here it will sent **406** and **201**

Description: If status code is

- **202 =>**
    - Successfully updated!!!
- **406 =>**
    - Internal error!!
    - Update failed!!!

7. Logged in user can delete his provided input data by airDataId

API: baseUrl/airData/update /:id

Type: REST

Method : PUT

Access: **: All**

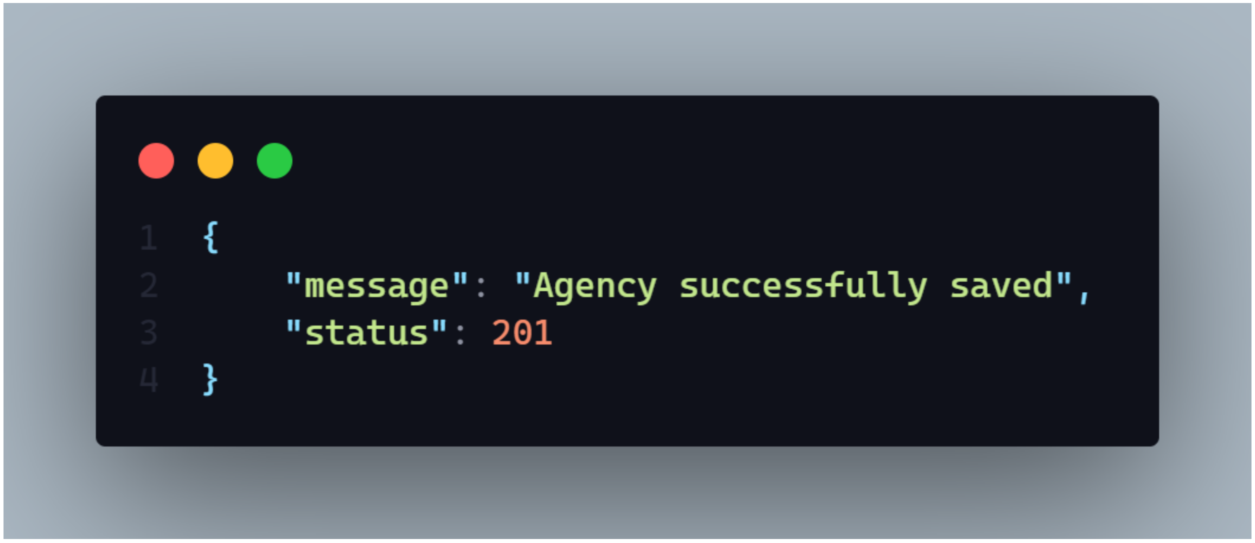
Body : N/A

isAuth: true

Params: id: string (airDataId)

Query: N/A

Response:



```
{
    "message": "Agency successfully saved",
    "status": 201
}
```

Here

- **message** will sent a response message. Every time it will come mandatory
- **status** will return a http status code here it will sent **406** and **201**

Description: If status code is

- **202 =>**
  - Successfully Delete!!!
- **406 =>**
  - Internal error!!
  - Delete failed!!!

8. Logged in user can input multiple daily air data

- e. **Input policy**
  - i. **by form (manual type)**
    1. **area**
    2. **date**
    3. **latitude**
    4. **longitude**
    5. **median**
    6. **mean**
    7. **max**
    8. **sum**
    9. **count**
  - ii. **by CSV file**

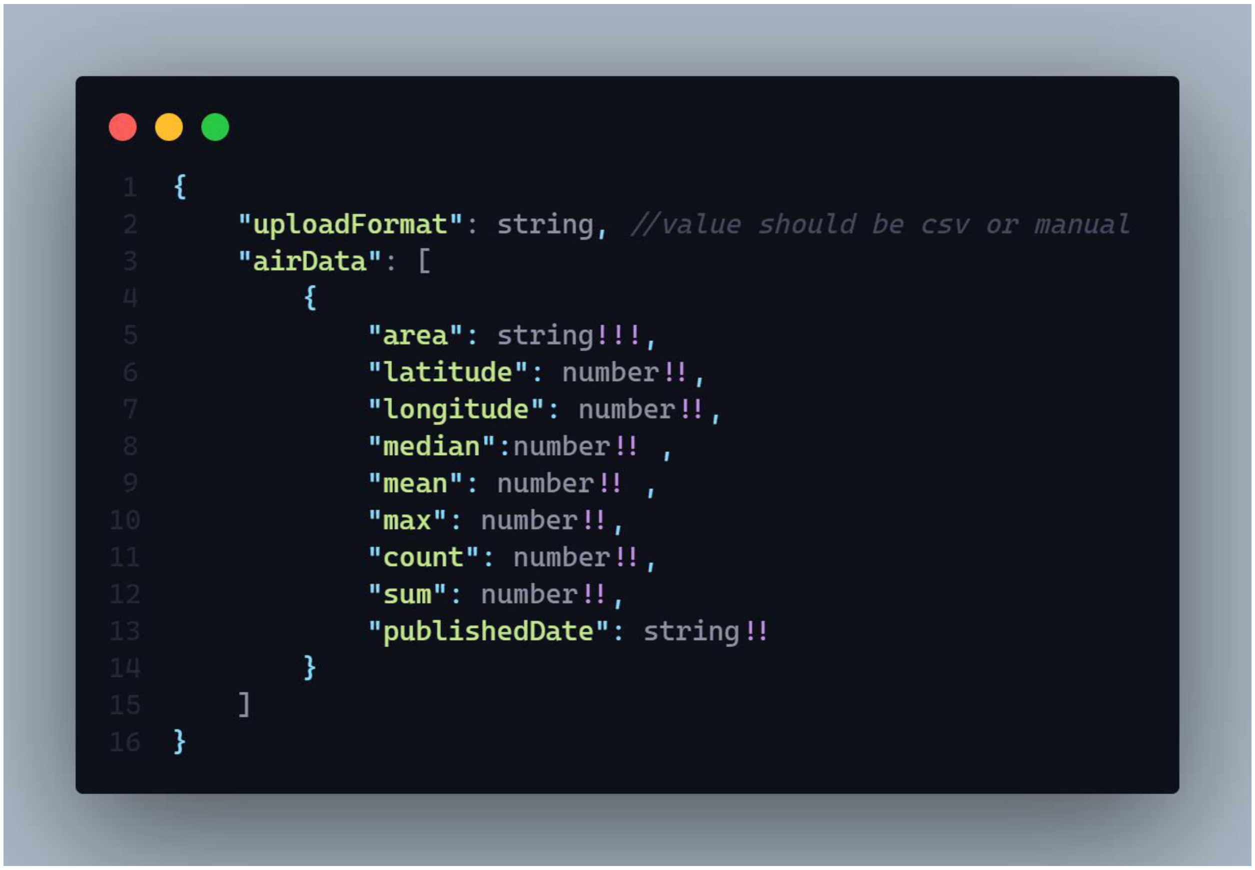
API: baseUrl/airData/daily/create

Type: REST

Method : POST

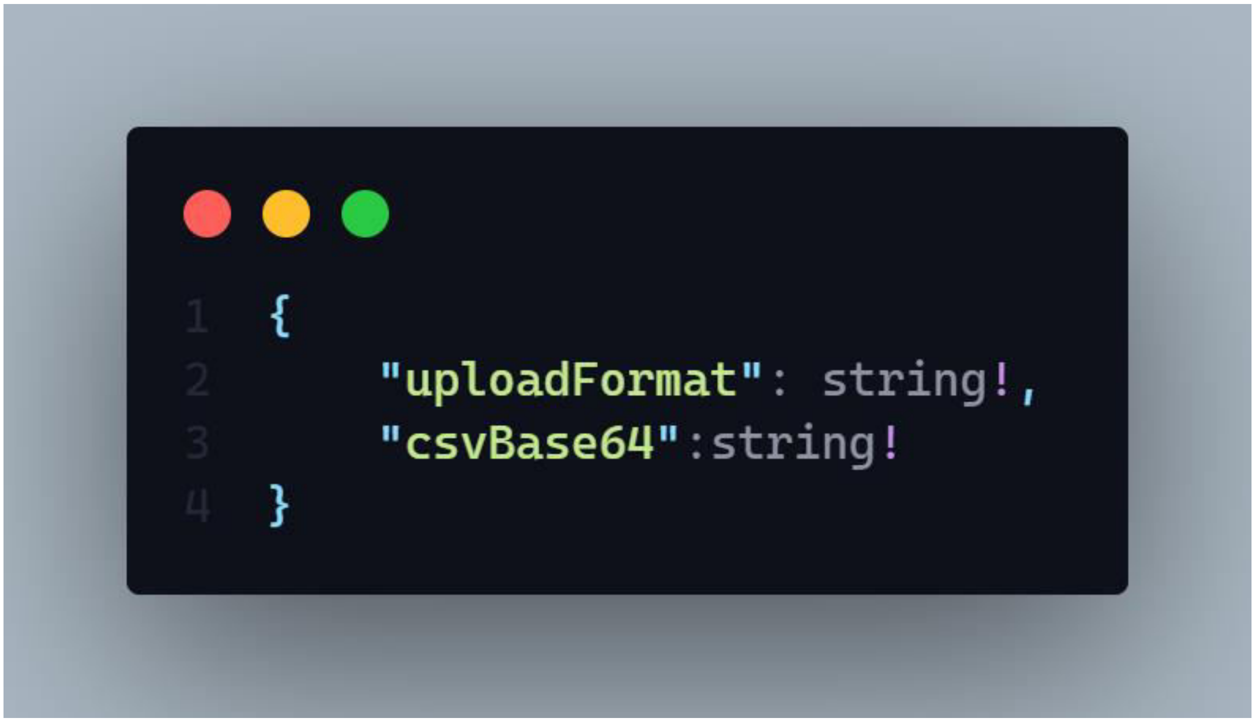
Access: **: All**

Body : For Manual Upload



```
{
    "uploadFormat": string, //value should be csv or manual
    "airData": [
        {
            "area": string!!!,
            "latitude": number!!,
            "longitude": number!!,
            "median":number!!  ,
            "mean": number!!  ,
            "max": number!!,
            "count": number!!,
            "sum": number!!,
            "publishedDate": string!!
        }
    ]
}
```

For CSV Upload:



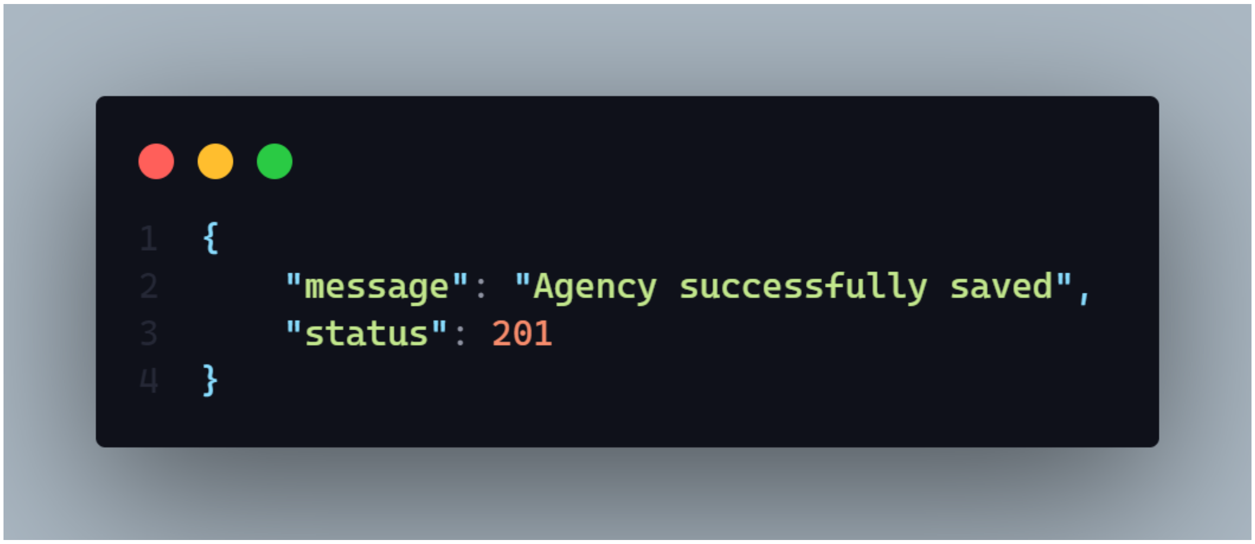
```
{
      "uploadFormat": string!,
      "csvBase64":string!
}
```

isAuth: true

Params: N/A

Query: N/A

Response:



```
{
      "message": "Agency successfully saved",
      "status": 201
}
```

Here

- **message** will sent a response message. Every time it will come mandatory
- **status** will return a http status code here it will sent **406** and **201**

Description: If status code is

- **201 =>**
  - New Air data collected!!!
- **406 =>**
  - Csv file delete failed from the server
  - CSV file parsing failed
  - CSV file upload failed to the server
  - New air data failed to collect please try again
  - Internal error!!
- **402 =**
  - Body input validation error

9. Logged in user can see his provided daily input data

a. **Sort by**
   iii. **Latest (default)**
   iv. **By median (Descending order)**
   v. **By mean (Descending order)**
   vi. **By max (Descending order)**
   vii. **By sum (Descending order)**
   viii. **By count (Descending order)**

b. **Search by**
   ix. **dataId**
   x. **Area**

c. **Pagination include**
   xi. **Default 5 data will be show**

d. **Get Data**
   1. **area**
   2. **date**
   3. **median**
   4. **mean**
   5. **max**
   6. **sum**
   7. **count**
   8. **dataId**

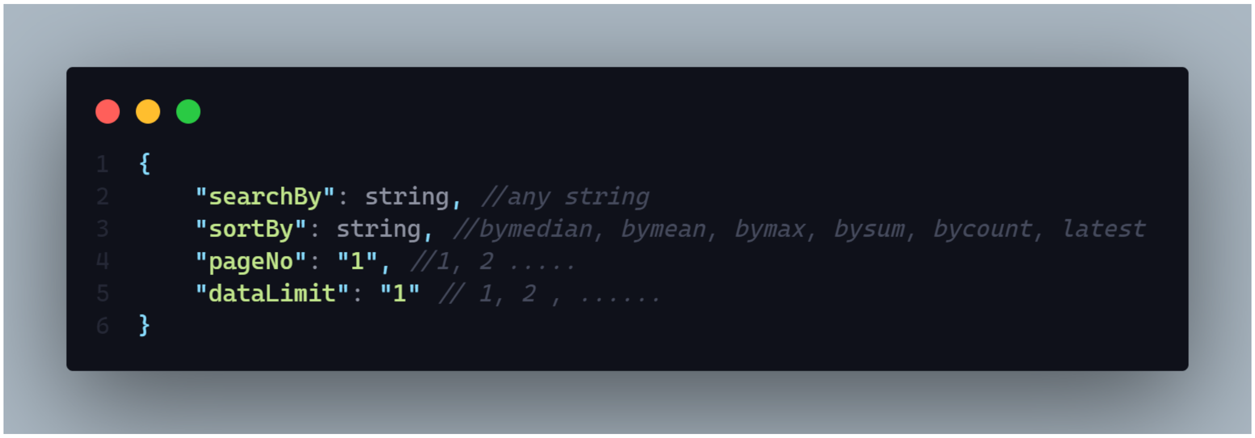
API: baseUrl/airData/showOwn/daily/data

Type: REST

Method : POST

Access: **: All**

Body :



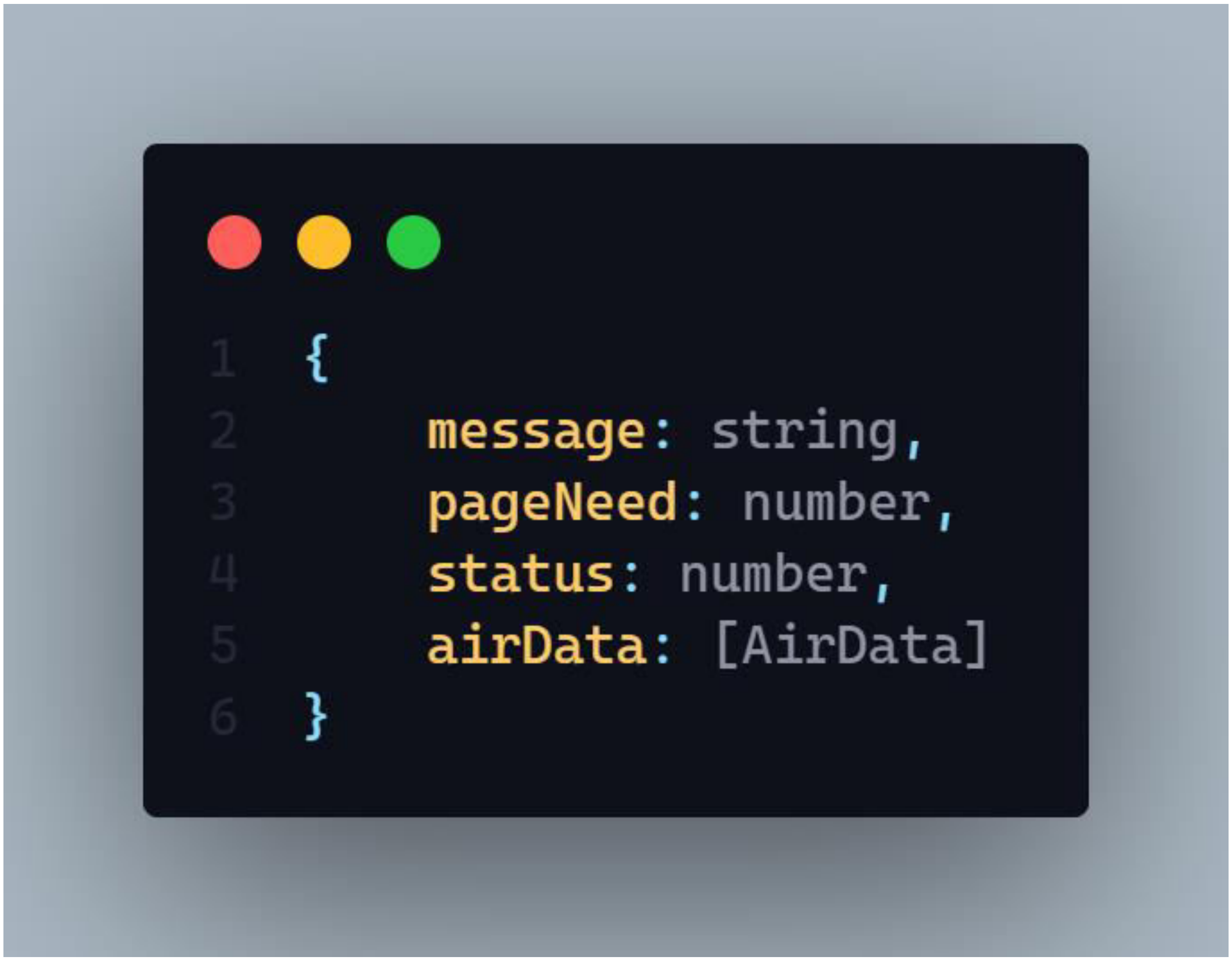
```
{
    "searchBy": string, //any string
    "sortBy": string, //bymedian, bymean, bymax, bysum, bycount, latest
    "pageNo": "1", //1, 2 .....
    "dataLimit": "1" // 1, 2 , ......
}
```

isAuth: true

Params: N/A

Query: N/A

Response:



```
{
      message: string,
      pageNeed: number,
      status: number,
      airData: [AirData]
}
```

Here

- **message** will sent a response message. Every time it will come mandatory
- **status** will return a http status code here it will sent **406** and **201**
- **pageNeed:** It will show how many page need to do this operation
- **airData:** here all match data will be show.

Description: If status code is

- **202 =>**
  - {numberOf} data found!!!
- **406 =>**
  - Internal error!!
- **404 =>**
  - No data has found!!!

10. Logged in user can delete his provided daily input data by dailyAirDataId

API: baseUrl/airData/delete /daily/data/:id

Type: REST

Method : PUT

Access: **: All**

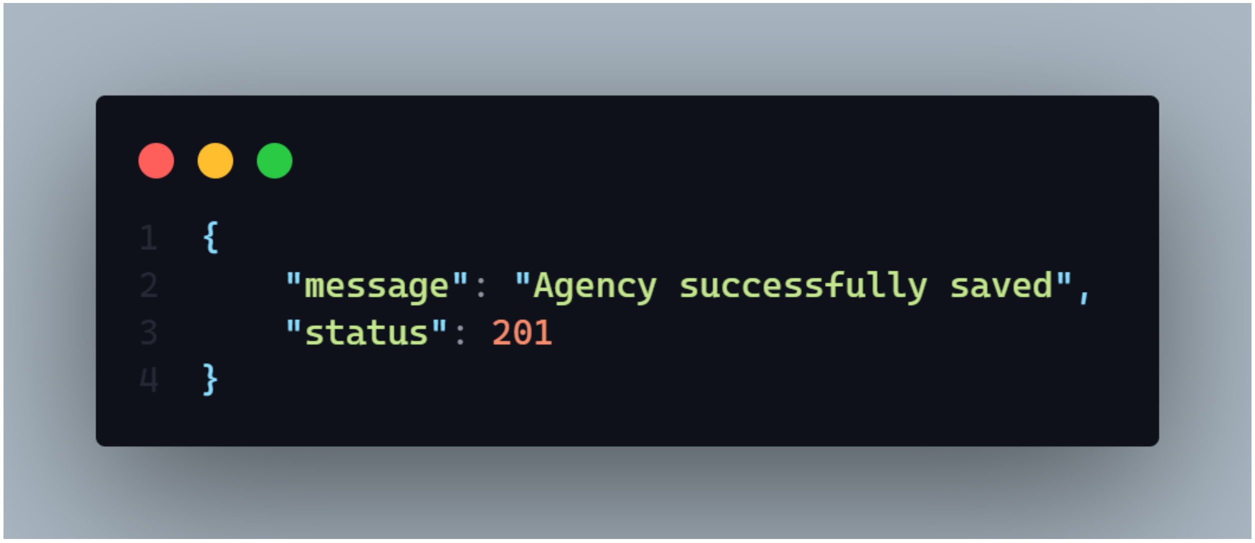
Body : N/A

isAuth: true

Params: id: string (dailyAirDataId)

Query: N/A

Response:



```
{
    "message": "Agency successfully saved",
    "status": 201
}
```

Here

- **message** will sent a response message. Every time it will come mandatory
- **status** will return a http status code here it will sent **406** and **201**

Description: If status code is

- **202 =>**
  - Successfully Delete!!!
- **406 =>**
  - Internal error!!
  - Delete failed!!!

11. Logged in user can update his provided input data by airDataId

API: baseUrl/airData/update /daily/data/:id

Type: REST

Method : PUT

Access: **: All**

Body :

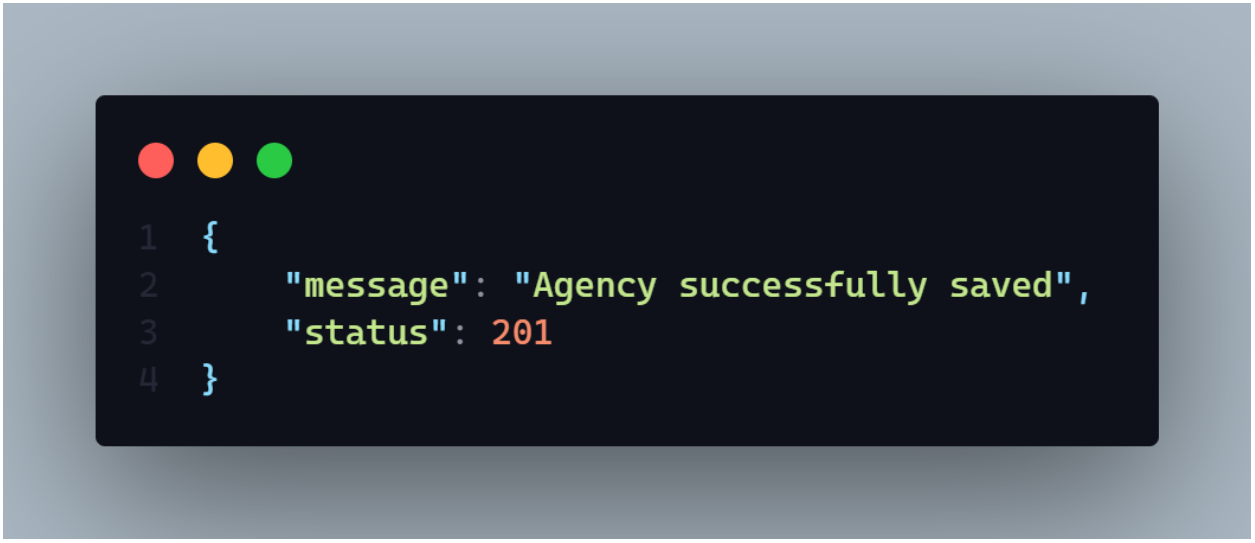
1. **area**
2. **date**
3. **median**
4. **mean**
5. **max**
6. **sum**
7. **count**
8. **dataId**

isAuth: true

Params: id: string

Query: N/A

Response:



```
{
      "message": "Agency successfully saved",
      "status": 201
}
```

Here

- **message** will sent a response message. Every time it will come mandatory
- **status** will return a http status code here it will sent **406** and **201**

Description: If status code is

- **202 =>**
  - Successfully updated!!!
- **406 =>**
  - Internal error!!
  - Update failed!!!

Global API’S

1. Register a new Agency

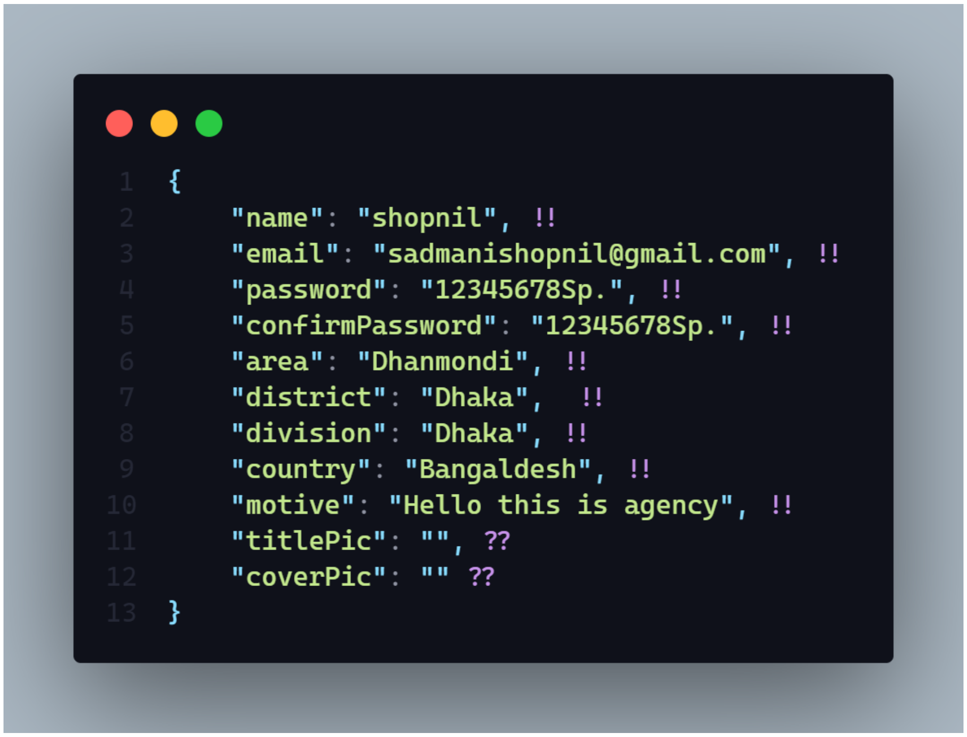
API: baseUrl/agecy/registration

Type: REST

Method : POST

Access: : All

Body :

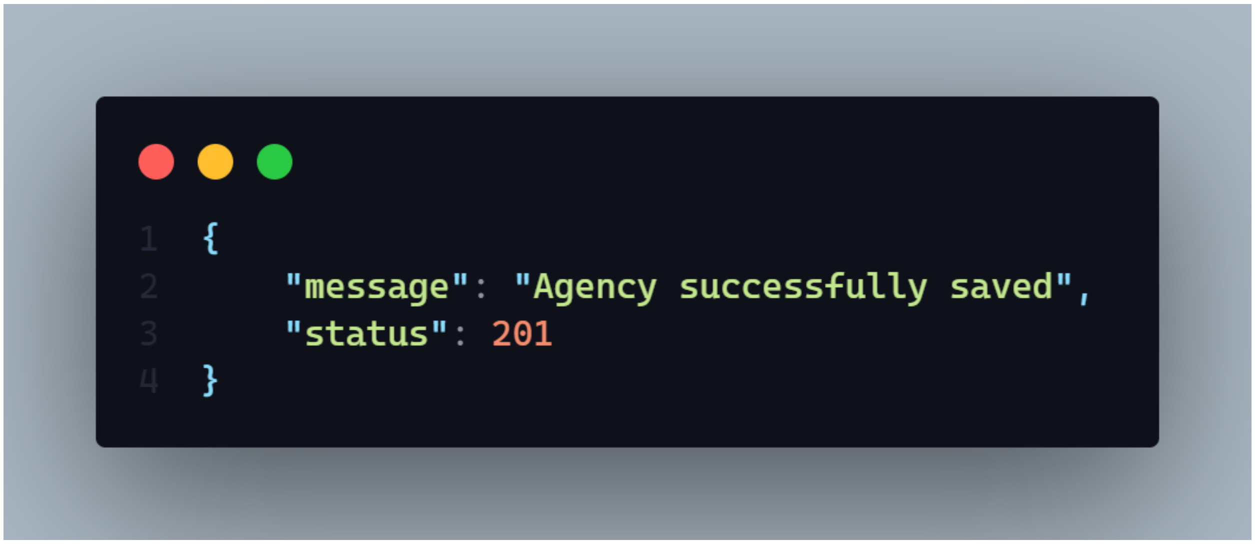


```
{
    "name": "shopnil", !!
    "email": "sadmanishopnil@gmail.com", !!
    "password": "12345678Sp.", !!
    "confirmPassword": "12345678Sp.", !!
    "area": "Dhanmondi", !!
    "district": "Dhaka",  !!
    "division": "Dhaka", !!
    "country": "Bangaldesh", !!
    "motive": "Hello this is agency", !!
    "titlePic": "", ??
    "coverPic": "" ??
}
```

Params: N/A

Query: N/A

Response:



```
{
    "message": "Agency successfully saved",
    "status": 201
}
```

Here

- **message** will sent a response message. Every time it will come mandatory
- **status** will return a http status code here it will sent **406** and **201**

Description: If status code is

- **201 =>**
  - Agency successfully created
- **406 =>**
  - Any validation error
  - Confirm password doesn't match with password
  - Email is already input please try with another email
  - Title Picture upload failed please try again
  - Cover Picture upload failed please try again
  - Title default Picture upload failed please try again
  - Cover default Picture upload failed please try again
  - Internal error!!
- **400 =>**
  - Agency failed to save
- **402 =>**
  - Body input validation error

2. Login API

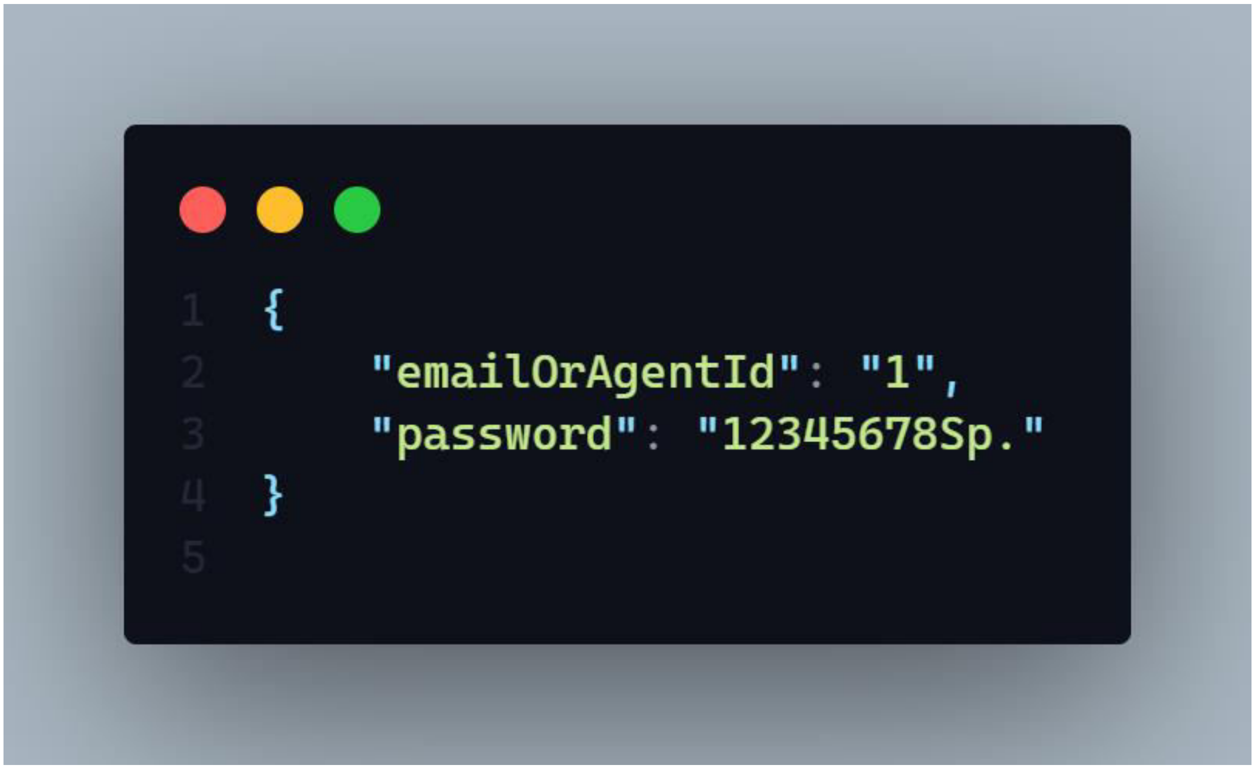
API: baseUrl/agecy/login

Type: REST

Method : POST

Access: : All

Body :



```
{
      "emailOrAgentId": "1",
      "password": "12345678Sp."
}

```

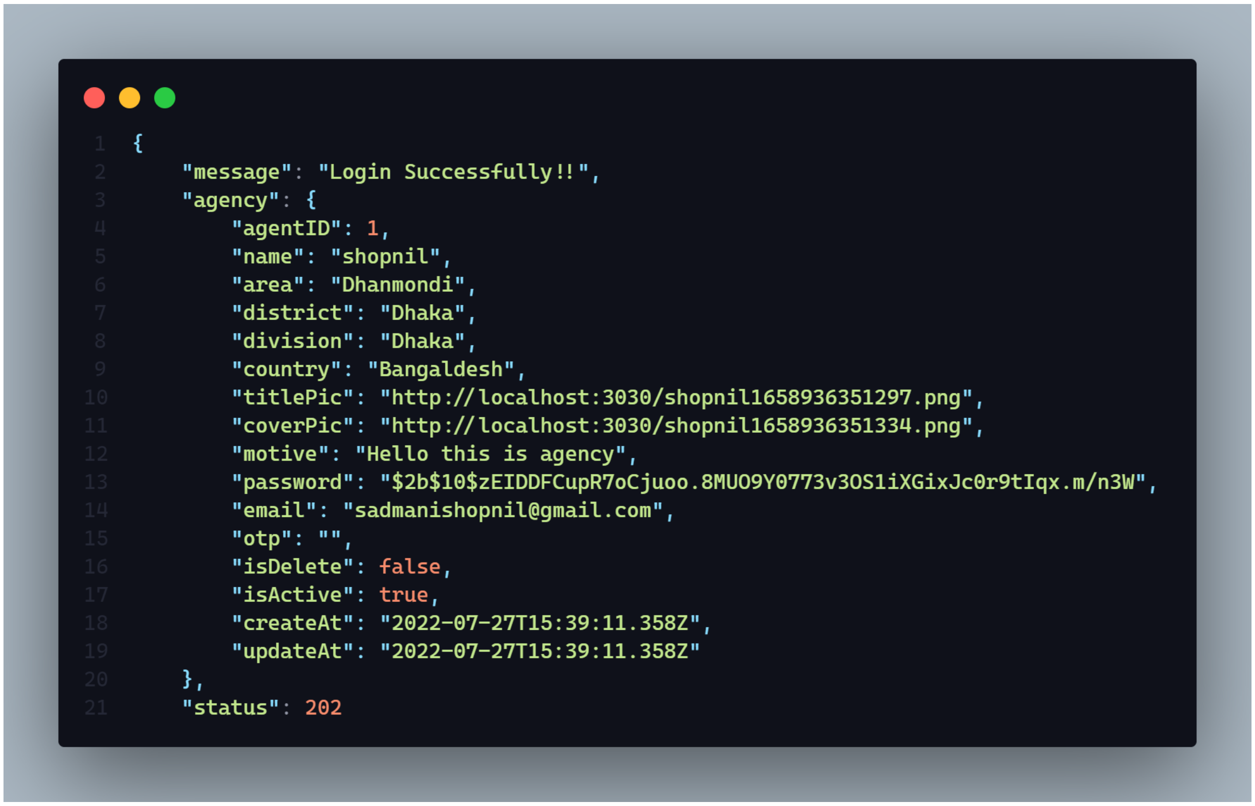
Password need to be =>

1. **Minimum One uppercase**
2. **Minimum One lowercase**
3. **Minimum One special character**
4. **Minimum One Digit**
5. **Length need to minimum 8**

Params: N/A

Query: N/A

Response:



```
{
    "message": "Login Successfully!!",
    "agency": {
        "agentID": 1,
        "name": "shopnil",
        "area": "Dhanmondi",
        "district": "Dhaka",
        "division": "Dhaka",
        "country": "Bangaldesh",
        "titlePic": "http://localhost:3030/shopnil1658936351297.png",
        "coverPic": "http://localhost:3030/shopnil1658936351334.png",
        "motive": "Hello this is agency",
        "password": "$2b$10$zEIDDFCupR7oCjuoo.8MUO9Y0773v3OS1iXGixJc0r9tIqx.m/n3W",
        "email": "sadmanishopnil@gmail.com",
        "otp": "",
        "isDelete": false,
        "isActive": true,
        "createAt": "2022-07-27T15:39:11.358Z",
        "updateAt": "2022-07-27T15:39:11.358Z"
    },
    "status": 202
```

Here

- **message** will sent a response message. Every time it will come mandatory
- **status** will return a http status code here it will sent **406** and **201**
- **Agency** will return the respective agency all data from database

Description: If status code is

- **202 =>**
    - Login successfully!!!
- **406 =>**
    - Password mismatch
    - Internal error!! / Runtime error
- **404 =>**
    - Agency not found
- **402 =>**
    - Body input validation error

3. Create Forgot password Part 1

a. Get the email and send a OTP to that email and set a JWT token

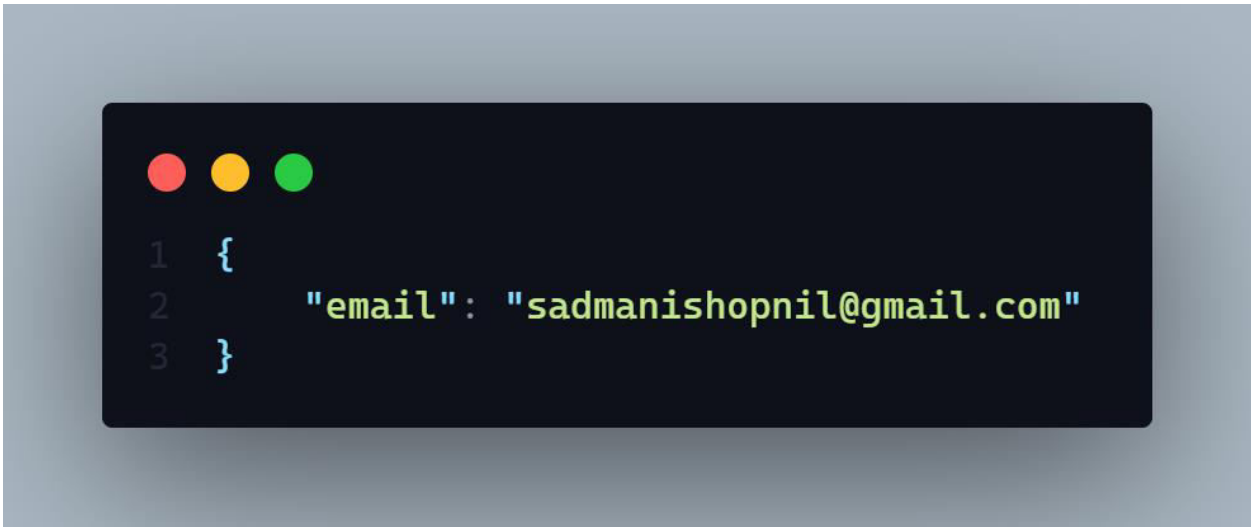
API: baseUrl/agecy/forgotPassword/verifyEmail

Type: REST

Method : POST

Access: : All

Body :



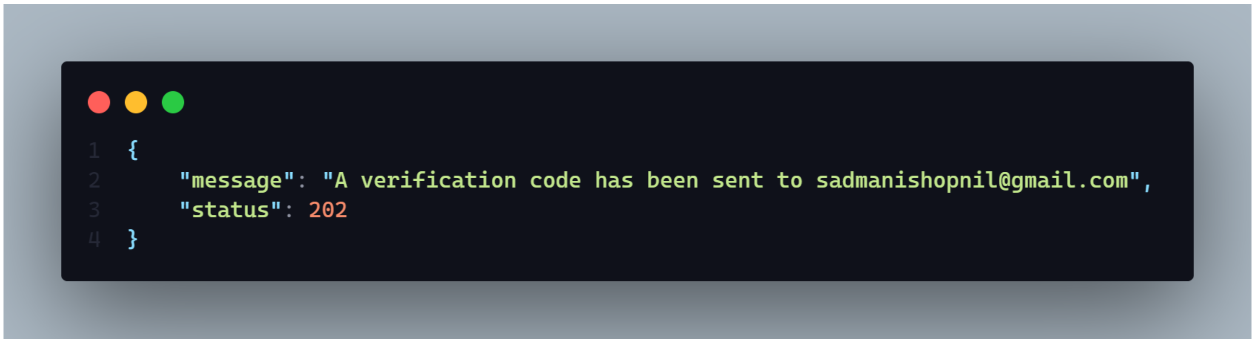
```
{
      "email": "sadmanishopnil@gmail.com"
}
```

Email need to be => A valid email

Params: N/A

Query: N/A

Response:



```
{
    "message": "A verification code has been sent to sadmanishopnil@gmail.com",
    "status": 202
}
```

Here

- **message** will sent a response message. Every time it will come mandatory
- **status** will return a http status code here it will sent **406** and **201**

Description: If status code is

- **202 =>**
  - A verification code has been sent to {email}
- **406 =>**
  - Otp update failed please try again
  - Runtime error
- **404 =>**
  - Agency not available with this email
- **402 =>**
  - Body input validation error

4. **Forgot password Part 2**
    a. **Take the OTP and send a positive message**

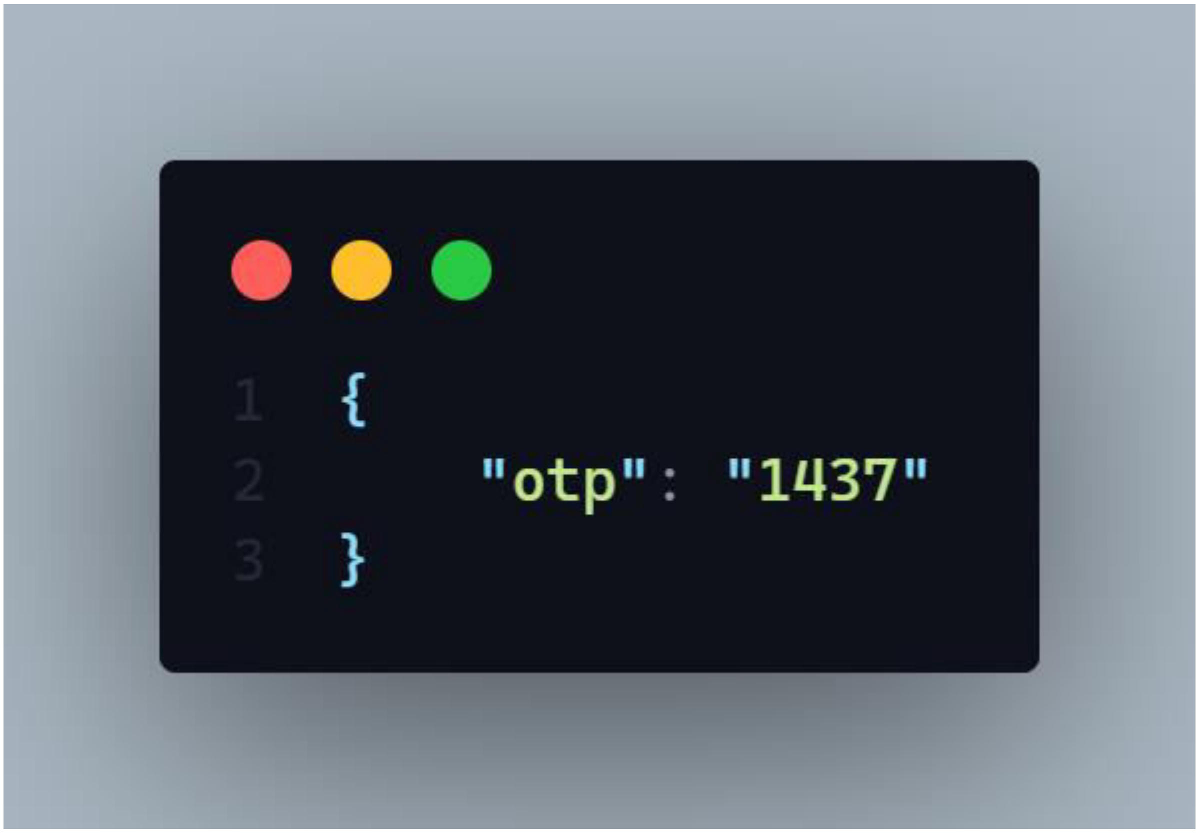
API: baseUrl/agecy/forgotPassword/verifyOtp

Type: REST

Method : POST

Access: : All

Body :

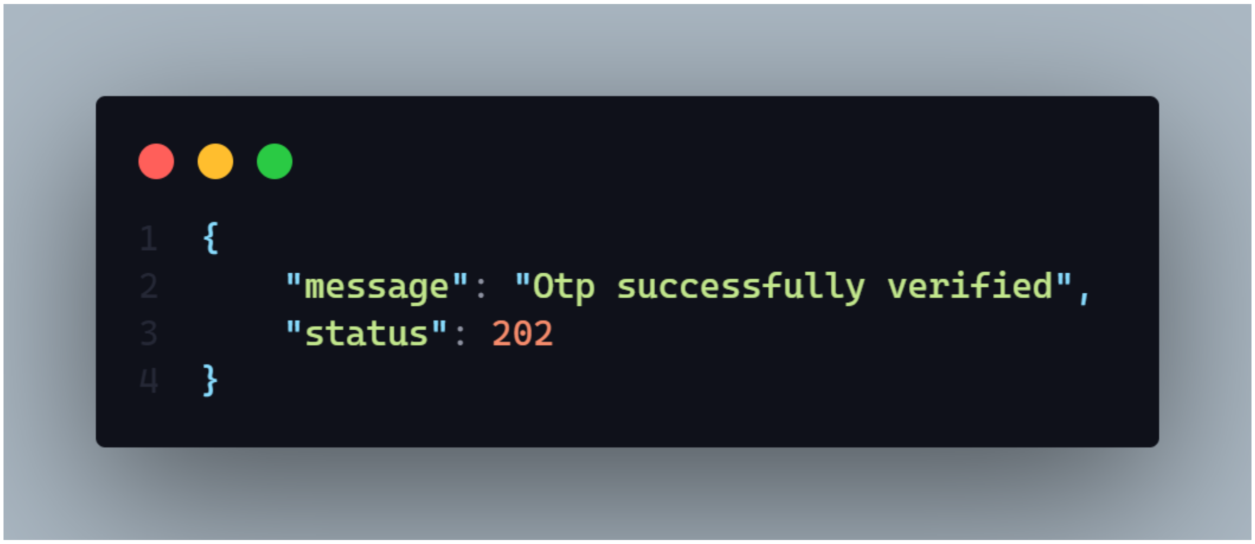


```
{
        "otp": "1437"
}
```

Params: N/A

Query: N/A

Response:



```
{
      "message": "Otp successfully verified",
      "status": 202
}
```

Here

- **message** will sent a response message. Every time it will come mandatory
- **status** will return a http status code here it will sent **406** and **201**

Description: If status code is

- **202 =>**
    - Otp successfully verified
- **406 =>**
    - Otp update failed please try again
    - OTP expired please try again
    - Runtime error / Internal error!!
- **402 =>**
    - Body input validation error

5. **Forgot password Part 3**
    a. **After verify the OTP now take new password including confirm password and change the password**

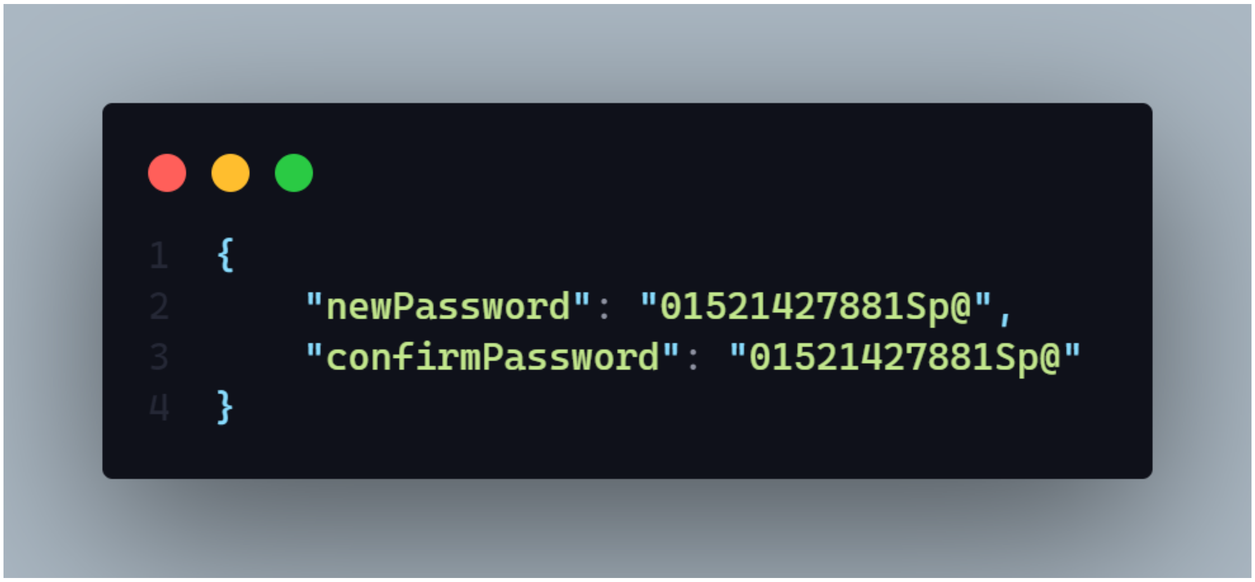
API: baseUrl/agecy/forgotPassword/resetPassword

Type: REST

Method : POST

Access: : All

Body :

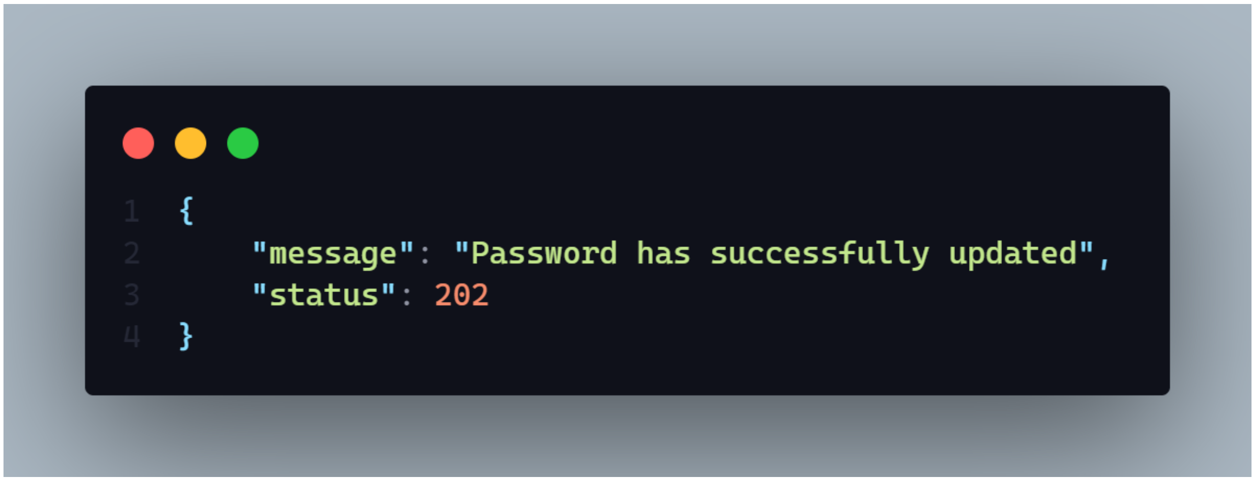


```
{
    "newPassword": "01521427881Sp@",
    "confirmPassword": "01521427881Sp@"
}
```

Params: N/A

Query: N/A

Response:



```
{
    "message": "Password has successfully updated",
    "status": 202
}
```

Here

- **message** will sent a response message. Every time it will come mandatory
- **status** will return a http status code here it will sent **406** and **201**

Description: If status code is

- **202 =>**
  - Password has successfully updated
- **406 =>**
  - OTP expired please try again
  - Password reset failed
  - Password hashing failed
  - Runtime error / Internal Error!!!
- **402 =>**
  - Body input validation error

6. Logut API

API: baseUrl/agecy/logout

Type: REST

Method : GET

Access: **: All**

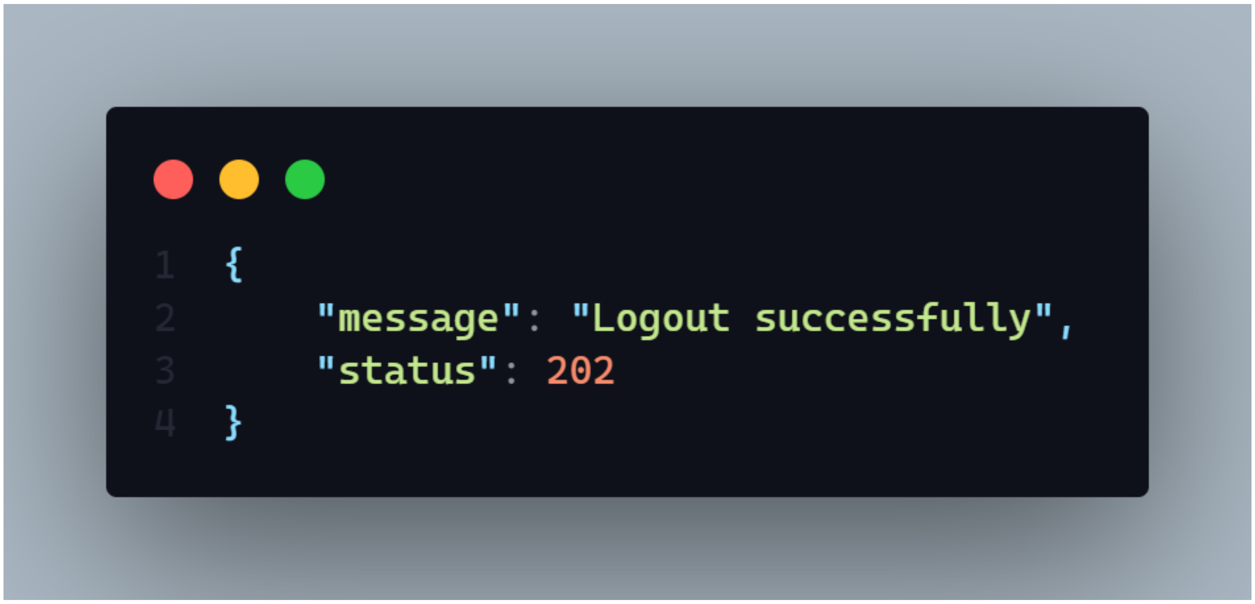
Body : N/A

Params: N/A

Auth: True

Query: N/A

Response:



```
{
    "message": "Logout successfully",
    "status": 202
}
```

Here

- **message** will sent a response message. Every time it will come mandatory
- **status** will return a http status code here it will sent **406** and **201**

Description: If status code is

- **202 =>**
  - Logout successfully
- **406 =>**
  - Runtime error / Internal Error!!!

7. Check Logged in session API

API: baseUrl/agecy/check/loggedIn/session

Type: REST

Method : POST

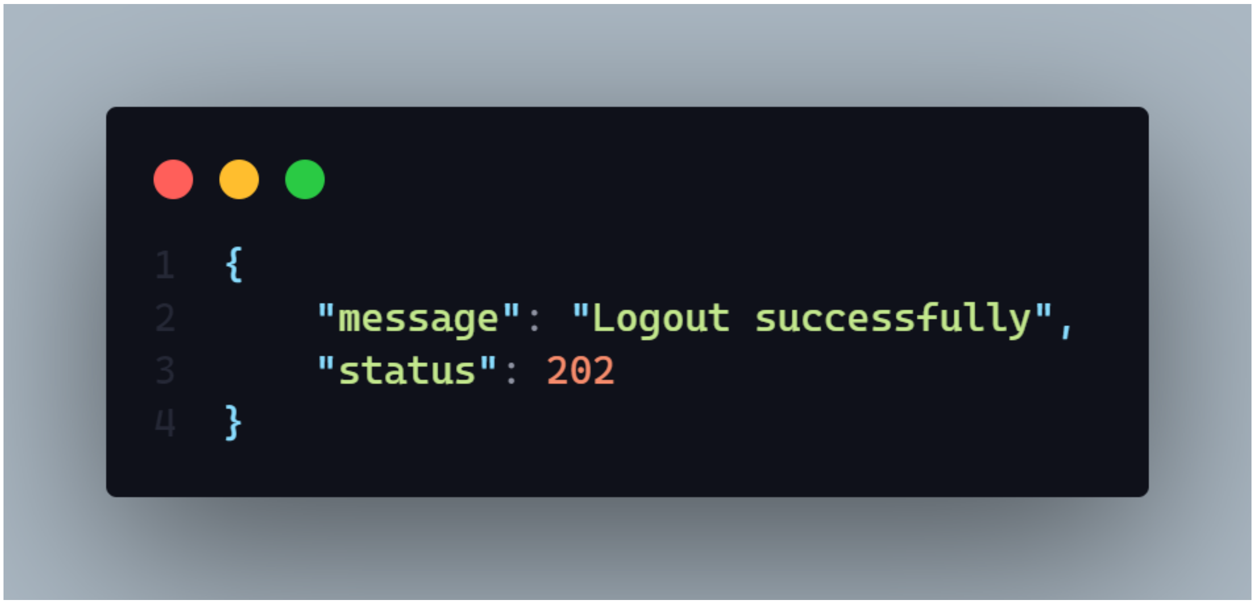
Access: : All

Body : N/A

Params: N/A

Query: N/A

Response:



```
{
      "message": "Logout successfully",
      "status": 202
}
```

Here

- **message** will sent a response message. Every time it will come mandatory
- **status** will return a http status code here it will sent **406** and **201**

Description: If status code is

- **202 =>**
  - User Logged In!!
- **406 =>**
  - Runtime error / Internal Error!!!
- **404 =>**
  - User not logged in please login

Fetcher’s API’S

1. **Get daily Air Quality Index (AQI) of PM2.5 average of a particular division.**
    a. **Need to show Daily Average of PM2.5 value of a particular place**
    b. **Place input will be a option input**

API: baseUrl/airData/get/avg/pm/:division

Type: REST

Method : GET

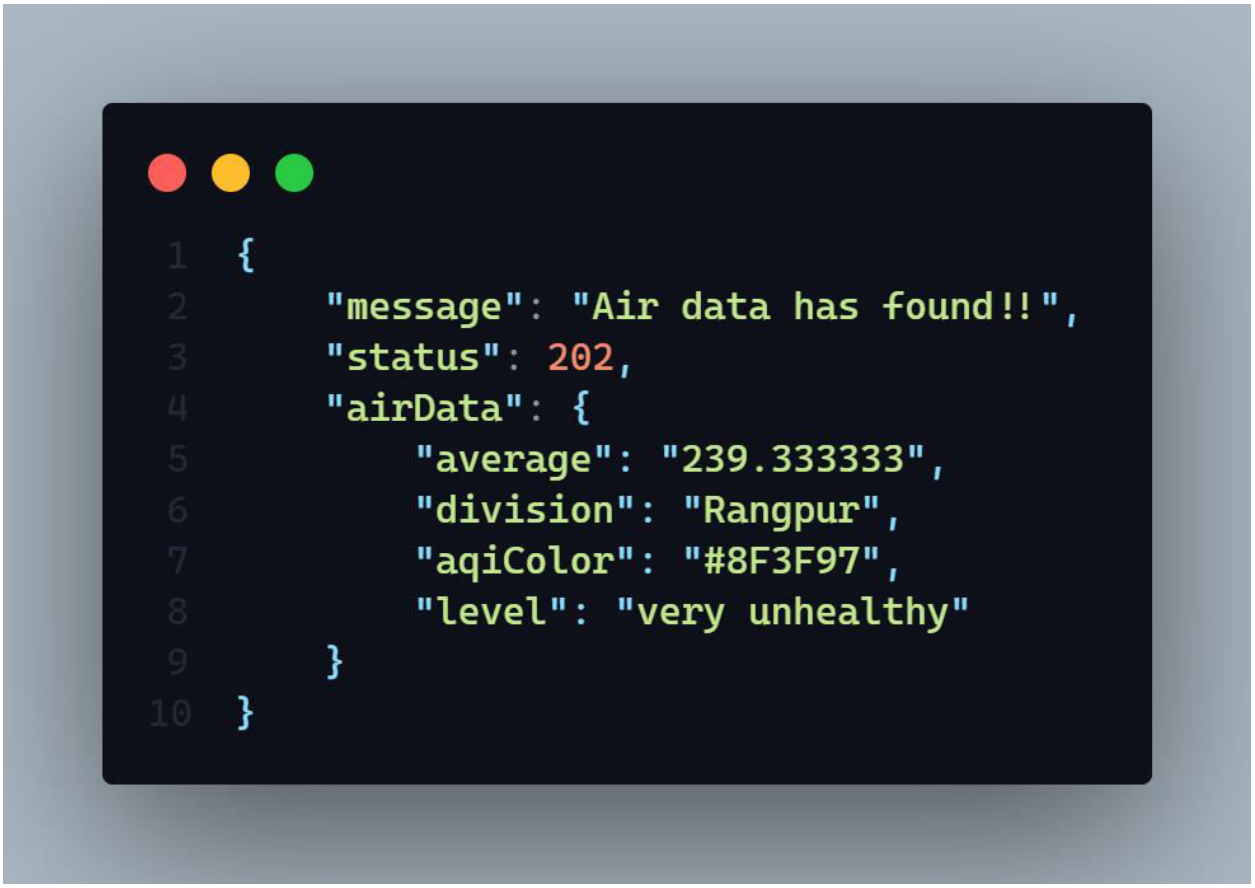
Access: : All

Body : N/A

Params: {division: string}

Query: N/A

Response:



```
{
    "message": "Air data has found!!",
    "status": 202,
    "airData": {
        "average": "239.333333",
        "division": "Rangpur",
        "aqiColor": "#8F3F97",
        "level": "very unhealthy"
    }
}
```

Description: If status code is

- **202 =>**
  - Air Data has been found
- **406 =>**
  - Runtime error / Internal Error!!!
- **404 =>**
  - **No data found**
  - **Division name required**

Requirement: 3

2. Get all available session of data

a. It will give all available session available from the inputted existing data

API: baseUrl/airData/get/available/session

Type: REST

Method : GET

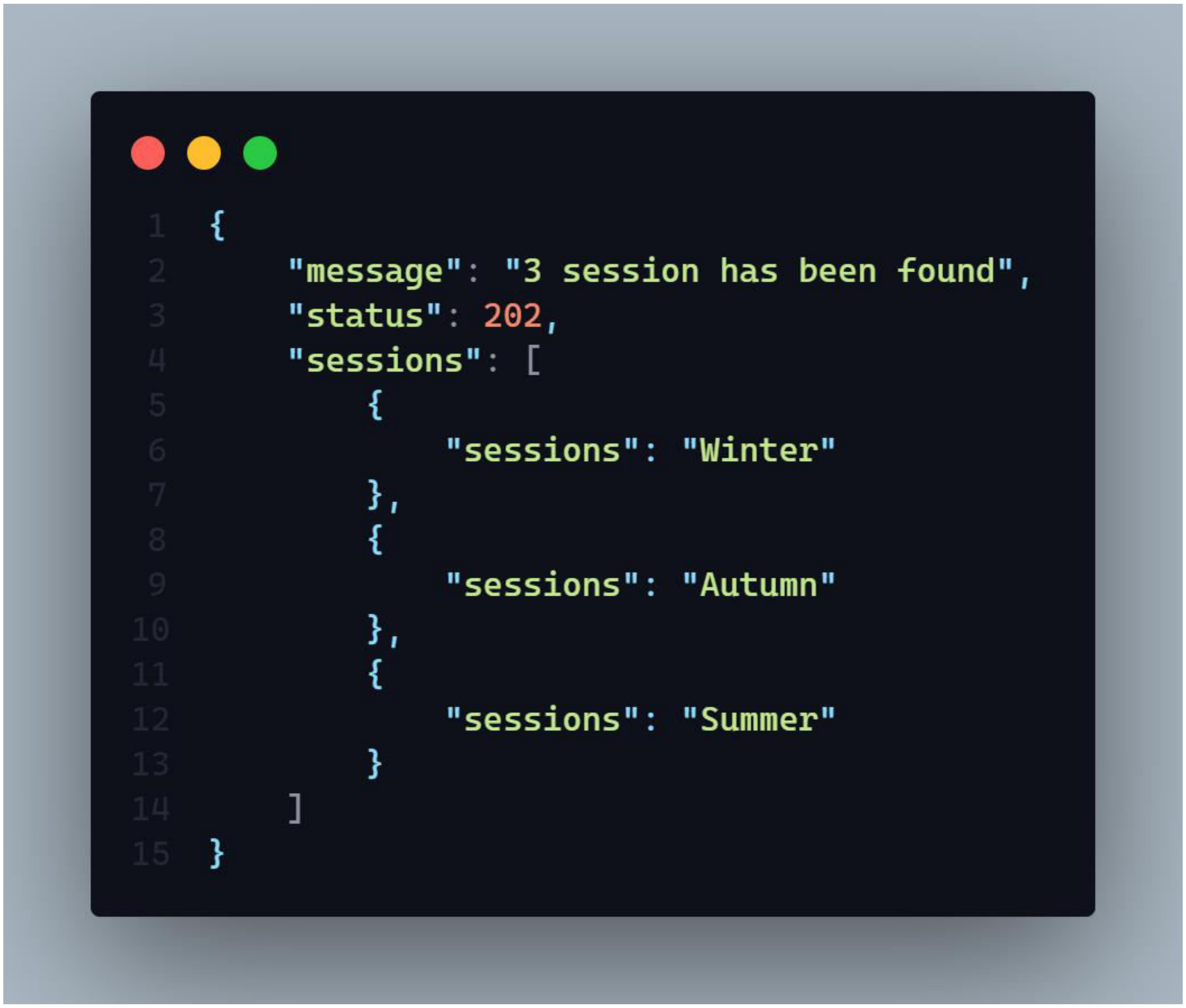
Access: **: All**

Body : N/A

Params: N/A

Query: N/A

Response:



```
{
    "message": "3 session has been found",
    "status": 202,
    "sessions": [
        {
            "sessions": "Winter"
        },
        {
            "sessions": "Autumn"
        },
        {
            "sessions": "Summer"
        }
    ]
}
```

Description: If status code is

- **202 =>**
  - {number} session found
- **406 =>**
  - Runtime error / Internal Error!!!
- **404 =>**
  - **No Session found**

3 . Get Two Agency's Daily basis Average of PM2.5 value by all year and get it like this [agencyOne Average PM2.5, AgencyTwo average PM2.5]

c. **Need to find Average Value of PM2.5 in a daily basis of Each Agency's**

d. **Then show only those value which year client user want to see.**

e. **Year will be a range like**

   i. **Client user want to see All data between 2018 to 2020**

   ii. **Then show all data between these two year range**

API: baseUrl/airData/get/daily/basis/mean/inRange/between/two

Type: REST

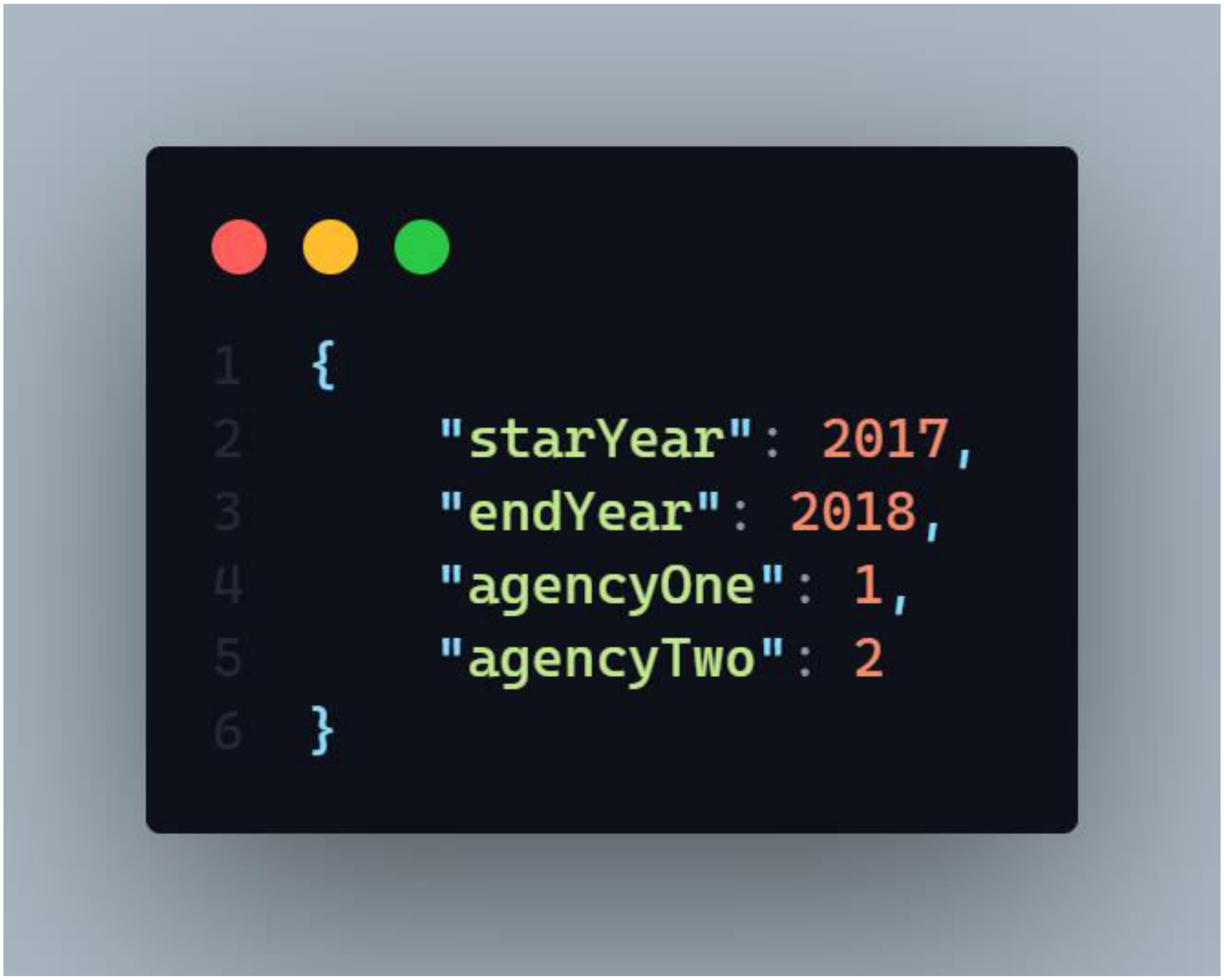
Method : POST

Access: : All

Requirement: 4(b)

(to get 4(a) see number 13 api details)

Body :

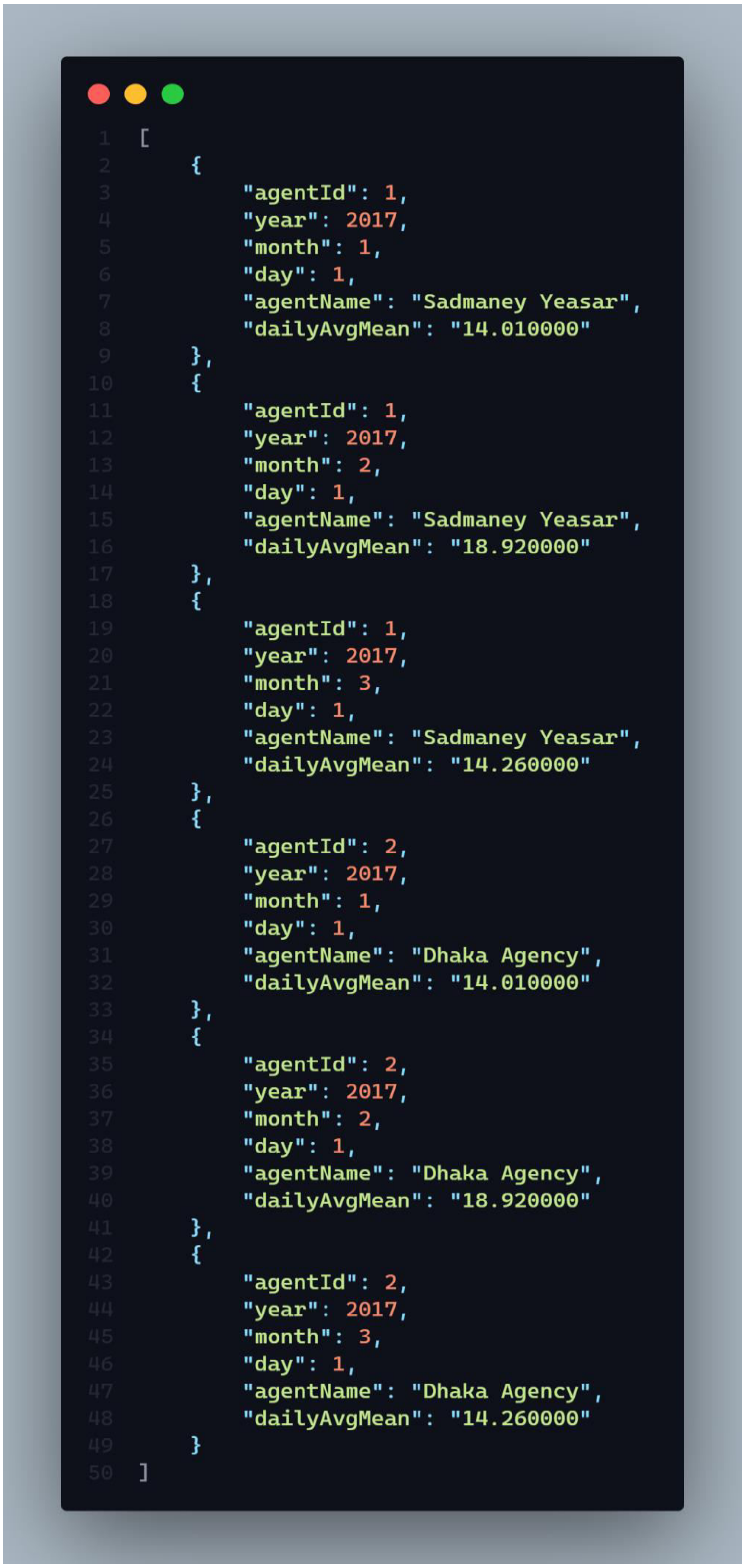


```
{
    "starYear": 2017,
    "endYear": 2018,
    "agencyOne": 1,
    "agencyTwo": 2
}
```

Params: N/A

Query: N/A

Response:

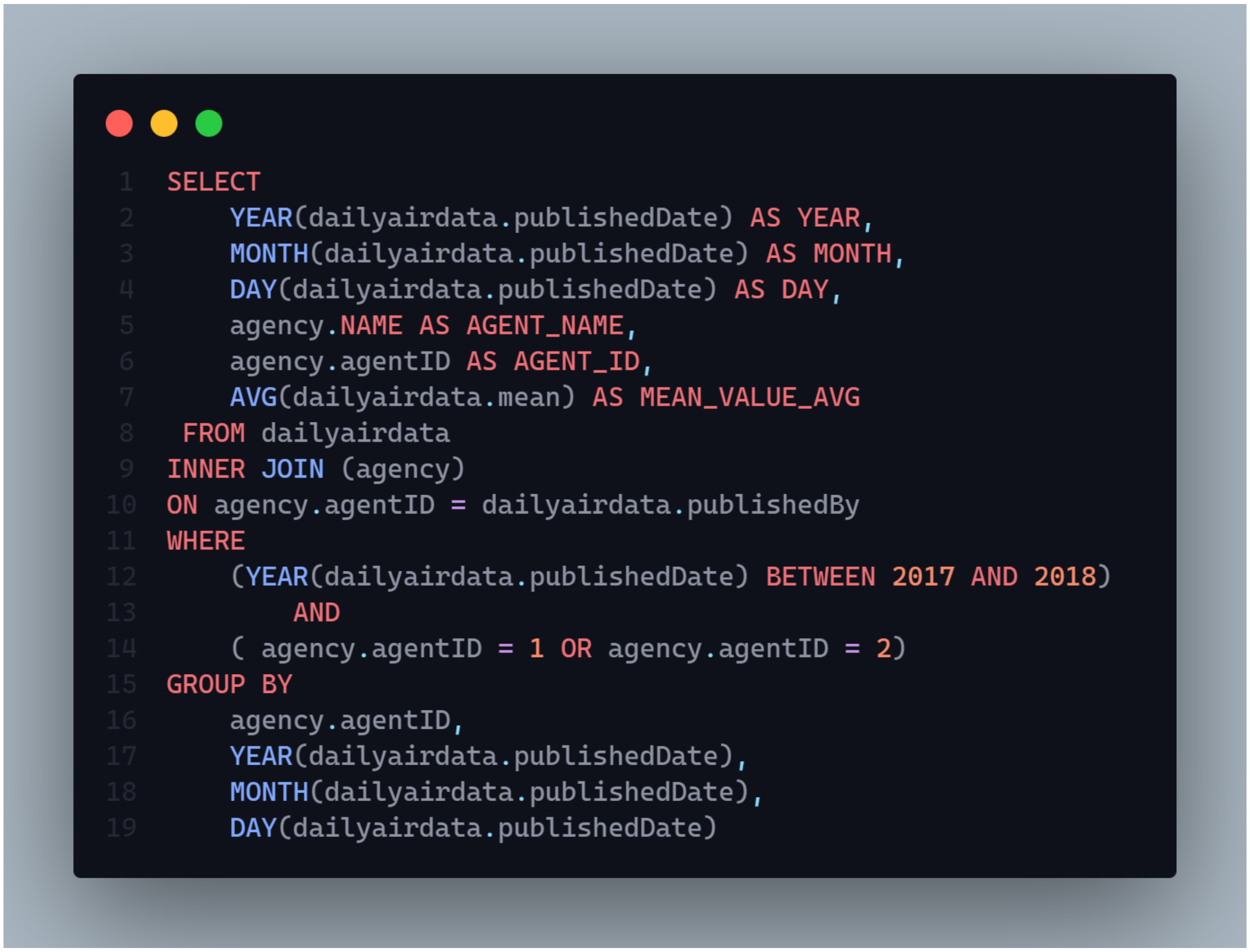


```
[
    {
        "agentId": 1,
        "year": 2017,
        "month": 1,
        "day": 1,
        "agentName": "Sadmaney Yeasar",
        "dailyAvgMean": "14.010000"
    },
    {
        "agentId": 1,
        "year": 2017,
        "month": 2,
        "day": 1,
        "agentName": "Sadmaney Yeasar",
        "dailyAvgMean": "18.920000"
    },
    {
        "agentId": 1,
        "year": 2017,
        "month": 3,
        "day": 1,
        "agentName": "Sadmaney Yeasar",
        "dailyAvgMean": "14.260000"
    },
    {
        "agentId": 2,
        "year": 2017,
        "month": 1,
        "day": 1,
        "agentName": "Dhaka Agency",
        "dailyAvgMean": "14.010000"
    },
    {
        "agentId": 2,
        "year": 2017,
        "month": 2,
        "day": 1,
        "agentName": "Dhaka Agency",
        "dailyAvgMean": "18.920000"
    },
    {
        "agentId": 2,
        "year": 2017,
        "month": 3,
        "day": 1,
        "agentName": "Dhaka Agency",
        "dailyAvgMean": "14.260000"
    }
]
```

Description: If status code is

- **202 =>**
  - Air data found!!!
- **406 =>**
  - Runtime error / Internal Error!!!
  - Start year is behind
- **404 =>**
  - **No air data has been found**
  - **A year range has been missed**

Database Query



```sql
SELECT
    YEAR(dailyairdata.publishedDate) AS YEAR,
    MONTH(dailyairdata.publishedDate) AS MONTH,
    DAY(dailyairdata.publishedDate) AS DAY,
    agency.NAME AS AGENT_NAME,
    agency.agentID AS AGENT_ID,
    AVG(dailyairdata.mean) AS MEAN_VALUE_AVG
 FROM dailyairdata
INNER JOIN (agency)
ON agency.agentID = dailyairdata.publishedBy
WHERE
    (YEAR(dailyairdata.publishedDate) BETWEEN 2017 AND 2018)
        AND
    ( agency.agentID = 1 OR agency.agentID = 2)
GROUP BY
    agency.agentID,
    YEAR(dailyairdata.publishedDate),
    MONTH(dailyairdata.publishedDate),
    DAY(dailyairdata.publishedDate)
```

Chart Link: https://plotly.com/javascript/line-and-scatter/

Type: Scatter + line

4 . Get Get all available Year From the existing data input

API: baseUrl/airData/get/available/published/year/:queryFor

Type: REST

Method : GET

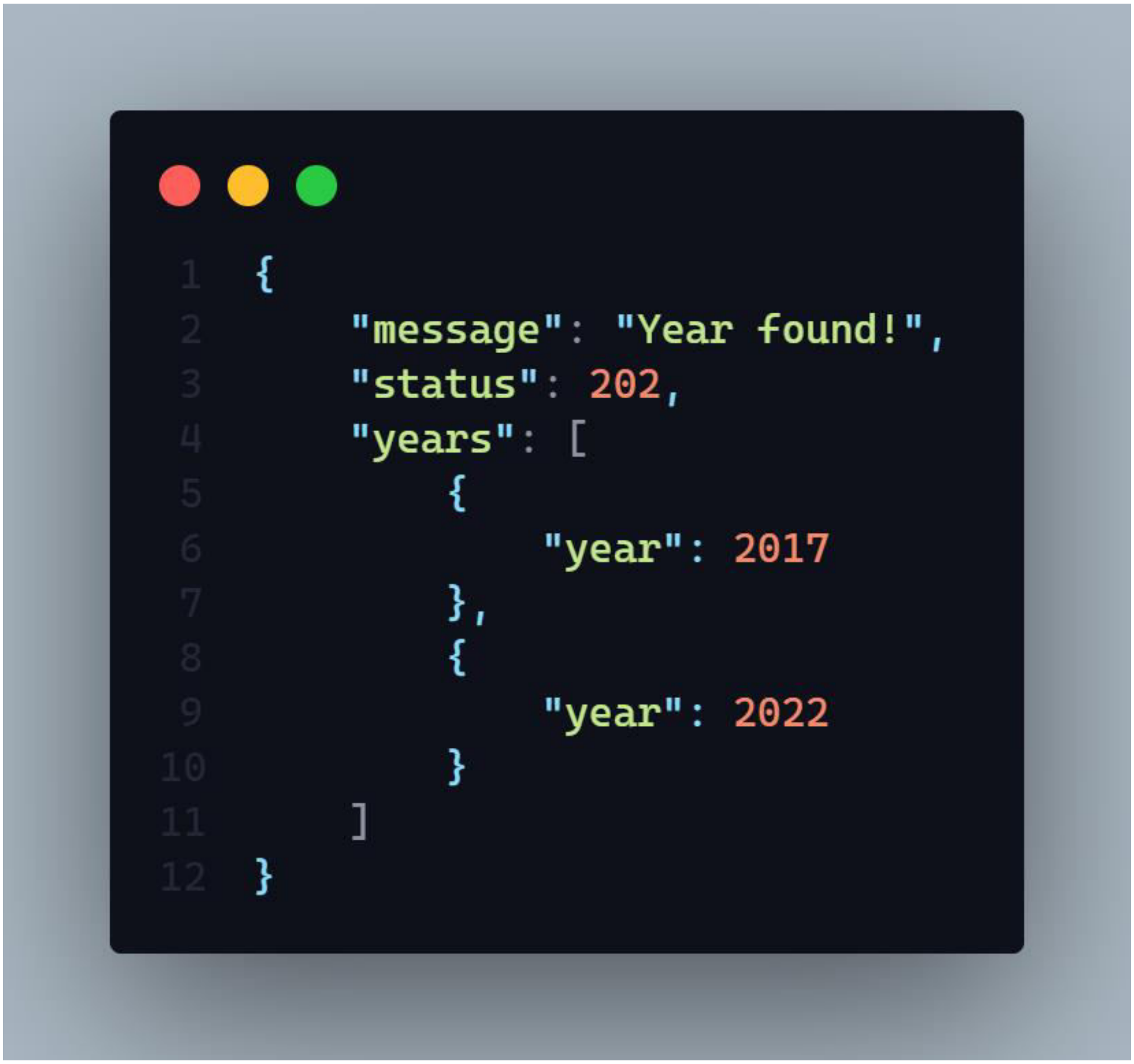
Access: **: All**

Body :

Params: {queryFor} // it should be “final” or “daily”

Query: N/A

Response:



```
{
    "message": "Year found!",
    "status": 202,
    "years": [
        {
            "year": 2017
        },
        {
            "year": 2022
        }
    ]
}
```

Description: If status code is

- **202 =>**
    - Year found!!!
- **406 =>**
    - Runtime error / Internal Error!!!
- **404 =>**
    - **No year has been found**

5 . Get all available agency name

API: baseUrl/airData/get/available/agency/:queryFor

Type: REST

Method : GET

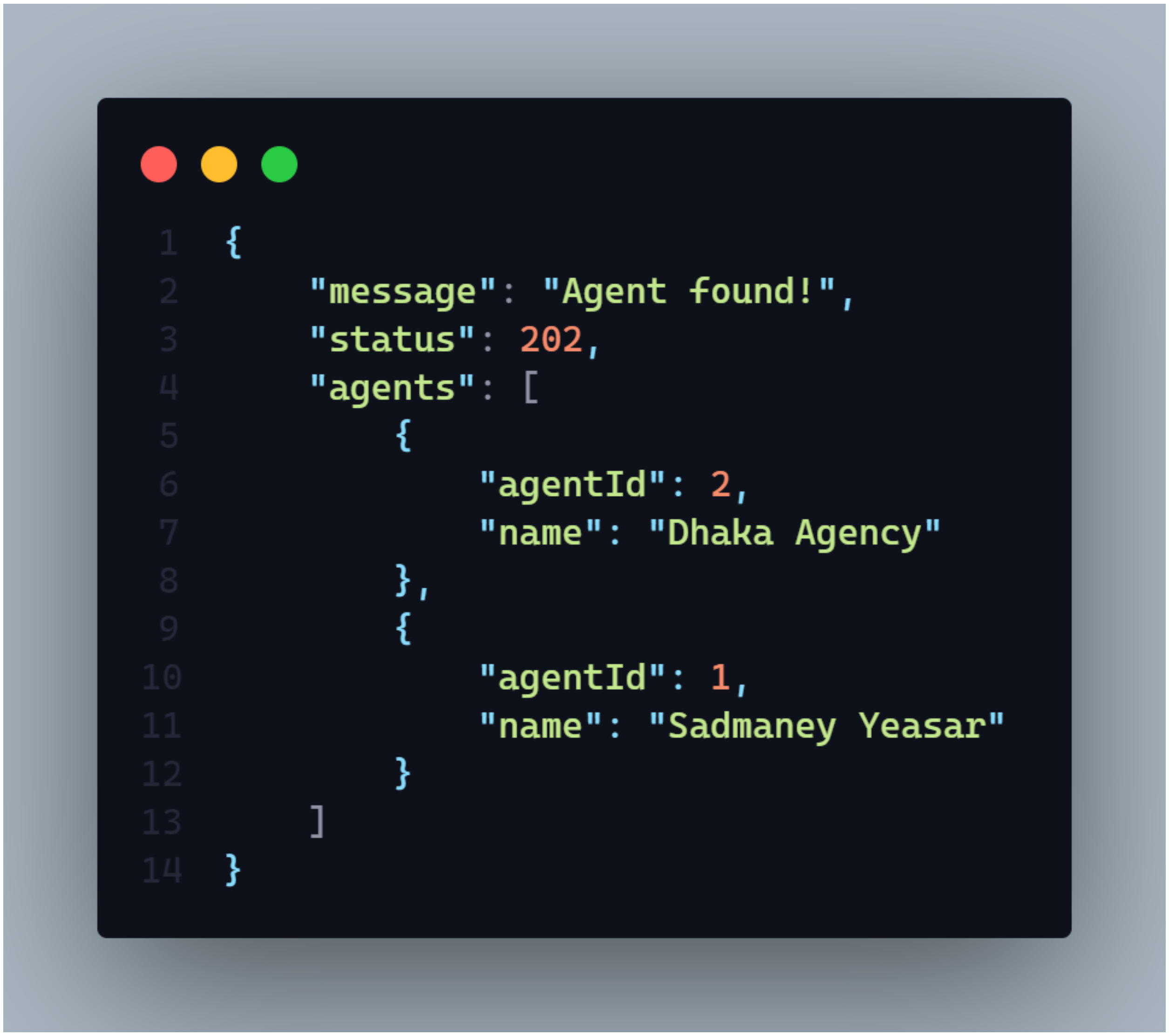
Access: : All

Body :

Params: {queryFor} // it should be “final” or “daily”

Query: N/A

Response:



```
{
    "message": "Agent found!",
    "status": 202,
    "agents": [
        {
            "agentId": 2,
            "name": "Dhaka Agency"
        },
        {
            "agentId": 1,
            "name": "Sadmaney Yeasar"
        }
    ]
}
```

Description: If status code is

- **202 =>**
  - Agency found!!!
- **406 =>**
  - Runtime error / Internal Error!!!
- **404 =>**
  - **No Agency has been found**

6 . Get Get all available Year From the existing data input

API: baseUrl/airData/get/available/published/year/:queryFor

Type: REST

Method : GET

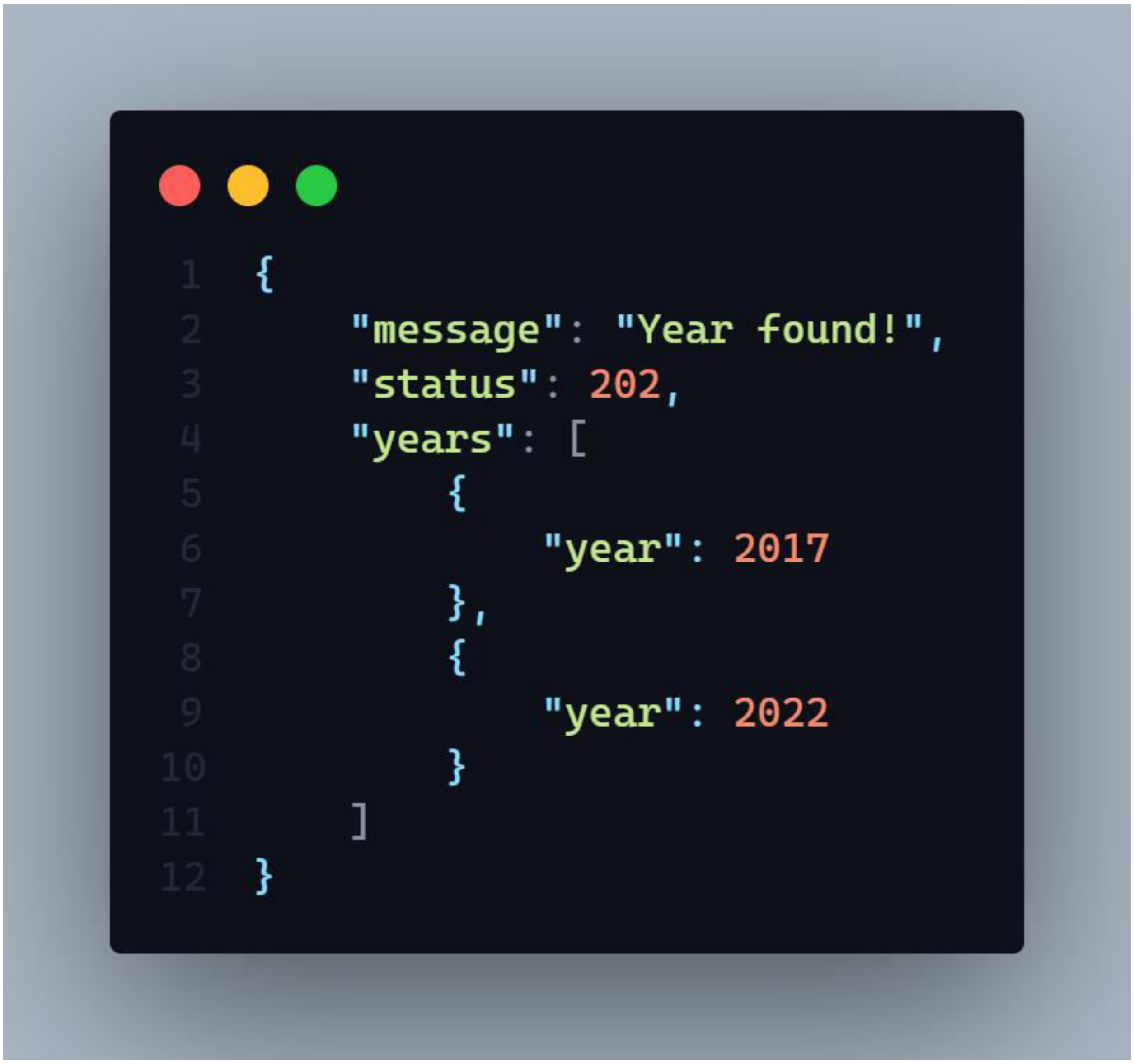
Access: **: All**

Body :

Params: {queryFor} // it should be “final” or “daily”

Query: N/A

Response:



```
{
    "message": "Year found!",
    "status": 202,
    "years": [
        {
            "year": 2017
        },
        {
            "year": 2022
        }
    ]
}
```

Description: If status code is

- **202 =>**
  - Year found!!!
- **406 =>**
  - Runtime error / Internal Error!!!
- **404 =>**
  - **No year has been found**

7. Get all available agency name

API: baseUrl/airData/get/available/agency/:queryFor

Type: REST

Method : GET

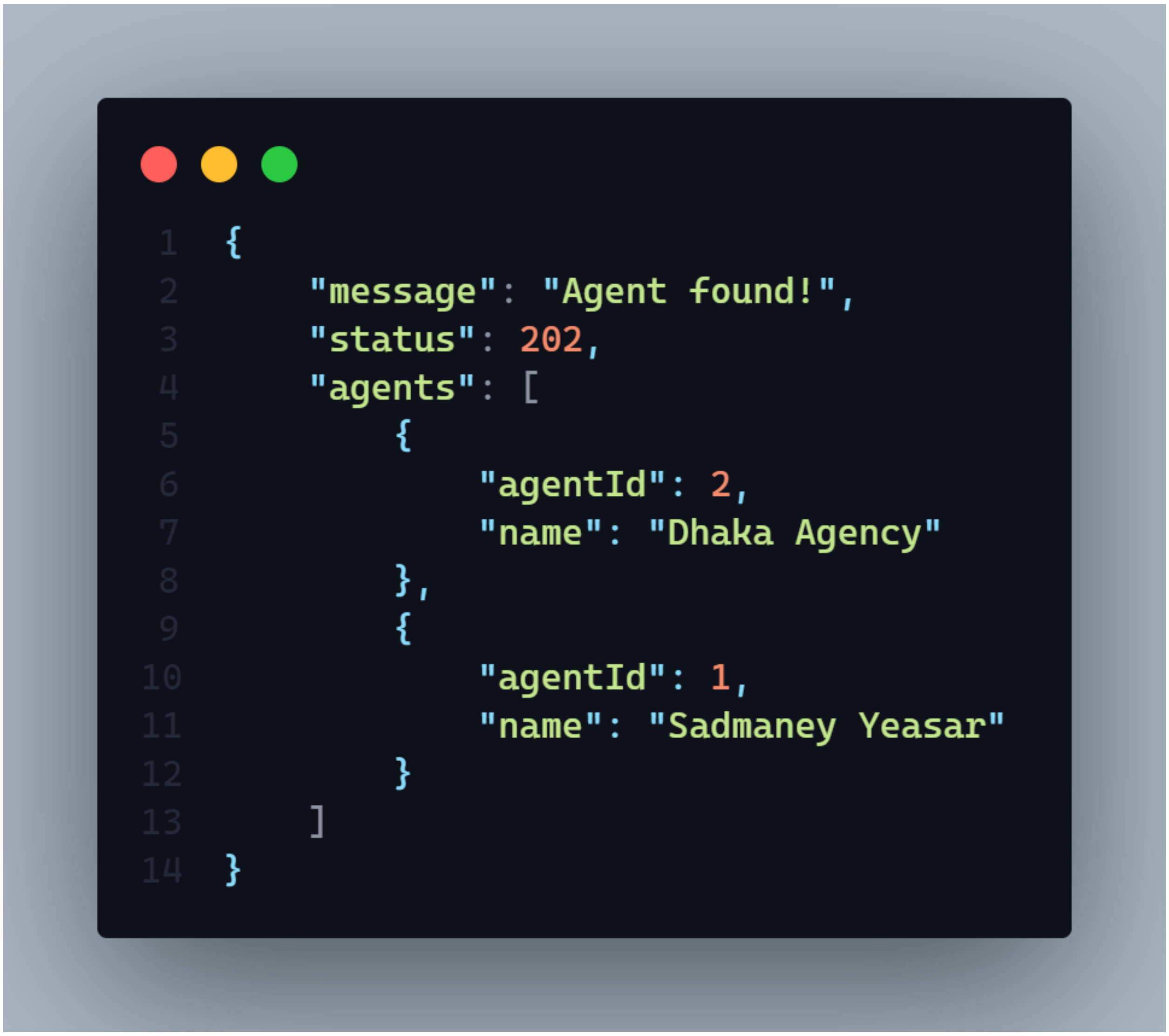
Access: : All

Body :

Params: {queryFor} // it should be “final” or “daily”

Query: N/A

Response:



```
{
    "message": "Agent found!",
    "status": 202,
    "agents": [
        {
            "agentId": 2,
            "name": "Dhaka Agency"
        },
        {
            "agentId": 1,
            "name": "Sadmaney Yeasar"
        }
    ]
}
```

Description: If status code is

- **202 =>**
  - Agency found!!!
- **406 =>**
  - Runtime error / Internal Error!!!
- **404 =>**
  - **No Agency has been found**

8. **Get Show AVG AQI of PM2.5 value of all division in many query like**

   f. **If client want to see Yearly data of All division then**
      i. **A range between two year will be passed**
      ii. **Then give all divisions AVG AQI data of between that two Year range.**
      iii. **In The Y axis PM 2.5 value will be show**
      iv. **In the X axis Year name will be show**

   g. **If Client want to see Monthly data of all division the**
      i. **Client need to mention the year name.**
      ii. **Then It will show that years all month's Data**
      iii. **In the Y axis PM2.5 value will be show**
      iv. **In the X axis Month Name will be show**

API: baseUrl/airData/airData/get/aqi/all

Type: REST

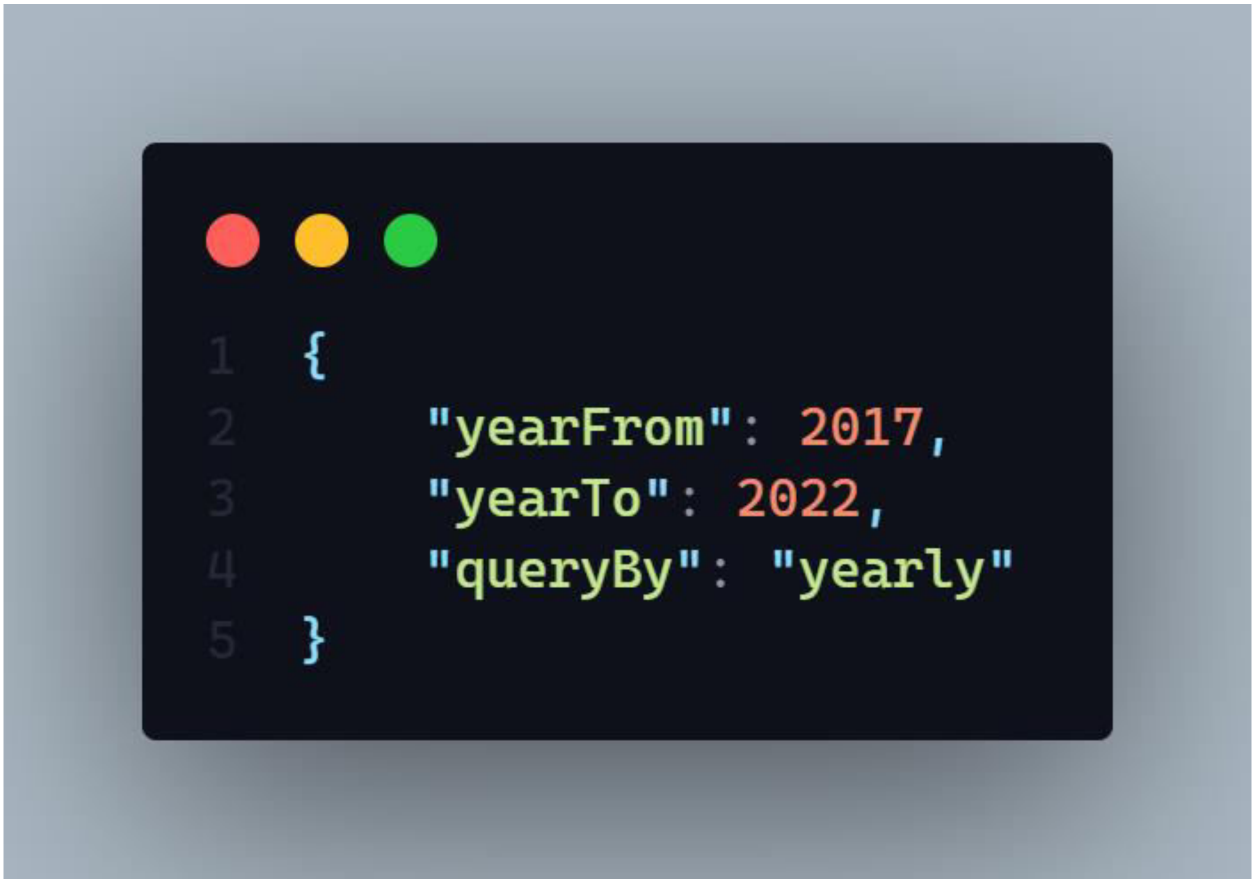
Requirement: 6

Method : POST

Access: **: All**

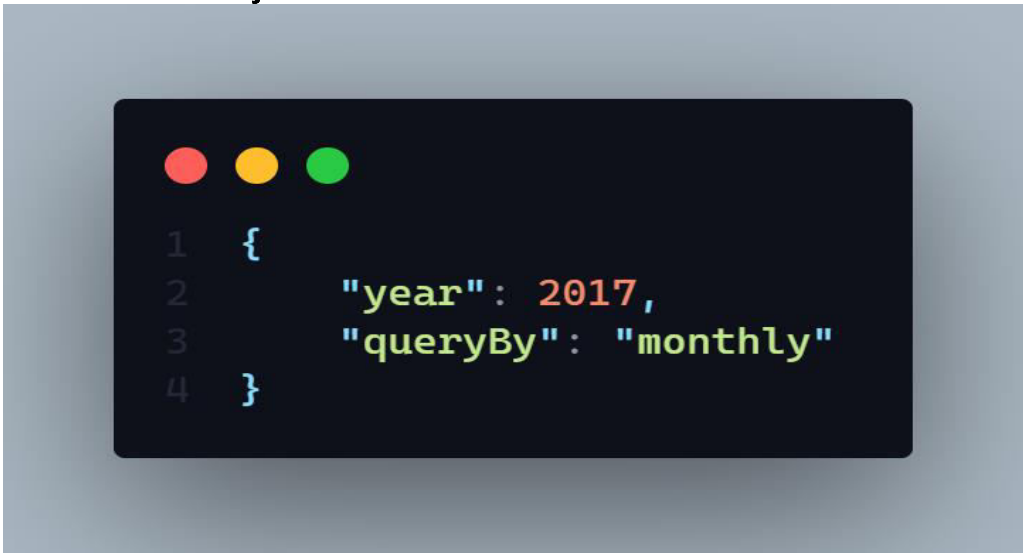
Body :

For Yearly data



```
{
      "yearFrom": 2017,
      "yearTo": 2022,
      "queryBy": "yearly"
}
```

For Monthly :



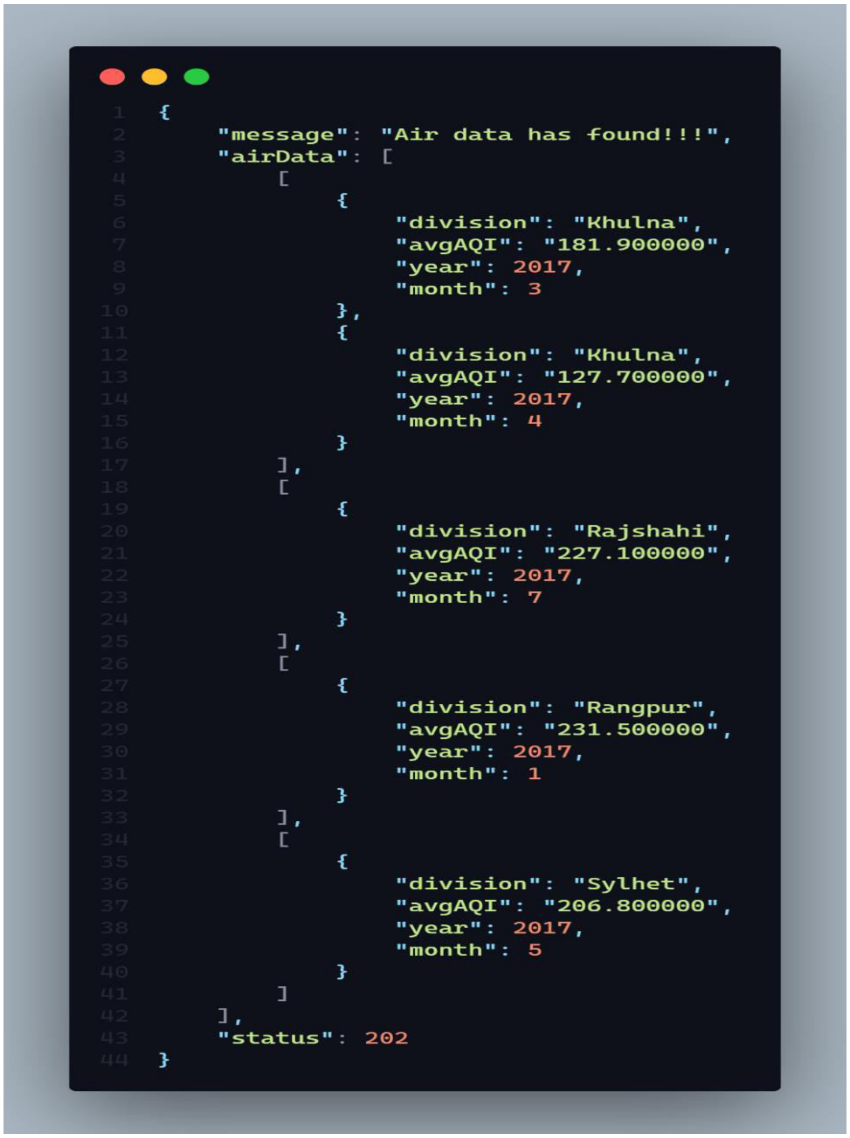
```
{
      "year": 2017,
      "queryBy": "monthly"
}
```

Params: N/A

Query: N/A

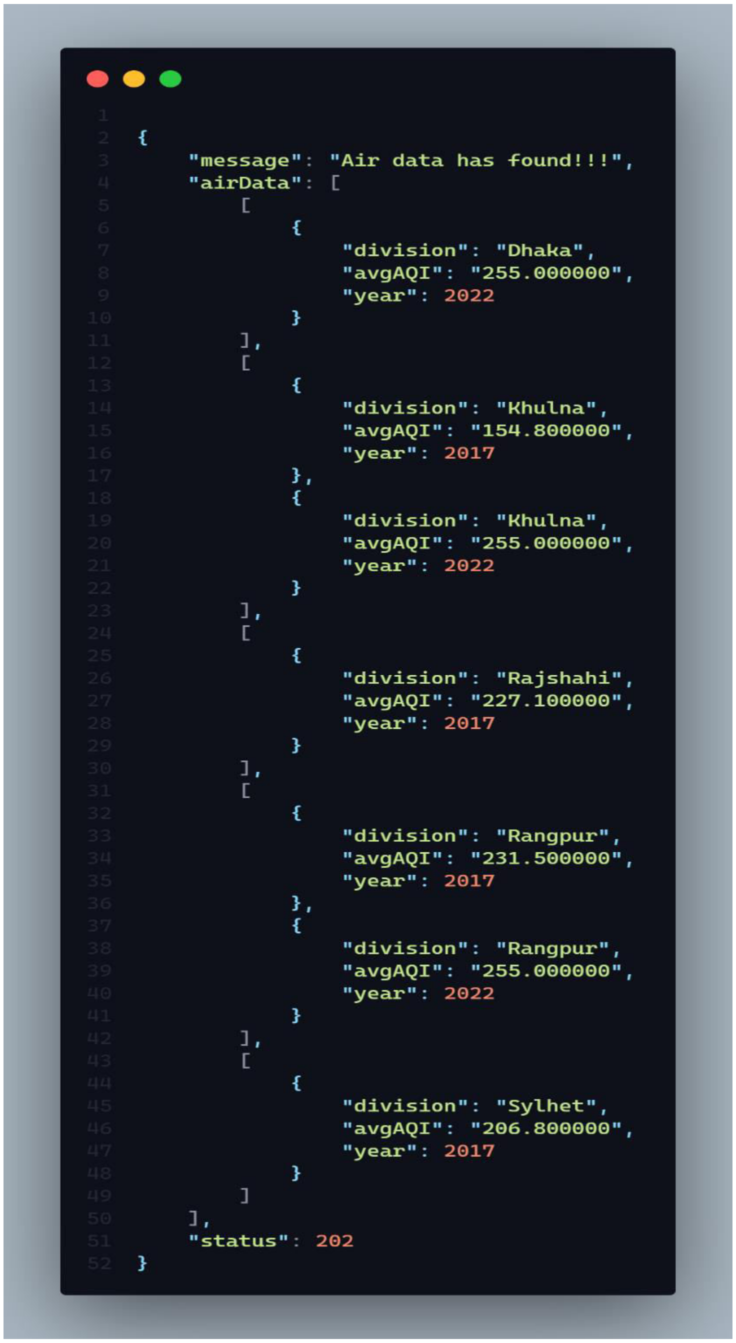
Response:

For Monthly :



```
{
    "message": "Air data has found!!!",
    "airData": [
        [
            {
                "division": "Khulna",
                "avgAQI": "181.900000",
                "year": 2017,
                "month": 3
            },
            {
                "division": "Khulna",
                "avgAQI": "127.700000",
                "year": 2017,
                "month": 4
            }
        ],
        [
            {
                "division": "Rajshahi",
                "avgAQI": "227.100000",
                "year": 2017,
                "month": 7
            }
        ],
        [
            {
                "division": "Rangpur",
                "avgAQI": "231.500000",
                "year": 2017,
                "month": 1
            }
        ],
        [
            {
                "division": "Sylhet",
                "avgAQI": "206.800000",
                "year": 2017,
                "month": 5
            }
        ]
    ],
    "status": 202
}
```

For Yearly:



```
{
    "message": "Air data has found!!!",
    "airData": [
        [
            {
                "division": "Dhaka",
                "avgAQI": "255.000000",
                "year": 2022
            }
        ],
        [
            {
                "division": "Khulna",
                "avgAQI": "154.800000",
                "year": 2017
            },
            {
                "division": "Khulna",
                "avgAQI": "255.000000",
                "year": 2022
            }
        ],
        [
            {
                "division": "Rajshahi",
                "avgAQI": "227.100000",
                "year": 2017
            }
        ],
        [
            {
                "division": "Rangpur",
                "avgAQI": "231.500000",
                "year": 2017
            },
            {
                "division": "Rangpur",
                "avgAQI": "255.000000",
                "year": 2022
            }
        ],
        [
            {
                "division": "Sylhet",
                "avgAQI": "206.800000",
                "year": 2017
            }
        ]
    ],
    "status": 202
}
```

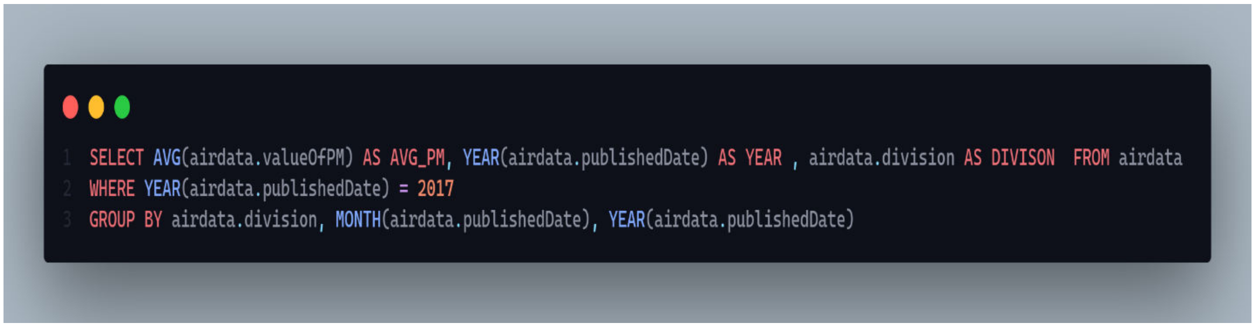
Description: If status code is

- **202 =>**
  - Air data found!!!
- **406 =>**
  - Runtime error / Internal Error!!!
- **404 =>**
  - **No Air data has been found**

Graph type; line Chart: https://plotly.com/javascript/line-charts/

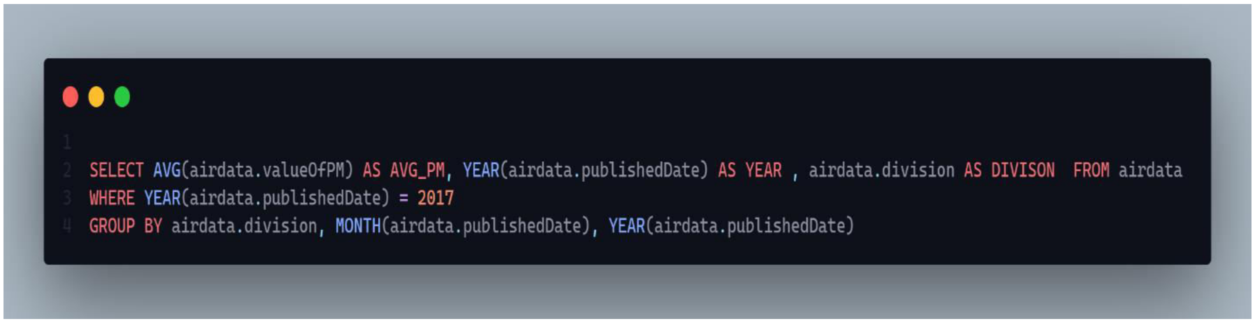
Data base query

For Yearly



```sql
SELECT AVG(airdata.valueOfPM) AS AVG_PM, YEAR(airdata.publishedDate) AS YEAR , airdata.division AS DIVISON  FROM airdata
WHERE YEAR(airdata.publishedDate) = 2017
GROUP BY airdata.division, MONTH(airdata.publishedDate), YEAR(airdata.publishedDate)
```

For Monthly



```sql

SELECT AVG(airdata.valueOfPM) AS AVG_PM, YEAR(airdata.publishedDate) AS YEAR , airdata.division AS DIVISON  FROM airdata
WHERE YEAR(airdata.publishedDate) = 2017
GROUP BY airdata.division, MONTH(airdata.publishedDate), YEAR(airdata.publishedDate)
```

9. Get Show Daily AVG AQI of PM2.5 value of all Division

h. **In the X axis Division Name**
i. **In the Y axis contains PM2.5 value**

API: baseUrl/airData/get/aqi/all/division

Type: REST

Requirement: 7

Method : GET

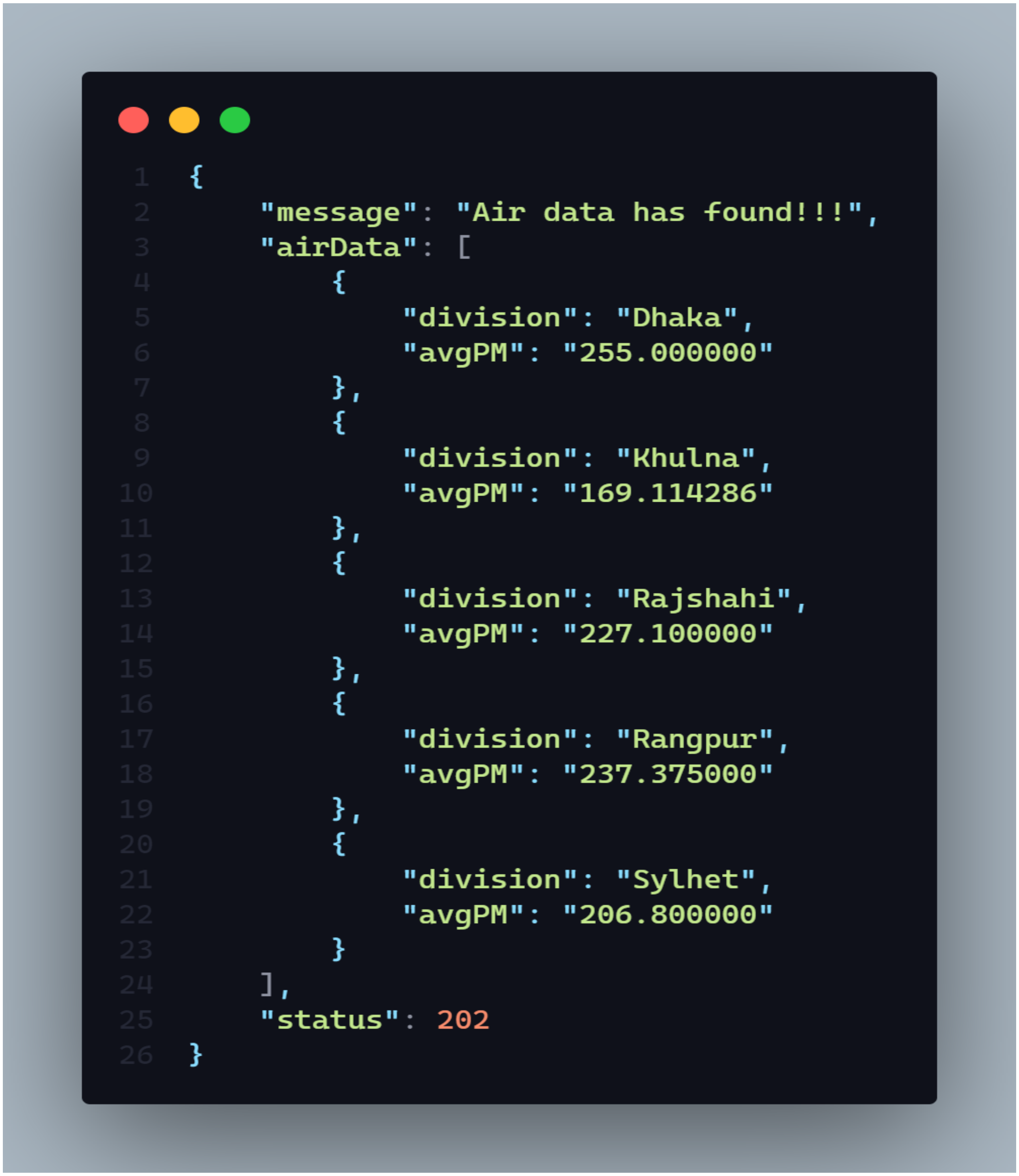
Access: **: All**

Body : N/A

Params: N/A

Query: N/A

Response:



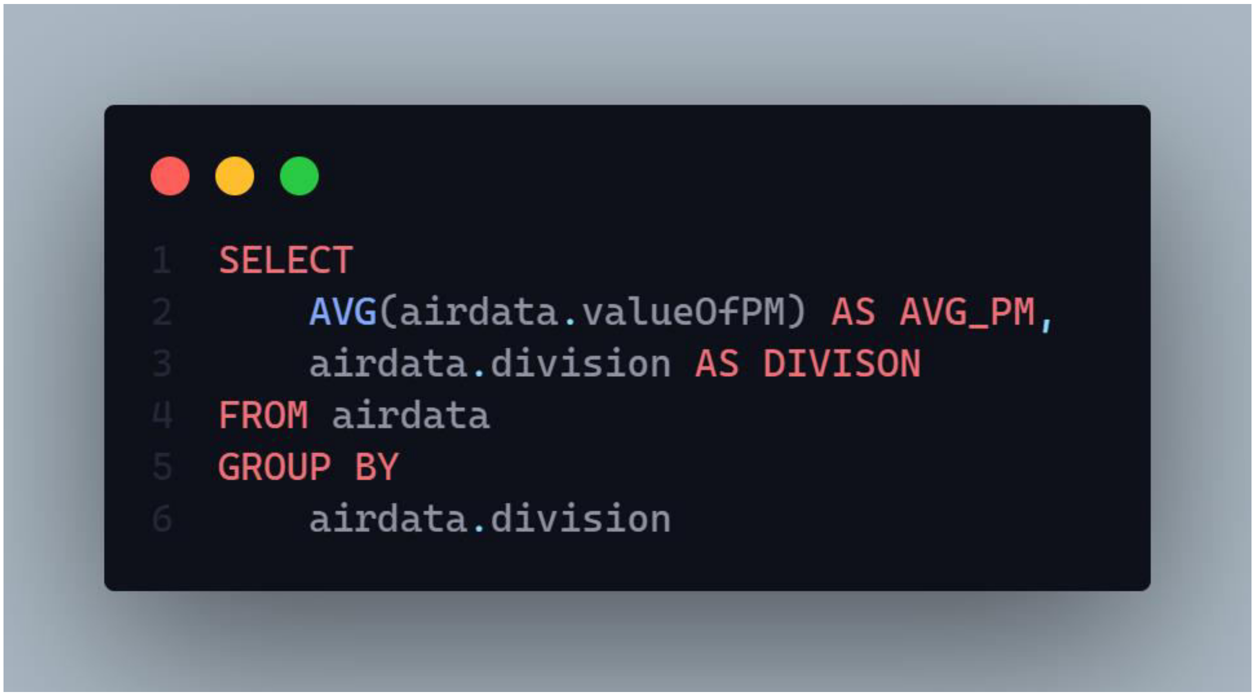
```
{
    "message": "Air data has found!!!",
    "airData": [
        {
            "division": "Dhaka",
            "avgPM": "255.000000"
        },
        {
            "division": "Khulna",
            "avgPM": "169.114286"
        },
        {
            "division": "Rajshahi",
            "avgPM": "227.100000"
        },
        {
            "division": "Rangpur",
            "avgPM": "237.375000"
        },
        {
            "division": "Sylhet",
            "avgPM": "206.800000"
        }
    ],
    "status": 202
}
```

Description: If status code is

- **202 =>**
  - Air data found!!!
- **406 =>**
  - Runtime error / Internal Error!!!
- **404 =>**
  - **No Air data has been found**

Graph type; line Chart: https://plotly.com/javascript/line-charts/

Data base query



```
SELECT
    AVG(airdata.valueOfPM) AS AVG_PM,
    airdata.division AS DIVISON
FROM airdata
GROUP BY
    airdata.division
```

10. Show Month Wise AVG AQI of PM2.5 value of a particular Station

j. In the Y axis contains PM 2.5 value
k. In the X axis contains Month number
l. Client need to give a Input of Year
m. With Agency Name and Station number

API: baseUrl/airData/ get/aqi/of/station/monthly

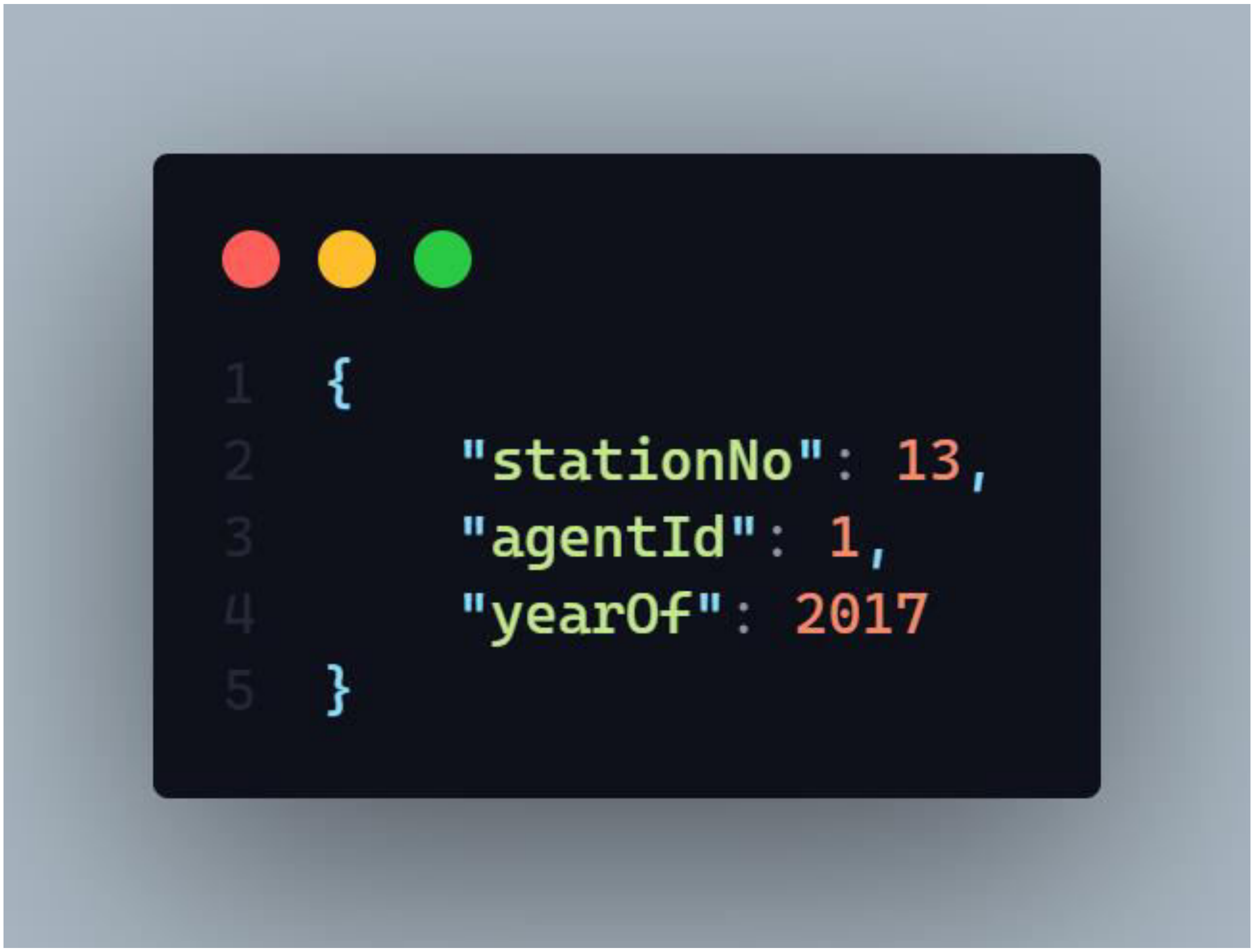
Type: REST

Method : Post

Access: : All

Requirement: 8(b)

Body :

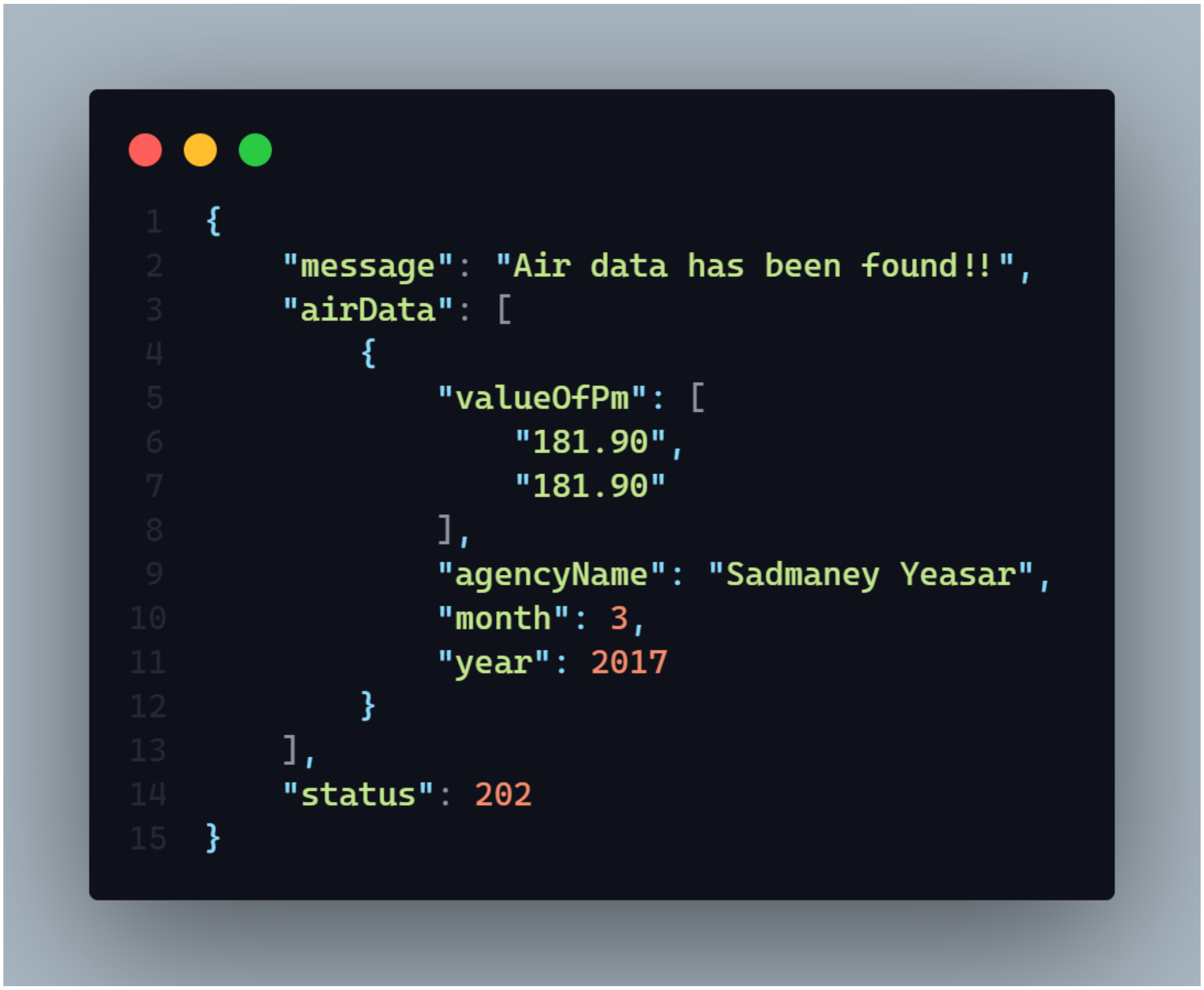


```
{
        "stationNo": 13,
        "agentId": 1,
        "yearOf": 2017
}
```

Params: N/A

Query: N/A

Response:



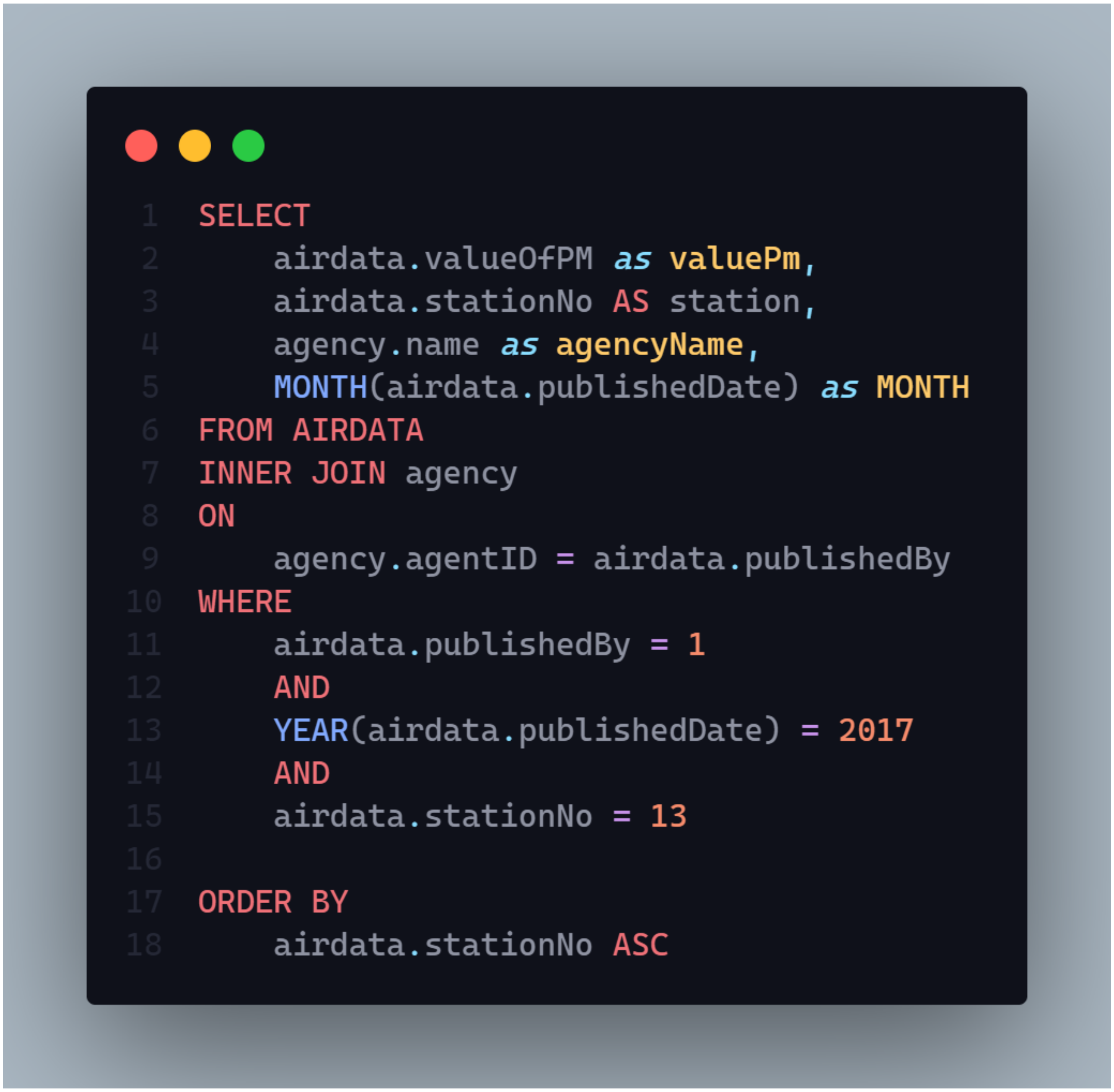
```json
{
    "message": "Air data has been found!!",
    "airData": [
        {
            "valueOfPm": [
                "181.90",
                "181.90"
            ],
            "agencyName": "Sadmaney Yeasar",
            "month": 3,
            "year": 2017
        }
    ],
    "status": 202
}
```

Description: If status code is

- **202 =>**
  - Air data found!!!
- **406 =>**
  - Runtime error / Internal Error!!!
- **404 =>**
  - **No Air data has been found**

Graph type; Box plot: https://plotly.com/javascript/box-plots/

Data base query



```sql
SELECT
    airdata.valueOfPM as valuePm,
    airdata.stationNo AS station,
    agency.name as agencyName,
    MONTH(airdata.publishedDate) as MONTH
FROM AIRDATA
INNER JOIN agency
ON
    agency.agentID = airdata.publishedBy
WHERE
    airdata.publishedBy = 1
    AND
    YEAR(airdata.publishedDate) = 2017
    AND
    airdata.stationNo = 13

ORDER BY
    airdata.stationNo ASC
```

11. Get All station number of by agency Id

n. Station Number

API: baseUrl/airData/get/all/station/of/1 **// id should be agency id**

Type: REST

Method : GET

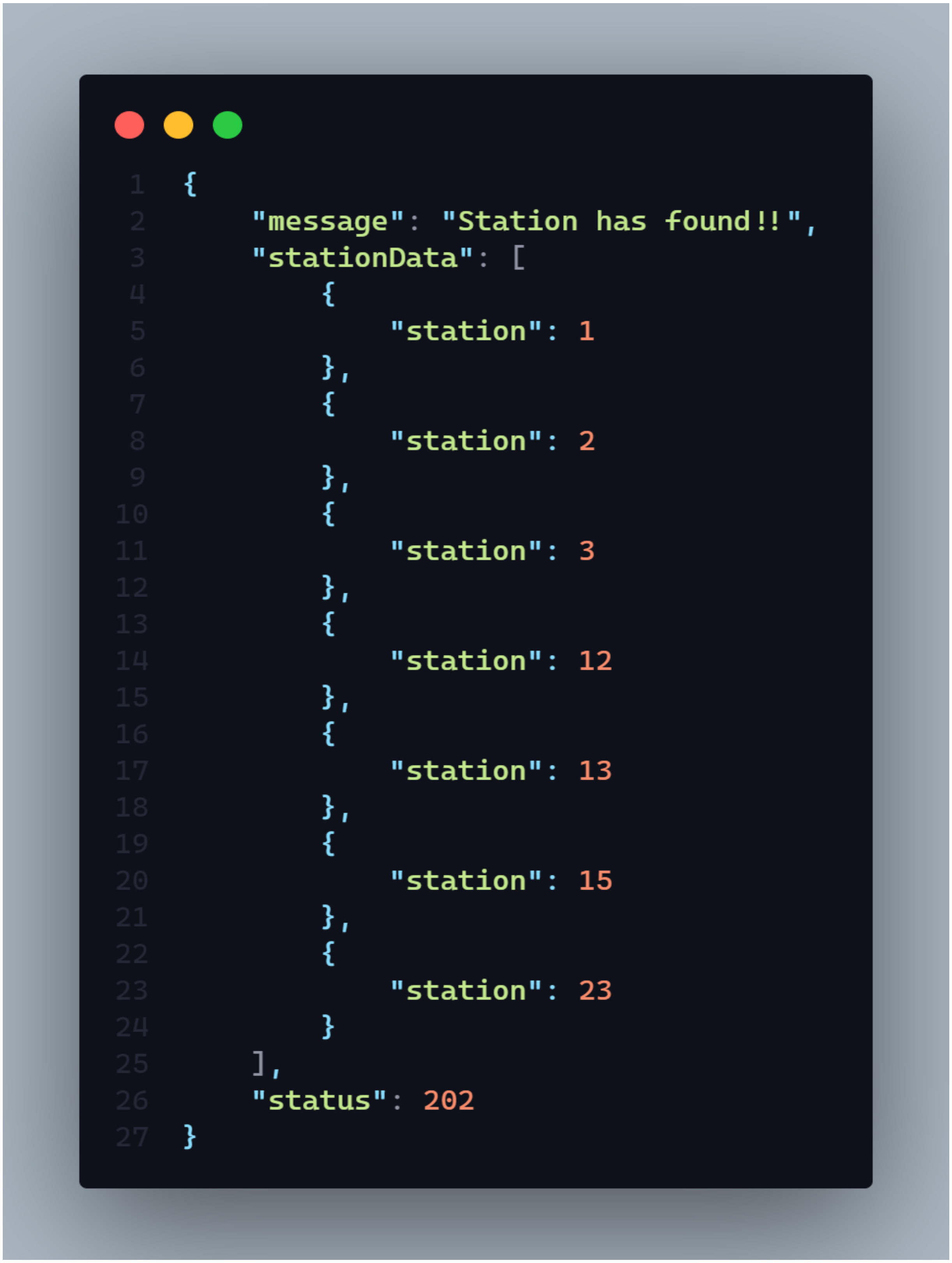
Access: **: All**

Body : N/A

Params: {id:string} // it should be agency id

Query: N/A

Response:

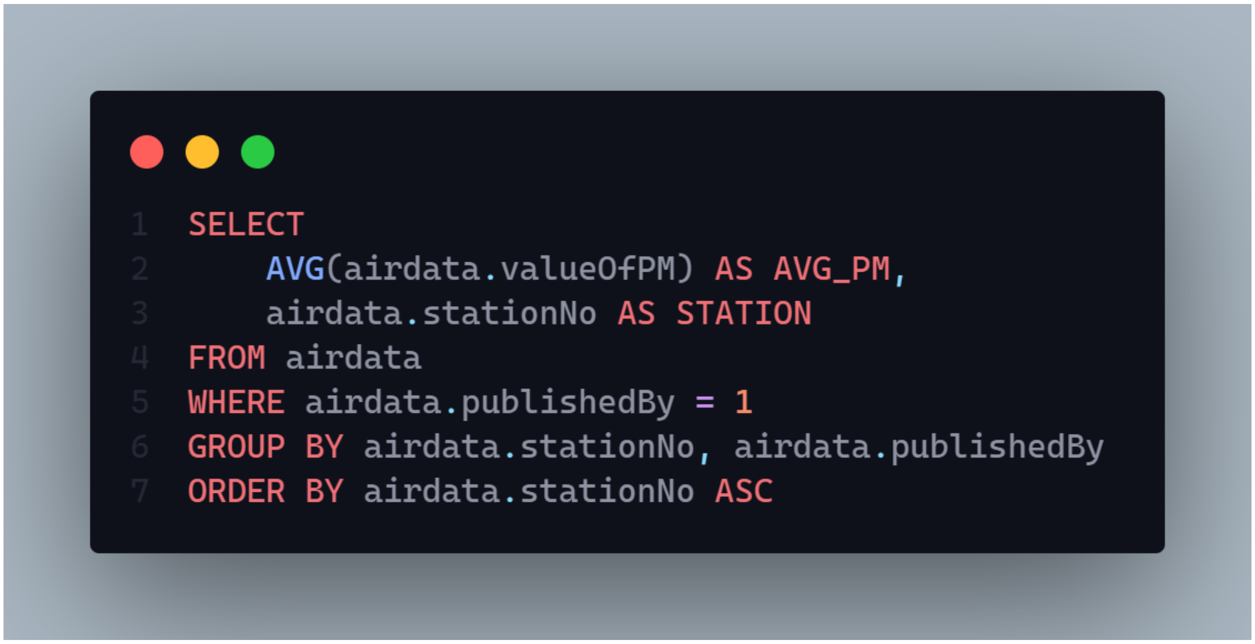


```json
{
    "message": "Station has found!!",
    "stationData": [
        {
            "station": 1
        },
        {
            "station": 2
        },
        {
            "station": 3
        },
        {
            "station": 12
        },
        {
            "station": 13
        },
        {
            "station": 15
        },
        {
            "station": 23
        }
    ],
    "status": 202
}
```

Description: If status code is

- **202 =>**
  - Station data found!!!
- **406 =>**
  - Runtime error / Internal Error!!!
- **404 =>**
  - **No Station data has been found**

Data base query



```
SELECT
    AVG(airdata.valueOfPM) AS AVG_PM,
    airdata.stationNo AS STATION
FROM airdata
WHERE airdata.publishedBy = 1
GROUP BY airdata.stationNo, airdata.publishedBy
ORDER BY airdata.stationNo ASC
```

12. Show Station wise AVG AQI of PM2.5 value of a particular Agency BY AGENCY ID

a. In the Y axis contains Pm2.5 value
b. In the X axis contains Station number

API: baseUrl/airData/ get/aqi/all/station/of/agency

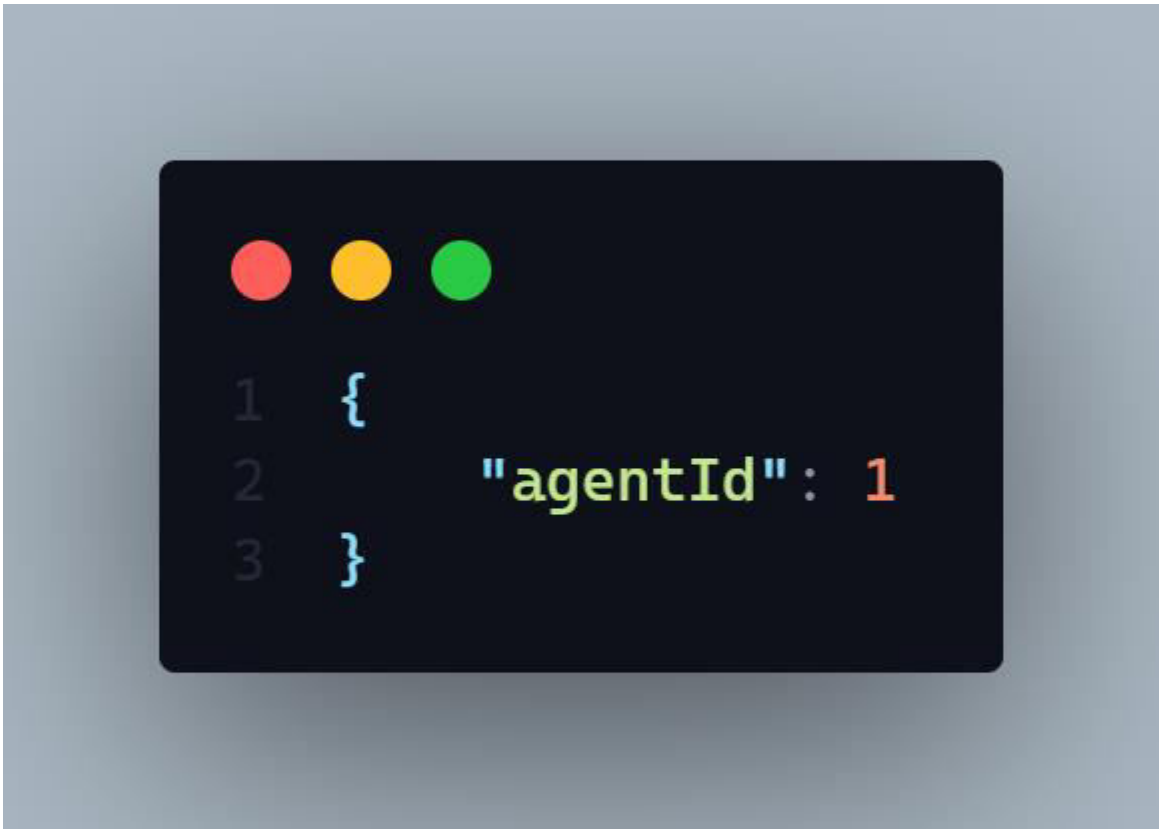
Type: REST

Method : POST

Access: **: All**

Requirement: 8(a)

Body :

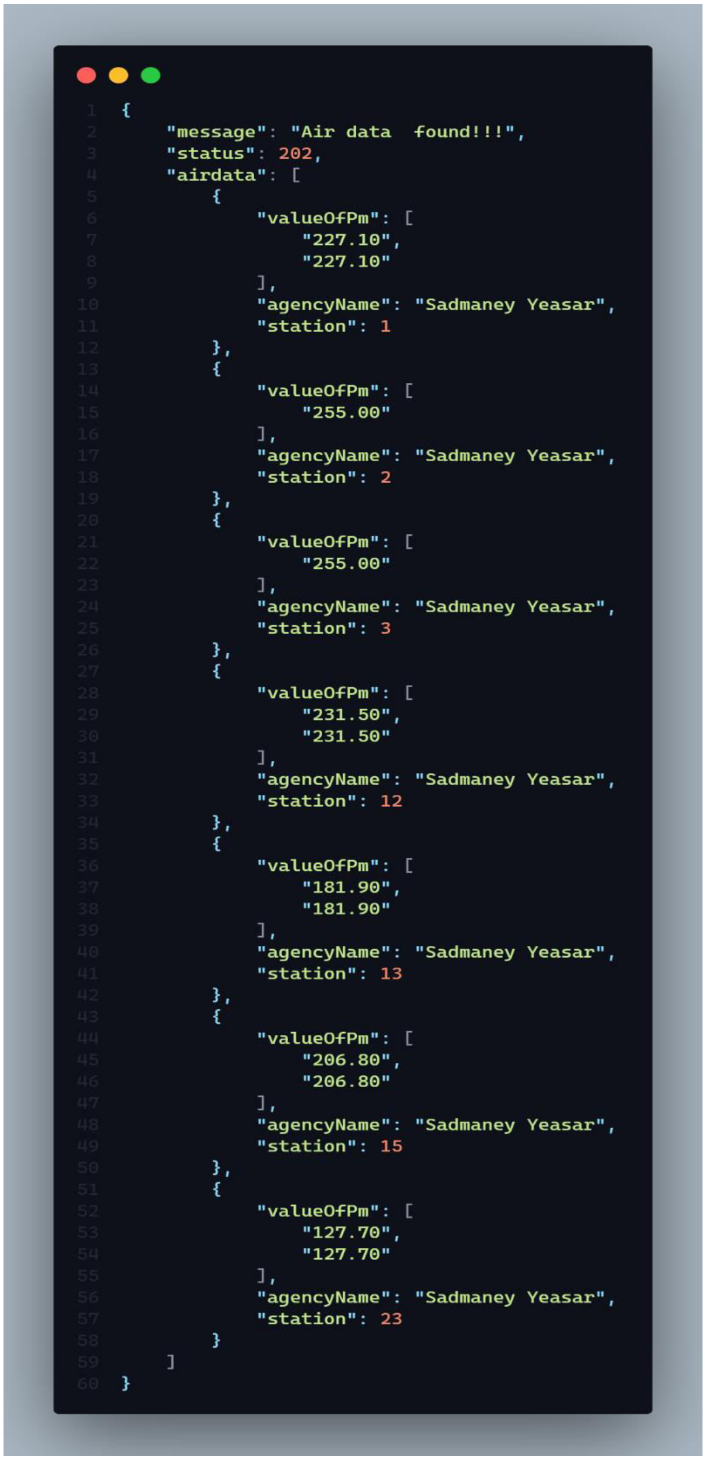


```
{
        "agentId": 1
}
```

Params: N/A

Query: N/A

Response:



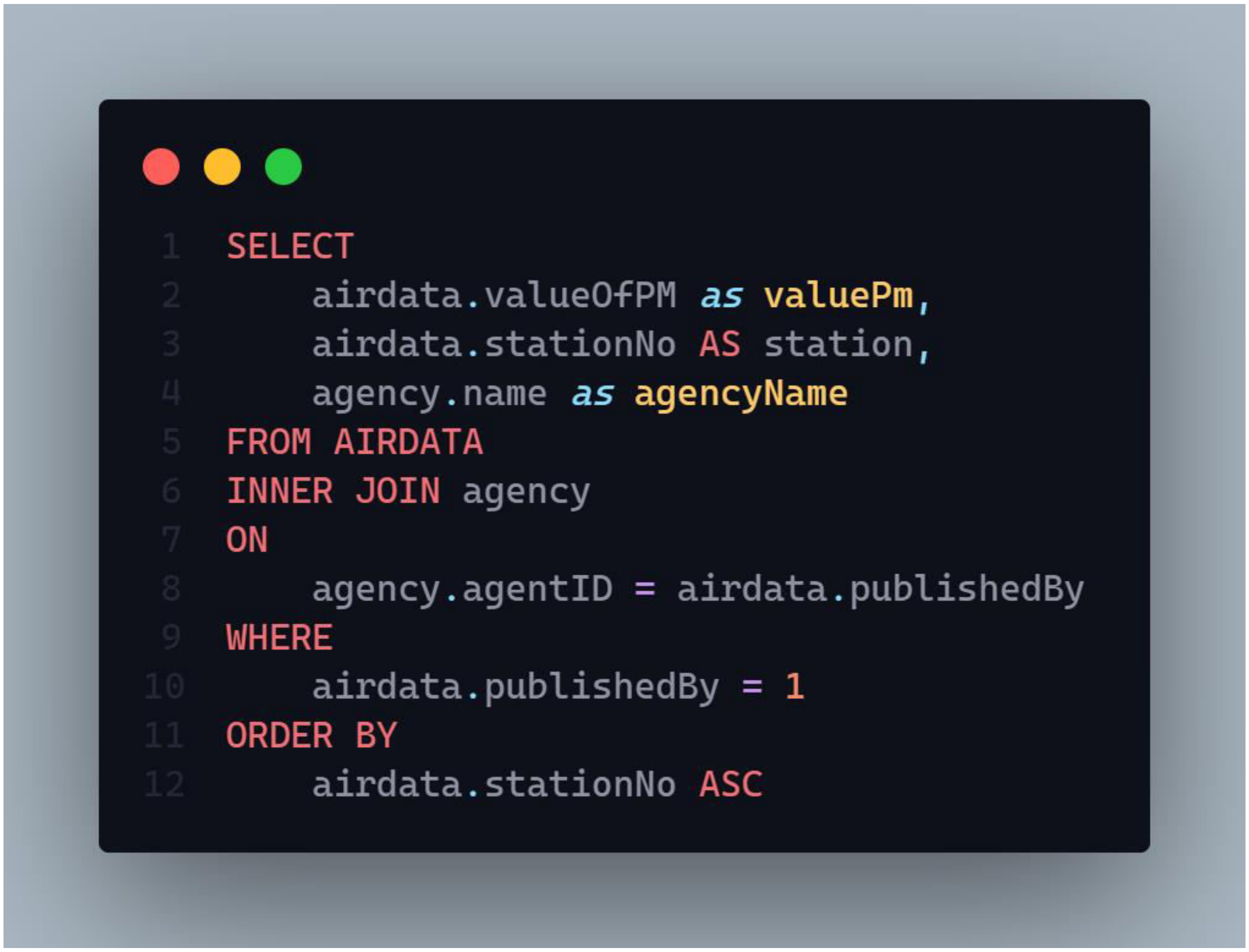
```
{
    "message": "Air data  found!!!",
    "status": 202,
    "airdata": [
        {
            "valueOfPm": [
                "227.10",
                "227.10"
            ],
            "agencyName": "Sadmaney Yeasar",
            "station": 1
        },
        {
            "valueOfPm": [
                "255.00"
            ],
            "agencyName": "Sadmaney Yeasar",
            "station": 2
        },
        {
            "valueOfPm": [
                "255.00"
            ],
            "agencyName": "Sadmaney Yeasar",
            "station": 3
        },
        {
            "valueOfPm": [
                "231.50",
                "231.50"
            ],
            "agencyName": "Sadmaney Yeasar",
            "station": 12
        },
        {
            "valueOfPm": [
                "181.90",
                "181.90"
            ],
            "agencyName": "Sadmaney Yeasar",
            "station": 13
        },
        {
            "valueOfPm": [
                "206.80",
                "206.80"
            ],
            "agencyName": "Sadmaney Yeasar",
            "station": 15
        },
        {
            "valueOfPm": [
                "127.70",
                "127.70"
            ],
            "agencyName": "Sadmaney Yeasar",
            "station": 23
        }
    ]
}
```

Description: If status code is

- **202 =>**
  - Air data found!!!
- **406 =>**
  - Runtime error / Internal Error!!!
- **404 =>**
  - **No Air data has been found**

Graph type; box plot Chart: https://plotly.com/javascript/box-plots/

Data base query



```sql
SELECT
    airdata.valueOfPM as valuePm,
    airdata.stationNo AS station,
    agency.name as agencyName
FROM AIRDATA
INNER JOIN agency
ON
    agency.agentID = airdata.publishedBy
WHERE
    airdata.publishedBy = 1
ORDER BY
    airdata.stationNo ASC
```

13. Show Session Wise AVG AQI of PM2.5 value in some query

o. **If user want to see monthly base**
   i. **Y axis contains PM 2.5 value**
   ii. **X axis contains Season Name**
   iii. **But AVG value will be count by individual months of all years**

p. **If user want to see yearly base**
   i. **Y axis contains PM 2.5 value**
   ii. **X axis contains Season name**
   iii. **But Value will be count by all year**

API: baseUrl/airData/get/aqi/of/session

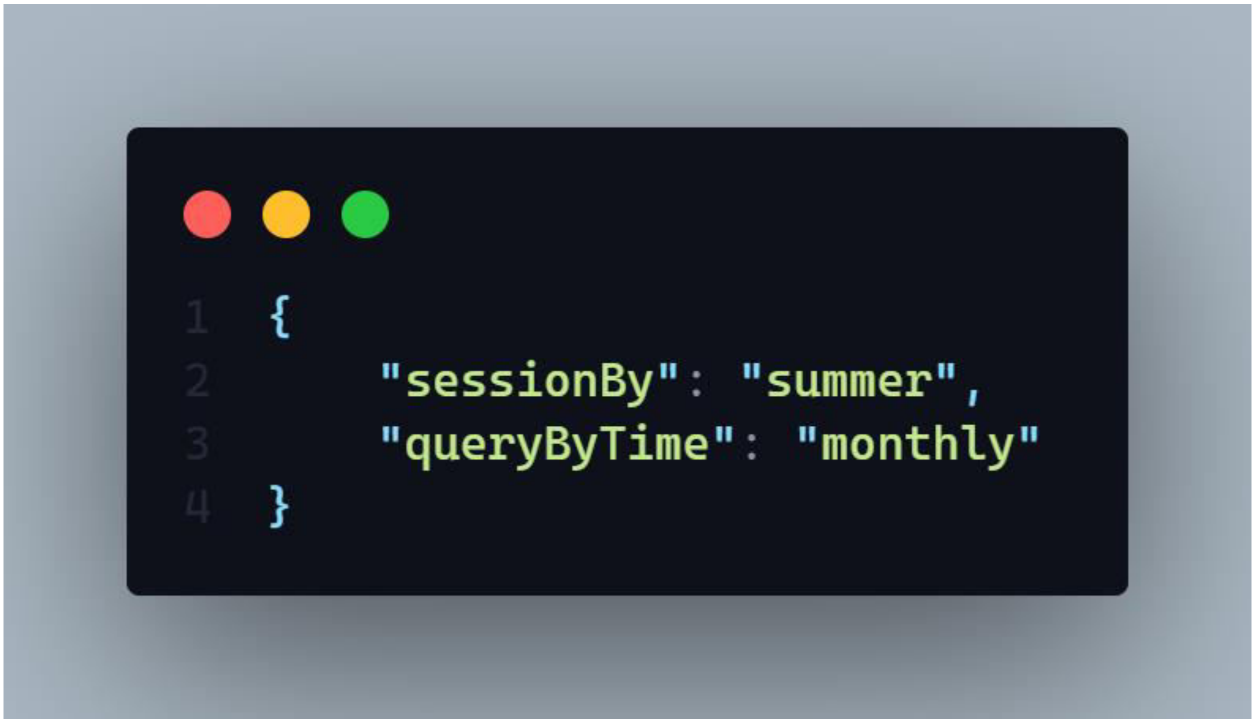
Type: REST

Method : POST

Access: **: All**

Requirement: 9

Body :



```
{
      "sessionBy": "summer",
      "queryByTime": "monthly"
}
```

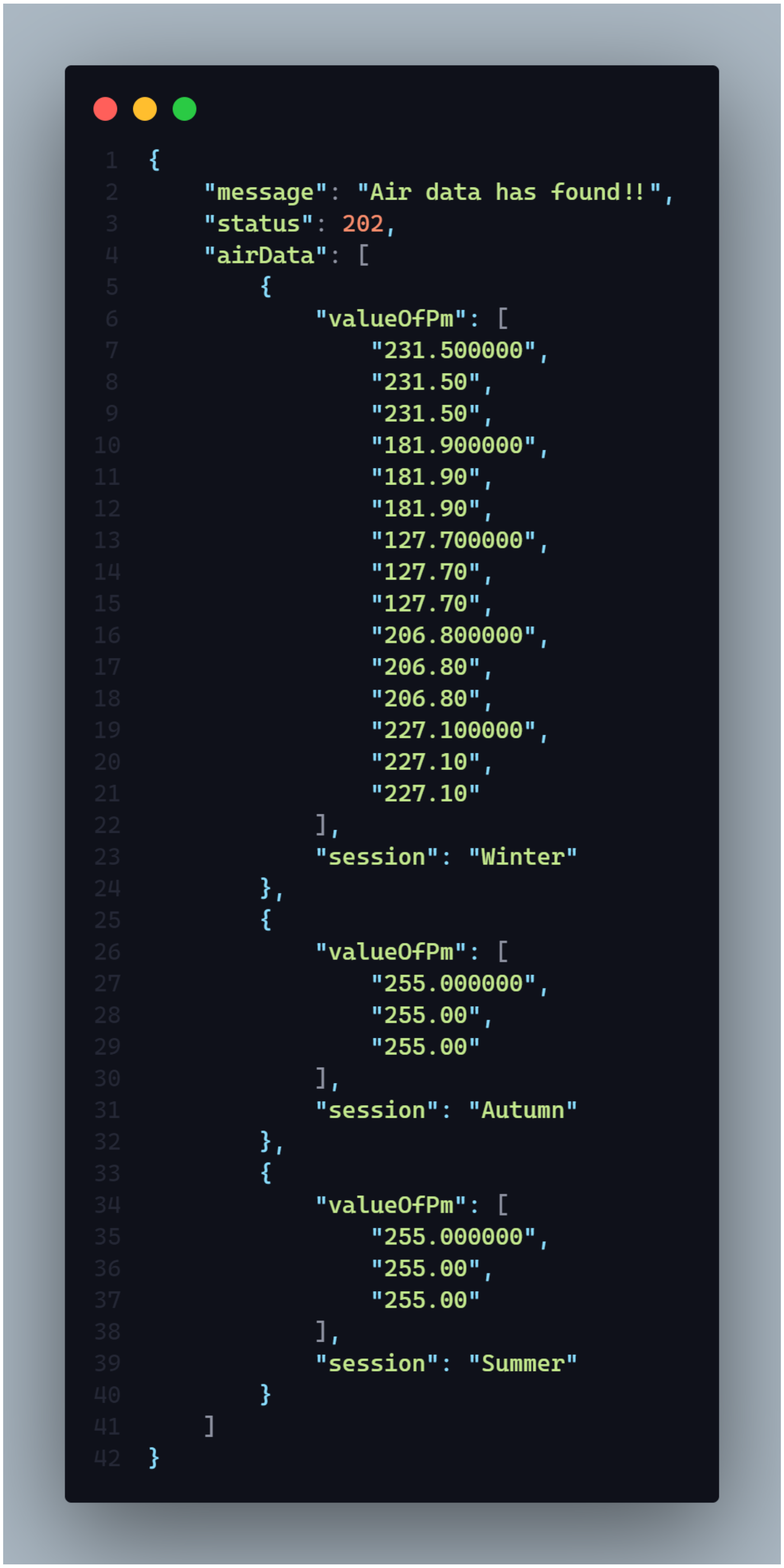
Session by => “summer”, “autunm” , “winter”

queryByTime => “monthly”, “yearly”

Params: N/A

Query: N/A

Response:



```json
{
    "message": "Air data has found!!",
    "status": 202,
    "airData": [
        {
            "valueOfPm": [
                "231.500000",
                "231.50",
                "231.50",
                "181.900000",
                "181.90",
                "181.90",
                "127.700000",
                "127.70",
                "127.70",
                "206.800000",
                "206.80",
                "206.80",
                "227.100000",
                "227.10",
                "227.10"
            ],
            "session": "Winter"
        },
        {
            "valueOfPm": [
                "255.000000",
                "255.00",
                "255.00"
            ],
            "session": "Autumn"
        },
        {
            "valueOfPm": [
                "255.000000",
                "255.00",
                "255.00"
            ],
            "session": "Summer"
        }
    ]
}
```

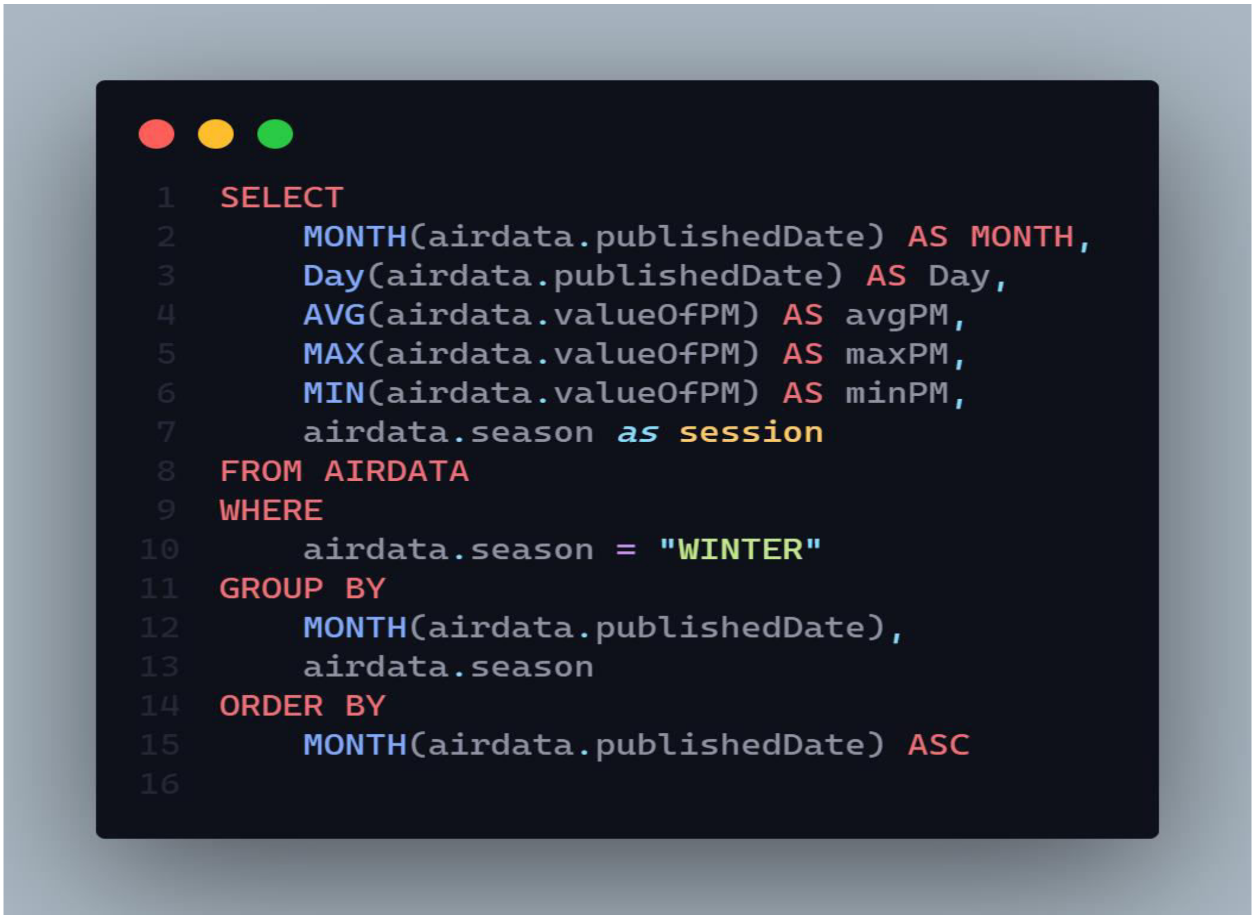
Description: If status code is

- **202 =>**
  - Air data found!!!
- **406 =>**
  - Runtime error / Internal Error!!!
- **404 =>**
  - **No Air data has been found**

Graph type; box plot Chart: [**https://plotly.com/javascript/box-plots/**](https://plotly.com/javascript/box-plots/)

Data base query

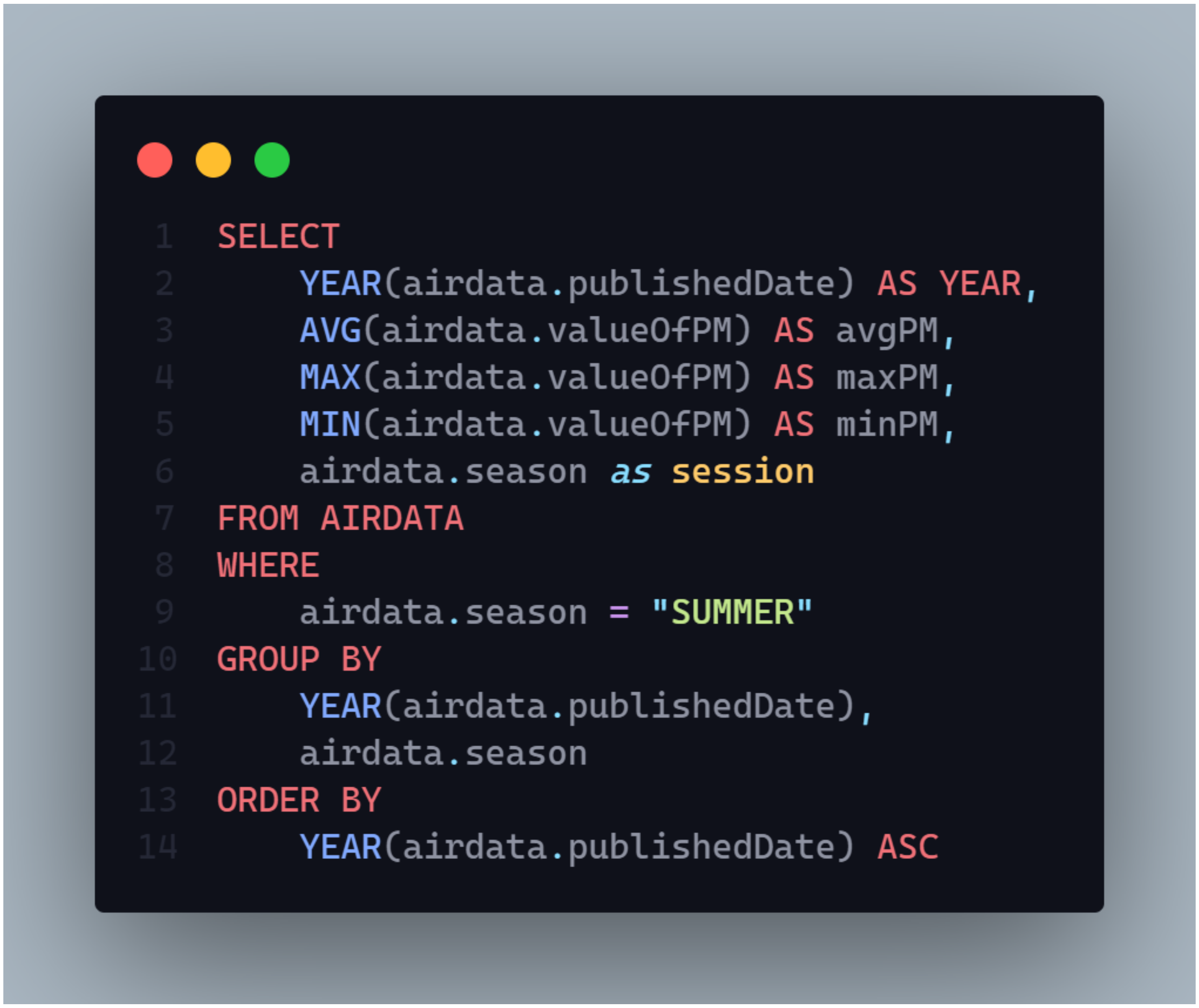
For Monthly



```sql
SELECT
    MONTH(airdata.publishedDate) AS MONTH,
    Day(airdata.publishedDate) AS Day,
    AVG(airdata.valueOfPM) AS avgPM,
    MAX(airdata.valueOfPM) AS maxPM,
    MIN(airdata.valueOfPM) AS minPM,
    airdata.season as session
FROM AIRDATA
WHERE
    airdata.season = "WINTER"
GROUP BY
    MONTH(airdata.publishedDate),
    airdata.season
ORDER BY
    MONTH(airdata.publishedDate) ASC

```

For yearly



```sql
SELECT
    YEAR(airdata.publishedDate) AS YEAR,
    AVG(airdata.valueOfPM) AS avgPM,
    MAX(airdata.valueOfPM) AS maxPM,
    MIN(airdata.valueOfPM) AS minPM,
    airdata.season as session
FROM AIRDATA
WHERE
    airdata.season = "SUMMER"
GROUP BY
    YEAR(airdata.publishedDate),
    airdata.season
ORDER BY
    YEAR(airdata.publishedDate) ASC
```

14. Show AVG PM 2.5 value of each year

q. Y axis contains PM 2.5 value
r. X axis contains each Year number
s. User need to provide a range of year which between he can see such type of data

API: baseUrl/airData/get/avg/aqi/by/year

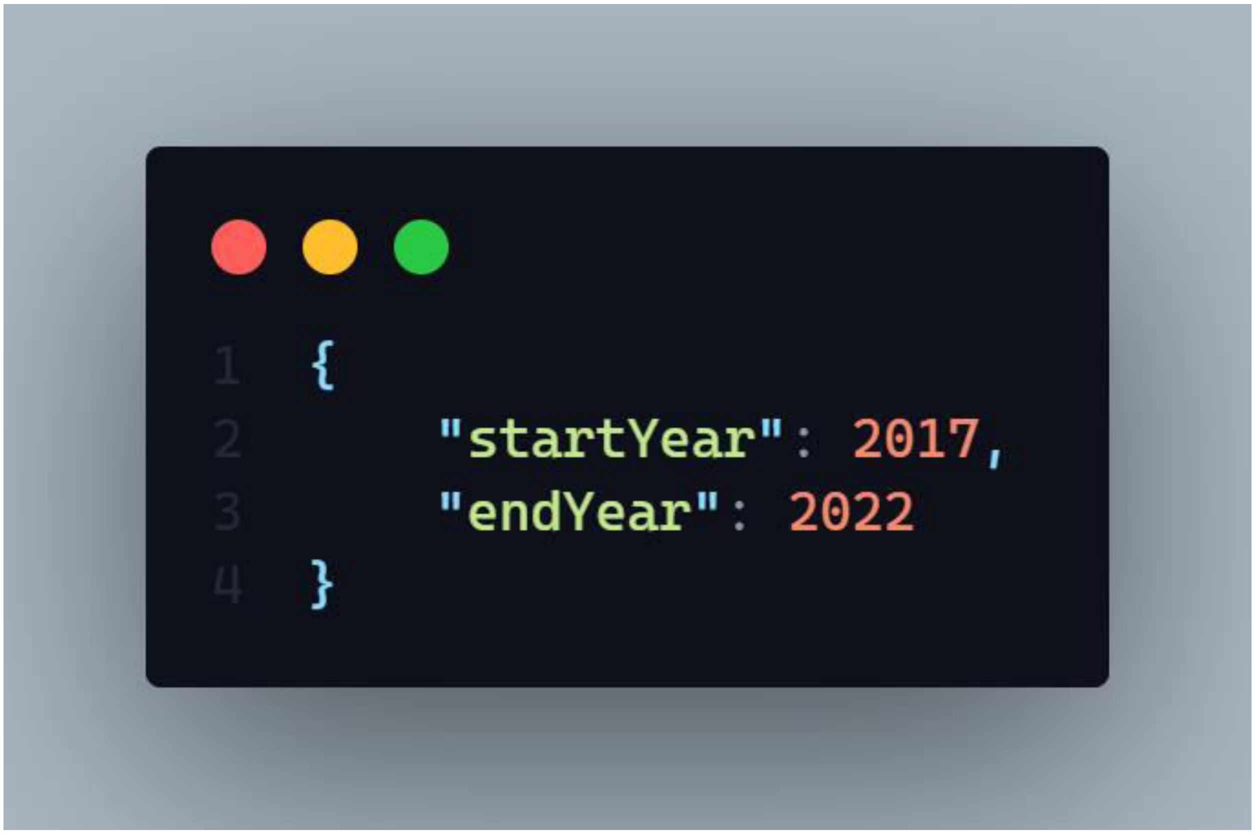
Type: REST

Method : POST

Access: : All

Requirement: 10

Body :

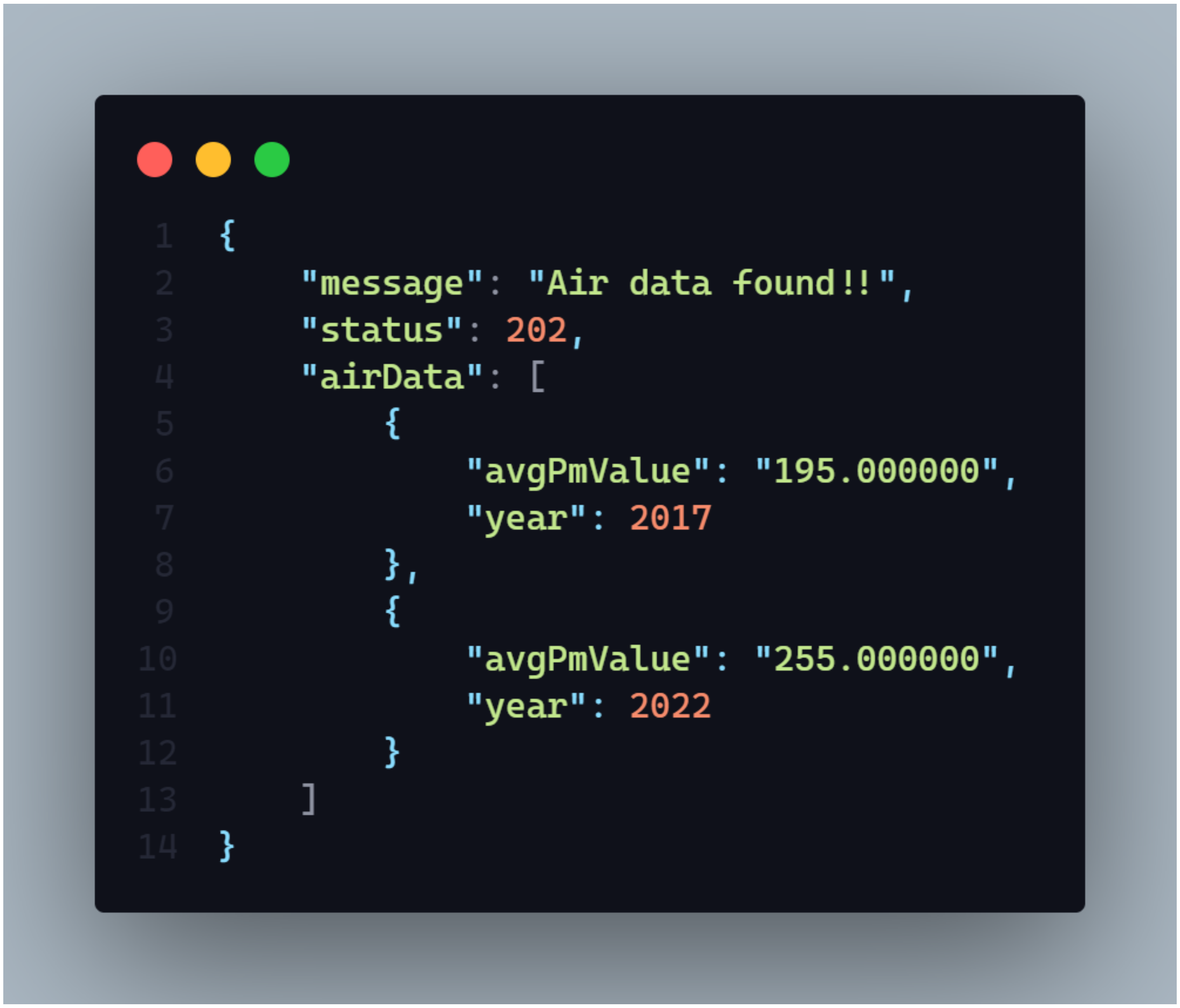


```
{
        "startYear": 2017,
        "endYear": 2022
}
```

Params: N/A

Query: N/A

Response:



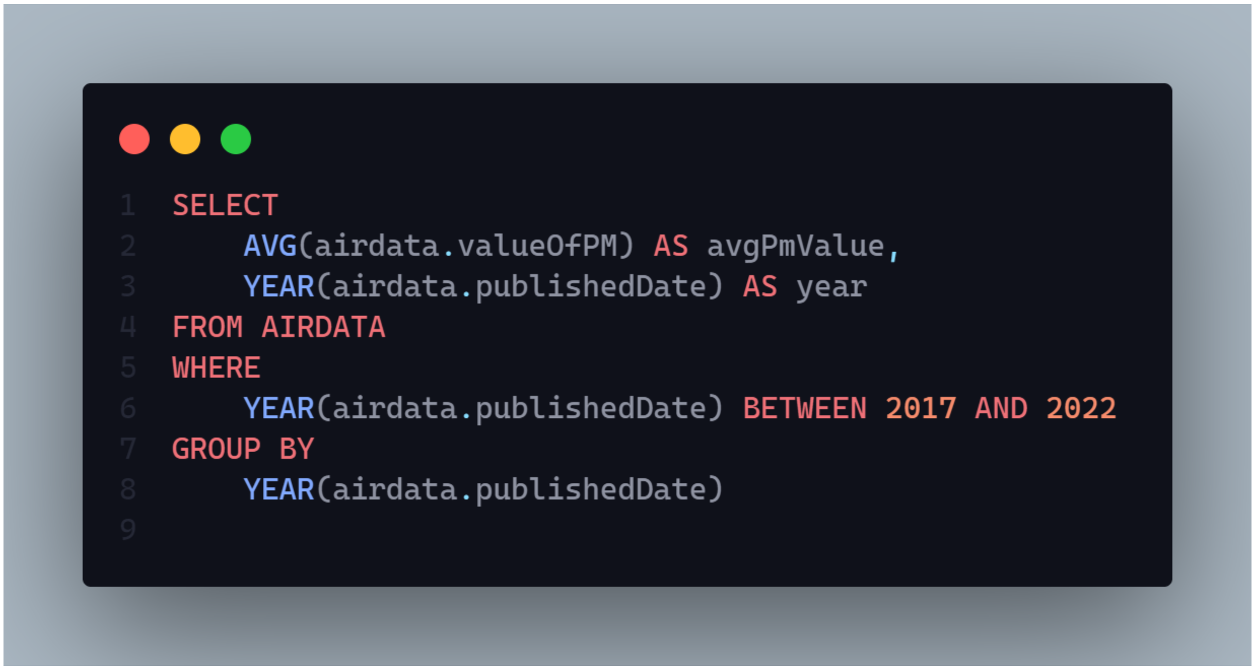
```
{
    "message": "Air data found!!",
    "status": 202,
    "airData": [
        {
            "avgPmValue": "195.000000",
            "year": 2017
        },
        {
            "avgPmValue": "255.000000",
            "year": 2022
        }
    ]
}
```

Description: If status code is

- **202 =>**
  - Air data found!!!
- **406 =>**
  - Runtime error / Internal Error!!!
- **404 =>**
  - **No Air data has been found**

Graph type; bar Chart: https://plotly.com/javascript/bar-charts/

Data base query



```
SELECT
    AVG(airdata.valueOfPM) AS avgPmValue,
    YEAR(airdata.publishedDate) AS year
FROM AIRDATA
WHERE
    YEAR(airdata.publishedDate) BETWEEN 2017 AND 2022
GROUP BY
    YEAR(airdata.publishedDate)

```

15. Get all agency’s Average PM2.5 in a daily basis of a particular Season.

t. Need to find all agency’s Daily Avg PM2.5
u. But only for a selected session data need to be find
v. Session input will be pass from client
w. Y axis contains Daily Avg PM2.5 value
x. X axis contains Daily date
i. Date will contains => YYYY-MM

API: baseUrl/airData/get/daily/basis/session

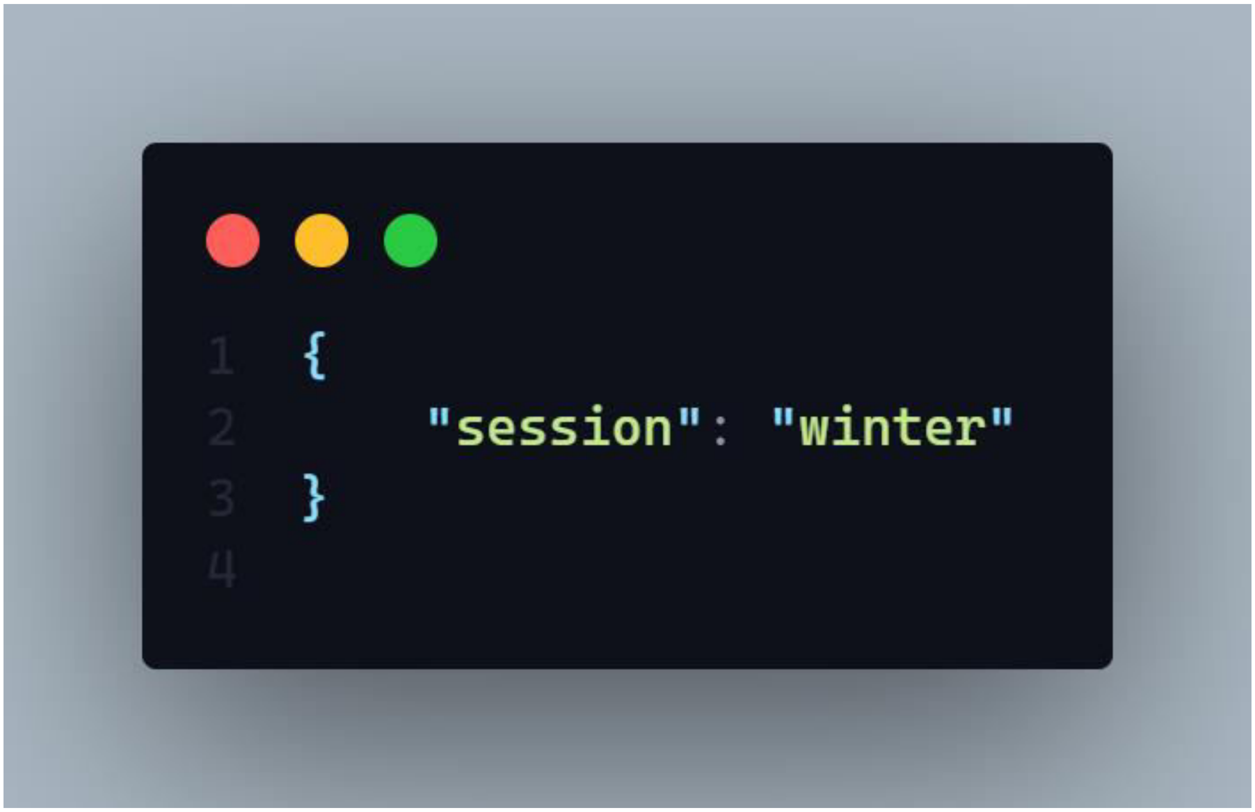
Type: REST

Method : POST

Access: : All

Requirement: 4(a)

Body :

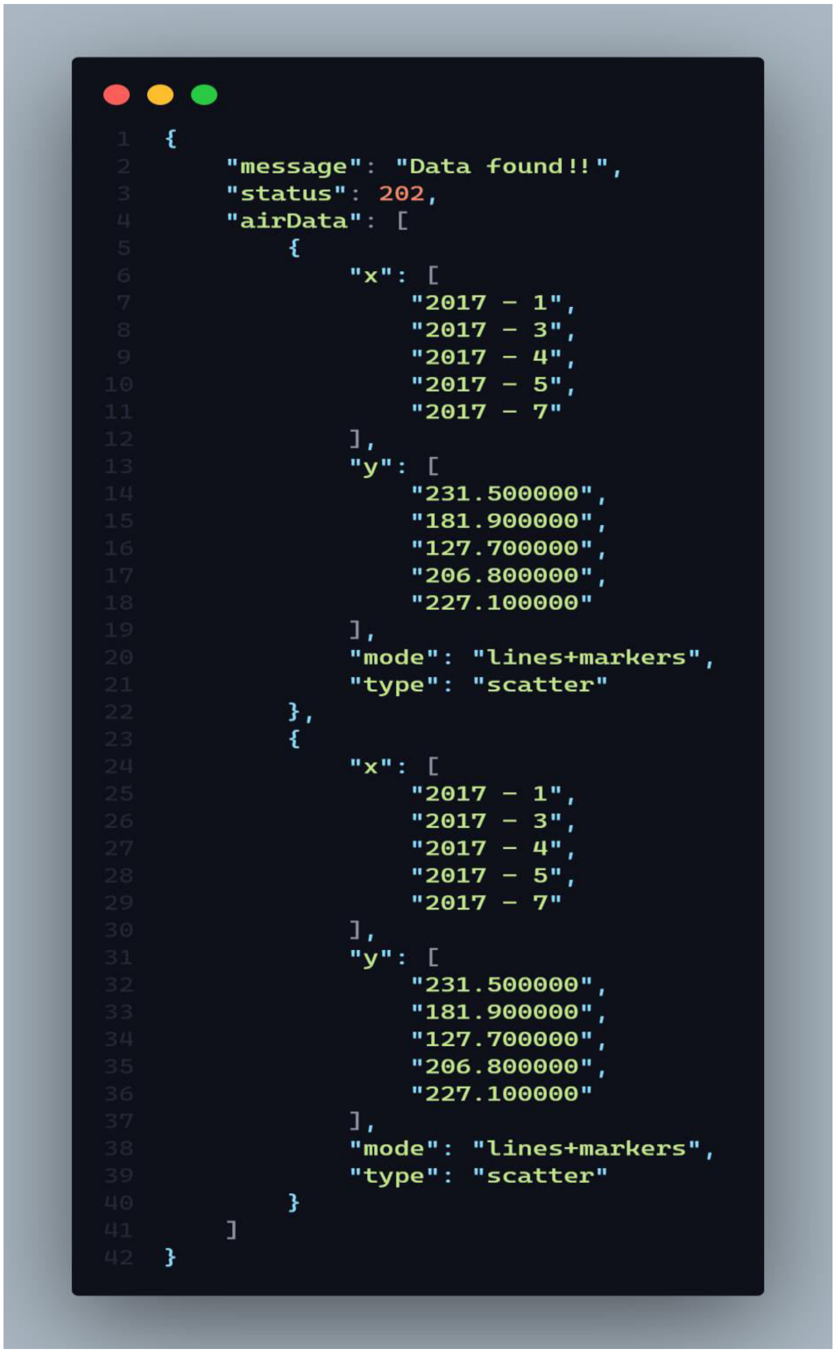


```
{
      "session": "winter"
}
```

Params: N/A

Query: N/A

Response:



```
{
    "message": "Data found!!",
    "status": 202,
    "airData": [
        {
            "x": [
                "2017 - 1",
                "2017 - 3",
                "2017 - 4",
                "2017 - 5",
                "2017 - 7"
            ],
            "y": [
                "231.500000",
                "181.900000",
                "127.700000",
                "206.800000",
                "227.100000"
            ],
            "mode": "lines+markers",
            "type": "scatter"
        },
        {
            "x": [
                "2017 - 1",
                "2017 - 3",
                "2017 - 4",
                "2017 - 5",
                "2017 - 7"
            ],
            "y": [
                "231.500000",
                "181.900000",
                "127.700000",
                "206.800000",
                "227.100000"
            ],
            "mode": "lines+markers",
            "type": "scatter"
        }
    ]
}
```

Description: If status code is

- **202 =>**
  - Air data found!!!
- **406 =>**
  - Runtime error / Internal Error!!!
- **404 =>**
  - **No Air data has been found**

Graph type; line Chart: https://plotly.com/javascript/line-charts/

16. Show Avg AQI of PM2.5 value of all Division

API: baseUrl/airData/get/aqi/all/division/data

Type: REST

Method : GET

Access: : All

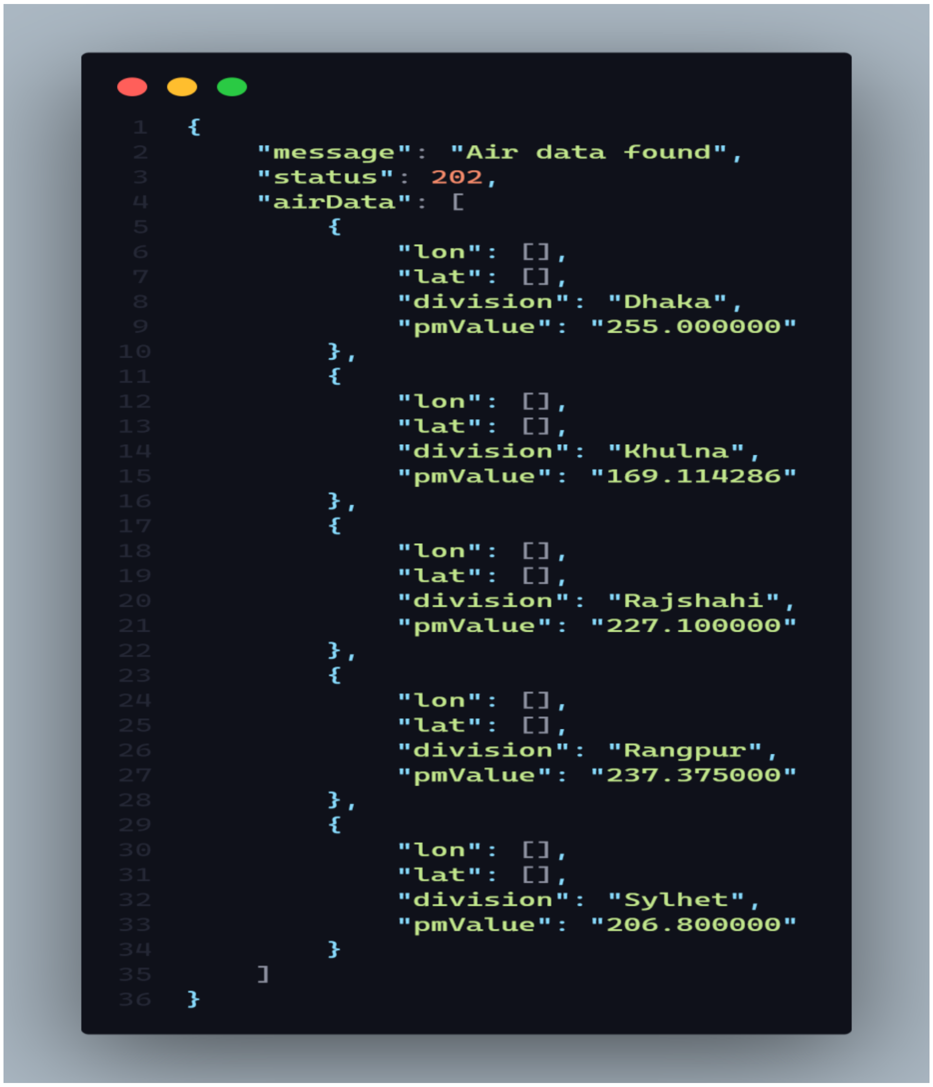
Requirement: 5

Body : N/A

Params: N/A

Query: N/A

Response:



```
{
        "message": "Air data found",
        "status": 202,
        "airData": [
            {
                "lon": [],
                "lat": [],
                "division": "Dhaka",
                "pmValue": "255.000000"
            },
            {
                "lon": [],
                "lat": [],
                "division": "Khulna",
                "pmValue": "169.114286"
            },
            {
                "lon": [],
                "lat": [],
                "division": "Rajshahi",
                "pmValue": "227.100000"
            },
            {
                "lon": [],
                "lat": [],
                "division": "Rangpur",
                "pmValue": "237.375000"
            },
            {
                "lon": [],
                "lat": [],
                "division": "Sylhet",
                "pmValue": "206.800000"
            }
        ]
}
```

Description: If status code is

- **202 =>**
  - Air data found!!!
- **406 =>**
  - Runtime error / Internal Error!!!
- **404 =>**
  - **No Air data has been found**

Graph type; Map box: https://plotly.com/javascript/filled-area-on-mapbox/