API Design

**Accident Data API**

**GET /accidents**

* **Description:** Retrieves all accident records.
* **Method**: GET
* **Authentication Required:** Yes (API-KEY)
* **Success Response:** JSON array of accident records.
* **Failure Response:** Error JSON with description and HTTP status code.

**GET /accidents/{id}**

* **Description:** Retrieves a specific accident record by its unique ID.
* **Method:** GET
* **Authentication Required:** Yes (API-KEY)
* **Parameters:** id (the unique identifier of the accident).
* **Success Response:** JSON of the accident record.
* **Failure Response:** Error JSON with description and HTTP status code.

**POST /accidents**

* **Description:** Creates a new accident record.
* **Method:** POST
* **Authentication Required:** Yes (API-KEY)
* **Body:** JSON containing location, timestamp, accidentType, severity, participants, weatherConditions, latitude, longitude, and optional geometry.
* **Success Response:** JSON with a success message.
* **Failure Response:** Error JSON with description and HTTP status code.
* **Example body**:

{

"latitude": 60.192059,

"longitude": 24.945831,

"timestamp": "2024-03-03T12:00:00",

"volume": 100,

"averageSpeed": 80,

"congestionLevel": "High",

"geometry": [

[

{"latitude": 60.192059, "longitude": 24.945831},

{"latitude": 60.193000, "longitude": 24.946000},

{"latitude": 60.194000, "longitude": 24.947000}

]

]

}

**OR**

{

"latitude": 60.192059,

"longitude": 24.945831,

"timestamp": "2024-03-03T12:00:00",

"accident\_type": "Collision",

"severity": "Minor",

"participants": 2,

"weather\_conditions": "Clear"

}

**DELETE /accidents**

* **Description:** Deletes all accident records.
* **Method:** DELETE
* **Authentication Required:** Yes (API-KEY)
* **Success Response:** JSON with a success message.
* **Failure Response:** Error JSON with description and HTTP status code

**DELETE /accidents/{id}**

* **Description:** Deletes a specific accident record by its unique ID.
* **Method:** DELETE
* **Authentication Required:** Yes (API-KEY)
* **Parameters:** id (the unique identifier of the accident).
* **Success Response:** JSON with a success message.
* **Failure Response:** Error JSON with description and HTTP status code.

**Traffic Data API**

**GET /traffic**

* **Description:** Retrieves all traffic records.
* **Method:** GET
* **Authentication Required:** Yes (API-KEY)
* **Success Response:** JSON array of traffic records.
* **Failure Response:** Error JSON with description and HTTP status code.

**GET /traffic/{id}**

* **Description:** Retrieves a specific traffic record by its unique ID.
* **Method:** GET
* **Authentication Required:** Yes (API-KEY)
* **Parameters:** id (the unique identifier of the traffic record).
* **Success Response:** JSON of the traffic record.
* **Failure Response:** Error JSON with description and HTTP status code.

**POST /traffic**

* **Description:** Creates a new traffic record.
* **Method:** POST
* **Authentication Required:** Yes (API-KEY)
* **Body:** JSON containing location, timestamp, volume, averageSpeed, congestionLevel, latitude, longitude, and optional geometry.
* **Success Response:** JSON with a success message.
* **Failure Response:** Error JSON with description and HTTP status code.
* **Example body:**

{

"latitude": 60.192059,

"longitude": 24.945831,

"timestamp": "2024-03-03T12:00:00",

"volume": 100,

"averageSpeed": 80,

"congestionLevel": "High",

"geometry": [

[

{"latitude": 60.192059, "longitude": 24.945831},

{"latitude": 60.193000, "longitude": 24.946000},

{"latitude": 60.194000, "longitude": 24.947000}

]

]

}

**OR**

{

"latitude": 60.192059,

"longitude": 24.945831,

"timestamp": "2024-03-03T12:00:00",

"volume": 100,

"averageSpeed": 80,

"congestionLevel": "High"

}

**DELETE /traffic**

* **Description:** Deletes all traffic records.
* **Method**: DELETE
* **Authentication Required:** Yes (API-KEY)
* **Success Response:** JSON with a success message.
* **Failure Response:** Error JSON with description and HTTP status code

**DELETE /traffic/{id}**

* **Description:** Deletes a specific traffic record by its unique ID.
* **Method:** DELETE
* **Authentication Required:** Yes (API-KEY)
* **Parameters:** id (the unique identifier of the traffic record).
* **Success Response:** JSON with a success message.
* **Failure Response:** Error JSON with description and HTTP status code.

**Testing**

For testing, you can use the Postman collections found in the Github repository. Copy them and paste (import) them into your Postman. Link to download Postman: <https://www.postman.com/downloads/>

You can also use similar tools such as Insomnia or Curl for testing.