**Voyage Report Service**

**Host URL:** *http://voyageservice.herokuapp.com/*

**Block Diagram:**

**LUIS**

**Mongo DB**

**GET**

**POST**

**Control Flow:**

* **/POST Request:** 
  + Service accept incoming request to submit a new report.
  + Request body which contains set of voyage report statements are validated against the specified schema.
  + The validation is done via node package ‘**Joi**’. Please refer to below URL for more information regarding the package; [*https://www.npmjs.com/package/joi*](https://www.npmjs.com/package/joi)
  + If there is any error in validation, corresponding error message with status code **400 bad request** returned to client.
  + If validation is successful then each report statement send to **LUIS** for analyzing and each analyzed statement get stored in memory along with the analyzed details in form of array of objects.
  + **Schema** is present to store data in **Mongo DB** and the schema is specified by using **mongoose ODM** (*https://www.npmjs.com/package/mongoose*).
  + After analyzing all statement, the resultant array structure is validated against the DB schema.
  + If there is any mismatch, corresponding error message with status code **500 internal server error** send to client.
  + If no mismatch is there, then first extract the **flight no** from first statement of report and then store the complete array to DB.
  + If there is any error while storing report on DB, server will throw an error and service will return a **500 internal server error** to client.
  + If the request is successful, then client gets informed about the same by returning status code **201 created**.
* **/GET Request:** 
  + This request gets fired from ‘**React dashboard application**’ to show existing voyage reports.
  + There are two different **GET** request gets fired, one to get **all reports** and another to get a **specific report**. The second one accepts report id as request parameter in URL.
  + For each request, service accepts the request first and then query the database to fetch the report.
  + If request is successful, then DB will return the report/reports to client application with status code **200 ok**.
  + If a specific report which is asked by the client not found in DB, then client will get informed about the same by sending **404 not found** status.
* **Credentials:**
  + **MLabs:** [*https://mlab.com/login/*](https://mlab.com/login/)
  + **Username:** *qatar*
  + **Password:** *qatar@1234*