Web Engineering Project Documentation

Dimitris Aktsoglou (s2979616), Emanuel Nae (s2931931), Daan Groot (s2958287)

March 16, 2019

1 Introduction

The decisions we make in the design of our API are based on RESTful Design principles. REST design is in general based upon the following principles:

1. Resources, which are any kind of object, data, or service that can be accessed by the client.
2. Resource has an identifier, which is a URI that uniquely identifies that resource.
3. Client interacts with a service by exchanging representation of resources.
4. Use a uniform interface, which helps to decouple the client and service implementations.

Furthermore there are four level of "Maturity" that have been defined in such a design. Our goal is to design an API as close as possible to Maturity level 3.

2 Milestone 1: API Design

2.1 Resources

Resources refers to the information that can be returned by an API. In our case, the various resources that can be returned are the following:

1. **Airports object**: represents all the airports that are available in the USA, which has parameters such as `code` and `name`.
2. **Statistics object**: returns statistics about flights. It has the following parameters: `flights`, `number of delays` and `minutes delayed`.
   - 2.1. **Flights object**: returns information about the number of `cancelled`, `on time`, `delayed`, `diverted` flights and their `total`.
   - 2.2. **Number of delays object**: returns the number of delays categorized by reasons such as: `late aircraft`, `weather`, `security`, `national aviation system` and `carrier`.
   - 2.3. **Minutes delayed object**: returns the number of minutes delayed per reason of delay: `late aircraft`, `weather`, `security`, `national aviation system` and `carrier`.
3. **Time object**: returns information about time based on the following parameters: `year`, `month` and `label`
4. **Carriers object**: returns the carriers that have flights in the USA, which have a `code` and a `name` parameter.

2.2 JSON Hyper-Schema

JSON Hyper-Schema is a JSON Schema vocabulary that allows the user to annotate JSON documents with hyperlinks and instructions for processing and manipulating remote JSON resources through hypermedia environments. In our case, since we are using REST API, the hypermedia environment that is being used is HTTP.

1. **Airports object**

```
{
    "type": "object",
    "properties": {
        "code": {
            "type": "string",
            "readOnly": true
        }
        "name": {
            "type": "string",
            "readOnly": true
        }
    },
    "links": [
        {
            "rel": "self",
            "href": "airports/{airport_code}"
        }
    ],
    "required": ["code"]
}
```

2. **Statistics object**

```
{
    "type": "object",
    "properties": {
        "flights": {
            "type": "string",
            "readOnly": true
        }
        "# of delays": {
            "type": "string",
            "readOnly": true
        }
        "minutes delayed": {
            "type": "string",
            "readOnly": true
        }
    },
    "links": [
        {
            "rel": "airports", "carriers",
```

```
            "href": "airports/carriers/stats/flights"
        }
    ]
}
```

2.1. **Flights object**:

```
{
    "type": "object",
    "properties": {
        "cancelled": {
            "type": "number",
        }
        "on time": {
            "type": "number",
        }
        "converted": {
            "type": "number",
        }
        "delayed": {
            "type": "number",
        }
        "total": {
            "type": "number",
        }
    },
    "links": [
        {
            "rel": "stats",
            "href": "airports/carriers/stats/flights"
        }
    ]
}
```

2.2. **Number of delays object**

```
{
    "type": "object",
    "reasons": {
        "late aircraft": {
            "type": "number",
        }
        "weather": {
            "type": "number",
        }
        "security": {
            "type": "number",
        }
        "national aviation system": {
            "type": "number",
        }
```

```
            "carrier": {
                "type": "number",
            }
        },
        "links": [
            {
                "rel": "stats",
                "href": "airports/carriers/stats/#ofdelays"
            }
        ]
    }
```

2.3. **Minutes delayed object**:

```
    {
        "type": "object",
        "reasons": {
            "late aircraft": {
                "type": "number",
            }
            "weather": {
                "type": "number",
            }
            "security": {
                "type": "number",
            }
            "national aviation system": {
                "type": "number",
            }
            "carrier": {
                "type": "number",
            }
            "total": {
                "type": "number",
            }
        },
        "links": [
            {
                "rel": "stats",
                "href": "airports/carriers/stats/minutesdelayed"
            }
        ]
    }
```

3. **Time object**:

```
{
    "type": "object",
    "properties": {
        "label": {
            "type": "string",
            "readOnly": true
```

```
            }
            "year": {
                "type": "number",
                "readOnly": true
            }
            "month": {
                "type": "number",
                "readOnly": true
            }
        },
        "links": [
            {
                "rel": "airports", "carriers", "stats",
                "href": "airports/carriers/stats/time?year={year}&month={month}"
            }
        ],
        "required": ["year","month"]
    }
```

4. **Carriers object**:

```
    {
        "type": "object",
        "properties": {
            "code": {
                "type": "string",
                "readOnly": true
            }
            "name": {
                "type": "string",
                "readOnly": true
            }
        },
        "links": [
            {
                "rel": "self",
                "href": "carriers\{carrier_code}"
            }
        ],
        "required": ["code"]
    }
```

2.3 Endpoints

Endpoints represent the entries that give access to the resource. Currently, the endpoints presented below refer strictly to the ones given in the Project Description document, but they may be modified in the future.

1. **GET /airports**
   Returns all airports available in the US.

Request headers:

- `Accept: application/json`
- `Accept: text/csv`

Request body:

`empty`

Response headers:

- `Content-Type: application/json`
- `Content-Type: text/csv`

Responses with content:

- 200 OK Successful Operation
  `: array[string]  The 3-letter codes of the airports.`
- 5xx Internal Server Error
  `empty`

2. **GET /carriers**
Returns all carriers operating in US airports.

Request headers:

- `Accept: application/json`
- `Accept: text/csv`

Request body:

`empty`

Response headers:

- `Content-Type: application/json`
- `Content-Type: text/csv`

Responses with content:

- 200 OK Successful Operation
  `: array[string]  The 3-letter codes of the carriers.`
- 5xx Internal Server Error
  `empty`

3. **GET /airports/{airport_code}/carriers**
Returns all carriers operating at a specific US airport.

Endpoint path parameters:

- `airport_code` The 3-letter code of this airport.

Request headers:

- `Accept: application/json`
- `Accept: text/csv`

Request body:

`empty`

Response headers:

- `Content-Type: application/json`
- `Content-Type: text/csv`

Responses with content:

- 200 OK Successful operation
  `: array[string]` The 3-letter codes of the carriers.
- 400 Bad Request
  `empty`
- 401 Unauthorized
  `empty`
- 404 Not Found
  `empty`
- 5xx Internal Server Error
  `empty`

4. **GET /airports/{airport_code}/carriers/{carrier_code}/stats?year={year}&month={month}**
   Returns statistics about flights of a carrier from/to an US airport for a given year and month.

   GET /airports/{airport_code}/carriers/{carrier_code}/stats
   Returns statistics about flights of a carrier from/to an US airport for all months available.

   Endpoint path parameters:

   - `airport_code` The 3-letter code of the airport.
   - `carrier_code` The 3-letter code of the carrier.
   - `year` The year for which data should be returned.
   - `month` The month for which data should be returned.

   Request headers:

   - `Accept: application/json`
   - `Accept: text/csv`

Request body:

```
empty
```

Response headers:

- `Content-Type: application/json`
- `Content-Type: text/csv`

Responses with content:

- 200 OK Successful operation
  /airports/{airport_code}/carriers/{carrier_code}/stats?year={year}&month={month}

```
flights: object
    cancelled: int
    on-time: int
    total: int
    delayed: int
    diverted: int
delays-counts: object
    late-aircraft: int
    weather: int
    security: int
    national-aviation-system: int
    carrier: int
minutes-delayed: object
    late-aircraft: int
    weather: int
    security: int
    national-aviation-system: int
    carrier: int
    total: int
```

/airports/{airport_code}/carriers/{carrier_code}/stats

```
: array[object]
    flights: object
        cancelled: int
        on-time: int
        total: int
        delayed: int
        diverted: int
    delays-counts: object
        late-aircraft: int
        weather: int
        security: int
        national-aviation-system: int
        carrier: int
    minutes-delayed: object
        late-aircraft: int
        weather: int
```

```
        security: int
        national-aviation-system: int
        carrier: int
        total: int
    time: object
        year: int
        month: int
```

- 400 Bad Request

  `empty`
- 401 Unauthorized

  `empty`
- 404 Not Found

  `empty`
- 5xx Internal Server Error

  `empty`

POST /airports/{airport_code}/carriers/{carrier_code}/stats?year={year}&month={month}
Add statistics about flights of a carrier from/to an US airport for a given year and month.

POST /airports/{airport_code}/carriers/{carrier_code}/stats
Add statistics about flights of a carrier from/to an US airport for all months available.

Endpoint path parameters:

- `airport_code` The 3-letter code of the airport.
- `carrier_code` The 3-letter code of the carrier.
- `year` The year for which data should be returned.
- `month` The month for which data should be returned.

Request headers:

- `Content-Type: application/json`
- `Content-Type: text/csv`

Request body:
/airports/{airport_code}/carriers/{carrier_code}/stats?year={year}&month={month}

```
flights: object
    cancelled: int
    on-time: int
    total: int
    delayed: int
    diverted: int
delays-counts: object
    late-aircraft: int
    weather: int
```

```
        security: int
        national-aviation-system: int
        carrier: int
    minutes-delayed: object
        late-aircraft: int
        weather: int
        security: int
        national-aviation-system: int
        carrier: int
        total: int
```

/airports/{airport_code}/carriers/{carrier_code}/stats

```
: array[object]
    flights: object
        cancelled: int
        on-time: int
        total: int
        delayed: int
        diverted: int
    delays-counts: object
        late-aircraft: int
        weather: int
        security: int
        national-aviation-system: int
        carrier: int
    minutes-delayed: object
        late-aircraft: int
        weather: int
        security: int
        national-aviation-system: int
        carrier: int
        total: int
    time: object
        year: int
        month: int
```

Response headers:

```
empty
```

Responses with content:

- 200 OK Successful operation
  ```
  empty
  ```
- 400 Bad Request
  ```
  empty
  ```
- 401 Unauthorized
  ```
  empty
  ```
- 404 Not Found

`empty`

- 5xx Internal Server Error

  `empty`

PUT /airports/{airport_code}/carriers/{carrier_code}/stats?year={year}&month={month}
Update statistics about flights of a carrier from/to an US airport for a given year and month.

PUT /airports/{airport_code}/carriers/{carrier_code}/stats
Update statistics about flights of a carrier from/to an US airport for all months available.

Endpoint path parameters:

- `airport_code` `The 3-letter code of the airport.`
- `carrier_code` `The 3-letter code of the carrier.`
- `year` `The year for which data should be returned.`
- `month` `The month for which data should be returned.`

Request headers:

- `Content-Type: application/json`
- `Content-Type: text/csv`

Request body:
/airports/{airport_code}/carriers/{carrier_code}/stats?year={year}&month={month}

```
flights: object
    cancelled: int
    on-time: int
    total: int
    delayed: int
    diverted: int
delays-counts: object
    late-aircraft: int
    weather: int
    security: int
    national-aviation-system: int
    carrier: int
minutes-delayed: object
    late-aircraft: int
    weather: int
    security: int
    national-aviation-system: int
    carrier: int
    total: int
```

/airports/{airport_code}/carriers/{carrier_code}/stats

```
: array[object]
    flights: object
        cancelled: int
        on-time: int
        total: int
        delayed: int
        diverted: int
    delays-counts: object
        late-aircraft: int
        weather: int
        security: int
        national-aviation-system: int
        carrier: int
    minutes-delayed: object
        late-aircraft: int
        weather: int
        security: int
        national-aviation-system: int
        carrier: int
        total: int
    time: object
        year: int
        month: int
```

Response headers:

```
empty
```

Responses with content:

- 200 OK Successful operation
  `empty`
- 400 Bad Request
  `empty`
- 401 Unauthorized
  `empty`
- 404 Not Found
  `empty`
- 5xx Internal Server Error
  `empty`

DELETE /airports/{airport_code}/carriers/{carrier_code}/stats?year={year}&month={month}
Delete statistics about flights of a carrier from/to an US airport for a given year and month.

DELETE /airports/{airport_code}/carriers/{carrier_code}/stats
Delete statistics about flights of a carrier from/to an US airport for all months available.

Endpoint path parameters:

- `airport_code` The 3-letter code of the airport.
- `carrier_code` The 3-letter code of the carrier.
- `year` The year for which data should be returned.
- `month` The month for which data should be returned.

Request headers:

- `Content-Type: application/json`
- `Content-Type: text/csv`

Request body:

/airports/{airport_code}/carriers/{carrier_code}/stats?year={year}&month={month}

```
empty
```

/airports/{airport_code}/carriers/{carrier_code}/stats

```
: array[object]
    time: object
        year: int
        month: int
```

Response headers:

```
empty
```

Responses with content:

- 200 OK Successful operation
  ```
  empty
  ```
- 400 Bad Request
  ```
  empty
  ```
- 401 Unauthorized
  ```
  empty
  ```
- 404 Not Found
  ```
  empty
  ```
- 5xx Internal Server Error
  ```
  empty
  ```

5. **GET /airports/{airport_code}/carriers/{carrier_code}/stats/flights?year={year} &month={month}**
   Returns the number of on-time, delayed, and cancelled flights of a carrier from/to an US airport for a given year and month.

   GET /airports/{airport_code}/carriers/{carrier_code}/stats/flights
   Returns the number of on-time, delayed, and cancelled flights of a carrier from/to an US airport for all months available.

   Endpoint path parameters:

- `airport_code` The 3-letter code of the airport.
- `carrier_code` The 3-letter code of the carrier.
- `year` The year for which data should be returned.
- `month` The month for which data should be returned.

Request headers:

- `Accept: application/json`
- `Accept: text/csv`

Request body:

`empty`

Response headers:

- `Content-Type: application/json`
- `Content-Type: text/csv`

Responses with content:

- 200 OK Successful operation
  /airports/{airport_code}/carriers/{carrier_code}/stats/flights?year={year}&month={month}

```
on time: int
delayed: int
cancelled: int
```

  /airports/{airport_code}/carriers/{carrier_code}/stats/flights

```
: array[object]
    on-time: int
    delayed: int
    cancelled: int
    time: object
        year: int
        month: int
```

- 400 Bad Request

  `empty`
- 401 Unauthorized

  `empty`
- 404 Not Found

  `empty`
- 5xx Internal Server Error

  `empty`

6. **GET /airports/{airport_code}/carriers/stats/delay-times?reason=carrier,late-aircraft&year={year}&month={month}**
   Returns the number of minutes of delay per carrier attributed to carrier-specific reasons, for a given year and month and for a specific airport.

   GET /airports/{airport_code}/carriers/stats/delay-times?year={year}&month={month}
   Returns the number of minutes of delay per carrier attributed to all reasons, for a given year and month and for a specific airport.

   GET /airports/carriers/stats/delay-times?reason=carrier,late-aircraft&year={year}&month={month}
   Returns the number of minutes of delay per carrier attributed to carrier-specific reasons, for a given year and month and for all US airports.

   GET /airports/carriers/stats/delay-times?year={year}&month={month}
   Returns the number of minutes of delay per carrier attributed to all reasons, for a given year and month and for all US airports.

   GET /airports/{airport_code}/carriers/stats/delay-times?reason=carrier,late-aircraft
   Returns the number of minutes of delay per carrier attributed to carrier-specific reasons, for all months available and for a specific airport.

   GET /airports/{airport_code}/carriers/stats/delay-times
   Returns the number of minutes of delay per carrier attributed to all reasons, for all months available and for a specific airport.

   GET /airports/carriers/stats/delay-times?reason=carrier,late-aircraft
   Returns the number of minutes of delay per carrier attributed to carrier-specific reasons, for all months available and for all US airports.

   GET /airports/carriers/stats/delay-times
   Returns the number of minutes of delay per carrier attributed to all reasons, for all months available and for all US airports.

   Endpoint path parameters:

   - `airport_code` The 3-letter code of the airport.
   - `carrier_code` The 3-letter code of the carrier.
   - `year` The year for which data should be returned.
   - `month` The month for which data should be returned.

   Request headers:

   - `Accept: application/json`
   - `Accept: text/csv`

Request body:

```
empty
```

Response headers:

- `Content-Type: application/json`
- `Content-Type: text/csv`

Responses with content:

- 200 OK Successful operation
  /airports/{airport_code}/carriers/stats/delay-times?reason=carrier,late-aircraft&year={year}&month={month}

```
: array[object]
    minutes-delayed: object
        carrier: int
        late-aircraft: int
    carrier: object
        code: string
```

  /airports/{airport_code}/carriers/stats/delay-times?year={year}&month={month}

```
: array[object]
    minutes-delayed: object
        late-aircraft: int
        weather: int
        carrier: int
        security: int
        national-aviation-system: int
    carrier: object
        code: string
```

  /airports/carriers/stats/delay-times?reason=carrier,late-aircraft&year={year}&month={month}

```
: array[object]
    carrier-delays: array[object]
        minutes-delayed: object
            carrier: int
            late-aircraft: int
        carrier: object
            code: string
    airport: object
        code: string
```

  /airports/carriers/stats/delay-times?&year={year}&month={month}

```
: array[object]
    carrier-delays: array[object]
        minutes-delayed: object
            late-aircraft: int
            weather: int
            carrier: int
```

```
            security: int
            national-aviation-system: int
        carrier: object
            code: string
    airport: object
        code: string
```

/airports/{airport_code}/carriers/stats/delay-times?reason=carrier,late-aircraft

```
: array[object]
    : array[object]
        minutes-delayed: object
            carrier: int
            late-aircraft: int
        carrier: object
            code: string
    time: object
        year: int
        month: int
```

/airports/{airport_code}/carriers/stats/delay-times

```
: array[object]
    : array[object]
        minutes-delayed: object
            late-aircraft: int
            weather: int
            carrier: int
            security: int
            national-aviation-system: int
        carrier: object
            code: string
    time: object
        year: int
        month: int
```

/airports/carriers/stats/delay-times?reason=carrier,late-aircraft

```
: array[object]
    : array[object]
        carrier-delays: array[object]
            minutes-delayed: object
                carrier: int
                late-aircraft: int
            carrier: object
                code: string
        airport: object
            code: string
    time: object
        year: int
        month: int
```

/airports/carriers/stats/delay-times

```
: array[object]
    : array[object]
        carrier-delays: array[object]
            minutes-delayed: object
                late-aircraft: int
                weather: int
                carrier: int
                security: int
                national-aviation-system: int
            carrier: object
                code: string
        airport: object
            code: string
    time: object
        year: int
        month: int
```

- 400 Bad Request

  `empty`
- 401 Unauthorized

  `empty`
- 404 Not Found

  `empty`
- 5xx Internal Server Error

  `empty`

7. **GET /airports/{airport_code_1}/{airport_code_2}/carriers/{carrier_code}/extra-stats/delay-times**
   Returns the descriptive statistics, mean, median, and standard deviation, for carrier-specific delays for a flight between any two airports in the USA for a specific carrier serving this route.

   GET /airports/{airport_code_1}/{airport_code_2}/carriers/extra-stats/delay-times
   Returns the descriptive statistics, mean, median, and standard deviation, for carrier-specific delays for a flight between any two airports in the USA for all carriers serving this route.

   Endpoint path parameters:

   - `airport_code_1` The 3-letter code of the first airport.
   - `airport_code_2` The 3-letter code of the second airport.
   - `carrier_code` The 3-letter code of the carrier.

   Request headers:

   - `Accept: application/json`
   - `Accept: text/csv`

   Request body:

```
empty
```

Response headers:

- `Content-Type: application/json`
- `Content-Type: text/csv`

Responses with content:

- 200 OK Successful operation
  /airports/{airport_code_1}/{airport_code_2}/carriers/{carrier_code}/extra-stats/delay-times

```
carrier: object
    mean: int
    median: int
    standard-deviation: int
late-aircraft: object
    mean: int
    median: int
    standard-deviation: int
```

  /airports/{airport_code_1}/{airport_code_2}/carriers/extra-stats/delay-times

```
: array[object]
    carrier-delay: object
        mean: int
        median: int
        standard-deviation: int
    late-aircraft: object
        mean: int
        median: int
        standard-deviation: int
    carrier: object
        code: string
```

- 400 Bad Request

  `empty`
- 401 Unauthorized

  `empty`
- 404 Not Found

  `empty`
- 5xx Internal Server Error

  `empty`

3 Server Responses Data

3.1 Data

The requested data will be served in JSON format. Next follow some examples of data in JSON returned by the server for a **GET** request to each endpoint:

1. **GET** request to:

   `/airports`

   JSON response example:

   `[ATL, CLT, SAN]`

   CSV response example:

ATL
CLT
SAN

2. **GET** request to:

   `/carriers`

   JSON response example:

   `[CO, VX, SAN]`

   CSV response example:

CO
VX
SAN

3. **GET** request to:

   `/airports/{airport_code}/carriers`

   JSON response example:

   `[CO, VX, SAN]`

   CSV response example:

CO
VX
SAN

4. **GET** request to:

   `/airports/{airport_code}/carriers/{carrier_code}/stats?year={year}&month={month}`

   JSON response example:

```
{
    "flights": {
        "cancelled": 5,
        "on time": 561,
        "total": 752,
        "delayed": 186,
        "diverted": 0
    },
    "# of delays": {
        "late aircraft": 18,
```

```
        "weather": 28,
        "security": 2,
        "national aviation system": 105,
        "carrier": 34
    },
    "minutes delayed": {
        "late aircraft": 1269,
        "weather": 1722,
        "carrier": 1367,
        "security": 139,
        "total": 8314,
        "national aviation system": 3817
    }
}
```


CSV response example:

Cancelled	OnTIme	Total	Delayed	Diverted
5	561	752	186	0

# of Delay(late aircraft)	# of Delay(weather)	# of Delay(security)	# of Delay(N.A.S)	# of Delay(carrier)
18	28	2	105	34

Min-Delayed(late aircraft)	Min-Delayed(weather)	Min-Delayed(carrier)	Min-Delayed(security)	Min-Delayed(total)
1269	1722	1367	139	8314

GET request to:

`/airports/{airport_code}/carriers/{carrier_code}/stats`

JSON response example:


```
[
    {
        "flights": {
            "cancelled": 5,
            "on time": 561,
            "total": 752,
            "delayed": 186,
            "diverted": 0
        },
        "# of delays": {
            "late aircraft": 18,
            "weather": 28,
            "security": 2,
            "national aviation system": 105,
            "carrier": 34
        },
        "minutes delayed": {
```

```
            "late aircraft": 1269,
            "weather": 1722,
            "carrier": 1367,
            "security": 139,
            "total": 8314,
            "national aviation system": 3817
        },
        "time": {
            "label": "2003/6",
            "year": 2003,
            "month": 6
        }
    },
    {
        "flights": {
            "cancelled": 7,
            "on time": 1034,
            "total": 1266,
            "delayed": 225,
            "diverted": 0
        },
        "# of delays": {
            "late aircraft": 46,
            "weather": 24,
            "security": 2,
            "national aviation system": 84,
            "carrier": 69
        },
        "minutes delayed": {
            "late aircraft": 3043,
            "weather": 1783,
            "carrier": 4201,
            "security": 45,
            "total": 12139,
            "national aviation system": 3067
        },
        "time": {
              "label": "2003/10",
              "year": 2003,
              "month": 10
        }
    }
]
```


5. **GET** request to:

   `/airports/{airport_code}/carriers/{carrier_code}/stats/flights?year={year}&month={month}`

   JSON response example:


   ```
   {
   ```

```
    "on time": 561,
    "delayed": 186,
    "cancelled": 5
}
```


GET request to:

`/airports/{airport_code}/carriers/{carrier_code}/stats/flights`

JSON response example:


```
[
    {
        "on time": 561,
        "delayed": 186,
        "cancelled": 5,
        "time": {
            "label": "2003/6",
            "year": 2003,
            "month": 6
        }
    },
    {
        "on time": 1034,
        "delayed": 225,
        "cancelled": 7,
        "time": {
              "label": "2003/10",
              "year": 2003,
              "month": 10
        }
    }
]
```


6. **GET** request to:

   `/airports/{airport_code}/carriers/stats/delay-times?reason=carrier,late-aircraft&year={year}&month={month}`

   JSON response example:


```
[
    {
        "minutes delayed": {
            "carrier": 1367,
            "late aircraft": 1269
        },
        "carrier": {
            "code": "AA",
            "name": "American Airlines Inc."
        }
    },
    {
        "minutes delayed": {
```

```
            "carrier": 69,
            "late aircraft": 277
        },
        "carrier": {
            "code": "XE",
            "name": "ExpressJet Airlines Inc."
        }
    }
]
```


GET request to:

`/airports/carriers/stats/delay-times?reason=carrier,late-aircraft&year={year}&month={month}`

JSON response example:


```
[
    {
        "carrier-delays": [
            {
                "minutes delayed": {
                    "carrier": 1367,
                    "late aircraft": 1269
                },
                "carrier": {
                    "code": "AA",
                    "name": "American Airlines Inc."
                }
            },
            {
                "minutes delayed": {
                    "carrier": 69,
                    "late aircraft": 277
                },
                "carrier": {
                    "code": "XE",
                    "name": "ExpressJet Airlines Inc."
                }
            }
        ],
        "airport": {
          "code": "ATL",
          "name": "Atlanta, GA: Hartsfield-Jackson Atlanta International"
        }
    },
    {
        "carrier-delays": [
            {
                "minutes delayed": {
                    "carrier": 5,
                    "late aircraft": 0
                },
```

```
                "carrier": {
                    "code": "AA",
                    "name": "American Airlines Inc."
                }
            },
            {
                "minutes delayed": {
                    "carrier": 690054684655,
                    "late aircraft": 15699900
                },
                "carrier": {
                    "code": "AL",
                    "name": "AlwaysLate Airlines Inc."
                }
            }
        ],
        "airport": {
          "code": "SEA",
          "name": "Seattle, WA: Seattle/Tacoma International"
        }
    }
]
```


GET request to:

`/airports/{airport_code}/carriers/stats/delay-times?year={year}&month={month}`

JSON response example:


```
[
    {
        "minutes delayed": {
            "late aircraft": 775,
            "weather": 155,
            "carrier": 1478,
            "security": 0,
            "national aviation system": 3343
        },
        "carrier": {
            "code": "AA",
            "name": "American Airlines Inc."
        }
    },
    {
        "minutes delayed": {
            "late aircraft": 456,
            "weather": 658,
            "carrier": 1414,
            "security": 2,
            "national aviation system": 3543
        },
        "carrier": {
```

```
            "code": "XE",
            "name": "ExpressJet Airlines Inc."
        }
    }
]
```


GET request to:

`/airports/carriers/stats/delay-times?year={year}&month={month}`

JSON response example:


```
[
    {
        "carrier-delays": [
            {
                "minutes delayed": {
                    "late aircraft": 775,
                    "weather": 155,
                    "carrier": 1478,
                    "security": 0,
                    "national aviation system": 3343
                },
                "carrier": {
                    "code": "AA",
                    "name": "American Airlines Inc."
                }
            },
            {
                "minutes delayed": {
                    "late aircraft": 123,
                    "weather": 321,
                    "carrier": 213,
                    "security": 312,
                    "national aviation system": 0
                },
                "carrier": {
                    "code": "XE",
                    "name": "ExpressJet Airlines Inc."
                }
            }
        ],
        "airport": {
          "code": "ATL",
          "name": "Atlanta, GA: Hartsfield-Jackson Atlanta International"
        }
    },
    {
        "carrier-delays": [
            {
                "minutes delayed": {
                    "late aircraft": 721,
```

```
                    "weather": 462,
                    "carrier": 5555,
                    "security": 10,
                    "national aviation system": 3343
                },
                "carrier": {
                    "code": "AA",
                    "name": "American Airlines Inc."
                }
            },
            {
                "minutes delayed": {
                    "late aircraft": 0,
                    "weather": 0,
                    "carrier": 0,
                    "security": 1000000000,
                    "national aviation system": 0
                },
                "carrier": {
                    "code": "AL",
                    "name": "AlwaysLate Airlines Inc."
                }
            }
        ],
        "airport": {
          "code": "SEA",
          "name": "Seattle, WA: Seattle/Tacoma International"
        }
    }
]
```


7. **GET** request to:

   `/airports/{airport_code_1}/{airport_code_2}/carriers/{carrier_code}/extra-stats/delay-times`

   JSON response example:


```
{
    "carrier": {
        "mean": 775,
        "median": 155,
        "standard deviation": 1478
    },
    "late aircraft": {
        "mean": 775,
        "median": 1554,
        "standard deviation": 1235
    }
}
```


   GET request to:

   `/airports/{airport_code_1}/{airport_code_2}/carriers/extra-stats/delay-times`

JSON response example:


```
[
    {
        "carrier delay": {
            "mean": 775,
            "median": 155,
            "standard deviation": 1478
        },
        "late aircraft": {
            "mean": 775,
            "median": 1554,
            "standard deviation": 1235
        },
        "carrier": {
            "code": "DL",
            "name": "Delta Air Lines Inc."
        }
    },
    {
        "carrier delay": {
            "mean": 775,
            "median": 155,
            "standard deviation": 1478
        },
        "late aircraft": {
            "mean": 775000,
            "median": 1589,
            "standard deviation": 1458
        },
        "carrier": {
            "code": "AL",
            "name": "AlwaysLate Airlines Inc."
        }
    }
]
```


4 ER Model

This is an alpha version of our ER model. We identify three main entities in our model: **Airports, Carries and Flights**. The first two entities are pretty straight forward. Each of contains a name and a code as their identifiers. For the third entity things are a bit more complicated. Flights are mainly identified by the flight code. However we also need to keep track of the statistics of all flights. To do that we decided to split the Flight entity into 4 sub-entities. **On-time, Cancelled Delayed and Diverted.** For each of those sub categories we keep track of the statistics. We have a large number of statistics. First of all we need to keep track the reason a flight has been delayed due to different reasons(N.A.S,aircraft,weather,carrier),which are provided to us in both json and csv files. Secondly we also need to keep track of the total time(in minutes) of each of those reason seperately and the combined time of all those reasons.

Since this is an Alpha version of our ER model, some of the entities and identifiers might change in the future. However we feel its important to provide a starting point for a visual representation of our thought process.