University of Groningen

Web Engineering

Group 5

# API Documentation

**Authors:**
**Alina Boshchenko**
**Ai Deng**
**Henry Salas**

March 4, 2019

# Contents

# 1

## 1.1 Get all airports available in the USA.

**Description:**
Get all airports available in the USA.

**URL:**
GET /airports

**Header Fields:**

| **Path Parameter** | **Value** |
|---|---|
| Content-Type | `application/json` or `text/csv` |

**Response Format:**

Code 200:
A JSON response which contains the links for all the airports in USA:

```
{
    airports: [
        {
            ”aiport_name”: string,
            ”aiport_code”: string,
            ”link”: string,
        },
        ...
    ]
}
```

For example:

```
{
    airports: [
        {
            ”aiport_name”: ”John F. Kennedy International Airport”,
            ”aiport_code”: ”JFK”,
            ”link”: ”airport/JFK”,
        },
        {
            ”aiport_name”: ”Los Angeles International Airport”,
            ”aiport_code”: ”LAX”,
            ”link”: ”airport/LAX”,
            }
        ]
}
```

## 1.2 Get a specific airport in the USA with its airport_code.

**Description:**
Gets a specific airport with its airport_code.

**URL:**
GET /airport/{airport_code}

**Request Parameters:**

| **Path Parameter** | **Value** |
|---|---|
| airport_code | The airport code, for example: "JFK" |

**Header Fields:**

| **Path Parameter** | **Value** |
|---|---|
| Content-Type | `application/json` or `text/csv` |

**Response Format:**

Code 200:
As JSON:

```
{
    "airport_name": string,
    "airport_code": string
}
```

Code 400:
Client syntax error.

Code 404:
Not found.

**Sample call:**

1. A query to get information for airport JFK:

```
/airport/JFK
```

# 2

## 2.1 Get all carriers available in USA.

**Description:**
Get all carriers available in USA.

**URL:**
GET /carriers

**Header Fields:**

| Path Parameter | Value |
|---|---|
| Content-Type | `application/json` or `text/csv` |

**Response Format:**

<u>Code 200:</u>

A JSON response which contains the links for all the carriers in the USA:

```
{
    ”carriers”: [
        {
            ”carrier_name”: string,
            ”carrier_code”: string,
            ”link”: string,
        },
        ...
    ]
}
```

## 2.2 Get a specific carrier with its carrier_code.

**Description:**
Gets a specific carrier with its carrier_code.

**URL:**
GET /carrier/{carrier_code}

**Request Parameters:**

| **Path Parameter** | **Value** |
|---|---|
| carrier_code | The carrier code, for example: "KLM" |

**Header Fields:**

| **Path Parameter** | **Value** |
|---|---|
| Content-Type | `application/json` or `text/csv` |

**Response Format:**

Code 200:

As JSON:

```
{
    "carrier_name": string
    "carrier_code": string
}
```

Code 400:
Client syntax error.

Code 404:
Not found.

**Sample call:**
A query to get information for carrier KLM:
`/carrier/KLM`

# 3

## 3.1 Get all carriers operating at a specific airport.

**Description:**
Gets all carriers operating at the specific airport.

**URL:**
GET /airport/{airport_code}/carriers

**Request Parameters:**

| **Path Parameter** | **Value** |
|---|---|
| airport_code | The airport code, for example: ”SVO” |

**Header Fields:**

| **Path Parameter** | **Value** |
|---|---|
| Content-Type | `application/json` or `text/csv` |

**Response Format:**

Code 200:

A JSON response which contains the information for all carriers for a specific airport:

```
{
    ”carriers”: [
        {
            ”carrier_code”: string,
            ”carrier_name”: string,
            ”link”: string,
        },
        ...
    ]
}
```

Code 400:

Client syntax error.

Code 404:

Not found.

**Sample call:**
A query to get information about all carriers in airport AMS:
`GET /airport/AMS/carriers`

# 4

## 4.1 Get statistics about flights of a carrier for a given date.

**Description**

Gets all the statistics about the flights of a carrier for a specific date and route. This endpoint allows for filtering the statistics you get.

**URL:**

```
GET /carrier/{carrier_code}/statistics/flights?
   route = {route}&
   month = {month}&
   year = {year}&
   statistics_filter = {statistics_filter}
```

**Request Parameters:**

| **Path Parameter** | **Value** |
|---|---|
| carrier_code | The carrier code. For example: ”klm” |

**Header Fields:**

| **Path Parameter** | **Value** |
|---|---|
| Content-Type | `application/json` or `text/csv` |

**Query Parameters:**

| Query Parameter | Value Type | Value |
|---|---|---|
| route | `string` | The type of route:<br>`"from"`: Get the data from a US airport<br>`"to"`: Get the data to a US airport |
| month | `unsigned int` | *Optional.*<br>The month for which to get the statistics.<br>A value from 1-12.<br>If month is not given, gets it for the given year.<br>If no month or year given, returns for all time. |
| year | `unsigned int` | *Optional.*<br>The year for which to get the statistics.<br>A 4 digit number representing a year.<br>If year is not given, gets it for the given months.<br>If no month or year given, returns for all time. |
| statistics_filter | `array` | *Optional.*<br>Allows you to filter statistics results. You only need to include the strings you wish to get results for:<br>Possible filters:<br>`[`<br>`"cancelled",`<br>`"on_time",`<br>`"total",`<br>`"delayed",`<br>`"diverted",`<br>`]` |

**Response Format:**

Code 200:

A complete JSON response (with no statistics filtering):

```
{
    ”route”: string ,
    ”month”: unsigned int ,
    ”year”: unsigned int ,
    ”statistics”: {
        ”cancelled”: unsigned int | null ,
        "on_time": unsigned int | null,
        "total": unsigned int | null,
        "delayed": unsigned int | null,
        "diverted": unsigned int | null
    }
}
```

Code 400:
Client syntax error.

Code 404:
Not found.

**Sample call:**

1. A query to get the statistics for KLM, for the date July, 2018, for flights that are "from":
`GET carrier/klm/flight_statistics?route=from&month=7&year=2018`

Returns a successful (code: 200) response with the structure described above. Note that since we did not specify `content-type` in the header field, we will receive a JSON format by default.

2. (as well a specific example of question 5) A query to get the number of on-time, delayed, and cancelled flights statistics for KLM, for the year 1999, for the flights that are "to":

```
GET /carrier/klm/flight_statistics?route=to&year=1999&
statistics_filter=[‘on_time’, ‘delayed’, ‘cancelled’]
```

Returns a 200 code HTTP response with the body described above as a JSON.

### 4.2 Insert new statistics about flights of a carrier for a given date.

**Description**

Insert new statistics about the flights of a carrier for a specific date and route.

**URL:**

`POST /carrier/{carrier_code}/statistics/flights`

**Request Parameters:**

| **Path Parameter** | **Value** |
|---|---|
| carrier_code | The carrier code. For example: ”klm”. |

**Header Fields:**

| **Path Parameter** | **Value** |
|---|---|
| Authorization | A valid access token |

**Body Parameters:**

| **Request Body Data** | **Value Type** | **Value** |
|---|---|---|
| route | `string` | The route type of data:<br>`"from"`: Get the data from a US airport<br>`"to"`: Get the data to a US airport |
| month | `unsigned int` | The month for which to get the statistics.<br>A value from 1-12. |
| year | `unsigned int` | The year for which to get the statistics.<br>A 4 digit number representing a year. |
| statistics | `array` | A **complete** `array` containing the statistics to add:<br>`[`<br>`"cancelled" => unsigned int` \| `null`,<br>`"on_time" => unsigned int` \| `null`,<br>`"total" => unsigned int` \| `null`,<br>`"delayed" => unsigned int` \| `null`,<br>`"diverted" => unsigned int` \| `null`,<br>`]`<br>NOTE:<br>- Complete means each key above should be included.<br>- `null` should be placed if key value is unknown. |

**Response Format:**

Code 200:
Signifies a successful addition of data.

Code 400:
Client syntax error or combination of route, and date is not unique.

**Sample call:**

1. Adding flight statistics for KLM for the date of March, 1997 with the route of ”to”. Note that if we do not know the value of one of the statistics, we are free to place `null` (we did so for `"total"` as an example).
`POST /carrier/klm/flight_statistics`

Body parameters:

```
[
    ”month” => 3,
    ”year” => 1997,
    ”route” => ”to”,
    ”statistics” => [
        ”cancelled” => 12412,
        ”on_time” => 1234,
        ”total” => null,
        ”delayed” => 3242,
        ”diverted” => 25436
    ]
]
```

Code 200: Successful addition, assuming that the combination of the given date, carrier, and route are unique in the database (otherwise, code 400).

## 4.3 Delete all statistics about flights of a carrier for a particular date and route.

**Description**

Deletes all the flight statistics of a carrier for a particular date and route.

**URL:**

```
DELETE /carrier/{carrier_code}/statistics/flights?
   route = {route}&
   month = {month}&
   year = {year}
```

**Request Parameters:**

| Path Parameter | Value |
|---|---|
| carrier_code | The carrier code, for example: ”klm” |

**Header Fields:**

| Path Parameter | Value |
|---|---|
| Authorization | A valid access token |

**Query Parameters:**

| Query Parameter | Value Type | Value |
|---|---|---|
| route | `string` | The type of data:<br>`"from"`: Delete the data from a US airport<br>`"to"`: Delete the data to a US airport |
| month | `unsigned int` | The month for which to get the statistics.<br>A value from 1-12. |
| year | `unsigned int` | The year for which to get the statistics.<br>A 4 digit number representing a year. |

**Response Format:**

Code 200: Signifies a successful deletion of data.

Code 400: Client syntax error or combination of route and date is invalid.

Code 404: Not found.

**Sample call:**

Deletes the flight statistics for KLM for July 2018 for ”from” US airports.

```
DELETE carrier/klm/flight_statistics?route=from&month=7&year=2018
```

## 4.4 Updates the flight statistics of a carrier for a particular date and route.

**Description**

Updates the flight statistics of a carrier using its carrier_code, a particular date and route. Notice that this endpoint only updates the statistics for a given carrier, date and route. It is not possible to update the carrier, date, or route themselves as they are essentially the keys for the statistics. If these have to be updated, the statistics should be first deleted and added over again with the updated information.

**URL:**

```
PUT /carrier/{carrier_code}/statistics/flights?
   route = {route}&
   month = {month}&
   year = {year}
```

**Request Parameters:**

| Path Parameter | Value |
|---|---|
| carrier_code | The carrier code. For example: ”klm”. |

**Header Fields:**

| Path Parameter | Value |
|---|---|
| Authorization | A valid access token |

**Query Parameters:**

| Query Parameter | Value Type | Value |
|---|---|---|
| route | `string` | The type of data:<br>`"from"`: Update the data from a US airport<br>`"to"`: Update the data to a US airport |
| month | `unsigned int` | The month for which to update the statistics.<br>A value from 1-12. |
| year | `unsigned int` | The year for which to update the statistics.<br>A 4 digit number representing a year. |

**Body Parameters:**

| Request Body Data | Value Type | Value |
|---|---|---|
| statistics | `array` | A **complete** `array` containing the statistics to add:<br>`[`<br>`"cancelled" => unsigned int` \| `null`,<br>`"on_time" => unsigned int` \| `null`,<br>`"total" => unsigned int` \| `null`,<br>`"delayed" => unsigned int` \| `null`,<br>`"diverted" => unsigned int` \| `null`,<br>`]`<br>Complete means each key above should be included.<br>Null should be placed if key value is unknown. |

**Response Format:**
Code 200:
Signifies a successful update of data.

Code 400:
Client syntax error or invalid request (such as choosing to update an already existing combination of date, carrier, and route).

**Sample call:**
1. Updating flight statistics for KLM for the date of February, 1996 with the route of ”to”.

```
PUT /carrier/klm/flight_statistics?route=to&month=2&year=1996
```

Body parameters:

```
[
    ”cancelled” => 1642,
    ”on_time” => 423,
    ”total” => 545,
    ”delayed” => 689,
    ”diverted” => null
]
```

Code 200:

Successful update, assuming that the combination of the given date, carrier, and route already exist in the database (otherwise, code 400).

# 5 Included in section 4.1

Included in section 4.1, as it is the same as requirement 4 but allows for filtering.

# 6

## 6.1 Get statistics in minutes about flights of a carrier for all carriers for a given date for selected airport.

**Description:**
Get number of minutes of delay per carrier attributed to carrier-specific reasons (i.e. attributes carrier and late aircraft in the dataset)/all reasons, for a given month/all months available and for a specific airport/across all US airports.

**URL:**

```
GET /carriers/statistics/minutes_delayed?
      airport_code={airport_code}&
      month={month}&
      year={year}
      reasons={reasons}
```

**Header Fields:**

| **Path Parameter** | **Value** |
|---|---|
| Content-Type | application/json or text/csv |

**Query Parameters:**

| **Query Parameter** | **Value Type** | **Value** |
|---|---|---|
| airport_code | `string` | *Optional.*<br>The code of the airport. |
| month | `unsigned int` | *Optional.*<br>The month for which to get the statistics.<br>A value from 1-12.<br>If month is not given, gets it for the given year.<br>If no month or year given, returns for all time. |
| year | `unsigned int` | *Optional.*<br>The year for which to get the statistics.<br>A 4 digit number representing a year.<br>If year is not given, gets it for the given months.<br>If no month or year given, returns for all time. |
| reasons | `array` | An `array` containing the reasons to add:<br>`[`<br>`"late_aircraft" => unsigned int` \| `null`,<br>`"carrier" => unsigned int` \| `null`,<br>`"total" => unsigned int` \| `null`,<br>`]`<br>Null should be placed if key value is unknown. |

**Response Format:**

Code 200:
As JSON:

```
{
    "carriers": [
        {
            "carrier": {
                "carrier_code": string,
                "carrier_name": string,
                "link": string
            }
            "airport_code": string,
            "month": unsigned int,
            "year": unsigned int,
            "reasons": {
                "late_aircraft": unsigned int | null,
                "carrier":unsigned int | null,
                "total": unsigned int | null
            }
        },
        ...
    ]
}
```

Code 400:
Client syntax error.

Code 404:
Not available.

**Sample call:**

1. A query to get the statistics of delays in minutes for all carriers for late previous aircraft reason, for the date June, 2017 in Schiphol airport:

```
/carriers/statistics/minutes_delayed?
  airport_code=AMS&
  month=6&year=2017&
  reasons=["late_aircraft"]
```

Returns a successful (code: 200) response with the array of structures described above. Note that since we did not specify `content-type` in the header field, we will receive a JSON format by default.

2. A query to get the statistics for all carriers for the year 2007, for delays for all reasons in Schiphol airport:

```
/carriers/statistics/minutes_delayed?
  airport_code=AMS&
  year=2007&
  reasons=["total"]
```

Returns a successful (code: 200) response with the structure described above. Note that since we did not specify `content-type` in the header field, we will receive a JSON format by default. Also month is not specified so it returns statistics for all months for given year.

# 7

## 7.1 Get statistics for all carriers delays between 2 specific airports.

**Description**
Get statistics for all carriers delays between 2 specific airports.

**URL:**

```
GET /carriers/statistics/delays?
  airport1 = {airport_code}&
  airport2 = {airport_code}
```

**Query Parameters:**

| Query Parameter | Value Type | Value |
|---|---|---|
| airport1 | `string` | airport_code |
| airport2 | `string` | airport_code |

**Header Fields:**

| Path Parameter | Value |
|---|---|
| Content-Type | `application/json` or `text/csv` |

**Response Format:**

Code 200:

A JSON response with descriptive statistics (mean, median, standard deviation) for carrier-specific delays (as above) for a flight between any two airports in the USA for all carriers serving this route:

```
{
    "airport1" : {
        "airport_name": string ,
        "airport_code": string ,
        "link": string
    },
    "airport2" : {
        "airport_name": string,
        "airport_code": string,
        "link": string
    },
    "mean": float,
    "median": float,
    "standard_deviation": float
}
```

<u>Code 400:</u>
Client syntax error.

<u>Code 404:</u>
Not available.

**Sample call:**

1. Get all delays statistics between airport LAX and JFK.

`/carriers/statistics/delays?airport1=LAX&aiport2=JFK`

## 7.2 Get statistics for a specific carrier delays between 2 airports.

**Description** Get statistics for a specific carrier delays between 2 specific airports.

**URL:**

```
GET  /carrier/{carrier_code}/statistics/delays?
  airport1 = {airport_code}&
  airport2 = {airport_code}
```

**Request Parameters:**

| Path Parameter | Value |
|---|---|
| carrier_code | The carrier code. For example: ”klm” |

**Header Fields:**

| Path Parameter | Value |
|---|---|
| Content-Type | `application/json` or `text/csv` |

**Query Parameters:**

| Query Parameter | Value Type | Value |
|---|---|---|
| airport1 | `string` | airport_code |
| airport2 | `string` | airport_code |

**Response Format:**

Code 200:

A JSON response with descriptive statistics (mean, median, standard deviation) for carrier-specific delays (as above) for a flight between any two airports in the USA for a specific carrier/all carriers serving this route:

```
{
    "carrier": {
        "carrier_name": string;
        "carrier_code": string;
        "link": string
    },
    "airport1": {
        ”airport_name”: string,
        ”airport_code”: string,
        ”link”: string
    },
    ”airport2”: {
        ”airport_name”: string,
        ”airport_code”: string,
        ”link”: string
    },
    ”mean”: float,
    ”median”: float,
    ”standard_deviation”: float

}
```

Code 400:
Client syntax error.

Code 404:
Not available.

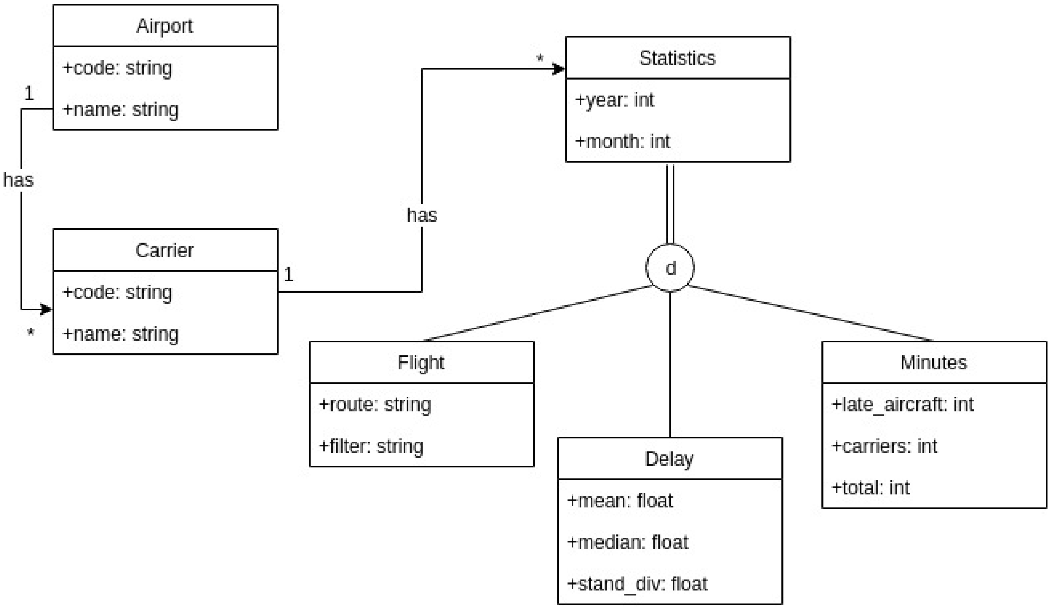
**Sample call:**

1. Get all delays statistics for the carrier KLM between airport LAX and JFK.

`/carrier/KLM/statistics/delays?airport1=LAX&aiport2=JFK`

Returns a successful (code: 200) response with the structure with statistics described above. Note that since we did not specify `content-type` in the header field, we will receive a JSON format by default.

## 8 Data model



NOTE: **d** in statistics relation means that statistics could not overlap with each other