University of Groningen

Web Engineering

Group 5

API Documentation

Authors:
Alina Boshchenko
Ai Deng
Henry Salas

March 21, 2019

rijksuniversiteit
groningen

Contents

Note: All API endpoints begin with the prefix: `/API/`

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
Accept	`application/json` or `text/csv`

Response Format:

Code 200:

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

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

For example:

```
[
    {
        "aiport_name": "John F. Kennedy International Airport",
        "aiport_code": "JFK",
        "link": "API/airports/JFK",
    },
    {
        "aiport_name": "Los Angeles International Airport",
        "aiport_code": "LAX",
        "link": "API/airports/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 /airports/{airport_code}`

Header Fields:

Path Parameter	Value
Accept	`application/json` or `text/csv`

Request Parameters:

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

Response Format:

Code 200:
   As JSON:

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

   Code 404:
   Not found.

Sample call:
1. A query to get information for airport JFK:

   `/API/airports/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
Accept	`application/json` or `text/csv`

Response Format:
Code 200:
A JSON response which contains the links for all the carriers in the USA:

```
[
    {
        "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
Accept	application/json or text/csv

Response Format:

Code 200:
    As JSON:

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

    Code 404:
    Not found.

Sample call:
A query to get information for carrier KLM:
API/carriers/KLM

3

3.1   Get all carriers operating at a specific airport.

Description:
Gets all carriers operating at the specific airport.

URL:
GET /airports/{airport_code}/carriers

Request Parameters:

Path Parameter	Value
airport_code	The airport code, for example: "SVO"

Header Fields:

Path Parameter	Value
Accept	`application/json` or `text/csv`

Response Format:
Code 200:
A JSON response which contains the information for all carriers for a specific airport:

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

Code 404:
Not found.

Sample call:
A query to get information about all carriers in airport AMS:
GET API/airports/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 airport code. This endpoint allows for filtering the statistics you get.

URL:

```
GET /carriers/{carrier_code}/statistics/flights?
    route = {route}&
    month = {month}&
    year = {year}&
    airport_code = {airport_code}&
    filter = {statistics_filter}
```

Request Parameters:

Path Parameter	Value
carrier_code	The carrier code. For example: "klm"

Header Fields:

Path Parameter	Value
Accept	`application/json` or `text/csv`

Query Parameters:

Query Parameter	Value Type	Value
route	`string`	The type of route: `"from"`: Get the data from a US airport `"to"`: Get the data to a US airport
month	`unsigned int`	*Optional.* The month for which to get the statistics. A value from 1-12. If month is not given, gets it for the given year. If no month or year given, returns for all time.
year	`unsigned int`	*Optional.* The year for which to get the statistics. A 4 digit number representing a year. If year is not given, gets it for the given months. If no month or year given, returns for all time.
airport_code	string	The airport code, for example: "JFK".

filter	`array`	*Optional.*
		
		 Allows you to filter statistics results. You only need to include the strings you wish to get results for:
		
		 Possible filters:
		
		 `[`
		   `"cancelled",`
		   `"on_time",`
		   `"total",`
		   `"delayed",`
		   `"diverted",`
		 `]`

Response Format:

<u>Code 200</u>:

A complete JSON response (with no statistics filtering):

```
{
    "route": string,
    "month": unsigned int,
    "year": unsigned int,
```

```
    "statistics": {
        "cancelled": unsigned int,
        "on_time": unsigned int,
        "total": unsigned int,
        "delayed": unsigned int,
        "diverted": unsigned int
    }
}
```

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" for airport LAX:
```
GET API/carriers/klm/flight_statistics?route=from&month=7&year=2018
&airport_code=LAX
```

Returns a successful (code: 200) response with the structure described above. Note that since we did not specify `Accept` 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" for airport LAX:
```
GET API/carriers/klm/flight_statistics?route=to&year=1999&
filter=['on_time', 'delayed', 'cancelled']
airport_code=LAX
```

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 airport code.

URL:

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

Request Parameters:

Path Parameter	Value
carrier_code	The carrier code. For example: "klm".

Body Parameters:

Request Body Data	Value Type	Value
month	unsigned int	The month for which to get the statistics. A value from 1-12.
year	unsigned int	The year for which to get the statistics. A 4 digit number representing a year.
airport_code	string	The airport code, for example: "JFK".
cancelled	unsigned int	The cancelled statistics
on_time	unsigned int	The on_time statistics
total	unsigned int	The total amount.
delayed	unsigned int	The number of delayed statistics.
diverted	unsigned int	The number of diverted statistics.

Response Format:

<u>Code 200:</u>
Signifies a successful addition of data.

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

Sample call:
1. Adding flight statistics for KLM for the date of March, 1997 for airport LAX.

```
POST API/carriers/klm/flight_statistics
```

<u>Body parameters:</u>

```
"month" => 3,
"year" => 1997,
"airport_code" => "LAX"
"cancelled" => 12412,
"on_time" => 1234,
"total" => 9833,
"delayed" => 3242,
"diverted" => 25436
```

Code 200.

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

Description

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

URL:
```
DELETE /carriers/{carrier_code}/statistics/flights?
    month = {month}&
    year = {year}
    airport_code = {airport_code}
```

Request Parameters:

Path Parameter	Value
carrier_code	The carrier code, for example: "klm"

Query Parameters:

Query Parameter	Value Type	Value
month	unsigned int	The month for which to get the statistics. A value from 1-12.
year	unsigned int	The year for which to get the statistics. A 4 digit number representing a year.
airport_code	string	The airport code, for example: "JFK".

Response Format:

Code 200: Signifies a successful deletion of data.

Code 400: Client syntax error.

Sample call:

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

```
DELETE API/carriers/klm/flight_statistics?&month=7
&year=2018&airport_code=LAX
```

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

Description

Updates the flight statistics of a carrier using its carrier_code, a particular date and airport code. Notice that this endpoint only updates the statistics for a given carrier, date and airport code. It is not possible to update the carrier, date, or airport code 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?
   month = {month}&
   year = {year}
   airport_code = {airport_code}
```

Request Parameters:

Path Parameter	Value
carrier_code	The carrier code. For example: "klm".

Query Parameters:

Query Parameter	Value Type	Value
month	unsigned int	The month for which to update the statistics. A value from 1-12.
year	unsigned int	The year for which to update the statistics. A 4 digit number representing a year.
airport_code	string	The airport code, for example: "JFK".

Body Parameters:

Request Body Data	Value Type	Value
cancelled	unsigned int	The cancelled statistics
on_time	unsigned int	The on_time statistics
total	unsigned int	The total amount.
delayed	unsigned int	The number of delayed statistics.
diverted	unsigned int	The number of diverted statistics.

Response Format:

Code 200:

Signifies a successful update of data.

Code 400:

Client syntax error or invalid request.

Sample call:

1. Updating flight statistics for KLM for the date of February, 1996 with airport LAX.

```
PUT API/carriers/klm/flight_statistics?airport_code=LAX&month=2&year=1996
```

Body parameters:

```
"cancelled" => 1642,
"on_time" => 423,
"total" => 545,
"delayed" => 689,
"diverted" => null
```

Code 200:
Successful update.

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
Accept	application/json or text/csv

Query Parameters:

Query Parameter	Value Type	Value
airport_code	string	*Optional.* The code of the airport.
month	unsigned int	*Optional.* The month for which to get the statistics. A value from 1-12. If month is not given, gets it for the given year. If no month or year given, returns for all time.
year	unsigned int	*Optional.* The year for which to get the statistics. A 4 digit number representing a year. If year is not given, gets it for the given months. If no month or year given, returns for all time.

reasons	array	An `array` containing the reasons to get:
		`[`
		`  "late_aircraft",`
		`  "carrier",`
		`  "total"`
		`]`

Response Format:

<u>Code 200:</u>
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.

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

Returns a successful (code: 200) response with the structure described above. Note that since we did not specify `Accept` 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:
```
API/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 `Accept` 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
Accept	`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:

```json
{
    "airport1" : {
        "airport_name": string,
        "airport_code": string,
        "link": string
    },
    "airport2" : {
        "airport_name": string,
        "airport_code": string,
        "link": string
```

19

```
        },
        "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.

```
API/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  /carriers/{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
Accept	`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 `Accept` in the header field, we will receive a JSON format by default.

8 Extras

8.1 Rankings of carriers for different combinations of data

Description Rankings of carriers for number of delays or ratio of cancellation for a given year.

URL:

```
GET  carriers/rankings/{ranking_type}?year={year}
```

Request Parameters:

Path Parameter	Value
ranking_type	The ranking type. e.g: "number_of_delays"

Header Fields:

Path Parameter	Value
Accept	`application/json` or `text/csv`

Query Parameters:

Query Parameter	Value Type	Value
year	`unsigned int`	year

Response Format:

Code 200:

A JSON response with descriptive ranking for the number of delays for all carrier in given year.

```
[
    {
        "rank": int,
        "carrier_code": string,
        "num_of_delays": int,
        "carrier_link": "string
    },
    ...
]
```

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

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

Sample call:

1. Get the number of delays ranking for year 2006.

```
carriers/rankings/number_of_delays?year=2006
```

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

2. Get the number of ratio of cancellation ranking for year 2003.

```
carriers/rankings/ratio_of_cancellation?year=2003
```

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

8.2   Posting reviews

Description Option for users to comment about their own experience about delays with a specific carrier (without login), including user ranking of carriers.

URL:
```
POST reviews/{user_name}
```

Request Parameters:

Path Parameter	Value
user_name	The user name. For example: "pikachu"

Header Fields:

Path Parameter	Value
Accept	`application/json` or `text/csv`

Body Parameters:

Query Parameter	Value Type	Value
review	`string`	The user's review.
carrier_code_rank_1	`string`	The user's most satisfied carrier.
carrier_code_rank_2	`string`	The user's second satisfied carrier.
carrier_code_rank_3	`string`	The user's third satisfied carrier.

Response Format:
Code 200: Signifies a successful deletion of data.

Code 400: Client syntax error.

Sample call:
1. Adding review for user "Henry Salas"
```
POST API/reviews/HenrySalas
```

Body parameters:
```
"review" => "The food provided on the plane is bad.",
"carrier_code_rank_1" => "AA",
```

```
"carrier_code_rank_1" => "AB",
"carrier_code_rank_1" => "AC",
```

Code 200.

8.3 Getting reviews

Description Get all previously posted reviews for a given user name (see section above)

URL:
```
GET reviews/{user_name}
```

Request Parameters:

Path Parameter	Value
user_name	The user name. For example: "Henry Salas"

Header Fields:

Path Parameter	Value
Accept	application/json or text/csv

Response Format:

<u>Code 200:</u>

A JSON response which contains the links for all the reviews for the given user.

```
[
    {
        "user_name": string,
        "reviews": string,
        "carrier_code_rank_1": string,
        "carrier_code_rank_2": string,
        "carrier_code_rank_3": string,
        "timestamp": {
            "date": string,
            "timezone_type": int,
            "timezone": string
        },
        "link": string
    },
    ...
]
```

<u>Code 400:</u>

Client syntax error.

Code 404:
Not available.

Sample call:
1. Getting reviews for user "Henry Salas"
`GET API/reviews/HenrySalas`

Code 200.

8.4 Getting a single review with given review id and user name

Description Getting a single review with given review id and user name.

URL:
```
GET reviews/{user_name}/{review_id}
```

Request Parameters:

Path Parameter	Value
user_name	The user name. For example: "Henry Salas"
review_id	The review id. For example: 1

Header Fields:

Path Parameter	Value
Accept	`application/json` or `text/csv`

Response Format:

Code 200:

A JSON response which contains the single review.

```
{
        "user_name": string,
        "reviews": string,
        "carrier_code_rank_1": string,
        "carrier_code_rank_2": string,
        "carrier_code_rank_3": string,
        "timestamp": {
            "date": string,
            "timezone_type": int,
            "timezone": string
        }
}
```
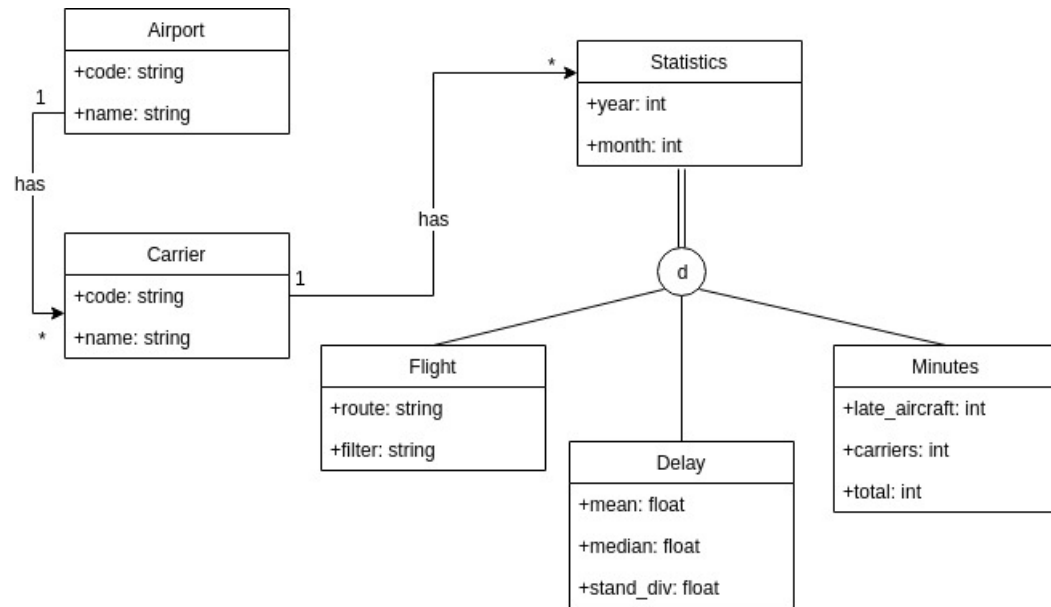
Code 400:

Client syntax error.

Code 404:

Not available.

Sample call:
1. Getting a review for user "HenrySalas"
`GET API/reviews/HenrySalas/1`

Code 200.

9   Data model

Airport
+code: string
+name: string

Carrier
+code: string
+name: string

Statistics
+year: int
+month: int

Flight
+route: string
+filter: string

Delay
+mean: float
+median: float
+stand_div: float

Minutes
+late_aircraft: int
+carriers: int
+total: int

has

has

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