PROG8170 SEC 3

Rutvik Gandhi (8809972)

Ricky Satyam (8826138)

Assignment #4 – API/Integration Testing

April 5th, 2023

1. Collection is uploaded with the assignment to eConestoga (also pasted below in the document as well).
2. 10 unique API calls using different endpoints.
3. GET /airports

Graphical user interface, text, application, email

Description automatically generated

1. GET /airports/:id (id = KIX)

Graphical user interface, text, application, email

Description automatically generated

1. POST /airports/distance (from = KIX to = NRT)

Graphical user interface, text, application, email

Description automatically generated

1. POST /tokens

Graphical user interface, text, application, email

Description automatically generated

1. GET /favorites

Graphical user interface, text, application, email

Description automatically generated

1. GET /favorites/:id (id = 22)

Graphical user interface, text, application, email

Description automatically generated

1. POST /favorites (airport\_id = JFK)

Graphical user interface, application, Teams

Description automatically generated

1. PATCH /favorites/:id (id = 8915)

Graphical user interface, text, application, email

Description automatically generated

1. DELETE /favorites/:id (id = 8915)

Graphical user interface, text, application

Description automatically generated

1. DELETE /favorites/clear\_all

Graphical user interface, text, application, email

Description automatically generated

1. The 20 unique requests:

|  |  |  |  |
| --- | --- | --- | --- |
| Test ID | Test Steps | Expected Outcome | Actual Outcomes |
| API001 | Create a GET request for airport information using POSTMan at https://airportgap.dev-tester.com/api/airports. | Verify that we receive a 200 – OK response along with the correct data. | PASS – see a. from previous section. |
| API002 | Create a GET request with POSTMan to get information about the Japan Osaka airport: https://airportgap.dev-tester.com/api/airports/KIX | Verify that we receive a 200 – OK response along with the correct data. | PASS – see b. from previous section. |
| API003 | Create a POST request for the following info using POSTMan:  "to": "NRT," "from": "KIX"  https://airportgap.dev-tester.com/api/airports/distance | Verify that we receive a 200 – OK response along with the correct data. | PASS – see c. from previous section. |
| API004 | Create a POST request for the following info using POSTMan:  "Password": "airportgappassword," "email": "test@airportgap.com"  https://airportgap.dev-tester.com/api/tokens | Verify that we receive a 200 – OK response along with the correct data. | PASS – see d. from previous section. |
| API005 | Create a GET request with POSTMan to obtain all stored information for preferred airports using the site's authorization token: https://airportgap.dev-tester.com/api/favorites | Verify that we receive a 200 – OK response along with the correct data. | PASS – see e. from previous section. |
| API006 | Using POSTMan, create a GET request to retrieve airport saved by ID data with authorization token provided from the site: using following data: {“id” : “22”} https://airportgap.dev-tester.com/api/favorites | Verify that we receive a 200 – OK response along with the correct data. | PASS – see f. from previous section. |
| API007 | Using POSTMan, create a POST request to create the following data with authorization token provided from the site:  {“airport\_id” : “KIX”} https://airportgap.dev-tester.com/api/favorites | Verify that we receive a 201 – Created response along with the correct data. | PASS – see g. from previous section. |
| API008 | Create a PATCH request using POSTMan to change the following data using the site's permission token:  {“id” : “8915”}  https://airportgap.dev-tester.com/api/favorites | Verify that we receive a 200 – OK response along with the correct data. | PASS – see h. from previous section. |
| API009 | Create a DELETE request for the following data using POSTMan and the site's supplied access token:  {“id” : “8915”}  https://airportgap.dev-tester.com/api/favorites | Verify that we receive a 204 – No Content response along with the correct data. | PASS – see i. from previous section. |
| API0010 | Create a DELETE request with the permission key from the website https://airportgap.dev-tester.com/api/favorites/clear\_all using POSTMan. | Verify that we receive a 204 – No Content response along with the correct data. | PASS – see j. from previous section. |
| API0011 | Create a GET request with POSTMan to get information about the JFK airport: https://airportgap.dev-tester.com/api/airports/JFK | Verify that we receive a 200 – OK response along with the correct data. | PASS – response status code was 200 OK |
| API0012 | generate a POST request using POSTMan to generate the following data using the site's permission token:  Airport code "JFK"  https://airportgap.dev-tester.com/api/favorites | Verify that we receive a 201 – Created response along with the correct data. | PASS – response status code was 201 Created |
| API0013 | Create a DELETE request for the following data using POSTMan and the site's supplied access token:  {“id” : “8938”}  https://airportgap.dev-tester.com/api/favorites | Verify that we receive a 204 – No Content response along with the correct data. | PASS – response status code was 204 No Content |
| API0014 | Create a DELETE request for the following data using POSTMan and the site's supplied access token:  {"id": "JFK"}  https://airportgap.dev-tester.com/api/favorites | Verify that we receive a 404 – Not Found response along with the correct data. | PASS – response status code was 404 Not Found |
| API0015 | generate a POST request using POSTMan to generate the following data WITHOUT using the site's supplied authorization token:  The airfield code is "NRT."  https://airportgap.dev-tester.com/api/favorites | Verify that we receive a 401 – Unauthorized response along with the correct data. | PASS – response status code was 401 Unauthorized |
| API0016 | Create a GET request for airport NRT info using POSTMan: https://airportgap.dev-tester.com/api/airports/NRT | Verify that we receive a 200 – OK response along with the correct data. | PASS – response status code was 200 OK |
| API0017 | Create a POST request for the following info using POSTMan:  "to": "JFK," "from": "KIX"  https://airportgap.dev-tester.com/api/airports/distance | Verify that we receive a 200 – OK response along with the correct data. | PASS – response status code was 200 OK |
| API0018 | Create a POST request for the following info using POSTMan:  "to": "JFK," "from": "NRT"  https://airportgap.dev-tester.com/api/airports/distance | Verify that we receive a 200 – OK response along with the correct data. | PASS – response status code was 200 OK |
| API0019 | generate a POST request using POSTMan to generate the following data using the site's permission token:  The airfield code is "NRT."  https://airportgap.dev-tester.com/api/favorites | Verify that we receive a 201 – Created response along with the correct data. | PASS – response status code was 201 Created |
| API0020 | Create a DELETE request for the following data using POSTMan and the site's supplied access token:  'id': 'fakeid'  https://airportgap.dev-tester.com/api/favorites | Verify that we receive a 404 – Not Found response along with the correct data. | PASS – response status code was 404 Not Found |

1. At least two failing tests(2):
2. Fails on authentication
3. Fails for another reason (404 not found)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| API0021 | Create a POST request to save your favourite airport using POSTMan and the site's changed authorization code (I appended a "1" to the end of it): using the info below:  Airport ID is "22."  https://airportgap.dev-tester.com/api/favorites | Verify that we receive a 201 – Created response along with the correct data. | FAIL – response status code was 401 Unauthorized (see figure 1 below) |  |  |  |  |
| API0022 | Create a DELETE request for the following data using POSTMan and the site's supplied access token:  {“id” : “25”}  https://airportgap.dev-tester.com/api/favorites | Verify that we receive a 204 – No Content response along with the correct data. | PASS – response status code was 404 Not Found (see figure 2 below) |  |  |  |  |

Graphical user interface, text, application, email

Description automatically generated

Figure 1: Expected 201 - Created but received 401 – Unauthorized

Graphical user interface, text, application, email

Description automatically generated

Figure 2: Expected 204 – No Content but received 404 – Not Found instead

Below is the JSON file from all the 10 tests:

{

    "info": {

        "\_postman\_id": "591a6d84-44cb-45e2-9f41-c496bfe009ed",

        "name": "PROG8170\_ RG\_RS",

        "schema": "https://schema.getpostman.com/json/collection/v2.1.0/collection.json",

        "\_exporter\_id": "25672691"

    },

    "item": [

        {

            "name": "Returns all airport in Airport DB",

            "request": {

                "method": "GET",

                "header": [],

                "url": {

                    "raw": "https://airportgap.dev-tester.com/api/airports",

                    "protocol": "https",

                    "host": [

                        "airportgap",

                        "dev-tester",

                        "com"

                    ],

                    "path": [

                        "api",

                        "airports"

                    ]

                }

            },

            "response": []

        },

        {

            "name": "Returns the airport specified by ID",

            "request": {

                "method": "GET",

                "header": [],

                "url": {

                    "raw": "https://airportgap.dev-tester.com/api/airports/KIX",

                    "protocol": "https",

                    "host": [

                        "airportgap",

                        "dev-tester",

                        "com"

                    ],

                    "path": [

                        "api",

                        "airports",

                        "KIX"

                    ]

                }

            },

            "response": []

        },

        {

            "name": "Airport specified by ID",

            "request": {

                "method": "POST",

                "header": [],

                "url": {

                    "raw": "https://airportgap.dev-tester.com/api/airports/distance?from=KIX&to=NRT",

                    "protocol": "https",

                    "host": [

                        "airportgap",

                        "dev-tester",

                        "com"

                    ],

                    "path": [

                        "api",

                        "airports",

                        "distance"

                    ],

                    "query": [

                        {

                            "key": "from",

                            "value": "KIX"

                        },

                        {

                            "key": "to",

                            "value": "NRT"

                        }

                    ]

                }

            },

            "response": []

        },

        {

            "name": "Returns the API token for the airport",

            "request": {

                "method": "POST",

                "header": [],

                "url": {

                    "raw": "https://airportgap.dev-tester.com/api/tokens?email=test@airportgap.com&password=airportgappassword",

                    "protocol": "https",

                    "host": [

                        "airportgap",

                        "dev-tester",

                        "com"

                    ],

                    "path": [

                        "api",

                        "tokens"

                    ],

                    "query": [

                        {

                            "key": "email",

                            "value": "test@airportgap.com"

                        },

                        {

                            "key": "password",

                            "value": "airportgappassword"

                        }

                    ]

                }

            },

            "response": []

        },

        {

            "name": "Returns all the favorite airports saved",

            "request": {

                "auth": {

                    "type": "bearer",

                    "bearer": [

                        {

                            "key": "token",

                            "value": "W5oe8joA6Fze2hJJcSp3fxGt",

                            "type": "string"

                        }

                    ]

                },

                "method": "GET",

                "header": [],

                "url": {

                    "raw": "https://airportgap.dev-tester.com/api/favorites",

                    "protocol": "https",

                    "host": [

                        "airportgap",

                        "dev-tester",

                        "com"

                    ],

                    "path": [

                        "api",

                        "favorites"

                    ]

                }

            },

            "response": []

        },

        {

            "name": "Returns the favorite airport from your Airport by ID",

            "request": {

                "auth": {

                    "type": "bearer",

                    "bearer": [

                        {

                            "key": "token",

                            "value": "W5oe8joA6Fze2hJJcSp3fxG",

                            "type": "string"

                        }

                    ]

                },

                "method": "GET",

                "header": [],

                "url": {

                    "raw": "https://airportgap.dev-tester.com/api/favorites?id=22",

                    "protocol": "https",

                    "host": [

                        "airportgap",

                        "dev-tester",

                        "com"

                    ],

                    "path": [

                        "api",

                        "favorites"

                    ],

                    "query": [

                        {

                            "key": "id",

                            "value": "22"

                        }

                    ]

                }

            },

            "response": []

        },

        {

            "name": "Allow you to save the Airport",

            "request": {

                "auth": {

                    "type": "bearer",

                    "bearer": [

                        {

                            "key": "token",

                            "value": "W5oe8joA6Fze2hJJcSp3fxGt",

                            "type": "string"

                        }

                    ]

                },

                "method": "GET",

                "header": [],

                "url": {

                    "raw": "https://airportgap.dev-tester.com/api/favorites?airport\_id=JFK",

                    "protocol": "https",

                    "host": [

                        "airportgap",

                        "dev-tester",

                        "com"

                    ],

                    "path": [

                        "api",

                        "favorites"

                    ],

                    "query": [

                        {

                            "key": "airport\_id",

                            "value": "JFK"

                        }

                    ]

                }

            },

            "response": []

        },

        {

            "name": "Allow you to update the note of one of your favorite airport",

            "request": {

                "auth": {

                    "type": "bearer",

                    "bearer": [

                        {

                            "key": "token",

                            "value": "W5oe8joA6Fze2hJJcSp3fxGt",

                            "type": "string"

                        }

                    ]

                },

                "method": "PATCH",

                "header": [],

                "url": {

                    "raw": "https://airportgap.dev-tester.com/api/favorites/8915",

                    "protocol": "https",

                    "host": [

                        "airportgap",

                        "dev-tester",

                        "com"

                    ],

                    "path": [

                        "api",

                        "favorites",

                        "8915"

                    ]

                }

            },

            "response": []

        },

        {

            "name": "Delete one of your favorite airport",

            "request": {

                "auth": {

                    "type": "bearer",

                    "bearer": [

                        {

                            "key": "token",

                            "value": "W5oe8joA6Fze2hJJcSp3fxGt",

                            "type": "string"

                        }

                    ]

                },

                "method": "DELETE",

                "header": [],

                "url": {

                    "raw": "https://airportgap.dev-tester.com/api/favorites/25",

                    "protocol": "https",

                    "host": [

                        "airportgap",

                        "dev-tester",

                        "com"

                    ],

                    "path": [

                        "api",

                        "favorites",

                        "25"

                    ]

                }

            },

            "response": []

        },

        {

            "name": "Clear all of your favorite airports",

            "request": {

                "auth": {

                    "type": "bearer",

                    "bearer": [

                        {

                            "key": "token",

                            "value": "W5oe8joA6Fze2hJJcSp3fxGt",

                            "type": "string"

                        }

                    ]

                },

                "method": "DELETE",

                "header": [],

                "url": {

                    "raw": "https://airportgap.dev-tester.com/api/favorites/clear\_all",

                    "protocol": "https",

                    "host": [

                        "airportgap",

                        "dev-tester",

                        "com"

                    ],

                    "path": [

                        "api",

                        "favorites",

                        "clear\_all"

                    ]

                }

            },

            "response": []

        }

    ]

}