**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**API Specification Doc**

**(*Web API- Fighting Falcons Web service*)**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Fighting Falcons WEB API SPECIFICATION

VERSION 1.0

20/03/2022

Created By: FIGHTING FALCONS TEAM

|  |
| --- |
|  |

Distribution:

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date Issued** | **Issued To** | **Description** |
| 1.0 | 20-Mar-2022 | FIGHTING FALCONS Web API Project Team Members | Initial draft |

**Index**

Contents

[Document Acceptance and Release Notice 2](#_Toc406495045)

[**Amendment History:** 2](#_Toc406495046)

[**Amendments in this release:** 3](#_Toc406495047)

[Fighting Falcons Web API 5](#_Toc406495048)

[1. Api 6](#_Toc406495049)

[Request 6](#_Toc406495050)

[Conventions 11](#_Toc406495056)

[Status Codes 11](#_Toc406495057)

## 

# FIGHTING FALCONS Web API

The purpose of this document is to provide a high level specification for the “FIGHTING FALCONS” Web Service API (Application Programming Interface). We expect the readers of this document will already have an understanding of basic API/Web Services concepts.

The FIGHTING FALCONS Web API web service will be used by FIGHTING FALCONS GUI to get details whenever it requires. Once the API is called by the client, we will validate and authorize the client and return information based on the level of access they have and information they required. **The response will be in Json format by default.**

There will be only one web service base URL be exposed and that particular url will have various paths and methods for clients to decide and call based on the data it requires.

**Web Service:**

When the GUI client requires data from the service, the client will establish a connection to the Web API authorization service (Yet to be Implemented) to generate a jwt token and passing the user credentials. The request will be validated and on success issues an access and refresh token with expiry (expiry yet to be determined, usually 5 minutes). Expired access token can be refreshed using a refresh\_token provided the session is still active**. A user can Login from different IP address at the same time and will be issued different tokens for each login.**

For audit purposes, the Web API will maintain an audit trail table for every request by logging user details, date & time, IP address, request and their downloaded record count.

The Web API will validate each user call against the user profile.

All requests except for the Authorization Endpoint must be decorated with an authorization header

“Authorization”:”Bearer {token from Authorization Endpoint}”

Web Methods

# Authorization

This web method will return ….

***Content Type***

application/json

### Request

|  |  |
| --- | --- |
| **Method** | **URL** |
| **POST** | Auth/Login |

INPUT Parameters for the request Json

|  |  |  |
| --- | --- | --- |
| **Parameters** | **Values** |  |
| Username | STRING |  |
| Password | STRING |  |

{

"username": "falcons",

"password": "1234567899",

}

Output Parameters for the response Json

{

"data":

{

"access\_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoxNTE2MjM5MDIyfQ.SflKxwRJSMeKKF2QT4fwpMeJf36POk6yJV\_adQssw5c…",

"refresh\_token":"falcons f",

"username": "falcons",

"email": "falcons@gmail.com",

"Name": "falcons f",

"Role": "Employee"

},

"isSuccessfull": "true",

"error": null

}

# GetExperiments

This web method will return ….

***Content Type***

application/json

### Request

|  |  |
| --- | --- |
| **Method** | **URL** |
| **GET** | Experiments/{pageNumber}/{pageSize} |

Output Parameters for the response Json

{

"data":

{

"data":

[

{

"id": "1",

"name": "falconsExp",

"created": "19-03-1990",

"updated": "19-03-1991",

"documents":

["file1.docx", " file1.png", "file3.jpg"]

"status": "ongoing",

"conclusion": null,

"concluded": null,

"Contributors": ["1", "2", "3"]

},

]

"page": "1",

"totalCount":

},

"isSuccessfull": "true",

"error": null

}

# AddExperiment

This web method will return ….

***Content Type***

application/json

### Request

|  |  |
| --- | --- |
| **Method** | **URL** |
| **POST** | Experiments |

INPUT Parameters for the request Json

|  |  |  |
| --- | --- | --- |
| **Parameters** | **Values** |  |
| Name | STRING |  |
| Documents | LIST<STRING> |  |
| Contributors | LIST<INT> |  |

{

"name": "falconsExp",

"documents":

["base64", " base64", " base64"]

"Contributors": ["1", "2", "3"]

}

Output Parameters for the response Json

{

"data":

{

"id": "1",

"name": "falconsExp",

"created": "19-03-1990",

"updated": "19-03-1991",

"documents":

["file1.docx", " file1.png", "file3.jpg"]

"status": "ongoing",

"conclusion": null,

"concluded": null,

"Contributors": ["1", "2", "3"]

},

"isSuccessfull": "true",

"error": null

}

# UpdateExperiment

This web method will return ….

***Content Type***

application/json

### Request

|  |  |
| --- | --- |
| **Method** | **URL** |
| **PUT** | Experiments |

INPUT Parameters for the request Json

|  |  |  |
| --- | --- | --- |
| **Parameters** | **Values** |  |
| Name | STRING |  |
| Documents | LIST<STRING> |  |
| Contributors | LIST<INT> |  |
| Status | INT |  |
| Conclusion | STRING |  |

{

"name": "falconsExp",

"documents":

["base64", " base64", " base64"],

"contributors": ["1", "2", "3"],

"status": 1,

"conclusion": " base64"

}

Output Parameters for the response Json

{

"data":

{

"id": "1",

"name": "falconsExp",

"created": "19-03-1990",

"updated": "19-03-1991",

"documents":

["file1.docx", " file1.png", "file3.jpg"]

"status": "ongoing",

"conclusion": null,

"concluded": null,

"Contributors": ["1", "2", "3"]

},

"isSuccessfull": "true",

"error": null

}

# DownloadConclusion

This web method will return ….

***Content Type***

application/json

### Request

|  |  |
| --- | --- |
| **Method** | **URL** |
| **GET** | Experiments/{experimentId} |

Output Parameters for the response Json

{

"data": "bloburl://…",

"isSuccessfull": "true",

"error": null

}

# GetStatus

This web method will return ….

***Content Type***

application/json

### Request

|  |  |
| --- | --- |
| **Method** | **URL** |
| **GET** | Experiments/Status |

Output Parameters for the response Json

{

"data": [

{"id":"1", "Name", "New"},

{"id":"2", "Name", "Ongoing"},

{"id":"3", "Name", "Review"},

{"id":"4", "Name", "Completed"},

{"id":"5", "Name", "Cancelled"}

],

"isSuccessfull": "true",

"error": null

}

Glossary

## Conventions

* **Client** - Client application.
* **Status** - HTTP status code of response.
* All the possible responses are listed under ‘Responses’ for each method. Only one of them is issued per request server.
* All response is in Json format.
* All request parameters are optional unless explicitly marked as [mandatory]

## Status Codes

All status codes are standard HTTP status codes. The below ones are used in this API.

2XX - Success of some kind

4XX - Error occurred in client’s part

5XX - Error occurred in server’s part

|  |  |
| --- | --- |
| **Status Code** | **Description** |
| 200 | OK |
| 201 | Created |
| 202 | Accepted (Request accepted, and queued for execution) |
| 400 | Bad request |
| 401 | Authentication failure |
| 403 | Forbidden |
| 404 | Resource not found |
| 405 | Method Not Allowed |
| 409 | Conflict |
| 412 | Precondition Failed |
| 413 | Request Entity Too Large |
| 500 | Internal Server Error |
| 501 | Not Implemented |
| 503 | Service Unavailable |