# Introduction

This document explains the technical requirements to connect and consume Blogging Platform API.

Blogging Platform API can be called by Web HTTP Requests (JSON) which supports HTTP

# Available API

* Register User (Sign up)
* Login.
* Get Blogs.
* Get Blog by Id
* Create Blog
* Delete Blog
* Update Blog

## Register User (Sign up)

API Description

* The main objective of this API is to provide a way to create a new account for customer.
* This API returns result to indicate if the API called and executed successfully or not

URL Format

**POST** **Method**

<http://localhost/auth/register>

**Body**

**{**

**“Username”: ”[ Set Username here]”,**

**“Password”: ”[ Set Password here]”,**

**}**

Result

* On Success: Http Code 200 with response body as follows:

{

    "result": 0,

    "message": "User Created Successfully",

    "data": **null**

}

* If username is already exist: Http Code 204 with an empty response body.
* If username is not filled in the request body: Http Code 500 whit response body as follows:

{

    "result": -1,

    "message": "The username should be filled",

    "data": **null**

}

* If password is not filled in the request body: Http Code 500 whit response body as follows:

{

    "result": -1,

    "message": "The password should be filled",

    "data": **null**

}

## Login (Sign in)

API Description

* The main objective of this API is to provide a way to login into account.
* This API returns result to indicate if the API called and executed successfully or not

URL Format

**POST** **Method**

<http://localhost/auth/login>

**Body**

**{**

**“Username”: ”[ Set Username here]”,**

**“Password”: ”[ Set Password here]”,**

**}**

Result

* On Success: Http Code 200 with response body as follows:

{

    "result": 0,

    "message": "",

    "data": {

        "userId": 1,

        "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJfaWQiOiIxIiwibmFtZSI6ImFkbWluIiwiaWF0IjoxNjg5NzkxNzIzLCJleHAiOjE2ODk4MTMzMjN9.8-ouulnKaJTmbYBJrm-dhW3qSjkTDsxHL1MGoeEB1Yo"

    }

}

* If the username or password is not correct: Http Code 200 with response body as follows:

{

    "result": 0,

    "message": "Username or Password is not correct",

    "data": **null**

}

* If username is not filled in the request body: Http Code 500 whit response body as follows:

{

    "result": -1,

    "message": "The username should be filled",

    "data": **null**

}

* If password is not filled in the request body: Http Code 500 whit response body as follows:

{

    "result": -1,

    "message": "The password should be filled",

    "data": **null**

}

## Get All Blogs

API Description

* The main objective of this API is to provide a list of blogs.
* This API returns result to indicate if the API called and executed successfully or not with the blogs.

URL Format

**GET Method**

<http://localhost/blogs>

**Headers that must to add**

We must to add the following header

Authorization : brear [Token that is return from the Login API]

Result

* On Success: Http Code 200 with response body as follows:

{

    "result": 0,

    "message": "",

    "data": [

        {

            "title": "Title1",

            "content": "Content1",

            "author\_Id": 1

        },

        {

            "title": "Title2",

            "content": "Content2",

            "author\_Id": 1

        }

    ]

}

* If user not authenticated (if the authorization is not added): Http Code 401 with response body as follows:

{

    "result": -1,

    "message": "Not Authenticated",

    "data": **null**

}

## Get Blog by Id

API Description

* The main objective of this API is to provide a blog for specific Id of blogs.
* This API returns result to indicate if the API called and executed successfully or not with the blogs.

URL Format

**GET Method**

<http://localhost/blogs/id>

**Headers that must to add**

We must to add the following header

Authorization : brear [Token that is return from the Login API]

Result

* On Success: Http Code 200 with response body as follows:

{

    "result": 0,

    "message": "",

    "data":

        {

            "title": "Title1",

            "content": "Content1",

            "author\_Id": 1

        }

}

* If user not authenticated (if the authorization is not added): Http Code 401 with response body as follows:

{

    "result": -1,

    "message": "Not Authenticated",

    "data": **null**

}

* If id that is passed to URL not found/ or the user that call the API is not the user who create this blog id: Http Code 200 with response body as follows:

{

    "result": -1,

    "message": "Item not found",

    "data": **null**

}

## Create Blog

API Description

* The main objective of this API is to provide a way to create a blog.
* This API returns result to indicate if the API called and executed successfully or not.

URL Format

**Post Method**

<http://localhost/blogs>

**Body**

**{**

**“title”: ”[ Set title of the blog here]”,**

**“content”: ”[ Set content of the blog here]”,**

**}**

**Headers that must to add**

We must to add the following header

Authorization : brear [Token that is return from the Login API]

Result

* On Success: Http Code 200 with response body as follows:

{

    "result": 0,

    "message": "Create a new blog",

    "data": {

        "\_id": -1,

        "title": "new title",

        "content": "new content",

        "author\_Id": "1"

    }

}

* If title is not filled in the request body: Http Code 500 whit response body as follows:

{

    "result": -1,

    "message": "The title should be filled",

    "data": **null**

}

* If content is not filled in the request body: Http Code 500 whit response body as follows:

{

    "result": -1,

    "message": "The content should be filled",

    "data": **null**

}

## Update Blog

API Description

* The main objective of this API is to provide a way to update a specific blog.
* This API returns result to indicate if the API called and executed successfully or not.

URL Format

**Put Method**

<http://localhost/blogs/id>

**Body**

**{**

**“title”: ”[ Set new title of the blog here]”,**

**“content”: ”[ Set new content of the blog here]”,**

**}**

**Headers that must to add**

We must to add the following header

Authorization : brear [Token that is return from the Login API]

Result

* On Success: Http Code 200 with response body as follows:

{

    "result": 0,

    "message": "Blog with ID (the passed id) Updated",

    "data": **null**

}

* If title is not filled in the request body: Http Code 500 whit response body as follows:

{

    "result": -1,

    "message": "The title should be filled",

    "data": **null**

}

* If content is not filled in the request body: Http Code 500 whit response body as follows:

{

    "result": -1,

    "message": "The content should be filled",

    "data": **null**

}

* If id that is passed to URL not found/ or the user that call the API is not the user who create this blog id: Http Code 200 with response body as follows:

{

    "result": -100,

    "message": "The Blog Not Found",

    "data": **null**

}

## Delete Blog

API Description

* The main objective of this API is to provide a way to delete blog for specific Id.
* This API returns result to indicate if the API called and executed successfully or not

URL Format

**Delete Method**

<http://localhost/blogs/id>

**Headers that must to add**

We must to add the following header

Authorization : brear [Token that is return from the Login API]

Result

* On Success: Http Code 200 with response body as follows:

{

    "result": 0,

    "message": "Blog with ID (the passed id) Deleted",

    "data": **null**

}

* If user not authenticated (if the authorization is not added): Http Code 401 with response body as follows:

{

    "result": -1,

    "message": "Not Authenticated",

    "data": **null**

}

* If id that is passed to URL not found/ or the user that call the API is not the user who create this blog id: Http Code 200 with response body as follows:

{

    "result": -1,

    "message": "Item not found",

    "data": **null**

}