API Documentation

Proxy server

- Proxy server acts as middleware to direct any API request.
- When an API call is made to send or receive any data through browser, browser blocks it (gives CORS error) for its own security purpose, if the site from which request is send (say https://myportfoloio.com) is not same as site to which request is send (say https://webflow.com)
- To resolve this, we need to create a server (proxy server), that will behave as if the browser is sending the request to same site.
- Proxy server will receive the request from browser (say https://myportfoloio.com) and then sends it to the other site (say https://webflow.com) and then receive the response from other site (say https://webflow.com) and send it back to browser (say https://myportfoloio.com)

Webyansh proxy server

> https://webyansh-proxy-server.vercel.app/

- Webyansh-proxy-server has been created to send and receive request from other sites, especially to utilize webflow APIs to get and update data to webflow CMS.
- The code is hosted on Webyansh GitHub (https://github.com/webyansh/proxy-server)
- The site/URL is hosted on vercel (https://vercel.com/webyanshs-projects)

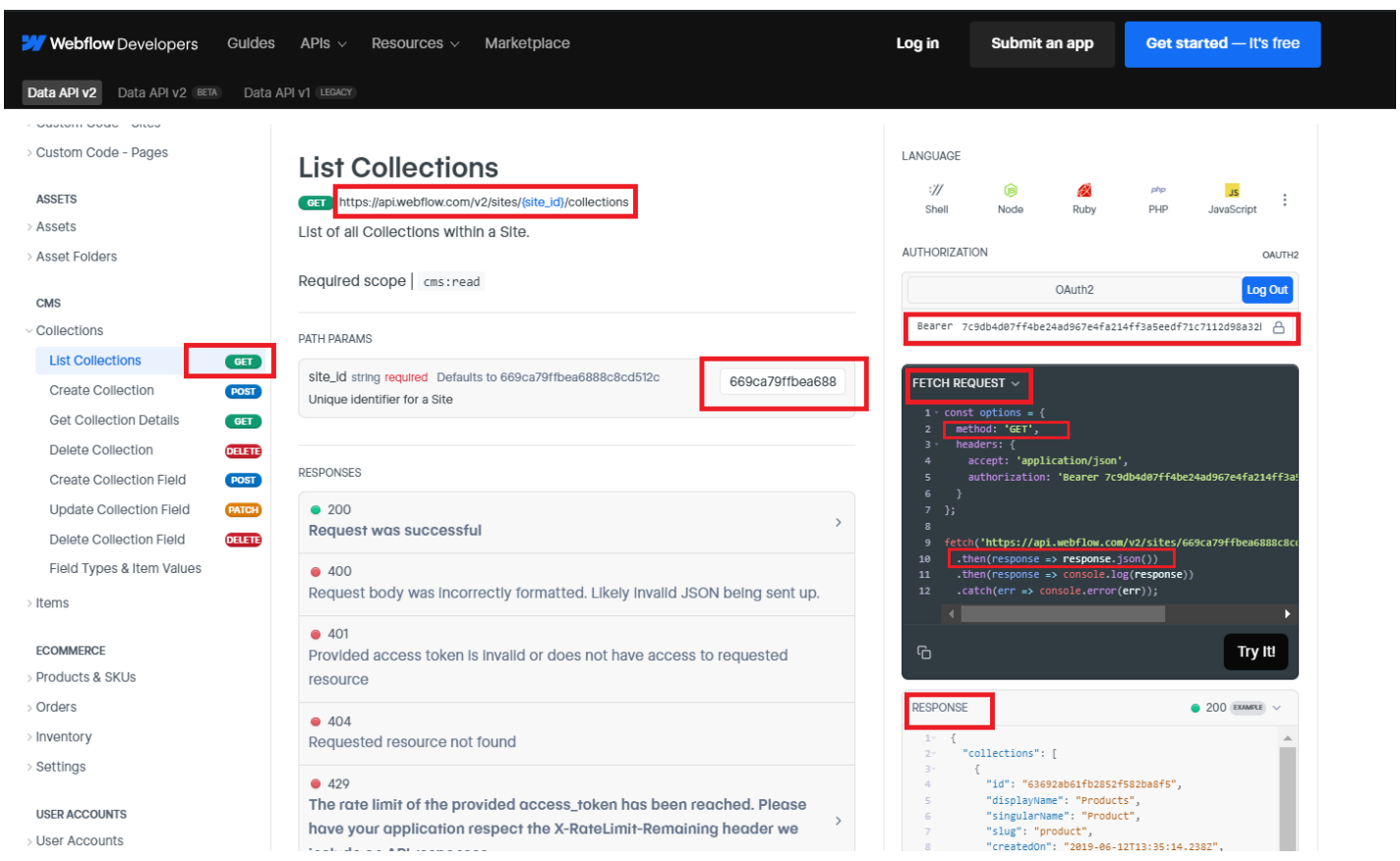
Methods

Different methods are designed for different purpose

- GET - To receive any data, such as list of items in a CMS or list of collections in a site
- POST - To send data.
- PATCH - To update data in CMS such as creating an item in CMS based on data user submits in a form.

API Structure

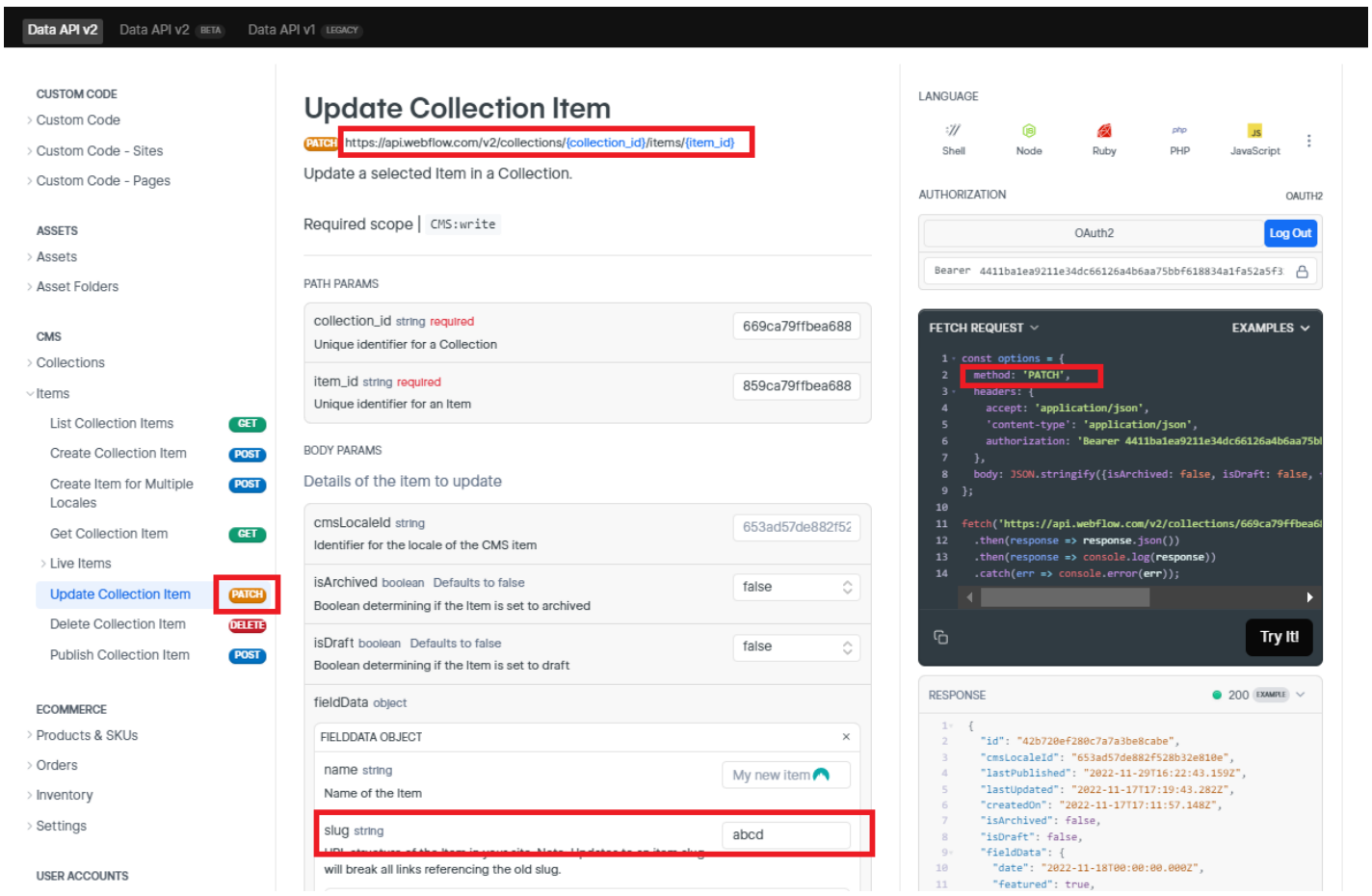
1. **GET : Get all collection list from a webflow site**



```
async function getCollectionList() {
    try {

            // STEP-1: Define webflow variables
        const webflowApiData = {
            site_url: "https://api.webflow.com/v2/sites/",
            collection_url: "https://api.webflow.com/v2/collections/",
            site_id: "669ca79ffbea6888c8cd512c",
            optionData: {   // This is the data that will be send to Webflow (matching
with format - developers.webflow.com)
                options: {
                    method: "GET",
                    headers: {
                        accept: "application/json",
                        authorization: "Bearer 7c9db4d07ff4be24ad967e4fa214ff3a5eedf71
c7112d98a32ba022552d53371"
                    }
                }
            }
        };

        // STEP-2: Defind request URL
        let requestURL = `https://api.webflow.com/v2/sites/669ca79ffbea6888c8cd512c/co
llections/`;

                // STEP-3: Define proxy server variables
        const proxyServerOptions = {    // This is the data required by webyansh-proxy
-server
            method: "POST",
            headers: {
                "Content-Type": "application/json"
            },
            body: JSON.stringify({
                request_url: requestURL,
                options: webflowApiData.optionData.options
            })
        };

                // STEP-4: Call API
        const response = await fetch('https://webyansh-proxy-server.vercel.app/webflow
-api', proxyServerOptions);
        const receivedData = await response.json();

        // STEP-5: Test data
                console.log(receivedData);   // Print the data if required

                // STEP-6: Perform operations on received data
                const dataArray = receivedData.collections;
        for(let i=0; i<dataArray.length; i++){
            // Get slug name of each collection and print
            const slugName = dataArray[i].slug;
            console.log(slugName);
        }

    } catch (error) {
        console.error("Error:", error);
        throw error;
    }
}
```

```
// Function to get site's collection list
getCollectionList();
```

2. **PATCH : Update slug name of all fields of a CMS collection [Prefix slug name with ‘testing’]**




```
async function updateCollectionSlug() {
    try {

        // STEP-1: Define webflow variables
        const webflowApiData = {
            site_url: "https://api.webflow.com/v2/sites/",
            collection_url: "https://api.webflow.com/v2/collections/",
            site_id: "669ca79ffbea6888c8cd512c",
            collection_id: "669ca79ffbea6888c8cd51d7",
            optionData: {   // This is the data that will be send to Webflow (matching
with format - developers.webflow.com)
                options: {
                    method: "GET",
                    headers: {
                        accept: "application/json",
                        'content-type': 'application/json',
                        authorization: "Bearer 7c9db4d07ff4be24ad967e4fa214ff3a5eedf71
c7112d98a32ba022552d53371"
                    }
                }
            }
        };

        // STEP-2: Defind request URL
        let requestURL = `https://api.webflow.com/v2/collections/669ca79ffbea6888c8cd5
1d7/items/`;

        // STEP-3: Define proxy server variables
        const proxyServerOptions = {    // This is the data required by webyansh-proxy
-server
            method: "POST",
```

```
            headers: {
                "Content-Type": "application/json"
            },
            body: JSON.stringify({
                request_url: requestURL,
                options: webflowApiData.optionData.options
            })
        };

        // STEP-4: Call API to Receive all data of collection
        const response = await fetch('https://webyansh-proxy-server.vercel.app/webflow
-api', proxyServerOptions);
        const receivedData = await response.json();

        // STEP-5: Test data
        console.log(receivedData);   // Print the data if required

        // STEP-6: Perform operations on received data
        const dataArray  = receivedData.items;
        for(let i=0; i<dataArray.length;i++){

            // Get slug name and item id of each item
            const prevSlug = dataArray[i].fieldData.slug;  // eg: 'mountain-view'
            const itemId = dataArray[i].id;                // eg: "669ca79ffbea6888c8c
d5277"

            // update slug name
            const newSlug = `testing-${prevSlug}`;           // eg: 'testing-mountain-
view'

            ////////////////////////// update new slug to webflow

            // step-1: change method to PATCH from GET
            webflowApiData.optionData.options.method = "PATCH";
            // step-2: save slug data
            webflowApiData.optionData.options.body = JSON.stringify({
                fieldData: {
                    slug: newSlug
                }
            });
            // step-3: update request_url to particular item id
            const newRequestURL = `${requestURL}${itemId}`;
            proxyServerOptions.body = JSON.stringify({
                request_url: newRequestURL,
                options: webflowApiData.optionData.options
            })
            // step-4: call API to update data
            const newResponse = await fetch('https://webyansh-proxy-server.vercel.app/
webflow-api', proxyServerOptions);
            const newReceivedData = await newResponse.json();

            // STEP-5: Test data
            console.log(newReceivedData);   // Print the data if required
        }


    } catch (error) {
        console.error("Error:", error);
        throw error;
    }
```

```
}

// Function to get site's collection list
updateCollectionSlug();
```

Important Note: The above structure has hard-coded the **collection-id and item-id** which may change on adding or deleting any field into CMS collection. Hence, it is better to get these values dynamically and only hard-code the **site-id** which remains constant through out the project as well as the **request URL**.

Better Approach:

```
async function updateCollectionSlug() {
    try {

        const webflowApiData = {
            site_url: "https://api.webflow.com/v2/sites/",
            collection_url: "https://api.webflow.com/v2/collections/",
            site_id: "669ca79ffbea6888c8cd512c",
            optionData: {
                options: {
                    method: "GET",
                    headers: {
                        accept: "application/json",
                        'content-type': 'application/json',
                        authorization: "Bearer 7c9db4d07ff4be24ad967e4fa214ff3a5eedf71
c7112d98a32ba022552d53371"
                    }
                }
            }
        };

        // Define Site URL to get all CMS collection
        let requestURL = `${webflowApiData.site_url}${webflowApiData.site_id}/collecti
ons`; // [https://api.webflow.com/v2/sites/669ca79ffbea6888c8cd512c/collections]

        const proxyServerOptions = {
            method: "POST",
            headers: {
                "Content-Type": "application/json"
            },
            body: JSON.stringify({
                request_url: requestURL,
                options: webflowApiData.optionData.options
            })
        };

        // Call API to Receive all CMS collection
        const response = await fetch('https://webyansh-proxy-server.vercel.app/webflow
-api', proxyServerOptions);
        const receivedData = await response.json();

        const collectionList  = receivedData.collections;

        // Get collection to be updated using its slug-name
        const propertyVibesColl = collectionList.find((collection) => collection.slug
=== "property-vibe") // Hard-code collection slug name from CMS
        const propertyVibesCollId = propertyVibesColl.id;

        // Get slug names of all items from property vibe collection
        webflowApiData.optionData.options.method = "GET";
```

```
        const propVibeCollUrl = `${webflowApiData.collection_url}${propertyVibesCollId}/items/`;  // [https://api.webflow.com/v2/collections/123456789/items/]
        proxyServerOptions.body = JSON.stringify({
            request_url: propVibeCollUrl,
            options: webflowApiData.optionData.options
        });
        const propVibeResponse = await fetch('https://webyansh-proxy-server.vercel.app/webflow-api', proxyServerOptions);
        const propVibeReceivedData = await propVibeResponse.json();
        const propVibeDataArray = propVibeReceivedData.items;

        for(let i=0; i<propVibeDataArray.length;i++){

            const prevSlug = propVibeDataArray[i].fieldData.slug;
            const itemId = propVibeDataArray[i].id;
            const newSlug = `testing-${prevSlug}` ;

            ////////////////////////////// update new slug to webflow

            webflowApiData.optionData.options.method = "PATCH";
            webflowApiData.optionData.options.body = JSON.stringify({
                fieldData: {
                    slug: newSlug
                }
            });
            const newRequestURL = `${propVibeCollUrl}${itemId}`;        // [https://api.webflow.com/v2/collections/123456789/items/987456321]
            proxyServerOptions.body = JSON.stringify({
                request_url: newRequestURL,
                options: webflowApiData.optionData.options
            })
            const newResponse = await fetch('https://webyansh-proxy-server.vercel.app/webflow-api', proxyServerOptions);
            const newReceivedData = await newResponse.json();

            // Test data
            console.log(newReceivedData);
        }


    } catch (error) {
        console.error("Error:", error);
        throw error;
    }
}

// Function to get site's collection list
updateCollectionSlug();
```

NOTE: Another advantage of using the above approach is that if there are multiple functions like updateCollectionSlug(), getCollectionSlug(), each calling separate API, then instead of defining the variables - webflowApiData, and proxyServerOptions within each function, we can define them globally and pass them into each function - updateCollectionSlug(webflowApiData, proxyServerOptions) as it would be same.

3. POST: Post methods are used to create new CMS collection or new item in a collection. Example - hopstack → directory forms: Creating a new item in directory form each time user submits a form

Sending data to sites other than webflow

- All the above API structure is used to deal with webflow APIs since their routes ended with /webflow-api （ https://webyansh-proxy-server.vercel.app/webflow-api ）
- To deal with other site APIs like teleCRM [sliceinn], package-tracker [shop-box] APIs etc., we need to change the route to /others （ https://webyansh-proxy-server.vercel.app/others).
- Methods will work similar as previous.