GenWrite ServerEnpoint Plugin

Set-up Credentials

- **Server Endpoint Url** : A route/endpoint created on the user's server to communicate with GenWrite.
  eg: http://localhost:4000/blogs

- **Frontend Url** : User's frontend url where the blogs are going to be posted publicly. It is needed for blog interlinking and search console metrics.
  eg: http://localhost:4000/

- **Auth Token** : To make the server communication safer we are going to use auth token validation, so the user will generate and use that authToken when setting up the plugin.
  eg: 1a2b3c4d5e

Note :- Make sure the Auth Token is safely stored and of at least 8 characters long. Also as we are encrypting auth tokens in db so you will not be able to see / edit the Auth Token once the ServerEndpoint integration is saved / updated.

Youtube Videos:

Wordpress Setup

Server-to-Server Setup

Server Paths to Implement

For better communication, the user has to create some following routes on the serverEndpoint he/she provided :

GET /ping

It's a ping route to check whether a connection between GenWrite server and the user's server is successfully established or not.

response status == 200 for proper connection establishment.

GET /categories

It's a category route to get user's specific categories so the user can use their categories for posting the blog.

**response status == 200,
response data must be an array of strings. eg.: ["Marketing", "AI", "Tech"]**

POST   /post

It's a post route to create a new blog post.

Request Body: PostObject

Response Status == 200 or 201

Response Data: PostedDetails

PUT   /post

It's a route to update an existing blog post.

Request Body: [PostObject](#)

Response Status == 200 or 201

Response Data: [PostedDetails](#)

PostObject
```
{
        id : null | undefined | string,          // posted id of blog in server
        title: string,                            // blog title
        tags: string[],                           // blog tags
        focusKeywords: string[],                  // blog primary keywords
        keywords: string[],                       // blog primary keywords
        category: string,                         // blog category
        content:string,                           // blog html content
        custom_summary: string,                   // blog summary
        meta_title:string,                        // blog meta title
        meta_description: string,                 // blog meta description
        meta_ketwords: string,                    // blog meta keywords (comma separated)
        featured_image_url: string,               // blog thumbnail image url
        featured_image_alt: string,               // blog thumbnail image alt text
        schema_markup: string                     // schema markup for blog
}
```

PostedDetails

```
{

        success: boolean,      // blog post creation or update is successful or not
        link: string,   // public posted blog link (for interlinking and search console)
        post_id: string        // unique blog id of db (for seamless updates)

}
```