# Connector Engine Configuration Guide

## Executive Summary

This document provides detailed technical instructions for installing, configuring, and maintaining the Formidable Forms Headless Connector Engine. The engine serves as the backend integration layer between WordPress (running Formidable Forms) and any headless front-end environment, agnostic of framework. It enables full hydration, rendering, and submission of Formidable forms and views via REST API, ensuring a secure, scalable, and extensible architecture.

## System Requirements

• WordPress 6.0 or later

• Formidable Forms (Pro)

• PHP 8.0 or later

• MySQL 5.7+ or MariaDB equivalent

• REST API enabled in WordPress

• SSL/TLS for secure API communication

## Installation

1. Download the Connector Engine source code from the Masterminds Enterprise Developer Series public GitHub repository:

https://github.com/vfontjr/enterprise-developer-series-code/tree/main

2. Upload the engine’s PHP integration files (e.g., form\_hydrator\_class.php, view-normalizer.php) to your WordPress installation under `/wp-content/plugins/{your-plugin}/`

3. Activate the plugin through the WordPress Admin > Plugins interface.

4. Ensure permalinks are enabled for REST API routing to function.

## Configuration Variables

|  |  |  |
| --- | --- | --- |
| Variable | Description | Default Value |
| FORMIDABLE\_API\_BASE | Base URL for WordPress REST API endpoints | https://yourdomain.com/wp-json |
| FORMIDABLE\_NONCE | Security nonce for authenticated requests | Generated per session |
| FORM\_RENDER\_ENGINE\_PATH | Relative path to form-renderer-engine.js in the headless front end | /integrations/formidable/vanilla-js/form-renderer-engine.js |
| FORM\_VIEW\_HANDLER\_PATH | Relative path to view-handler.js in the headless front end | /integrations/formidable/vanilla-js/view-handler.js |

## REST API Endpoints

The following endpoints are exposed by the connector engine:

• `/wp-json/custom/v1/form-id/{formKey}` – Resolves a Formidable form key to its numeric ID.

• `/wp-json/frm/v2/forms/{formId}` – Retrieves metadata for a specific form.

• `/wp-json/frm/v2/forms/{formId}/fields` – Retrieves the field definitions for a specific form.

• `/wp-json/custom/v1/view-id/{viewKey}` – Resolves a Formidable view key to its numeric ID.

• `/wp-json/custom/v1/view-normalized/{viewKey}` – Returns normalized JSON data for the specified view.

## Execution Flow

1. Front end requests form/view data by key.  
2. Connector engine resolves the key to an ID.  
3. Engine retrieves metadata and fields from the Formidable API.  
4. JSON payload is returned to the headless front end.  
5. Rendering engine on the front end constructs the DOM.

## Maintenance & Troubleshooting

• If REST requests return 404, verify permalinks are enabled and endpoint routes are registered.

• If nonce validation fails, ensure the correct nonce is passed via wpApiSettings or generated server-side.

• Check PHP error logs for backend processing errors.

## References

GitHub Repository: https://github.com/vfontjr/enterprise-developer-series-code/tree/main

Formidable Forms REST API Docs: https://formidableforms.com/knowledgebase/rest-api/

WordPress REST API Handbook: https://developer.wordpress.org/rest-api/