Your Development Journey

- Your Development Journey
- Step 1: Set Up Your Development Environment
  - Essential Requirements
  - Quick Setup Instructions
- Step 2: Choose Your Add-on Type
  - Which Type Should You Build?
  - Extension Examples
  - Integration Examples
- Step 3a: Create a Basic Extension
- Step 3b: Create a Basic Integration
- Step 4: Test and Deploy Your Add-on
- Essential Resources
  - Documentation
  - Tools and Templates
  - Community Support
- Next Steps

Getting Started with MainWP Development

Welcome to MainWP development! This guide will walk you through the essential steps to start building add-ons for MainWP whether you're creating a standalone extension or integrating with a third-party service.

Your Development Journey

Follow these steps to begin your MainWP development journey:

flowchart LR A[1. Setup Environment] --> B[2. Choose Add-on Type] B --> C1[3a. Create Basic Extension] B --> C2[3b. Create Basic Integration] C1 --> D[4. Test & Deploy] C2 --> D

```
style A fill:#4CAF50stroke:#388E3Ccolor:white
style B fill:#2196F3stroke:#1976D2color:white
style C1 fill:#FFC107stroke:#FFA000color:white
style C2 fill:#FFC107stroke:#FFA000color:white
style D fill:#9C27B0stroke:#7B1FA2color:white
```

Step 1: Set Up Your Development Environment

You are here: Step 1 of 4

Before you start coding you'll need to set up a proper development environment.

Essential Requirements

- ☐ WordPress installation (local or development server)
- ☐ MainWP Dashboard plugin installed and activated
- ☐ At least one child site connected to your MainWP Dashboard
- ☐ Code editor (VS Code PhpStorm or similar)
- ☐ Basic understanding of PHP and WordPress plugin development

Quick Setup Instructions

1. Install a local WordPress development environment like:
   - LocalWP (recommended for beginners)
   - XAMPP
   - MAMP
2. Install and activate the MainWP Dashboard plugin:
   - Download from WordPress.org
   - Or install directly from your WordPress admin area
3. Connect at least one child site:
   - Install WordPress on another local site
   - Install and activate the MainWP Child plugin
   - Connect it to your MainWP Dashboard
4. Install the MainWP Development Extension:
   - Clone from GitHub

- Or download the ZIP file and install manually

For detailed instructions see the Setting Up a Development Environment guide.

Step 2: Choose Your Add-on Type

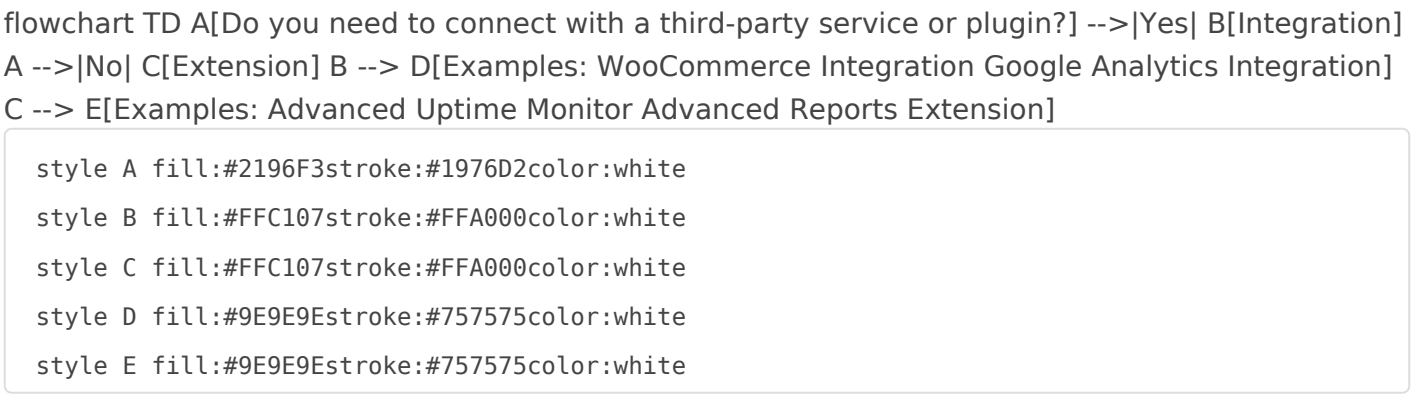
Next step: Step 2 of 4

MainWP uses specific terminology to distinguish between different types of add-ons:

- **Extension**: An add-on that works WITHOUT requiring a third-party plugin or API
- **Integration**: An add-on that works WITH a third-party plugin or API

Which Type Should You Build?

Use this decision tree to determine whether you should build an Extension or an Integration:



Extension Examples

Extensions are standalone add-ons that extend MainWP's functionality without requiring third-party services:

- **Advanced Uptime Monitor**: Monitors website uptime directly through MainWP
- **Advanced Reports**: Generates custom reports based on MainWP data
- **Maintenance Extension**: Schedules and manages maintenance tasks

Integration Examples

Integrations connect MainWP with third-party plugins or services:

- **WooCommerce Integration**: Manages WooCommerce stores through MainWP
- **Google Analytics Integration**: Displays Google Analytics data in MainWP
- **Yoast SEO Integration**: Manages Yoast SEO settings across sites

Step 3a: Create a Basic Extension

Next step (if building an Extension): Step 3a of 4

If you're building a standalone extension follow these steps:

1. Start with the MainWP Development Extension template
2. Customize the plugin information
3. Implement your core functionality
4. Create admin pages and settings
5. Test thoroughly

For detailed instructions see the Creating a Basic Extension guide.

Step 3b: Create a Basic Integration

Next step (if building an Integration): Step 3b of 4

If you're building an integration with a third-party service follow these steps:

1. Start with the MainWP Development Extension template
2. Customize the plugin information
3. Implement third-party API connectivity
4. Create admin pages and settings
5. Implement data synchronization
6. Test thoroughly

For detailed instructions see the Creating a Basic Integration guide.

Step 4: Test and Deploy Your Add-on

Final step: Step 4 of 4

Before releasing your add-on make sure to:

1. Test on multiple child sites
2. Check for any errors or warnings
3. Ensure compatibility with different WordPress versions
4. Package your add-on for distribution
5. Create documentation for users

For detailed instructions see the Packaging and Distribution guide.

Essential Resources

Documentation

- MainWP Dashboard API Documentation
- MainWP Child API Documentation
- MainWP Dashboard Hooks
- MainWP Child Hooks

Tools and Templates

- MainWP Development Extension on GitHub
- MainWP API Client PHP on GitHub

Community Support

- MainWP Community Forum
- MainWP Documentation
- MainWP on GitHub

Next Steps

After completing the getting started process explore these topics to enhance your MainWP development skills:

- Using MainWP Actions & Filters
- Building Admin Interfaces

- Data Storage and Retrieval
- Debugging Add-ons
- Working with Third-Party APIs (for Integrations)

□□□ #2
□ □□□ □□□ 2026-08-29 22:03:44 UTC
□ □□□ □□□ 2026-08-29 22:03:44 UTC