# Setting Up a MainWP Development Environment


- Quick Start for Experienced Developers
- Why Use a Local Development Environment?
- Recommended Development Tools
  - Local Development Environment
  - Code Editor
  - Version Control
  - Browser Developer Tools
- Step-by-Step Setup Guide
  - 1. Install Laragon
  - 2. Create a WordPress Installation
  - 3. Install MainWP Dashboard
  - 4. Set Up Test Child Sites
    - Option 1: Create Local Child Sites
    - Option 2: Use Existing Remote Sites
  - 5. Clone the MainWP Development Extension
  - 6. Configure PHP Development Settings
  - 7. Install Additional Development Tools
    - Composer
    - WP-CLI
    - Query Monitor
- Development Workflow
- Debugging Tips
  - WordPress Debugging
  - MainWP Specific Debugging
- Common Issues and Solutions
  - Connection Issues Between Dashboard and Child Sites

    - PHP Version Compatibility
- Next Steps
- Related Resources


---

This guide walks you through setting up a development environment for MainWP extension development. A proper development environment will make your development process more efficient and help you avoid common issues.

# Quick Start for Experienced Developers

1. **Install Laragon** (or similar local development environment)
2. **Set up WordPress** (latest version)
3. **Install and configure MainWP Dashboard plugin**
4. **Add at least one test child site** (local or remote)
5. **Clone the MainWP Development Extension**:

   ```
   git clone https://github.com/mainwp/mainwp-development-extension.git
   ```

6. **Start developing your extension**

# Why Use a Local Development Environment?

Developing MainWP extensions in a local environment offers several advantages:

- **Speed**: Local development is faster than working on a remote server
- **Safety**: You can experiment without affecting live sites
- **Offline Work**: You can work without an internet connection
- **Version Control**: Easier integration with Git or other version control systems
- **Debugging**: Better access to logs and debugging tools

# Recommended Development Tools

# Local Development Environment

We recommend using Laragon for MainWP extension development because it:

- Is easy to set up and use
- Includes all necessary components (Apache, MySQL, PHP)
- Supports multiple PHP versions
- Includes helpful developer tools
- Makes it easy to create and manage multiple local sites

Other options include:

- LocalWP
- XAMPP
- MAMP
- Docker with WordPress containers

# Code Editor

A good code editor will make your development more efficient. We recommend:

- Visual Studio Code with PHP extensions
- PhpStorm
- Sublime Text

# Version Control

Using Git for version control is highly recommended:

- Git
- GitHub Desktop (for a user-friendly interface)
- GitKraken

# Browser Developer Tools

Modern browsers include developer tools that are essential for debugging:

- Chrome DevTools
- Firefox Developer Tools

# Step-by-Step Setup Guide

## 1. Install Laragon

1. Download Laragon from laragon.org
2. Install Laragon following the installation wizard
3. Launch Laragon

## 2. Create a WordPress Installation

1. In Laragon, click "Menu" > "Quick app" > "WordPress"
2. Enter a name for your site (e.g., "mainwp-dev")
3. Laragon will create a new WordPress installation
4. Once complete, click the link to open your new WordPress site
5. Complete the WordPress installation process

## 3. Install MainWP Dashboard

1. Log in to your WordPress admin dashboard
2. Go to "Plugins" > "Add New"
3. Search for "MainWP Dashboard"
4. Click "Install Now" and then "Activate"
5. Follow the MainWP setup wizard to configure the dashboard

## 4. Set Up Test Child Sites

For a complete development environment, you'll need at least one child site to test your extension with.

### Option 1: Create Local Child Sites

1. In Laragon, create additional WordPress installations for child sites
2. Install and activate the MainWP Child plugin on each site
3. Connect these sites to your MainWP Dashboard

## Option 2: Use Existing Remote Sites

If you have existing WordPress sites, you can use them as test child sites:

1. Install and activate the MainWP Child plugin on each site
2. Connect these sites to your MainWP Dashboard

# 5. Clone the MainWP Development Extension

1. Open a terminal or command prompt
2. Navigate to your WordPress plugins directory:

   ```
   cd path/to/laragon/www/mainwp-dev/wp-content/plugins
   ```

3. Clone the MainWP Development Extension:

   ```
   git clone https://github.com/mainwp/mainwp-development-extension.git
   ```

4. Activate the extension in your WordPress admin dashboard

# 6. Configure PHP Development Settings

For effective development and debugging, configure PHP with appropriate settings:

1. In Laragon, click "Menu" > "PHP" > "php.ini"
2. Ensure the following settings are enabled:

   ```
   display_errors = On
   error_reporting = E_ALL
   ```

3. Save the file and restart Laragon

# 7. Install Additional Development Tools

## Composer

Composer is a dependency manager for PHP that makes it easy to manage libraries and dependencies:

1. Laragon includes Composer by default

2. To use it, open Laragon Terminal and run:

```
composer --version
```

## WP-CLI

WP-CLI is a command-line tool for managing WordPress:

1. Laragon includes WP-CLI by default
2. To use it, open Laragon Terminal and run:

```
wp --version
```

## Query Monitor

Query Monitor is a debugging plugin for WordPress:

1. In your WordPress admin, go to "Plugins" > "Add New"
2. Search for "Query Monitor"
3. Install and activate the plugin

# Development Workflow

Once your environment is set up, here's a recommended workflow for developing MainWP extensions:

1. **Plan your extension**: Define its purpose, features, and how it will integrate with MainWP
2. **Create a new extension** based on the MainWP Development Extension template
3. **Implement features** incrementally, testing each one as you go
4. **Test thoroughly** with different configurations and child sites
5. **Debug and refine** your code
6. **Package for distribution** when ready

# Debugging Tips

## WordPress Debugging

Enable WordPress debugging by adding these lines to your `wp-config.php` file:

```
define('WP_DEBUG', true);
define('WP_DEBUG_LOG', true);
define('WP_DEBUG_DISPLAY', true);
```

Debug logs will be saved to `wp-content/debug.log`.

# MainWP Specific Debugging

MainWP includes several debugging options:

1. Go to "MainWP" > "Settings" > "Advanced Options"
2. Enable "Development Mode" for additional debugging information
3. Use the "MainWP Tools" section for diagnostics

# Common Issues and Solutions

## Connection Issues Between Dashboard and Child Sites

If you're having trouble connecting your MainWP Dashboard to child sites:

1. Ensure both sites are accessible from the internet (or use a tool like ngrok for local sites)
2. Check that the MainWP Child plugin is properly installed and activated
3. Verify that your server meets the MainWP requirements
4. Try using the Connection Recovery process in MainWP Dashboard

## PHP Version Compatibility

MainWP requires PHP 7.0 or higher. To change PHP versions in Laragon:

1. Click "Menu" > "PHP"
2. Select the desired PHP version
3. Restart Laragon

# Next Steps

Now that your development environment is set up, you're ready to start creating your MainWP extension:

- Creating a Basic MainWP Extension
- Understanding the MainWP Development Extension

# Related Resources

- MainWP Knowledge Base
- WordPress Developer Resources
- PHP Documentation

---

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