Tools

目录
通用工具
Homebrew
构建工具
Iterm2 与 Oh-My-Zsh（可选）
Git / GitHub
本地环境
Sequel Pro
gettext
项目工具
PHP
PECL
Brew PHP Switcher（可选）
Composer
在 Local 环境中使用 XDebug
通过 NVM 安装 Node
Yarn
Grunt
调试工具栏

本页介绍了 Yoast 在开发各类插件和库时使用的工具。

:::note 以下说明假设您使用的是 OSX 系统。 :::

通用工具

Homebrew

Homebrew 是 OSX 的包管理器，类似于 `apt`、`yum` 和 `pip`。它允许你从命令行轻松安装和更新软件。

安装 Homebrew 非常简单。只需打开终端（`应用程序/实用工具/终端`）并运行以下命令：

```
/bin/bash -c "$(curl -fsSL
https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

构建工具

在 Mac 上进行开发时，建议安装一些额外的开发工具（`libtool`），以便从源代码构建代码：

```
xcode-select --install
brew install autoconf automake libtool
```

Iterm2 与 Oh-My-Zsh (可选)

iTerm2 是 Mac OSX 默认终端应用的替代品。我们建议安装它，因为它在终端操作中提供了许多额外功能。

Oh-My-Zsh 是一个命令行 shell，它添加了一些在普通 shell 中难以实现的功能。它还让你能更轻松地管理 zsh 配置。

你可以通过以下命令安装两者：

```
brew install --cask iterm2
sh -c "$(curl -fsSL https://raw.githubusercontent.com/robbyrussell/oh-my-
zsh/master/tools/install.sh)"
```

安装完成后，你需要通过运行 `chsh -s /bin/zsh` 将 shell 从 bsh 切换到 zsh。否则，你将无法体验 Oh-My-Zsh 的所有炫酷功能。

示例：

- ZSH 可以在 shell 中显示当前活动的 Git 分支。
- 更好的目录和文件名自动补全功能。

Git / GitHub

在 Yoast，我们使用 Git 进行版本控制。不了解 Git 如何工作？请查看 此教程。我们的 Git 仓库都托管在 GitHub 上。如果您还没有账户，请前往 创建账户。

通过运行以下命令安装 Git：

```
brew install git
```

然后，进一步阅读 如何配置本地 Git 设置 和 如何将 SSH 密钥添加到 GitHub。

最后一个建议： 添加全局 git 忽略文件，以免意外将 `.DS_Store` 等系统文件添加到仓库中。

本地环境

在 Yoast，我们使用 Local 作为开发环境的基础。

- 前往 Flywheel 的偏好设置 » 高级 » 路由器模式，并将其设置为 “站点域名”。
- 创建一个新站点，命名为 `basic.wordpress.test`。
- 可选：确保你的浏览器 /系统信任新创建站点的 SSL 证书。
  - 对于 macOS，需要几个步骤。

遇到 通过 Local 配置站点的问题？如果你的芯片组与 Local（或任何其他软件）开发时所用的不同，安装 Rosetta 的这个命令会有所帮助：

```
softwareupdate --install-rosetta
```

Sequel Pro

要以简单快捷的方式管理和搜索数据库， Sequel Pro 是推荐工具。您可以从其官网下载最新版本。

要了解如何连接到 Docker 数据库，请查阅 Plugin Development Docker 仓库中的 连接说明。

gettext

gettext 是一个 GNU 国际化 (i18n) 和本地化 (l10n) 库。在撰写本文时，构建 `wordpress-seo` 插件需要此软件包。

```
brew install gettext
brew link gettext --force
```

项目工具

PHP

由于插件使用 PHP 编写，需要在本地安装 PHP。我们目前使用 PHP 7.4 进行插件开发。由于该版本未包含在 `brew` 官方源中，需要先添加非官方的 Homebrew PHP 仓库再安装 PHP：

```
brew tap shivammathur/php
brew install php@7.4
```

PECL

PHP 扩展通过 PECL 分发。PECL 随 PHP 一同安装，但你可能需要将其添加到计算机的路径中以便访问。找到 PHP 的 `bin` 目录（通常遵循以下模式），并按如下方式将其添加到 `PATH` 变量中：

```
export PATH=$PATH:/usr/local/Cellar/php@7.4/7.4.33_4/bin/
```

Brew PHP Switcher (可选)

:::note 目前仅适用于 Homebrew。 :::

Brew PHP Switcher 是一个简单的 shell 脚本，可快速在 PHP 主要版本之间切换 Apache 和 CLI 配置。如果你需要同时维护使用新版或旧版 PHP 功能构建的多个产品或项目，这将非常有用。

假设你当前运行的是最新版 PHP，并希望降级到较早版本，可以按以下步骤操作：

- 使用 `brew install php@7.2` 安装 PHP 7.2 (或更早版本)
- 安装 Brew PHP Switcher `brew install brew-php-switcher`
- 切换到 PHP 7.2 (或更早版本) `brew-php-switcher 7.2`
- 重新打开终端并检查 PHP 版本 `php -v`

Composer

Composer 是一个 PHP 包管理器，在我们的 WordPress 插件中用于轻松管理代码依赖。要安装它，请在终端中运行以下命令：

```
brew install composer
```

安装 Composer 后，您可以在包含 `composer.json` 文件的插件目录中运行以下命令来安装所有必要的依赖项：

```
composer install
```

:::note 要更新依赖项，您可以运行 `composer update`。但通常无需这样做——我们使用 `composer.lock` 中列出的固定版本。 :::

在 Local 环境中使用 XDebug

如果您使用 Local 作为开发环境，XDebug 应该已经安装在您的本地站点上。要在 IDE 中启用 XDebug，请按照以下步骤操作。

确保以下路径已包含在您的 PHP 包含路径中：

- `~/Local Sites/<site-name>/app/public`：WordPress 安装所在的文件夹。
- `/Applications/Local.app/Contents/Resources/extraResources/local-bootstrap.php`：Local 用于引导站点的文件。

您可以使用 这些步骤 将这些路径包含到您的 PHPStorm 项目中。

确保这些路径映射到服务器上的正确路径：

- `~/Local Sites/<site-name>/app/public` 应映射到服务器上的相同路径（`~/Local Sites/<site-name>/app/public`）。
- `/Applications/Local.app/Contents/Resources/extraResources/local-bootstrap.php` 应映射到服务器上的相同路径（`/Applications/Local.app/Contents/Resources/extraResources/local-bootstrap.php`）。

为避免 PHPStorm 在每次请求时因断点而频繁中断，请禁用 `Break at first line in PHP scripts`。同时禁用 `Force break at first line when no path mapping specified` 和 `Force break at first line when a script is outside the project`。这些设置可以通过打开 `Settings`（`cmd ,`）-> `PHP` -> `Debug` 找到。

您可以使用 这些步骤 在您的 PHPStorm 项目中添加这些路径映射。

通过 NVM 安装 Node

Node.js® 是一个基于 Chrome V8 JavaScript 引擎构建的 JavaScript 运行时。我们的工具使用它来编译 JavaScript。Node 版本管理器 (NVM) 是一个简单的 bash 脚本，用于管理多个活跃的 Node.js 版本。

首先，安装 NVM。之后，你可以通过运行以下命令来安装一个 Node 版本

```
nvm install <version>
```

查看 Node.js 官网 以了解最新的 LTS 版本是什么。

Yarn

Yarn 是一个用于管理 JavaScript 依赖项的包管理器。要安装 Yarn，请确保您已先安装 Brew，然后运行以下命令在您的机器上全局安装它。请注意，此方法也会安装 Node，但我们不使用那个版本。

```
brew install yarn
```

与 Composer 类似，您需要导航到包含特定文件的目录。对于 Yarn，您需要在运行前确保位于 `package.json` 文件所在的目录：

```
yarn
```

Grunt

Grunt 是一款帮助运行重复性任务的工具，例如运行代码检查器、压缩代码或任何其他可以为了开发目的而自动化的任务。一般来说，Grunt 用于管理 JavaScript 代码。

要安装 Grunt 命令行界面 (CLI)，请在终端中运行以下命令：

```
yarn global add grunt-cli
```

安装 Grunt 后，您可以在项目目录中运行各种命令来处理不同的任务。一个常用的命令是 `grunt watch`。该命令会监视目录中的更改，并对源代码运行一组预定义的命令，以确保其正确输出。

由于我们的代码库使用了许多类似的任务，我们最终 创建了一个代码库 来将这些相似的任务集中在一起。

请务必也查看 Grunt 目录，特别是 `aliases.yml` 文件，以了解有哪些可用的命令。

调试工具栏

为协助调试 WordPress（插件），您可以使用 Debug Bar 插件。

该插件可以记录已运行的查询和已定义的环境参数，还能帮助您发现其他隐藏（但有用）的数据。

□□□ #2
□ □□□ □□□ 2026-03-26 17:08:40 UTC
□ □□□ □□□ 2026-03-26 17:08:40 UTC