

今後の開発において必要となるであろうエディタ、デプロイメントツール、デバッガがひとまとまりになった IDE（統合開発環境）のセットアップ手順を紹介します。

とても複雑な手順ですが、一度導入してしまえば開発などが非常に楽になりますので導入をおすすめします。もちろん自前の環境を用いても構いません。

■ PHPStorm について

PHPStorm は JetBrains 社の提供する PHP 専用の IDE です。本来有償ソフトですが、JetBrains 社は現在アカデミック向けにフリーライセンスを提供していますのでこちらで登録すると 1 年間無料で使用することができます。

[Free individual licenses for students and faculty member](#)

■ 目次

- 1 [テスト環境](#)
- 2 [開発環境の準備](#)
 - 2.1 [PHPStorm](#)
 - 2.2 [MAMP](#)
- 3 [MAMP の初期設定](#)
 - 3.1 [MAMP の設定](#)
 - 3.2 [XDebugger の設定](#)
- 4 [PHPStorm との連携](#)
 - 4.1 [PHP インタプリタの設定](#)
 - 4.2 [デバッグサーバの設定](#)
 - 4.3 [デプロイメントの設定](#)
- 5 [実際にデプロイから表示までやってみよう](#)
- 6 [デバッグをやってみよう](#)

■ テスト環境

Windows 10 Pro x64 と macOS 10.13 の二者にて実行を確認しました。

ここでは Windows をメインに行いますが、基本的には macOS でも同様の操作で実現できます。

■ 開発環境の準備

1. PHPStorm

PHP を編集及び、デバッグする環境として IDE である JetBrains 社の提供する PHPStorm を用います。

JetBrains 社の公式ページより PHPStorm(<https://www.jetbrains.com/phpstorm/>)をダウンロードの後、インストールします。特に注記することはないので指示通りで問題ありません。

2. MAMP

Web サーバの実行環境として MAMP を用いた Web サーバを構築します。

MAMP の公式ページより MAMP(<https://www.mamp.info/en/>)をダウンロードし、インストールします。なお、MAMP Pro をインストールするよう促されますが、無料版で事足りるのでインストールする必要はありません。

ただ、GUI で設定できる項目が増えるようなので常備する場合は検討しても良いかもしれません。

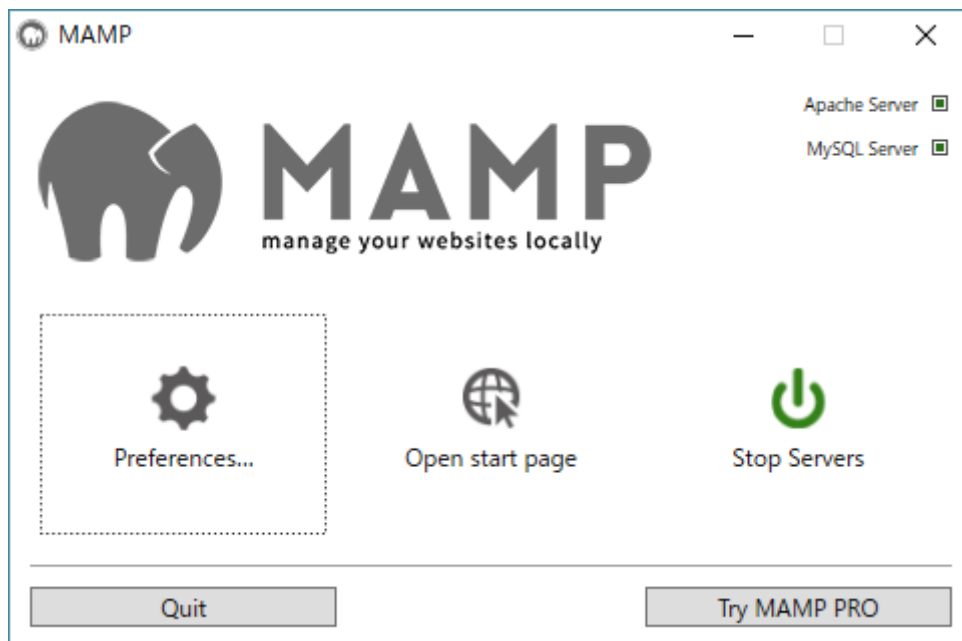
なお、ここではインストール先にデフォルトの C:\MAMP を指定することとします。
macOS では Applications にインストールします。

■ MAMP の初期設定

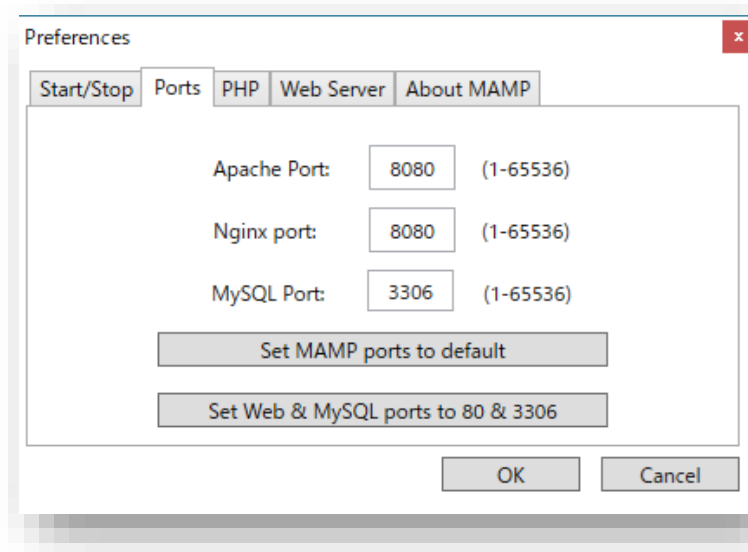
➤ MAMP の設定

MAMP を起動すると Start Servers をクリックするだけで全てのサーバを起動することができますが、ここでは統一のために設定を変更します。

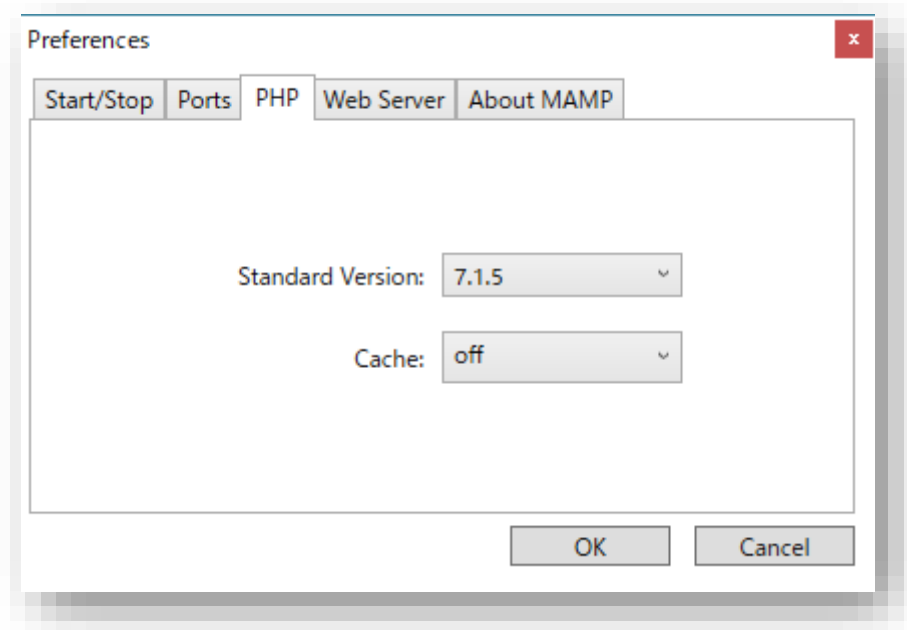
まずは Preferences...よりポートを設定します。



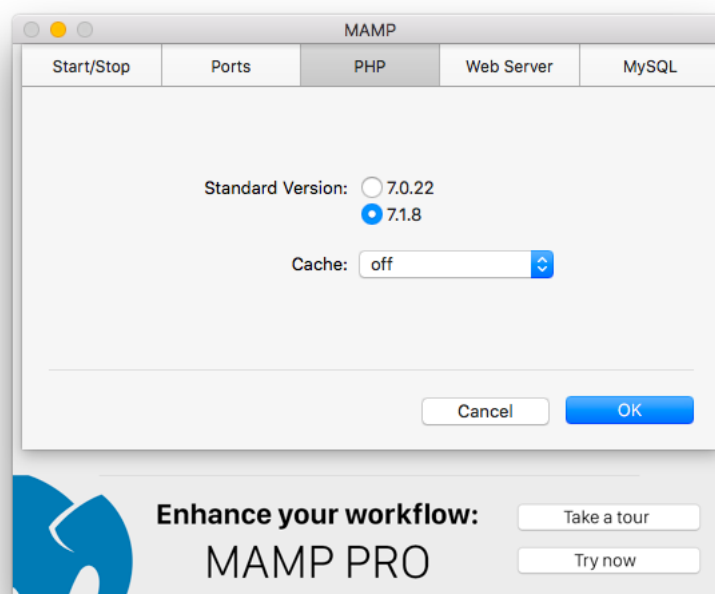
ここでは Apache のポートを 8080 にし、MySQL のポートを 3306 とします。なお、今回 Nginx は使用しないのでどのポートでも問題ありません。



ちなみにですが、PHP のバージョンは 7.1.5 としています。



ただし、macOS では 7.1.8 を選択します。



あとは Start Servers よりサーバを起動すると完了です。

この後 Web Server タブより Document Root を確認し、phpinfo();を実行するファイルを作成します。

```
<?php
phpinfo();
```

*phpinfo.php

その後それを実行し、Loaded Configuration File の値を控えておきましょう。

Configuration File (php.ini) Path	C:\WINDOWS
Loaded Configuration File	C:\MAMP\conf\php7.1.5\php.ini

➤ XDebugger の設定

次に、デバッグで必要となる XDebugger を設定します。

先程控えておいた Loaded Configuration File の php.ini ファイルに以下の設定を追記します。

```
[xdebug]
zend_extension="C:\MAMP\bin\php\php7.1.5\ext\php_xdebug.dll"
xdebug.coverage_enable=1
xdebug.default_enable=1
xdebug.profiler_enable=1
xdebug.profiler_output_dir="C:\MAMP/tmp"
xdebug.remote_autostart=1
xdebug.remote_enable=1
xdebug.remote_host=localhost
xdebug.remote_port=9000
xdebug.idekey="PHPSTORM"
```

*Windows 用

```
[xdebug]
zend_extension="/Applications/MAMP/bin/php/php7.1.8/lib/php/extensions/
no-debug-non-zts-20160303/xdebug.so"
xdebug.coverage_enable=1
xdebug.default_enable=1
xdebug.profiler_enable=1
xdebug.remote_autostart=1
xdebug.remote_enable=1
xdebug.remote_host=localhost
xdebug.remote_port=9000
xdebug.idekey="PHPSTORM"
xdebug.profiler_output_dir="/Applications/MAMP/tmp/xdebug/"
```

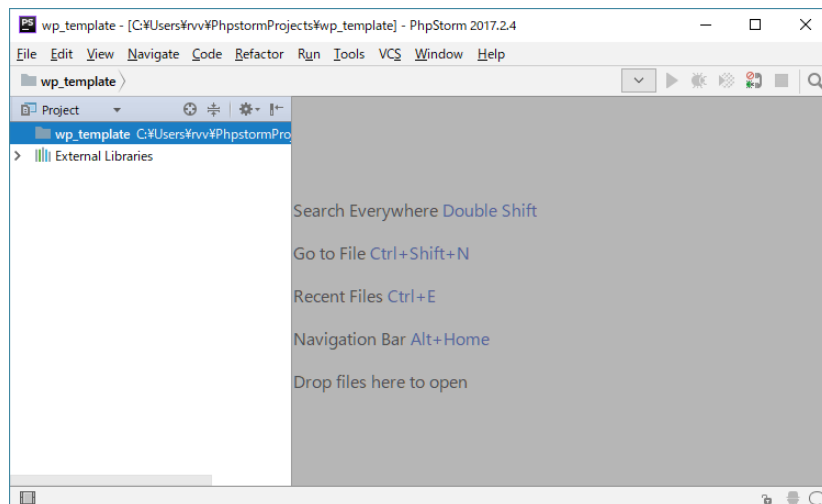
*macOS 用

これでとりあえず設定は完了です。あとはサーバを再起動しておきましょう。

■ PhpStorm との連携

➤ PHP インタプリタの設定

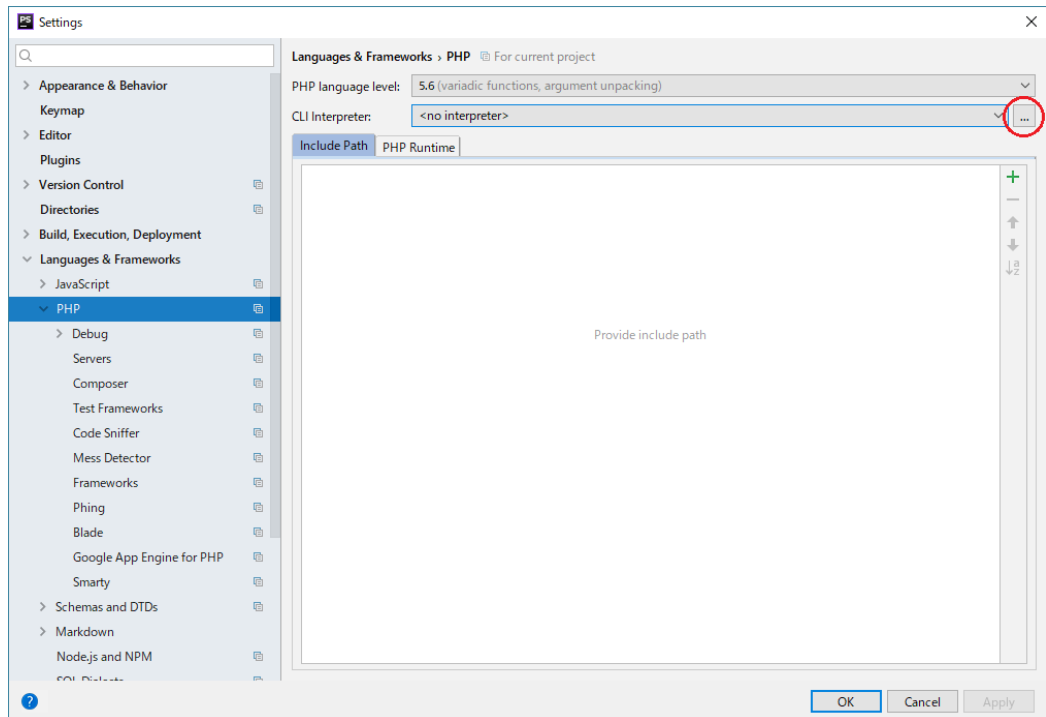
続いて PhpStorm の設定を行います。何かしらのプロジェクトを生成します。
するとこのような画面が出ると思います。



あえて載せるものでもないですが、バージョンにより UI が異なるようなので参考程度に掲載しておきます。

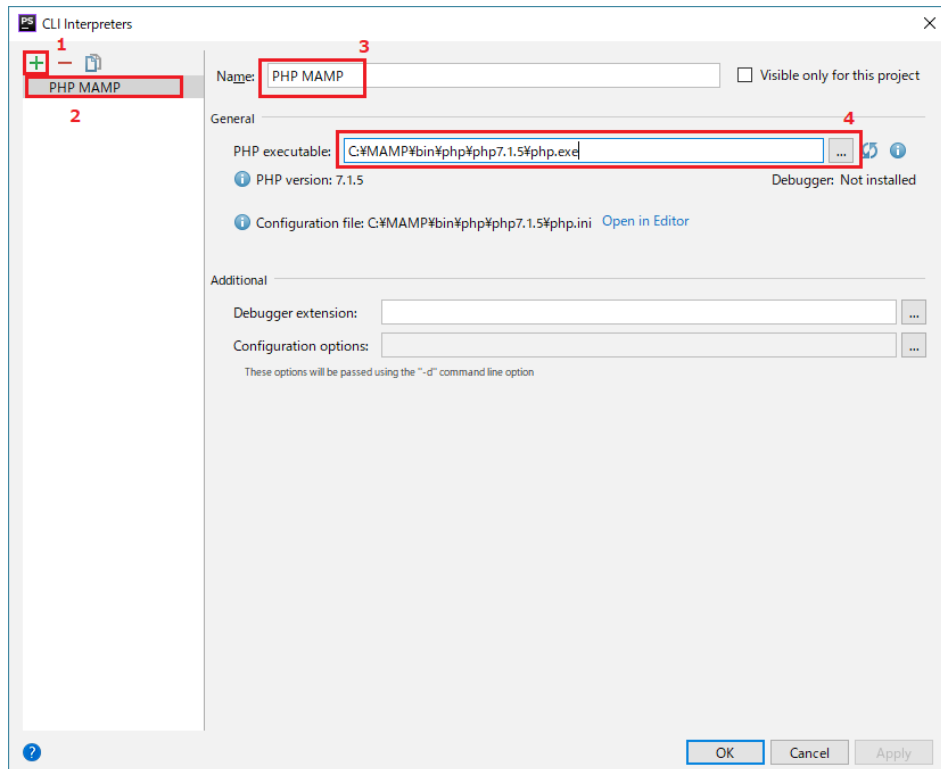
続いて「**File→Settings (macOS は PhpStorm -> Preferences)**」を開きます。すると設定画面がズラっと出てきますので「**Languages & Frameworks→PHP**」を開きます。

すると次のような画面が開かれます。



ここで **CLI Interpreter** を選択することで PHP のバージョンを選択しますが、初期設定ではないので...ボタンをクリックします。

またしても設定画面が開くのでここに MAMP の PHP を設定します。

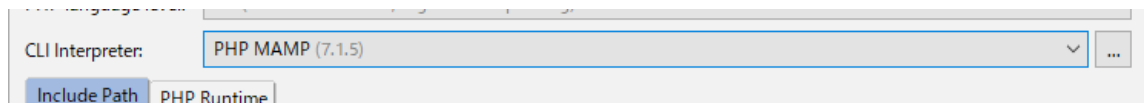


なお、Debugger がインストールされていないことになっていますが、Windows 版の MAMP は php.ini が php.exe 直下ではないため、インストールされていないことになっています。

MAMP を使用する場合はこのままで問題ありませんが、使用しない場合は直下の php.ini も設定しておきましょう。

設定が完了すると OK を押して閉じましょう。

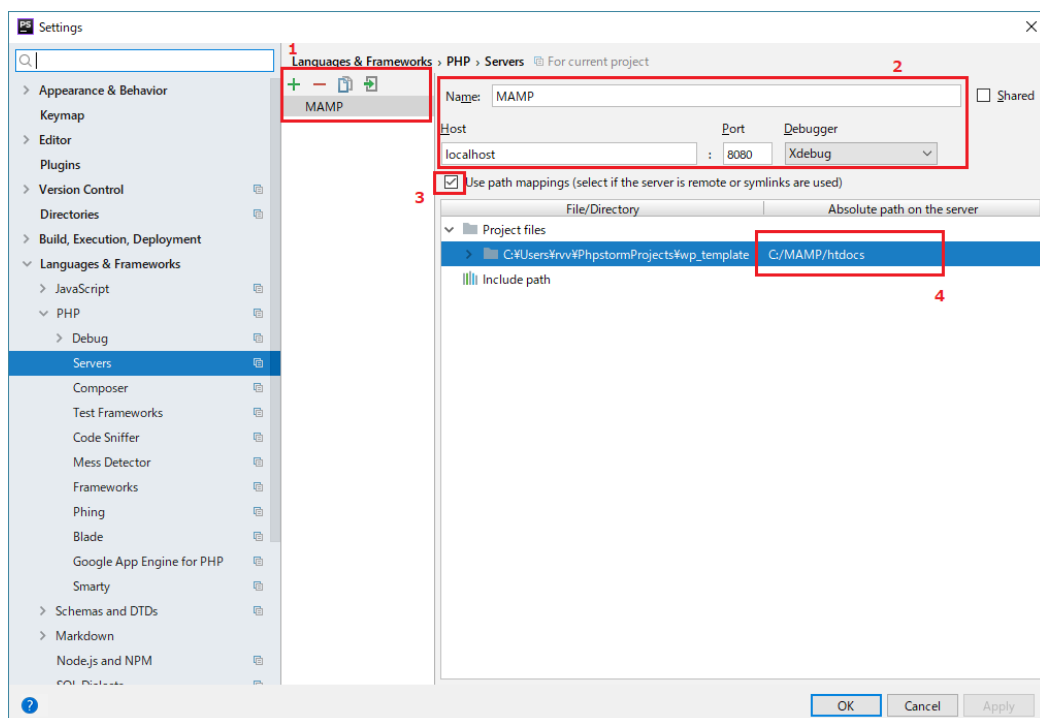
すると CLI Interpreter に先程設定した PHP が現れるので設定しておきます。



➤ デバッグサーバの設定

続いて、デバッグサーバの設定を行います。先程の PHP の更に深い階層に、**Servers** という項目があるのでそれを開きます。

先ほどと同様に MAMP のサーバ情報を記述します。



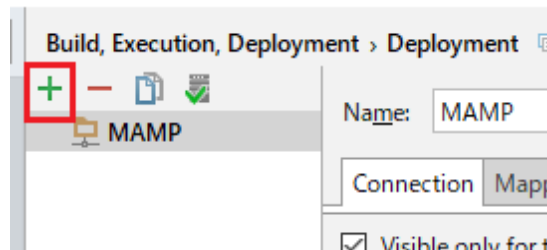
3 番以降はデバッグ時に IDE 上のファイルと紐付ける際に必要みたいで、**Absolute path on the server** にドキュメントルートを指定します。なお、ドキュメントルートでない部分のファイルを作成する場合はそれに応じて変更しましょう。

とりあえずドキュメントルートで実際に動かしてみるとわかると思います。

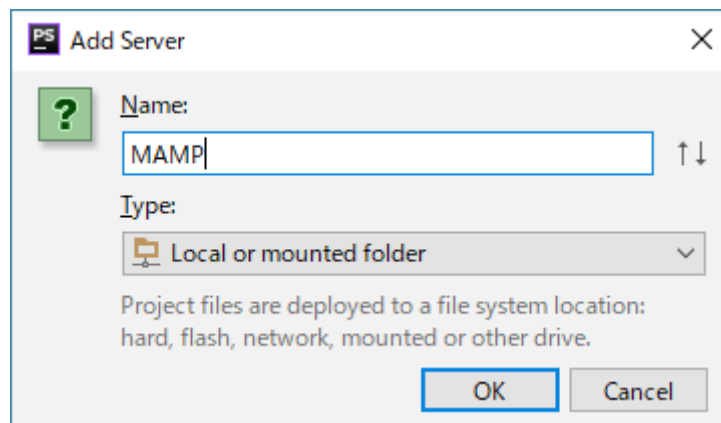
➤ デプロイメントの設定

次に、デプロイメントの設定を行います。

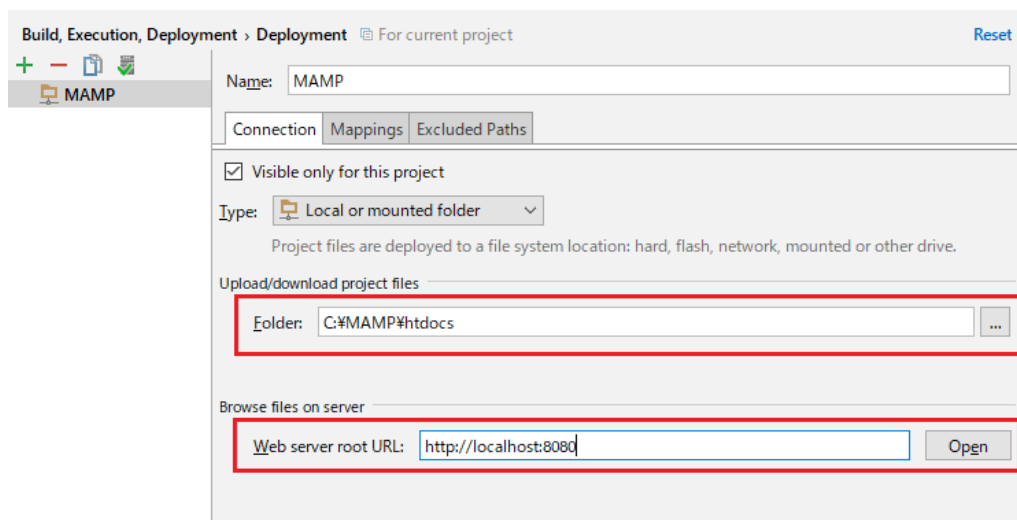
「Build, Execution, Deployment→Deployment」と進むとまたしても設定画面が現れるのでプラスボタンを押します。



すると次のような入力画面が出るため、適当な名前とともに **Local or mounted folder** を選択します。



すると細かい設定画面が出てくるので **Connection** の **Upload/Download project files** の **Folder** にドキュメントルートを設定します。



また、**Web server root URL** も MAMP のものに合わせてください。

次に、**Mappings タブ**にて **Deployment path on server** のパスを設定します。

The screenshot shows the 'Mappings' tab for a server named 'MAMP'. The 'Local path' is set to 'C:\Users\rvv\Projects\wp_template'. The 'Deployment path on server 'MAMP' (relative to folder 'C:\MAMP\htdocs')' is highlighted with a red box and contains a backslash '\'. The 'Web path on server 'MAMP'' is set to '/'. The 'Project URL' is 'http://localhost/'. There are buttons for 'Use this server as default', 'Add another mapping', and 'Reset'.

このパスはデプロイしたいファイルをどこにアップロード/ダウンロードするかの設定です。今回はルート直下から設定したいので「¥」（**macOS は「/」**）を設定します。なお、ここに「¥test¥」みたいに設定すると IDE では直下にあっても¥test¥にアップロードされることになります。

また、複数の設定を行いたい場合は **Add another mapping** にて更に細かい設定ができます。例えば、wordpress の theme 用と plugin 用にパスで分けるとデプロイの際に勝手に定位置にアップロードしてくれます。

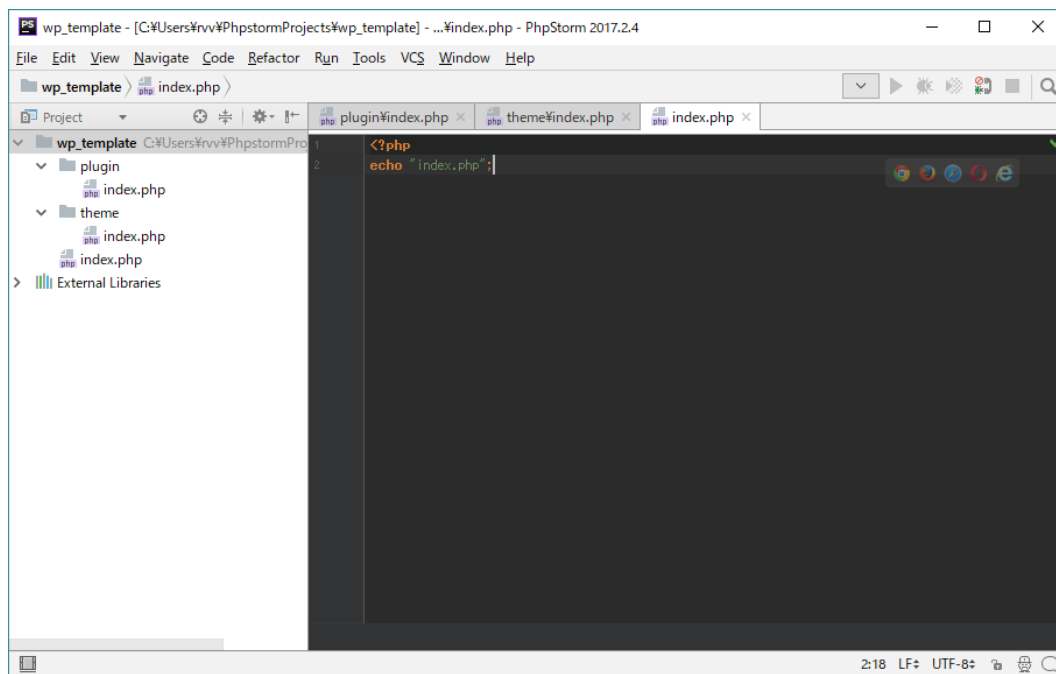
The screenshot shows the 'Paths' section with a table of mappings. The table has four columns: 'Local Path', 'Deployment Path', 'Web Path', and a '+'/- column. The first row shows the default mapping. The second and third rows show mappings for 'theme' and 'plugin' subdirectories.

Local Path	Deployment Path	Web Path	
C:\Users\rvv\Projects\wp_template	¥	/	+
C:\Users\rvv\Projects\wp_template¥theme	¥theme	/	-
C:\Users\rvv\Projects\wp_template¥plugin	¥plugin	/	

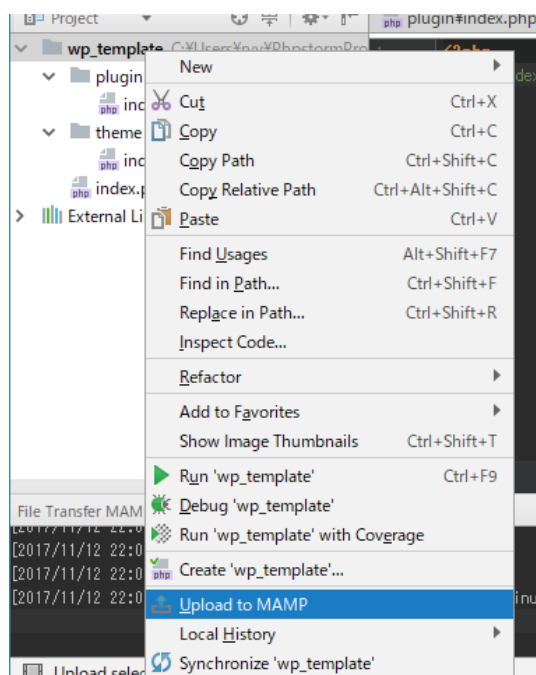
■ 実際にデプロイから表示までやってみよう

急ぎ目でババッと設定しましたが、実際に動かしてみましよう。

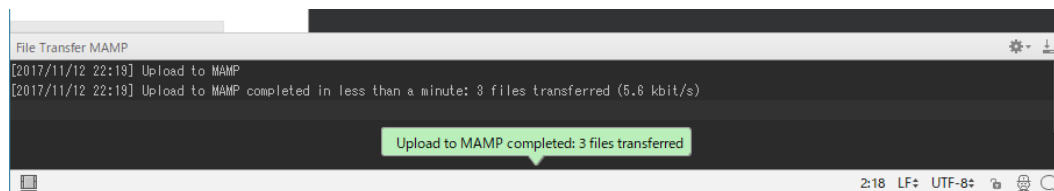
とりあえず適当に自分の名前を出力するようなコードを書いてみました。また、ファイル構造は先程マッピングの例で示したようにしています。



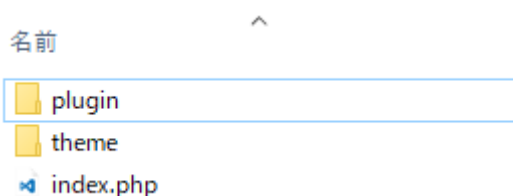
コードが完成すると次にアップロードをしなければなりません。必要なファイルを選択してアップロードもできますが、今回は全てアップロードしたいので上位ディレクトリを右クリックし、**Upload to** ○○を選択します。



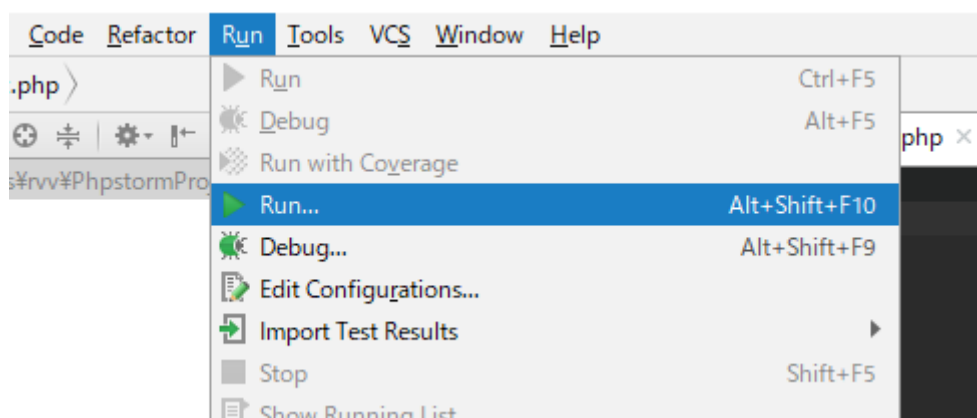
するとアップロードした旨のメッセージが流れるのでこれでアップロード完了です。



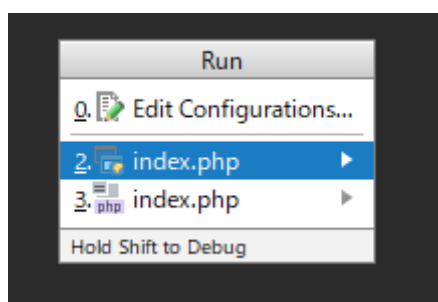
なお、ファイル構造もちゃんと指示通りになっています。



次に、実行ですが、「Run→Run」をクリックすると



Run ウィンドウが出てくるのでハイライトされているアイコンのものを選択するとただただ実行するだけの設定が作成されます。

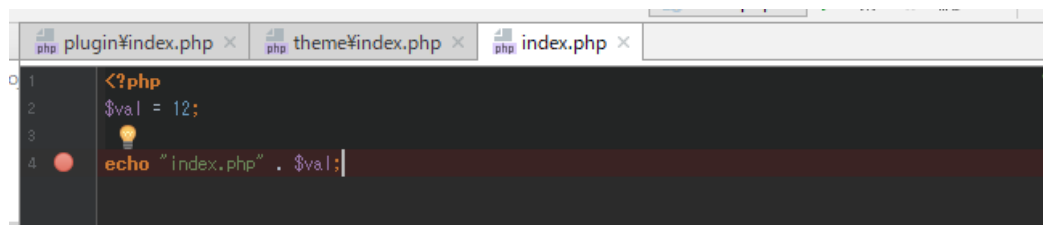


下のやつを選ぶと PHP Script として動作するので間違えないようにしましょう。設定を見るとわかりますが、JavaScript Debug として動いているのでこれだけで PHP デバッグはできません。ただ表示させるだけとして使用します。

■ デバッグをやってみよう

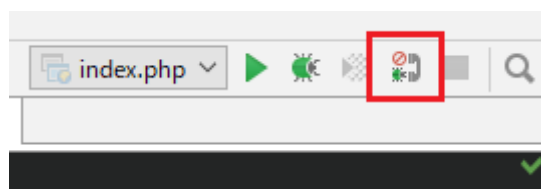
続いて肝心のデバッグですが、ここまで設定しておくとは簡単にできます。

まずは値を拾えるかどうかを確認するためにコードを若干書き換えます。なお、必要はないのでそのままで問題ありません。

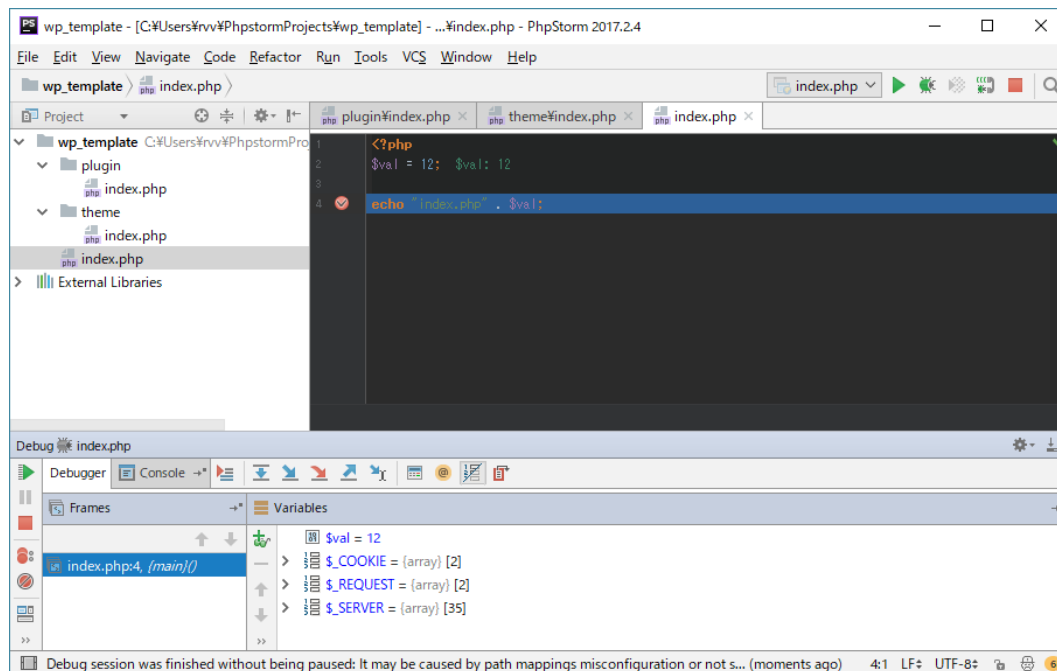


続いて、行のちょっと右あたりをクリックしてブレークポイントを起きます。

あとは右上の電話アイコンをクリックして接続し、実行します。



するとこのように実行を一時停止し、デバッグウィンドウが開きます。



あとはデバッグウィンドウの停止や再生ボタンなどをクリックするとそれに応じて処理が続行されます。とても簡単。

なお、この方法だと JavaScript Debug で表示させてもデバッガは動作します。