Getting Started with QuickDRY

# What you will need

* IIS (Apache also works, or any other HTTP server that can pipe PHP)
* PHP 7.0
* GIT for Windows - https://git-scm.com/download/win
* TortoiseGIT (64bit recommended) - https://tortoisegit.org/download/
* PuttyGen (for creating SSH Keys to connect to Git with) - https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html
* Composer - https://getcomposer.org/download/
* IIS comes with Windows 10
* Windows Web Platform Installer - https://www.microsoft.com/web/downloads/platform.aspx

# Install PHP

Use Windows Web Platform Installer to install

* PHP 7.0 64 bit
* Microsoft Drivers 4.0 (x64) for PHP v7.0 for SQL Server in IIS

Note that PHP 7.1 does not yet work with IIS

You will need composer installed in order to set up the dependencies

# Install SQL Server Management Studio

Note: The minimum install is Microsoft Command Line Utilities for SQL Server

SQLCMD is required for large query support in QuickDRY

# Install an IDE

PHP Storm is recommended as it is far better at code completion and helping you find your way through the code

EclipsePDT will also work

# Setting Up Your Local Hosts

Add a shortcut to

C:\Windows\System32\drivers\etc\hosts

To your desktop.

Modify the permissions of the file so you can edit it. It should be readable and writable by users. Otherwise you will need to open it in a text editor as an administrator

This is where you will point your dev domains to 127.0.0.1 so you can preview your work on your local machine

It is recommended that you use the format of

mysite.localhost.com

For your hosts. Some browsers do not like domains like “mysite” which will limit your ability to test the site.

Your settings file will then be

settings.mysite.localhost.com.php

but more on that later

# If You’re Using Linux

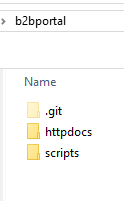
That’s the beauty of cross-platform languages like PHP. It will work just as well on Linux as it does on Windows. But I will leave it as an exercise for the reader on how to set the environment up.

# If You’re Using Subversion

As long as you’re using some form of source control. Git is preferred simply because of the resources out there and it is easy enough to set up a dedicated GitLab server on an old desktop running Debian.

# Out of the Box

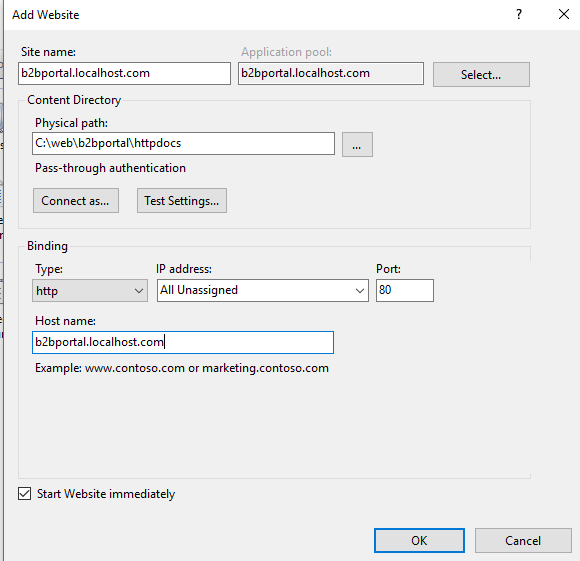
After you create and checkout your initial repository in GitLab, you will copy the httpdocs and scripts folder from QuickDRY into it.



Edit your hosts file and set up your localhost.com address to point to 127.0.0.1



Open up IIS Manager and point the host at the httpdocs folder



# Getting the Site Running

When setting up the project in PHPStorm, you will set the httpdocs folder as the root folder. You do not need the scripts folder as part of the project.

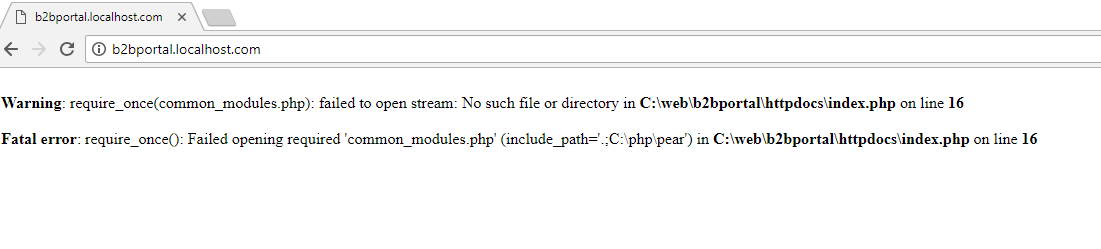
You can then rename the project as needed.

Make sure composer is installed and then open a command window in the httpdocs folder and run

composer install

In run\_composer.bat you’ll find three commands including the one above. Those are the commands you’ll use to install the third-party libraries and keep them up to date.

When you first load the site you will see

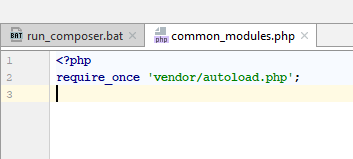


In the root folder you will need to add the file

common\_modules.php

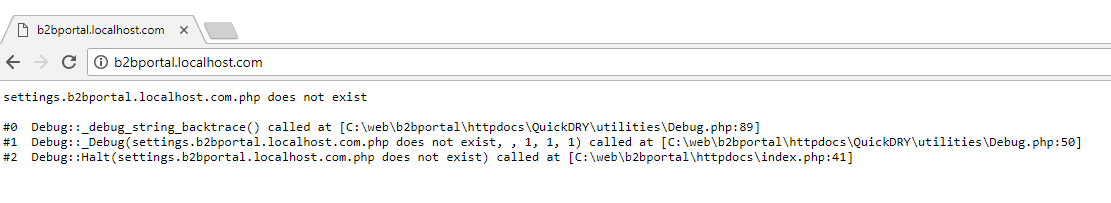
And you will add

require\_once ‘vendor/autoload.php’;



This will enable all the third-party libraries.

Reload the page and you will see

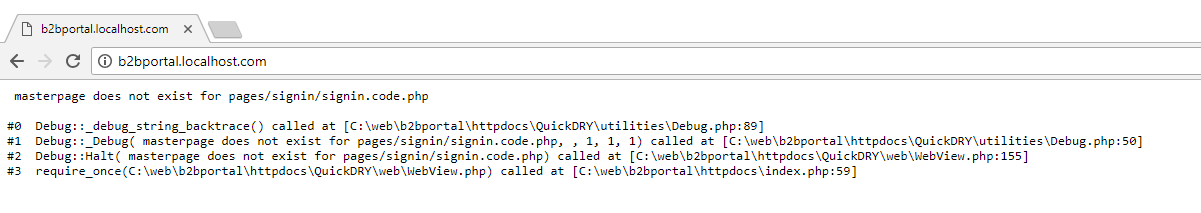


So now you need to create the file

settings.b2bportal.localhost.com.php

In the root httpdocs folder

For now, it can remain empty.  
  
Reload the page and you will see



What this is telling you is that there is no master page set and it’s trying to load the signin page.

And this is where we start building our specific application.