EXAMPLES

This directory has a number of examples of how to use the framework. I put the 'vendor/bartonlp/site-class/includes' directory in the /var/www that Apache2 creates when it
is installed on a Ubuntu system. By default the Apache2 install makes /var/www/html its
DocumentRoot (look at /etc/apache2/sites-enabled/000-default.conf the default site
configuration file).

The following examples are provided.

The first examples either 'require_once' the SiteClass.class.php
or the siteautoload.class.php via '../vendor/bartonlp/site-class/includes/'.

1. test1.php
2. test2.php
3. test3.php
4. test4.php
5. test5.php

The next five examples use the composer autoloader:

1. composer-test1.php
2. composer-test2.php
3. composer-test3.php
4. composer-test4.php
5. composer-test5.php

The next two examples show insertion and updating of the database and dbTables useage.

1. insert-update.php
2. composer-insert-update.php

You can use other frameworks or templeting engines. Here we will use Twig a popular
templet engine. Twig is a super powerful templet engine with looping and conditional
statements and much more. Here we do just about the minimum just as an example.

To use this example you need to install Twig in the 'examples' directory as it is NOT
part of this package by default.

```
composer require twig/twig:~1.0
```

composer-with-twig.php

If you need ReST routing (or pretty routing or SEO friendly routing as it is sometime called)
you could use one of the popular routing engines like **Meteor** or **Laravel**
but then again by that point you might as well just bite the bullet and spend the hours or
days trying to figure those frameworks out.

There is a pretty simple router called Altorouter which can be used without too much work. To
install Altorouter do `composer require altorouter/altorouter:1.1.0` in the 'examples'
directory.

The file 'composer-route.php' would normally be your 'index.php' in a production environment.
When using a server with Apache2 you would need a '.htaccess' file in the directory where
the 'index.php' lives. The '.htaccess' file would look like this:

```
RewriteEngine On
RewriteCond %{REQUEST_FILENAME} !-f
RewriteRule . index.php [L]
```

This says that if the requested filename does not exist go to the 'index.php' file instead.

Now to test this without renaming the 'composer-route.php' you can use the PHP server like this:

```
php -S localhost:8080 composer-route.php
```

The PHP server uses the composer-route.php file and you don't need a '.htaccess' file.

Now you can run the program. It will display a table and a form you can use to insert new
records. Also a button lets you reset the database table to its original state. If you click
on a number in the 'ID' column you get an edit page where you can change the names.

You can also get to the edit page by entering the URI '/edit/3' for example. That will take
you to the edit page for 'ID' three. From that page you and enter the URI '/home' which will
take you back to the home page.