# New Codeigniter Installation (v. 3.1.8):

1. Copy “application” and “themes” directory to your root. Allow the filesystem to overwrite the files.
2. Visit the URL for your current site installation

# Adding to an existing Codeigniter installation:

1. Copy the following files over to your current CI install:
   * themes/\*
   * application/config/layout.php
   * application/core/Layout\_Controller.php
   * application/core/MY\_Controller.php
   * application/helpers/MY\_path\_helper.php
   * application/language/english/error\_lang.php
   * application/language/english/site\_lang.php
   * application/libraries/Layout.php
2. Files to update:
   * application/config/autoload.php
     1. Make sure the following libraries are loaded: 'layout' i.e. $autoload['libraries'] = array('layout');
     2. Make sure the following configs are loaded: ‘layout’ i.e. $autoload['config'] = array('layout');
   * application/config/migration.php
     1. Make sure migration is enabled i.e. $config['migration\_enabled'] = TRUE;
     2. Make sure migration type is sequential i.e. $config['migration\_type'] = 'sequential';
     3. Depending on your current application, increment the migration version by 1. i.e. $config['migration\_version'] = 3;
     4. (only need to do if migration version is greater than 1), if migration version is greater than one, rename file application/migrations/001\_install\_ion\_auth.php to the next migration number i.e. 003\_install\_ion\_auth.php
     5. Run the migration, open bash shell, and in the site root type the following to perform a migration: “php index.php migrate/now”
3. Sample controller file: application/controllers/Welcome.php
   * Extend by Layout\_Controller
   * Added a \_remap method
   * In controller method, can load the layout object
   * In the view, load data array to the template for display

# User Guide

## Adding Local Javasript File

Javascript files are loaded from the [root]/js folder. To improve readability and organization, I suggest that you should create a folder with the controller name, and name the js file with the method.

* $this->layout->addJS(‘welcome/index.js’);

## Adding External Javascript File

You can load an external javascript file using the follow line of code:

* $this->layout->addExternalJS(‘https://www.fontawesome.com/test\_file.js’);

## Adding Local CSS File

CSS files are loaded from the [root]/css folder. To improve readability and organization, I suggest that you should create a folder with the controller name, and name the js file with the method.

* $this->layout->addCSS(‘welcome/index.css’);

## Adding Packages

Packages are js libraries that are contained as a single package with both js and css files. Packages are generally stored and loaded in [root]/package folder.

We have 3 methods for loading packages:

* Add Package: This loads both the js and css file in one method – i.e. $this->layout->addPackage(‘bootstrap/dist/js/bootstrap.js’, ’bootstrap/dist/css/bootstrap.css’);
* Add JS Package: This just loads the js file i.e. $this->layout->addJSPackage(‘bootstrap/dist/js/bootstrap.js’);
* Add CSS Package: This just loads the css file i.e. $this->layout->addCSSPackage(‘bootstrap/dist/css/bootstrap.css’);

## Additional Methods

See more methods in application/libraries/layout.php