**Using Yii2 code in other frameworks**

If you want to use Yii2 framework code with other frameworks just add Yii2-specific parameters in composer.j son:

{

"extra": {

"asset-installer-paths": {

"npm-asset-library": "vendor/npm",

"bower-asset-library": "vendor/bower"

}

}

}

And install the framework: composer require yiisoft/yii2

Now open the entry script of your application (on ZendFramework, Laravel, Symfony, and many more), require the Yii2 autoloader, and create the Yii application instance:

require( DIR . '/../vendor/autoload.php');

require( DIR . '/../vendor/yiisoft/yii2/Yii.php');

$config = require( DIR . '/../config/yii/web.php');

new yii\web\Application($config);

That’s it! Now you can use Yii::$app instances, models, widgets and other components from Yii2.

How it works...

In the first case we just install a new Composer package in our project and use it, because its composer. j son file defines all aspects of autoloading library files.

But in the second case we did not have Composer packages and registered the files in the autoloading mechanism manually. In Yii2 we can use aliases and Yii:: $classMap for registering the roots of PSR-4 namespaces and for single files.

But as an alternative we can use Composer autoloader for all cases. Just define an extended autoload section in the composer.json file like this:

"autoload": {

"psr-0": { "": "old/" },

"psr-4": {"awesome\\": "awesome/"},

"files": ["old/functions.php"]

}

Apply the changes using this command: composer update

Right now you can remove aliases and $classMap definitions from your configuration files and ensure the example page still works correctly: