

Tips and Tricks

Setup Tips

If you don't plan to execute the system on your local PC then all you need to do is just cloning the system Repository, and you're done.

If you plan to try executing the system on you pc. You need to have a webserver and database setup with PHP enabled.

- On Windows you may use [EasyPHP](#) or [XAMPP](#)
- On Linux you just need to install php5, apache and mysql packages from your package manager (also phpmyadmin is very useful)
- On Mac: [MAMP](#)

Remember to install at least PHP version 5.4 that is the minimum for using the Yii framework and make sure your Apache server has the **mod_rewrite** module enabled.

Also make sure your PHP installation has the **intl** extension enabled.

Then you should do the following actions:

1. Download this archive:
<https://github.com/yiisoft/yii2/releases/download/2.0.0/yii-basic-app-2.0.0.tgz> and extract ONLY the vendor folder in the root of the system (don't worry, it will not be tracked by git)
A better alternative to this step is described below in the Composer section
2. Configure your web server to have its DocumentRoot in the /web directory of the framework (and make sure it has write permissions to the framework folders and files)
3. Create a new database in your local DBMS (optionally also a new user with permissions on that database)
 - use utf8-general-ci encoding for the database, and, if available, InnoDB tables
4. Make a copy of the /config/db.php.example and call it /config/db.php Change it's contents to match the name of your database (keep localhost as host name) and put your database user and password to make the framework connect to the DBMS
5. Create the database structure by executing the command: `yii migrate` (See "Executables" section below)
 - If you are using MAMP and get error `Exception 'yii\db\Exception' with message 'SQLSTATE[HY000] [2002] No such file or directory, simply change the line within the config file db.php from 'dsn' => 'mysql:host=localhost;dbname=yii2basic', to 'dsn' => 'mysql:dbname=yii2basic;unix_socket=/Applications/MAMP/tmp/mysql/mysql.sock',`

6. Now open a shell (terminal or commands-prompt) and move to the root folder of the project. You should then execute the following commands to fill the database with the necessary tables:

- `./yii migrate --migrationPath=@yii/rbac/migrations/`
- `./yii migrate`

Development Tips

Here is a short collection of links and suggestions for using Yii:

Composer:

Yii, like many other PHP projects, uses composer for downloading packages and managing dependences.

You can get it from: <https://getcomposer.org/>

Once you have installed it, open your shell in the project's folder and execute this command:

```
composer global require "fxp/composer-asset-plugin:1.0.0-beta4"
```

if the composer command is not recognised, use "composer.phar" instead of "composer"

After this, you can execute:

```
composer install
```

This will download all the necessary packages and dependencies in the vendor folder. It can be used instead of downloading the yii-basic archive during the installation process. This command will also pull all the additional packages that we may require and are not included in the basic download.

Executables:

The framework has multiple execution points. One of course is the browser.

The other one is a command-line application, used for making life easier for the developers.

You can access this command line executing one of these files found in the root of the framework:

- *yii*: this is a Linux/Mac/Unix compatible shell script
- *yii.bat*: this is the Windows compatible shell script

They can be used for making lots of actions, like managing migrations and database versioning

Migrations:

Migrations are a way to put database structure under version control, just like we do with the normal code. This is achieved by modifying the database structure only using PHP code stored in the Framework folders. Basically every time you need to change the database structure you create a new migration Class, put in its `up()` method the code to make the database change and then execute it.

To create a migration class you can execute the command:

```
yii migrate/create
```

It will create a new migration class in the *migrations* folder, where you can write the code to modify the database.

To apply a migration you just need to run the command:

```
yii migrate
```

It will automatically detect which migrations have not yet been applied to the database and run them.

It is also possible to revert migrations to put the database in a previous state, **only** if the migrations have a `down()` method

<http://www.yiiframework.com/doc-2.0/guide-db-migrations.html>

NOTE: if you are creating a new database table please make sure you use the InnoDB engine, so we can use transactions if we need them in the future. Look at the `/migrations/m141120_092523_database_creation.php` file.

Gii - Generating common code

The Yii Framework has also a tool usable from the browser to create code, it is called Gii. It is reachable at this url (assuming your site URL is <http://localhost>):

<http://localhost/gii>

From gii you can automatically create Model classes from Database tables, controllers for your features and even controllers with some basic actions for each model (Creating a database record, updating it, deleting and showing it on the browser)

TIP: when you create a Model with Gii, please flag the Internationalization checkbox, so it is already made compatible with having a multilanguage website

More on Gii: <http://www.yiiframework.com/doc-2.0/guide-start-gii.html>

<http://www.yiiframework.com/doc-2.0/guide-tool-gii.html>

How to Develop:

The normal process for developing a function of the system usually follows these steps:

1. Create all the Model classes you may need for the current function. Example: if you need to manipulate Simulators and TimeSlots in the code, you will need the Simulator and TimeSlot model classes (this can be done with Gii from the corresponding database tables)
2. Find the right controller for your function. Controllers are logical containers of functions of the system. Example: all the functions that act on the Simulators should be methods of the SimulatorsController class. If it doesn't exist yet you should create it (Gii may be useful in this case too)

3. Then develop the function you want. Example: if you are to develop the function to add a new simulator to the system you will add the `addAction()` method to the `SimulatorController`
4. After you have the code, you actually need the html page to show on the browser, you can create the View file for this function in the `/views/<ControllerID>/<viewName>.php`. Example: for the add function of a simulator you would create the `/views/simulators/add.php` file containing all the html and php to actually render the data on the browser.

For more info on the components of the framework see:

<http://www.yiiframework.com/doc-2.0/guide-structure-overview.html>

Seeing a page on the browser:

One of the first questions you may have is how to actually see on the browser a specific page you have developed. Once you have made your controller and view of the function you are developing you can see it by visiting the following URL (assuming your website is

<http://localhost/>):

<http://localhost/<ControllerID>/<ViewName>>

Example: in the previous example of an Add function for the simulators, you would open the web page at this url: <http://localhost/simulators/add>

For more information on associating and customizing the url of an action:

<http://www.yiiframework.com/doc-2.0/guide-runtime-routing.html>

Translation:

Yii has a powerful internationalization engine, it can support translation of every single string in the whole application, so long as you surround it with the `Yii::t()` method.

Every string that is passed in the `Yii::t()` method before printing it on the page will be translatable in translation files. It is possible to have special placeholders in the string passed to `Yii::t()` for managing date formats, plurals, numbering, and many other things.

It is also possible to have completely different view files (i.e. html pages) for each language if needed.

More on i18n: <http://www.yiiframework.com/doc-2.0/guide-tutorial-i18n.html>

Question: isn't this bad for performance?

Answer: Yes, but in a production environment you would have Caching enabled and this solves the problem

Bootstrap and jQuery:

Bootstrap and jQuery are already enabled and included by default in all the pages of the application.

REST Interface:

Yii already provides a full set of facilities for easily implementing REST services without too much effort: <http://www.yiiframework.com/doc-2.0/guide-rest-quick-start.html>

Helpers and Widgets:

Yii comes with a full set of widgets and helpers that help you creating easily maintainable and correct html code.

There are classes provided easily for making forms based on models.

These classes also help in creating secure code

<http://www.yiiframework.com/doc-2.0/guide-input-forms.html>

When you need to insert URLs in the html code, you may find the `\yii\helpers\Url::to()` method very useful. It automatically composes urls for you, you just need to pass informations about the controller, action and data that will handle the request of that url.

<http://www.yiiframework.com/doc-2.0/guide-runtime-routing.html>