

CakePHP Console Cheat Sheet



Details extracted from CakePHP 3.2.x command prompt.

Installation

```
composer require --dev cakephp/bake:~1.0
```

PhpStorm comes with **composer** built in.

To install CakePHP via PhpStorm, use the File → New → Composer Project.

Then filter for *CakePHP* and select the `cakephp\app` composer project.

CakePHP Console

Run the CakePHP console from project root `bin/cake` (use `bin\cake` for Windows)

Possible "shells" `bake, migrations, i18n, orm_cache, plugin, routes, server, behavior_time, console, orm`

Run Cake Server `bin/cake server` (usually on localhost port 8765)

Run migrations shell `bin/cake migrations -h` (with help)

Run bake with plugin prefix `bin/cake bake.bake -h`

⚠ PHP CLI must be installed and in the path.

Bake Shell Common Options

help	<code>--help</code> or <code>-h</code>
Verbose output	<code>--verbose</code> or <code>-v</code>
Quiet output	<code>--quiet</code> or <code>-q</code>
Plugin to bake into	<code>--plugin</code> or <code>-p</code>
Force overwrite of existing files	<code>--force</code> or <code>-f</code>
Datasource connection to get data from (default: default)	<code>--connection</code> or <code>-c</code>
Theme to use when baking	<code>--theme</code> or <code>-t</code>
Do not bake a test skeleton	<code>--no-test</code>
Do not bake generic CRUD action methods	<code>--no-actions</code>

📌 These options are common to most of the cake shell commands.

Server Shell

Run server shell with (defaults)	<code>bin/cake server</code>
Run server shell and get help	<code>bin/cake server --help</code>
Run server shell on port 9999	<code>bin/cake server --port 9999</code>
Run server shell using IP of 10.0.0.2	<code>bin/cake server --host 10.0.0.2</code>
Run server shell with document root dummy/app/root	<code>bin/cake server --document_root dummy/app/root</code>
Run server shell with verbose output	<code>bin/cake server --verbose</code>
*** example	<code>bin/cake -H 10.0.0.2 -p 9999 -d dummy/app/root -v</code>

Bake Shell: Model

Run bake model shell and list all tables	<code>bin/cake bake model</code>
Bake model class for Friends table	<code>bin/cake bake model Friends</code>
Run bake model shell and get help	<code>bin/cake bake model --help</code>
Generalised use:	<code>bin/cake bake model [subcommand] [options] [<modelName>]</code>

Options *other than the default ones*

Table name if "non-conventional" table naming used	<code>--table <TableName></code>
Disable generation of Entity class	<code>--no-entity</code>
Disable generation of Table class	<code>--no-table</code>
Disable generation of validation rules	<code>--no-validation</code>
Disable generation of a rules checker	<code>--no-rules</code>
Disable generatin of associations	<code>--no-associations</code>
Disable generation of fields	<code>--no-fields</code>
Comma separated list of fields to make accessible	<code>--fields <fieldname>[, <fieldname>...]</code>
Comma separated list of hidden fields	<code>--hidden <fieldname>[, <fieldname>...]</code>
Set the primary key field or fields (comma separated)	<code>--primary-key <key-field>[, <key-field>...]</code>
Display field if you would like to select one	<code>--display-field <field-name></code>

Cake Routes Shell

List of all Routes	<code>bin/cake routes</code>
URL Parsing Testing	<code>bin/cake routes check route-and-options</code>
*** example	<code>bin/bake routes check /bookmarks/edit/1</code>
URL Generation Testing	<code>bin/cake routes generate controller:controllername action:action other options</code>
*** example	<code>bin/cake routes generate controller:Bookmarks action:edit 1</code>
*** example with string parameters	<code>bin/cake routes check "/bookmarks/?page=1&sort=title&direction=desc"</code>

Bake Shell: Controller

Bake a controller skeleton	<code>cake bake controller [subcommand] [options] [<name>]</code>
Bake the My controller with CRUD actions	<code>bin/cake bake controller MyController</code>
Bake all controllers with CRUD methods	<code>cake bake controller all</code>
⚠ Beware: This may overwrite existing controllers	
Options	
Plugin to bake into	<code>--plugin or -p</code>
Force overwrite of existing files	<code>--force or -f</code>
Datasource connection to get data from (default: default)	<code>--connection or -c</code>
Theme to use when baking	<code>--theme or -t</code>



Bake Shell: Controller (cont)

Components to use (*comma separated*) `--components`

Helpers to use (*comma separated*) `--helpers`

Namespace/routing prefix to use `--prefix`

Do not bake a test skeleton `--no-test`

Do not bake generic CRUD action methods `--no-actions`

`<name>` is the name of the controller to bake, and you may use `Plugin.name` to bake controllers into plugins (optional).

Version Control

Version	Date	Notes
1.0	25 April 2016	Initial version, there are many more commands to add

i This will be updated each time we add new content to the cheat sheet.



By **Ady G** (AdyG)
cheatography.com/adyg/

Not published yet.
Last updated 25th April, 2016.
Page 3 of 3.

Sponsored by **Readability-Score.com**
Measure your website readability!
<https://readability-score.com>