],

'runtimePaths' => [

'@backend/runtime',

'@frontend/runtime', '@console/runtime',

],

],

],

// ...

];

How it works...

All console commands should be extended from the yii\console\controller class. Since all console commands are run in yii\console\Application instead of yii\web\Application, we don’t have a way to determine the value of the @webroot alias. Also, in the yii2-app-advanced template we have backend, frontend, and console subdirectories by default. For this purpose, we are creating configurable public properties called assetPaths and runtimePaths.

The console command structure itself is like a typical controller. We are defining several actions we can run via yii <console command>/<command action>.

As you can see, there are no views used, so we can focus on programming tasks instead of design, markup, and so on. Still, you need to provide some useful output so that users will know what is going on. This is done through simple PHP echo statements.

If your command is relatively complex such as message or migrate bundled with Yii, it’s a good decision to provide some extra description of the available options and actions. It can be done by overriding the getHelp method:

public function getHelp()

{

$out = "Clean command allows you to clean up various temporary data Yii and an application are generating.\n\n";

return $out . parent::getHelp();

}

Run the following command:

./yii help clean

You can see the full output as follows:

DESCRIPTION

Clean command allows you to clean up various temporary data Yii and an application are generating.

Removes content of assets and runtime directories.

SUB-COMMANDS

* clean/assets Removes temporary assets.
* clean/runtime Removes runtime content.

By default, when we run the shell command:

. /yii