Moving configuration parts into separate files

In the basic application template we have separated web and console configuration files. And usually we  
set some application components in the both the configuration files.

Moreover, when we develop a big application, we may face some inconvenience. For example, if we need  
to adjust some settings, we would most probably end up repeating the changes in both the web application  
config and console application config.

Getting ready

Create a new yii2 -app-basic application by using the Composer package manager, as described in the  
official guide at <http://www.yiiframework.com/doc-2.0/guidestart-installation.html>.

How to do it...

1. Open the config/web. php file and add the urlManager section to the components configuration:

'components' => [

// ...

'db' => require( DIR . '/db.php'),

'urlManager' => [

'class' => 'yii\web\UrlManager',

'enablePrettyUrl' => true,

'showScriptName' => false,

'rules' => [

'' => 'site/index',

'<\_c:[\w\-]+>/<id:\d+>' => '<\_c>/view',  
'<\_c:[\w\-]+/<\_a:[\w\-]+>>/<id:\d+>' => '<\_c>/<\_a>',

'<\_c:[\w\-]+>' =>

'<\_c>/index',

],

],

],

1. Create the config/urlRules.php file and move rules array into it:

<?php  
return [

'' => 'site/index',

'<\_c:[\w\-]+>/<id:\d+>' => '<\_c>/view',

'<\_c:[\w\-]+/<\_a:[\w\-]+>>/<id:\d+>' => '<\_c>/<\_a>',

'<\_c:[\w\-]+>' => '<\_c>/index',

];

1. Replace the rule array with the file that requires this:

'urlManager' => [

'class' => 'yii\web\UrlManager',

'enablePrettyUrl' => true,

'showScriptName' => false,

'rules' => require( DIR . '/urlRules.php'),

],

How it works.

The preceding technique relies on the fact that Yii configuration files are native PHP files with arrays:

<?php

return [...];

Let’s look at the require construct:

'rules' => require( DIR . '/urlRules.php'),

When we use this, it reads the file specified, and, if there is a return statement inside this file, it returns a  
value.

Therefore, moving a part out of the main configuration file into a separate file requires creating a separate  
file, moving the configuration part into it right after the return statement, and using require in the main  
configuration file.

If separate applications (in our example, these are web applications and console applications) require some  
common configuration parts, then we can use require to move them into a separate file.

See also

In order to learn more about PHP require and include statements, refer to the following URLs:

* [http://php.net/mannal/en/function.require.php](http://php.net/manual/en/function.require.php)
* [http://php.net./mani]a]/en/function.include.php](http://php.net/manual/en/function.include.php)