Changing an advanced application template

By default, Yii2’s Advanced template has console, frontend, and backend applications. However, in your  
specific case, you can rename the existing ones and create your own applications. For example you can  
add the api application if you develop an API for your site.

Getting ready

Create a new yii2-app-advanced project by using the Composer package manager, as described in the  
official guide at <https://github.com/yiisoft/yii2-app-advanced/blob/master/docs/guide/start-installation.md>.

How to do it...

1. Copy the backend directory content to a new api directory in the root of your application.
2. Open the api/config/main.php file and change the controllerNamespace option value:

return [

'id' => 'app-manager',

'basePath' => dirname( DIR ),

'controllerNamespace' =>

'api\controllers',

//

]

1. Open api/assets/AppAsset. php and api/controllers/SiteController . php and change the  
   namespaces from backend to api like this:

namespaces api\assets;  
namespaces api\controllers;

1. Open the api/views/layouts/main. php file and find the following row:  
   use backend\assets\AppAsset;

Change it to this:

use api\assets\AppAsset;

1. Open common/config/bootstrap.php and add the @api alias for the new application:

<?php

Yii::setAlias('@common', dirname( DIR ));

Yii::setAlias('@frontend', dirname(dirname( DIR )) . '/frontend');

Yii::setAlias('@backend', dirname(dirname( DIR )) . '/backend');

Yii::setAlias('@console', dirname(dirname( DIR )) . '/console');

Yii::setAlias('@api', dirname(dirname( DIR )) . '/api);

1. Open the environments directory, and in the dev and prod subdirectories make the api directories  
   copies of backend.
2. Open the environments/index. php file and add rows for the api application:  
   return [

'Development' => [

'path' => ' dev',

'setWritable' => [

'backend/runtime',

'backend/web/assets',

'frontend/runtime',

'frontend/web/assets',

'api/runtime',

'api/web/assets',

],

'setExecutable' => [

' yii',

'tests/codeception/bin/yii',

],

'setCookieValidationKey' => [

'backend/config/main-local.php',

'frontend/config/main-local.php',

'api/config/main-local.php',

],

],

'Production' => [

'path' => ' prod ',

'setWritable' => [

'backend/runtime',

'backend/web/assets',

'frontend/runtime',

'frontend/web/assets',

'api/runtime',

'api/web/assets',

],

'setExecutable' => [

'yii',

],

'setCookieValidationKey' => [

'backend/config/main-local.php',

'frontend/config/main-local.php',

'api/config/main-local.php',

],

],

];

Now you have the console, frontend, backend, and api applications.

How it works...

The Advanced application template is a set of applications with custom aliases, such as @frontend,  
@backend, @common, and @console and corresponding namespaces instead of the simple @app alias for the

Basic template.

You can easily add, remove, or rename this applications (with their aliases and namespaces) if needed.

See also

For getting more information about the usage of application directory structures refer to  
[https://github.com/yiisoft/yri2-app-advanced/tree/master/docs/gurde](https://github.com/yiisoft/yii2-app-advanced/tree/master/docs/guide).