Creating components

If you have some code that looks like it can be reused but you don’t know if it’s a behavior, widget, or something else, most probably it’s a component. A component should be inherited from the yii\base\component class. Later on, the component can be attached to the application and configured using the components section of the configuration file. That’s the main advantage compared with using just a plain PHP class. Additionally, we are getting behavior, event, getter, and setter support.

For our example, we’ll implement a simple Exchange application component that will be able to get currency rates from the <http://fixer.io> site, attach it to the application, and use it.

Getting ready

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

How to do it...

For getting the currency rates, our component should send a HTTP GET query to a service URL such as <http://api.fixer.io/2016-05-14?base=USD>.

The service must return all supported rates on the nearest working day:

{

"base":"USD",

"date":"2016-05-13",

"rates": {

"AUD":1.3728,

"BGN":1.7235,

"ZAR":15.168,

"EUR":0.88121

}

}

The component should extract needle currency from the response in JSON format and return a target rate:

1. Create the components directory in your application structure.
2. Create the component class example with the following interface:

<?php

namespace app\components;

use yii\base\Component;

class Exchange extends Component {

public function getRate($source, $destination, $date = null)

{

}

}