3. Implement the component functional:

<?php

namespace app\components;

use yii\base\Component;

use yii\base\InvalidConfigException;

use yii\base\InvalidParamException;

use yii\caching\Cache;

use yii\di\Instance;

use yii\helpers\Json;

class Exchange extends Component {

/\*\*

\* @var string remote host \*/

public $host = 'http://api.fixer.io';

/\*\*

\* @var bool cache results or not \*/

public $enableCaching = false;

/\*\*

\* @var string|Cache component ID

\*/

public $cache = 'cache';

public function init()

{

if (empty($this->host)) {

throw new InvalidConfigException('Host must be set.');

}

if ($this->enableCaching) {

$this->cache = Instance::ensure($this->cache, Cache::className());

}

parent::init();

}

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

{

$this->validateCurrency($source);

$this->validateCurrency($destination);

$date = $this->validateDate($date);

$cacheKey = $this->generateCacheKey($source, $destination, $date); if (!$this->enableCaching || ($result = $this->cache->get($cacheKey)) ===

false) {

$result = $this->getRemoteRate($source, $destination, $date); if ($this->enableCaching) {

$this->cache->set($cacheKey, $result);

}

}

return $result;

}

private function getRemoteRate($source, $destination, $date)

{

$url = $this->host . '/' . $date . '?base=' . $source;

$response = Json::decode(file\_get\_contents($url)); if (!isset($response['rates'][$destination])) {

throw new \RuntimeException('Rate not found.');

}

return $response['rates'][$destination];

}

private function validateCurrency($source)