Custom panel for debug extension

The Yii2 - debug extension is a powerful tool for debugging own code, analyzing request information or  
database queries, and so on. Therefore, you can add your own panel for any custom report.

Getting ready

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

How to do it...

1. Create the panels directory on the root path of your site.
2. Add a new userPanel class:

<?php

namespace app\panels;

use yii\debug\Panel;  
use Yii;

class UserPanel extends Panel  
{

public function getName()

{

return ' User ' ;

}

public function getSummary()

{

return Yii::$app->view->render('@app/panels/views/summary', ['panel' =>

$this]);

}

public function getDetail()

{

return Yii::$app->view->render('@app/panels/views/detail', ['panel' =>

$this]);

}

public function save()

{

$user = Yii::$app->user;

return !$user->isGuest ? [

'id' => $user->id,

'username' => $user->identity->username,

] : null;

}

}

1. Create the panels/view/summary.php view with the following code:

<?php

/\* @var $panel app\panels\UserPanel \*/  
use yii\helpers\Html;

?>

<div class="yii-debug-toolbar block">

<?php if (!empty($panel->data)): ?>

<a href="<?= $panel->getUrl() ?>">

User

<span class="yii-debug-toolbar label yii-debug- toolbar label\_info">

<?= Html::encode($panel->data['username']) ?>

</span>

</a>

<?php else: ?>

<a href="<?= $panel->getUrl() ?>">Guest session</a>

<?php endif; ?>

</div>

1. Add the panels/view/detail.php view with the following code:

<?php

/\* @var $panel app\panels\UserPanel \*/  
use yii\widgets\DetailView;

?>

<h1>User profile</h1>

<?php if (!empty($panel->data)): ?>

<?= DetailView::widget([

'model' => $panel->data,

'attributes' => [

'id',

'username',

]

]) ?>

<?php else: ?>

<p>Guest session.</p>

<?php endif;?>

1. Turn on your toolbar in the config/web. php configuration file:

if (YII\_ENV\_DEV) {

$config['bootstrap'][] = 'debug';

$config['modules']['debug'] = [

'class' => 'yii\debug\Module',

'panels' => [

'views' => ['class' => 'app\panels\UserPanel'],

],

];

$config['bootstrap'][] = 'gii';

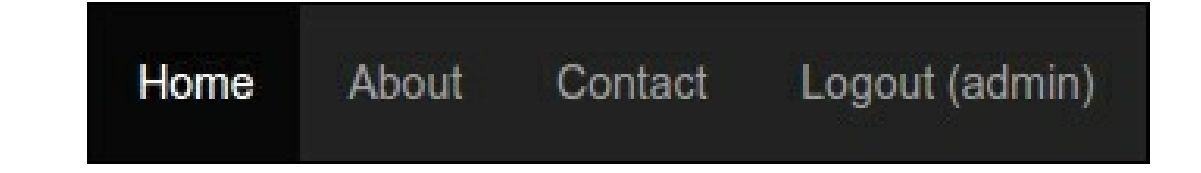
$config['modules']['gii'] = 'yii\gii\Module';

}

1. Reload the index page and look for the Guest Session cell at the end of the debug panel:



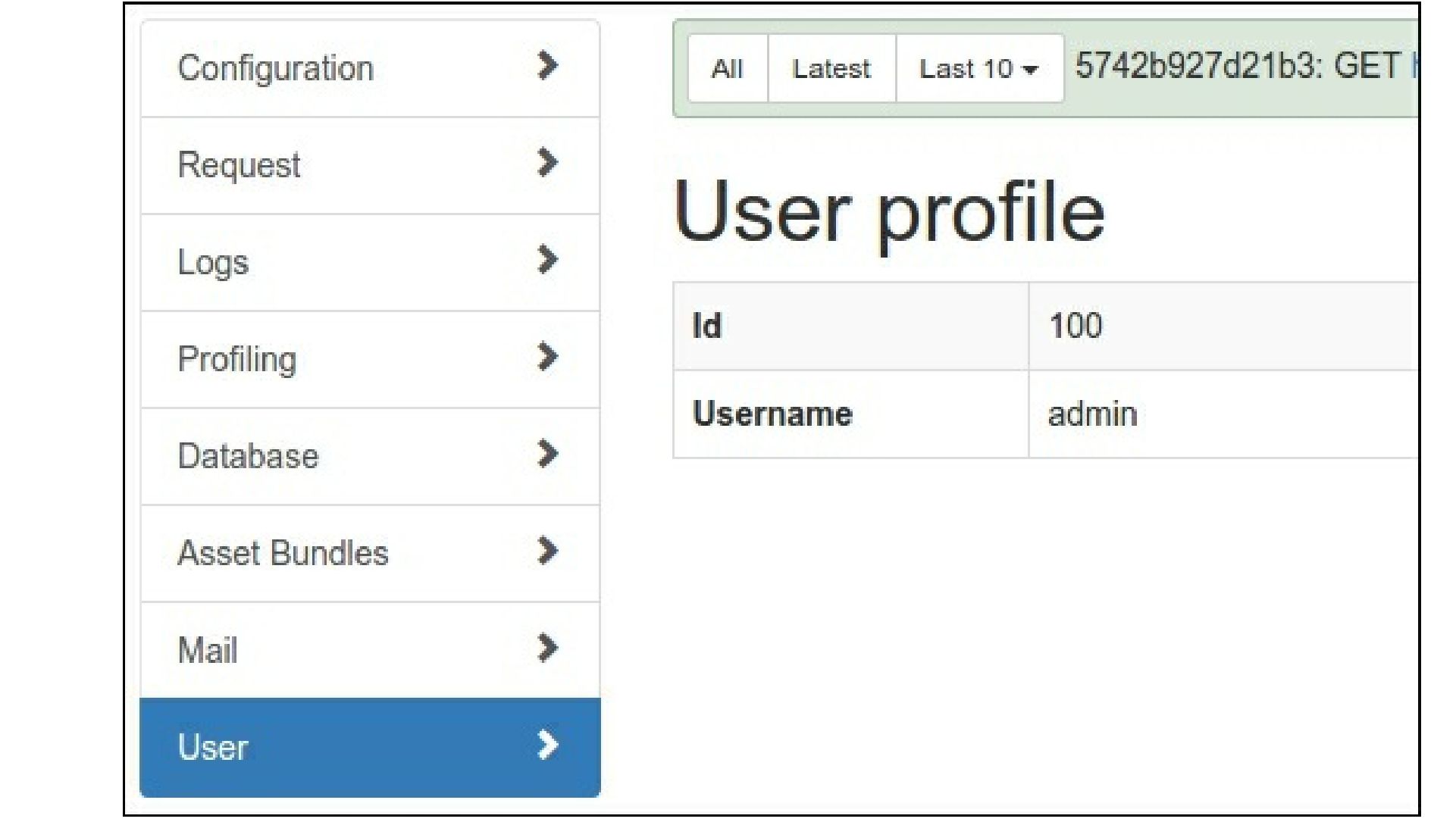
1. Log in to your site with the admin username and the admin password. In a success case, you must  
   see your username in the main menu:



8. Observe the debug panel again. Right now, you will see the admin username:



9. You can click on the username in the debug panel and see the detailed user information:



How it works...

To create our own panel for the yii2-debug module, we need to extend the yii\debug\Panel class and  
override its template methods:

* getName (): The label for menu item on debug detail page
* getsummary(): The debug panel cell code
* getDetail(): The detail page view code
* save (): Your information which will be saved in debug storage and load back into the $panel-  
  >data field

Your object can store any debug data and display it on the summary block of panel and on the detail page.  
In our example, we store user information:

public function save()

{

$user = Yii::$app->user;  
return !$user->isGuest ? [

'id' => $user->id,

'username' => $user->identity->username,

] : null;

}

Display it on summary and detail pages from the $panel->data field.

Handling events

You can subscribe to any events of application or any component in the init () method. For example, the  
built-in yii\debug\panels\MailPanel panel collects and stores all sent messages:

class MailPanel extends Panel  
{

private $\_messages = [];

public function init()

{

parent::init();

Event::on(

BaseMailer::className(),

BaseMailer::EVENT\_AFTER\_SEND,  
function ($event) {

$message = $event->message;

$messageData = [

// ...

];

$this->\_messages[] = $messageData;

}

);

}

// ...

public function save()

{

return $this->\_messages;

}

}

Also, it displays a grid with the list of stored messages on our own detail page.

See also

* [In order to learn more about yii2 - debug extension, refer to http://www. yiiframework. c om/doc -](http://www.yiiframework.com/doc-2.0/ext-debug-index.html)  
  2.0/ext-debug-index.html
* [For more information about creating a views counter panel, refer to https://github.com/viisoft/vii2-](https://github.com/yiisoft/yii2-debug/blob/master/docs/guide/topics-creating-your-own-panels.md)  
  debug/blob/master/docs/guide/topics-creating-your-own-panels.md