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/gurde/topics-creating-your-own-panels.md