Magento Live

UK | 2016

Sergey Lysak
CEO at Eltrino

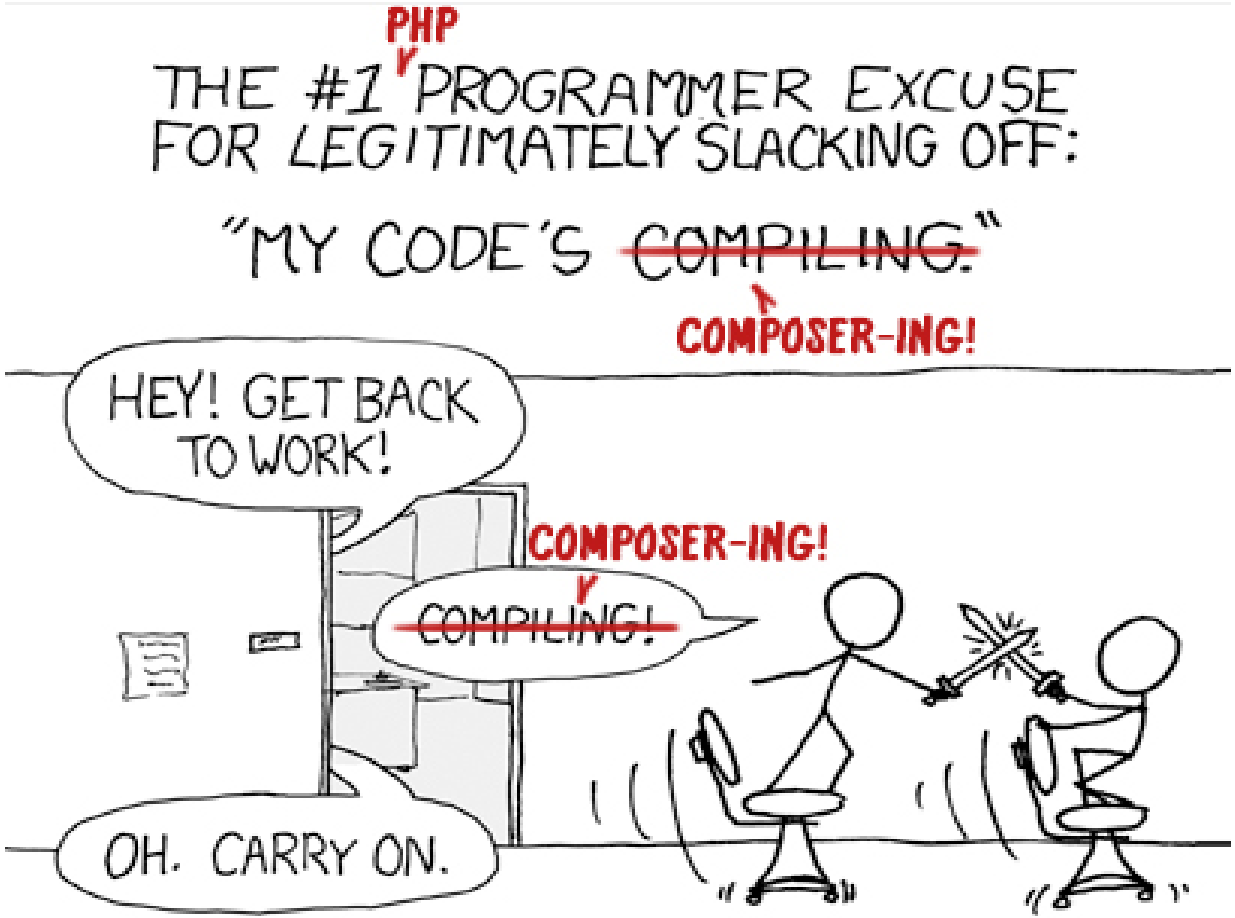
Lightning Fast Development with the Power of Composer

MagentoLive
UK | 2016

Agenda

1. Typical issues in Magento 2 that can be solved by Composer
2. Magic Magento-composer-installer
3. Example of implementation of third-party libraries
4. How to create new module
5. Backend validation comparison
6. Conclusions

Composer –
Dependency Manager for PHP


THE #1 PHP PROGRAMMER EXCUSE FOR LEGITIMATELY SLACKING OFF:
"MY CODE'S ~~COMPILING.~~ COMPOSER-ING!"
HEY! GET BACK TO WORK!
~~COMPILING!~~ COMPOSER-ING!
OH. CARRY ON.

Typical issues that can be solved by Composer

- Project dependency management of third-party libraries.
- Resolution of conflicts of libraries and priorities.
- Find and download into the project correct versions of libraries
- Generation autoload.php

Magento 2 and Composer combination features:

- Provides the ability to use third-party libraries while you don’t have to bundle them with source code of Magento 2.
- Offers a component-based architecture as well as reliable dependency management.
- Reduces extension conflicts as well as various compatibility issues.
- Streamlines your work with versioned dependencies.
- Introduces semantic versioning.
- Supports some useful standards, such as PHP Framework Interoperability.

Composer in Magento 2

- Magento 2 installer
- Magento-composer-installer

THE WALL

Composer in Magento 2

Avaliable Composer packages in Magento 2

- **magento2-module** - code inserted into **app/code**
- **magento2-theme** - code inserted into **app/design**
- **magento2-language** - code inserted into **app/i18n**
- **magento2-library** - code inserted into **lib/internal**
- **magento2-component** - code inserted into **root of magento installation**

Composer in Magento 2

MagentoLive
UK | 2016

Module installation

```
bash-4.3$
bash-4.3$ composer require vendor/module:version
bash-4.3$ *** some magic ***
bash-4.3$ Done.
bash-4.3$
```

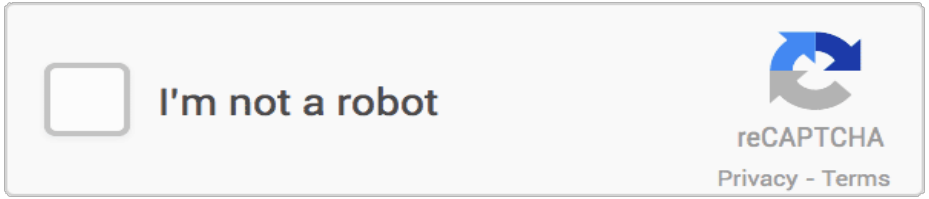
Magic Magento-composer-installer

Module configuration for Composer

```
{
    "name": "eltrino/recaptcha",
    "description": "Magento 2 Eltrino ReCaptcha",
    "require": {"php": "~5.5.0|~5.6.0|~7.0.0"...},
    "type": "magento2-module",
    "version": "100.0.0",
    "license": [...],
    "extra": {
        "map": [
            [
                "*",
                "Eltrino/ReCaptcha"
            ]
        ]
    }
}
```

Example of implementation of third-party libraries

Google ReCaptcha



Search and and installing the necessary libraries

http://packagist.org
google/recaptcha

```
bash-4.3$ composer require google/recaptcha
bash-4.3$ *** some magic ***
bash-4.3$ Done.
```

Existing complications in creating modules for Magento 2

- To build structure from scratch - **too long**
- The use of simple modules - **deprecated**
- ihb/moduleCreator - **ok**

Module installation ihb/moduleCreator

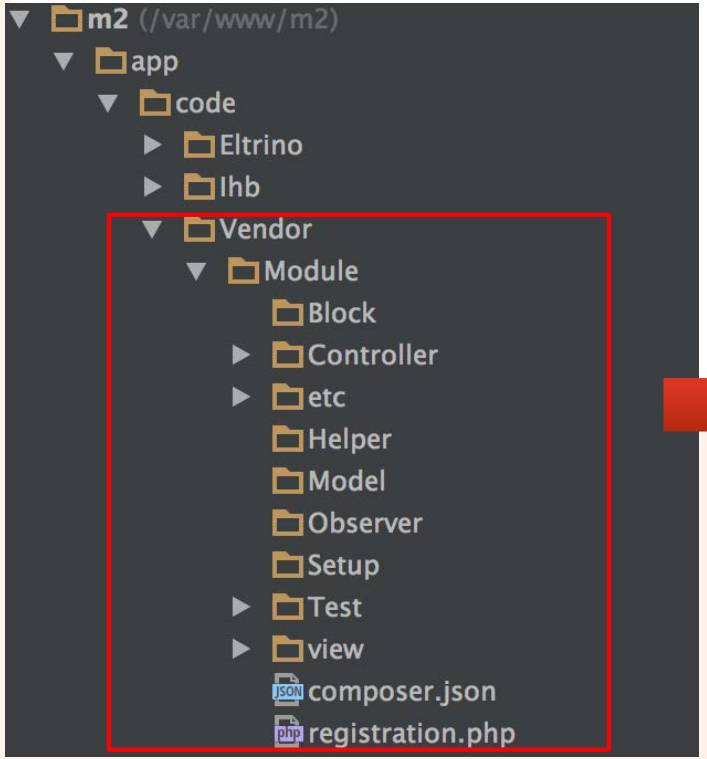
```
bash-4.3$
bash-4.3$ composer require ihb/moduleCreator:dev-master
bash-4.3$ *** some magic ***
bash-4.3$ Done.
bash-4.3$
bash-4.3$ bin/magento setup:upgrade
```

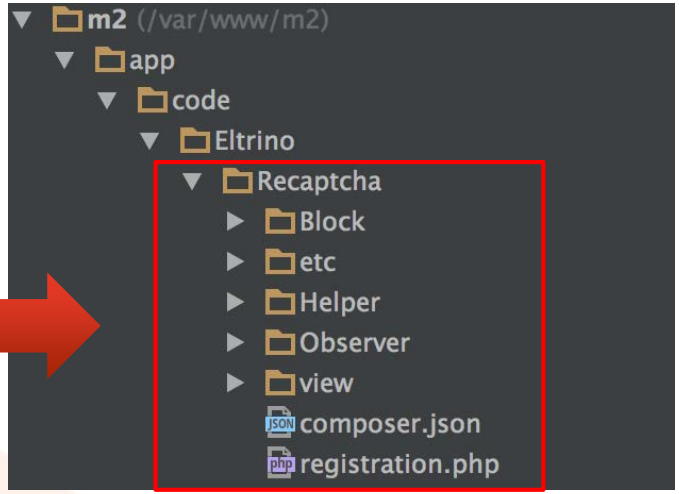
How to create new module

```
bash-4.3$
bash-4.3$ bin/magento ihb:module-create Vendor_Module
bash-4.3$
```

Simple module structure in app/code/Vendor/Module folder.

The structure of the new module





Add captcha output to contact form

magento2/app/code/Eltrino/ReCaptcha/view/frontend/layout/contact_index_index.xml

```
<page xmlns:xsi="..." xsi:noNamespaceSchemaLocation="...">
    <body>
        <referenceContainer name="form.additional.info">
            <block class="Eltrino\ReCaptcha\Block\Captcha" name="contactCaptcha" template="captcha.phtml"/>
        </referenceContainer>
    </body>
</page>
```

magento2/app/code/Eltrino/ReCaptcha/view/frontend/templates/captcha.phtml

```
<?php /** @var $this \Eltrino\ReCaptcha\Block\Captcha */ ?>
<div class="g-recaptcha" data-sitekey="<?php echo $this->getSiteCode() ?>"></div>
<script type="text/javascript" src="https://www.google.com/recaptcha/api.js?hl=en_US"></script>
```

https://github.com/google/recaptcha or http://www.google.com/recaptcha/intro/index.html

Result

Contact Us

Write Us

Jot us a note and we'll get back to you as quickly as possible.

Name *

Email *

Phone Number

What's on your mind? *

I'm not a robot

reCAPTCHA
Privacy - Terms

Submit

Backend validation – Observer

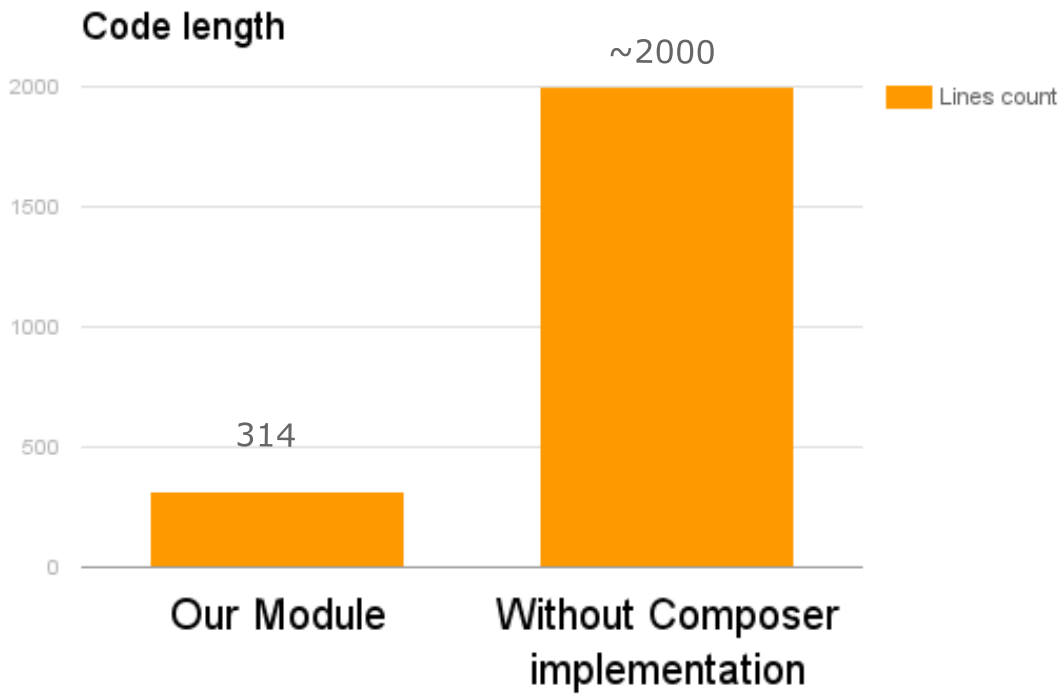
magento2/app/code/Eltrino/ReCaptcha/etc/events.xml

```
<config xmlns:xsi="..." xsi:noNamespaceSchemaLocation="...">
    <event name="controller_action_predispatch_contact_index_post">
        <observer name="captchaPreDispatch" instance="Eltrino\ReCaptcha\Observer\Predispatch" />
    </event>
</config>
```

Validation

```php
/**
 * @param \Magento\Framework\Event\Observer $observer
 * @return void
 * @SuppressWarnings(PHPMD.UnusedFormalParameter)
 */
public function execute(\Magento\Framework\Event\Observer $observer)
{
    if (!$this->helper->isEnabled()) {
        return;
    }

    /** @var \Magento\Framework\App\Action\Action $controller */
    $controllerAction = $observer->getControllerAction();
    /** @var \Magento\Framework\App\Request\Http $request */
    $request = $observer->getRequest();

    $reCaptchaCode = $request->getParam('g-recaptcha-response');

    if (!$reCaptchaCode) {
        $this->reCaptchaFailed($controllerAction);
        return;
    }

    $reCaptcha = new \ReCaptcha\ReCaptcha($this->helper->getSecretKey());
    $response = $reCaptcha->verify($reCaptchaCode, $request->getServer('REMOTE_ADDR'));

    if (!$response->isSuccess()) {
        $this->reCaptchaFailed($controllerAction);
        return;
    }
}
```

Comparison


Code length
~2000
314
2000
1500
1000
500
0
Lines count
Our Module
Without Composer implementation

Conclusions

Thank you for attention!

sergey@eltrino.com