

Shopware 6

for

Magento Developers

No silence

Don't Mute, Get a Better Headset

-- Matt Mullenweg

Shopware is *Symfony!*

Installation

- <https://docs.shopware.com/de/shopware-platform--dev-de>
- <https://dockware.io/docsx>

DevMode

```
<project root>  
└── .env
```

```
APP_ENV=dev
```

Plugins in Shopware 6

composer

<https://github.com/shopware/production>

Minimum Plugin

Minimum Plugin

```
<project root>  
└─ custom  
    └─ plugins  
        └─ PluginQuickStart  
            └─ src  
                └─ PluginQuickStart.php  
            └─ composer.json
```

composer.json

```
{
  "name": "swag/plugin-quick-start",
  "description": "Plugin quick start plugin",
  "version": "v1.0.0",
  "type": "shopware-platform-plugin",
  "license": "MIT",
  "authors": [
    {
      "name": "shopware AG",
      "role": "Manufacturer"
    }
  ],
  "require": {
    "shopware/core": "6.1.*"
  },
  "autoload": {
    "psr-4": {
      "Swag\\PluginQuickStart\\": "src/"
    }
  }
}
```

composer.json 2

```
{
  "extra": {
    "shopware-plugin-class": "Swag\\PluginQuickStart\\PluginQuickStart",
    "label": {
      "de-DE": "Beispiel Plugin",
      "en-GB": "Example Plugin"
    },
    "description": {
      "de-DE": "Deutsche Beschreibung des Plugins",
      "en-GB": "English description of the plugin"
    }
  }
}
```

Magento:

- `app/etc/modules/Namespace_Module.xml`
- `app/code/community/Namespace/Module/config.xml`

Shopware

- `PluginQuickStart.php`

PluginQuickStart.php

```
<?php declare(strict_types=1);
```

```
namespace Swag\PluginQuickStart;
```

```
use Shopware\Core\Framework\Plugin;
```

```
class PluginQuickStart extends Plugin {}
```

Models

Blocks

Helper

Symfony: Freedom!

```
<project root>
├── custom
│   ├── plugins
│   │   ├── PluginQuickStart
│   │   │   └── src
│   │   │       ├── Controller
│   │   │       │   └── MyController.php
│   │   │       ├── Service
│   │   │       │   └── MyService.php
│   │   │       ├── Subscriber
│   │   │       │   └── MySubscriber.php
│   │   │       └── WhateverYouWant
│   │   │           └── MyCoolClass.php
```

data/install Scripts

```
<?xml version="1.0"?>
<config>
  <global>
    <resources>
      <catalog_setup>
        <setup>
          <module>Mage_Catalog</module>
          <class>Mage_Catalog_Model_Resource_Setup</class>
        </setup>
      </catalog_setup>
    </resources>
  </global>
</config>
```


-> Migrations!

```
bin/console database:create-migration  
--plugin PluginQuickStart  
--name ExampleDescription
```

Creates Migration file

```
<project root>  
└─ custom  
    └─ plugins  
        └─ PluginQuickStart  
            └─ Migration  
                └─ Migration1546422281ExampleDescription.php
```

Migration1546422281ExampleDescription.php

```
<?php declare(strict_types=1);

namespace Swag\PluginQuickStart;

use Doctrine\DBAL\Connection;
use Shopware\Core\Framework\Migration\MigrationStep;

class Migration1546422281ExampleDescription extends MigrationStep
{
    public function getCreationTimestamp(): int
    {
        return 1546422281;
    }

    public function update(Connection $connection): void
    {
        $query = <<<SQL
CREATE TABLE IF NOT EXISTS
// ...
SQL;

        $connection->executeQuery($query);
    }

    public function updateDestructive(Connection $connection): void {}
}
```

PluginQuickStart.php

```
abstract class Plugin extends Bundle
{
    public function install(InstallContext $installContext): void {}

    public function postInstall(InstallContext $installContext): void {}

    public function update(UpdateContext $updateContext): void {}

    public function postUpdate(UpdateContext $updateContext): void {}

    public function activate(ActivateContext $activateContext): void {}

    public function deactivate(DeactivateContext $deactivateContext): void {}

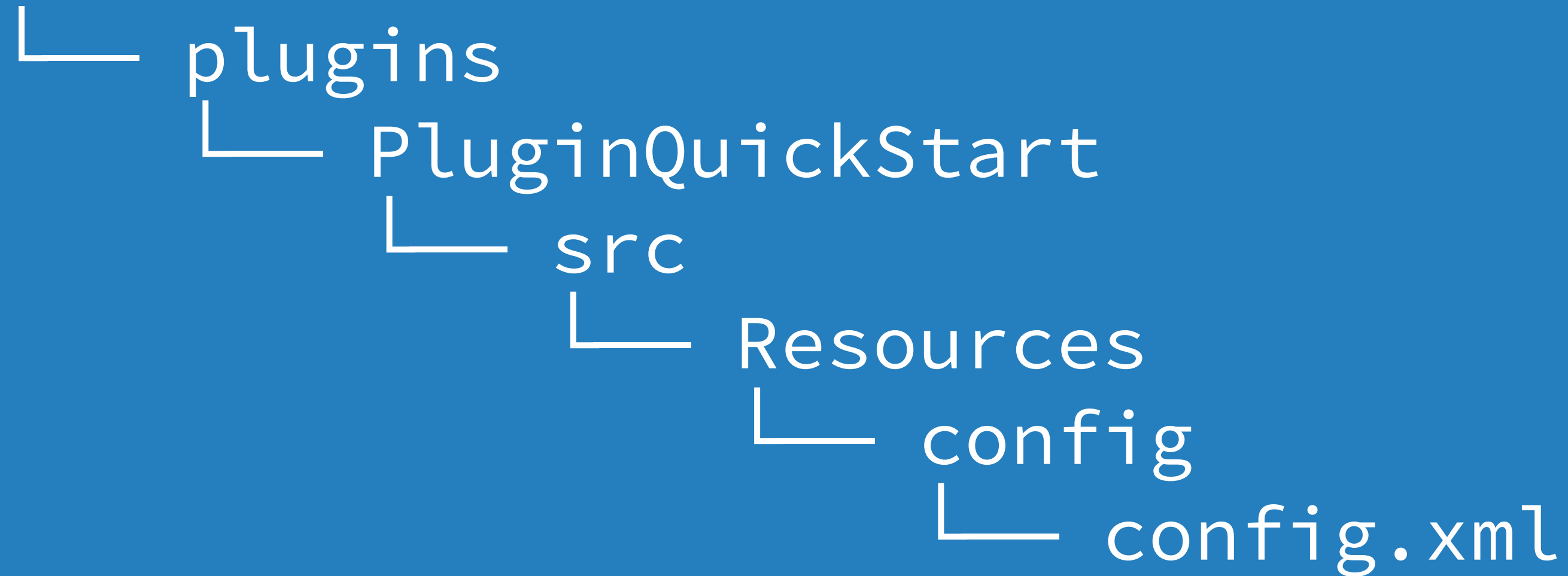
    public function uninstall(UninstallContext $uninstallContext): void {}
}
```

Configuration (system.xml)

Configuration (system.xml)

```
<?xml version="1.0"?>
<config>
  <sections>
    <contacts translate="label" module="contacts">
      <label>Contacts</label>
      <tab>general</tab>
      <frontend_type>text</frontend_type>
      <sort_order>100</sort_order>
      <show_in_default>1</show_in_default>
      <show_in_website>1</show_in_website>
      <show_in_store>1</show_in_store>
      <groups>
        <contacts translate="label">
          <label>Contact Us</label>
          <frontend_type>text</frontend_type>
          <sort_order>10</sort_order>
          <show_in_default>1</show_in_default>
          <show_in_website>1</show_in_website>
          <show_in_store>1</show_in_store>
          <fields>
            <enabled translate="label">
              <label>Enable Contact Us</label>
              <frontend_type>select</frontend_type>
              <source_model>adminhtml/system_config_source_yesno</source_model>
              <backend_model>contacts/system_config_backend_links</backend_model>
              <sort_order>10</sort_order>
              <show_in_default>1</show_in_default>
              <show_in_website>1</show_in_website>
              <show_in_store>1</show_in_store>
            </enabled>
          </fields>
        </contacts>
      </groups>
    </contacts>
  </sections>
</config>
```

Plugin configurations - cards



SW6 - config.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<config [...]>
  <card>
    <title>Advanced Configuration</title>
    <title lang="de-DE">Erweiterte Einstellungen</title>
    <input-field type="password">
      <name>secret</name>
      <label>Secret token</label>
      <label lang="de-DE">Geheim Schlüssel</label>
      <helpText>Your secret token for xyz...</helpText>
      <helpText lang="de-DE">Dein geheimer Schlüssel für xyz...</helpText>
    </input-field>
  </card>
</config>
```

— Namespaces removed, see SW6 docs!

Cron (Magento)

```
<config>
    <crontab>
        <jobs>
            <catalog_product_index_price_reindex_all>
                <schedule>
                    <cron_expr>0 2 * * *</cron_expr>
                </schedule>
                <run>
                    <model>catalog/observer::reindexProductPrices</model>
                </run>
            </catalog_product_index_price_reindex_all>
        </jobs>
    </crontab>
</config>
```

1. Command + „real“ cron

2. Scheduled task

Command + „real“ cron

```
<project root>
├── custom
│   ├── plugins
│   │   ├── PluginQuickStart
│   │   │   └── src
│   │   │       ├── Resources
│   │   │       │   ├── config
│   │   │       │   └── services.xml
│   │   │       └── Command
│   │   │           └── ExampleCommand.php
```

services.xml

```
<?xml version="1.0" ?>
```

```
<container [...]>
```

```
  <services>
```

```
    <service id="Swag\PluginQuickStart\Command\ExampleCommand">
```

```
      <tag name="console.command"/>
```

```
    </service>
```

```
  </services>
```

```
</container>
```

ExampleCommand.php

```
<?php declare(strict_types = 1);

namespace Swag\PluginQuickStart\Command;

use Symfony\Component\Console\Command\Command;
use Symfony\Component\Console\Input\InputInterface;
use Symfony\Component\Console\Output\OutputInterface;

class ExampleCommand extends Command
{
    protected static $defaultName = 'plugin-quickstart:example';

    protected function execute(InputInterface $input, OutputInterface $output)
    {
        $output->writeln('Do something...');
    }
}
```

Native cronjob

```
$ crontab -e
```

```
*/10 * * * * php /shopware/bin/console [...]:example
```

Scheduled task

```
<project root>
├── custom
│   ├── plugins
│   │   ├── PluginQuickStart
│   │   │   └── src
│   │   │       ├── Resources
│   │   │       │   ├── config
│   │   │       │   └── services.xml
│   │   │       └── ScheduledTask
│   │   │           ├── MyTask.php
│   │   │           └── MyTaskHandler.php
```

services.xml

```
<?xml version="1.0" ?>

<container [...]>

    <services>
        <service id="Swag\ PluginQuickStart\ScheduledTask\MyTask">
            <tag name="shopware.scheduled.task" />
        </service>

        <service id="Swag\ PluginQuickStart\ScheduledTask\MyTaskHandler">
            <argument type="service" id="scheduled_task.repository" />
            <tag name="messenger.message_handler" />
        </service>
    </services>
</container>
```


MyTask.php

```
<?php declare(strict_types=1);

namespace Swag\PluginQuickStart\ScheduledTask;

use Shopware\Core\Framework\MessageQueue\ScheduledTask\ScheduledTask;

class MyTask extends ScheduledTask
{
    public static function getTaskName(): string
    {
        return 'vendor_prefix.my_task';
    }

    public static function getDefaultInterval(): int
    {
        return 300; // 5 minutes
    }
}
```

MyTaskHandler.php

```
<?php declare(strict_types=1);

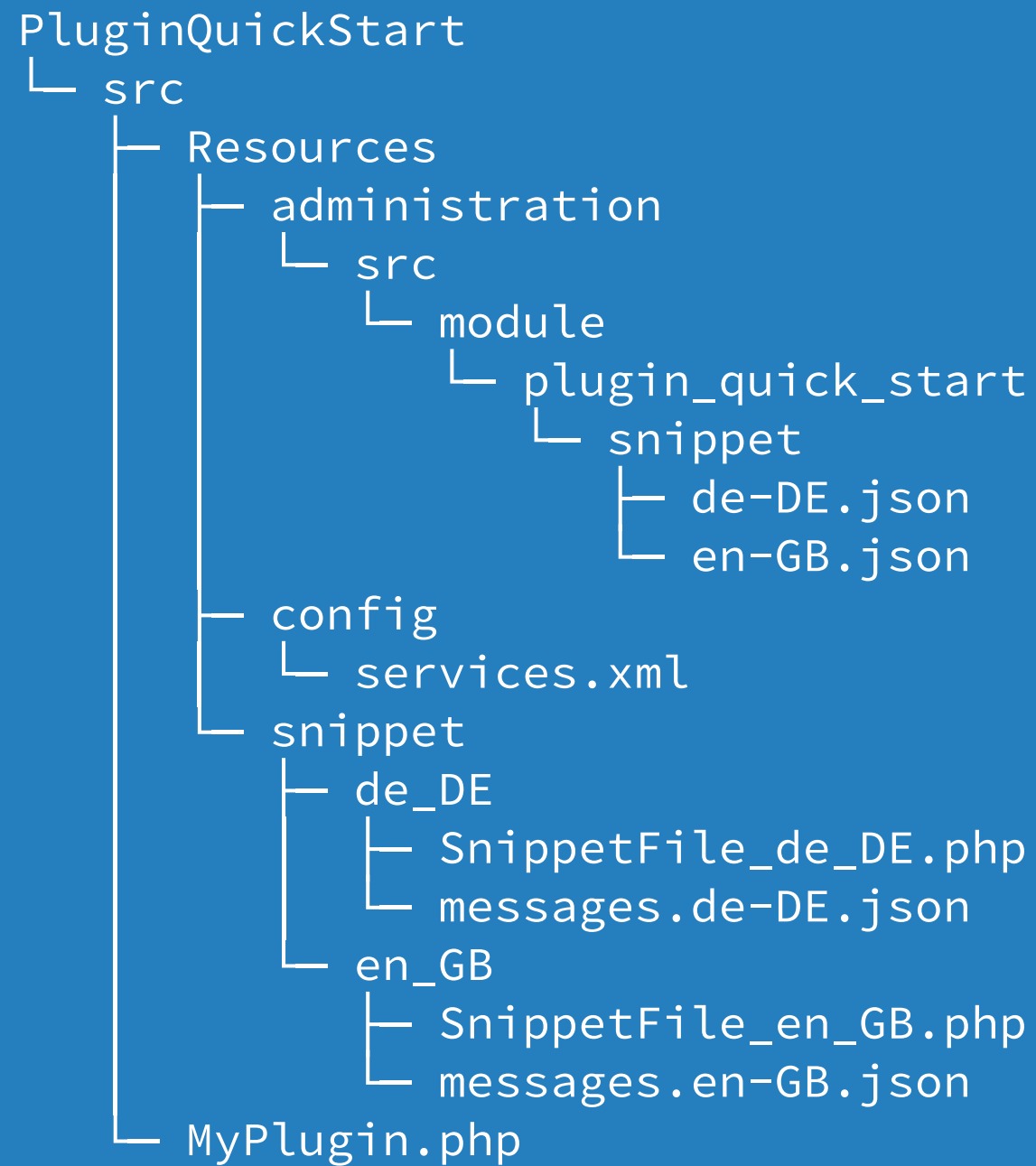
namespace Swag\PluginQuickStart\ScheduledTask;

use Shopware\Core\Framework\MessageQueue\ScheduledTask\ScheduledTaskHandler;

class MyTaskHandler extends ScheduledTaskHandler
{
    public static function getHandledMessages(): iterable
    {
        return [MyTask::class];
    }

    public function run(): void
    {
        echo 'Do stuff!';
    }
}
```

Translation



Event/Observer

Magento

```
<config/global>
<events>
  <log_log_clean_after>
    <observers>
      <catalog_product_compare_item>
        <class>catalog/observer</class>
        <method>catalogProductCompareClean</method>
      </catalog_product_compare_item>
    </observers>
  </log_log_clean_after>
</events>
</config/global>
```

Shopware 6

```
<project root>
├── custom
│   ├── plugins
│   │   ├── PluginQuickStart
│   │   │   └── src
│   │   │       ├── Resources
│   │   │       │   ├── config
│   │   │       │   └── services.xml
│   │   │       └── Subscriber
│   │   │           └── MySubscriber.php
```

services.xml

```
<?xml version="1.0" ?>  
<container [...]>  
  <services>  
    <service id="Swag\ PluginQuickStart\Subscriber\MySubscriber">  
      <tag name="kernel.event_subscriber"/>  
    </service>  
  </services>  
</container>
```


Routing

Magento

```
<config>  
    <frontend>  
        <rovers>  
            <atalog>  
                <args>  
                    <odule>Mage_Catalog</odule>  
                    <frontName>atalog</frontName>  
                </args>  
            </atalog>  
        </rovers>  
    </frontend>  
</config>
```

Magento 2

```
<config [...]>  
    <router id="standard">  
        <route id="catalog" frontName="catalog">  
            <module name="Magento_Catalog" />  
        </route>  
    </router>  
</config>
```

Shopware 6

```
<project root>
├── custom
│   ├── plugins
│   │   ├── PluginQuickStart
│   │   │   ├── Resources
│   │   │   │   ├── config
│   │   │   │   └── routes.xml
│   │   └── src
│   │       ├── Storefront
│   │       │   ├── Controller
│   │       │   └── ClearCartController.php
```

Shopware 6

```
<?xml version="1.0" encoding="UTF-8" ?>  
<routes [...]>  
  
    <import  
        resource="../../**/Storefront/Controller/*Controller.php"  
        type="annotation" />  
</routes>
```

Controller

```
<?php declare(strict_types=1);

namespace Swag\StorefrontController\Storefront\Controller;

use Shopware\Core\Framework\Routing\Annotation\RouteScope;
use Shopware\Storefront\Controller\StorefrontController;
use Symfony\Component\Routing\Annotation\Route;

/**
 * @RouteScope(scopes={"storefront"})
 */
class ClearCartController extends StorefrontController
{
    /**
     * @Route("/cart/clear", name="frontend.checkout.clearCart", methods={"GET"})
     */
    public function clearCart() {}
}
```

And on the next episode

- Emails
- Caching
- **Themes**
- Templates
- Layout
- Emails
- JavaScript
- CSS