

HALF A RUSH HOUR IN MAGENTO 2

SOMETHING TO SAY?
ON TWITTER PLEASE TAG WITH
#MHRH AND #MM19NL



BY AND WITH RIBIAN BLECHTZEL

RICO NEITZEL AND FABIAN BLECHSCHMIDT

3 — Attend mageunconference.org starting 29.11.2019 — Rico Neitzel and Fabian Blechschmidt

SLIDES ARE ONLINE

Slides are here:

<http://theslidesareonline.de/m2>

**WE'LL START
RIGHT AWAY!**

FIRST TALK FRONTEND

PRODUCT TYPES

Product Types can change on the fly!

- Simple is the default

PRODUCT TYPES - VIRTUAL

Weight
[global]

lbs

This item has no weight



PRODUCT TYPES - DOWNLOADABLE

Downloadable Information



☐ Is this downloadable Product?

To enable the option set the weight to no

PRODUCT TYPES - CONFIGURABLE

Configurations



Configurable products allow customers to choose options (Ex: shirt color). You need to create a simple product for each configuration (Ex: a product for each color).

Create Configurations

PRODUCT TYPES - CONFIGURABLE

Product Name *
[store view]

This is a required field.

SKU *
[global]

This is a required field.

Price *
[global]

This is a required field.

PRODUCT TYPES - GROUPED + BUNDLE

Products



10

demo ▾

Add Product ▲

Simple Product

Configurable Product

Grouped Product

Virtual Product

Bundle Product

Downloadable Product

Filters

 Default View

Search by keyword 

Actions ▾ 2046 records found

20 ▾ per page <

☐ ▾	ID ↓	Thumbnail	Name	Type	Attribute Set	SKU	Price	Quantity	Visibility			
☐	1		Joust Duffle Bag	Simple Product	Bag	24-MB01	₫34.00	100.0000	Catalog, Search	Enabled	Main Website	Edit
☐	2		Strive Shoulder	Simple	Bag	24-	₫32.00	100.0000	Catalog,	Enabled	Main	Edit

PRODUCT TYPES - GROUPED

Grouped Products 

A grouped product is made up of multiple, standalone products that are presented as a group. You can offer variations of a single product, or group them by season or theme to create a coordinated set. Each product can be purchased separately, or as part of the group.

Add Products to Group

<

1

of 1

>

	ID	Thumbnail	Name	Attribute Set	Status	SKU	Price	Default Quantity	Actions
	1		Joust Duffle Bag	Bag	Enabled	24-MB01	₫34.00		Remove
	2		Strive Shoulder Pack	Bag	Enabled	24-MB04	₫32.00		Remove
	3		Crown Summit Backpack	Bag	Enabled	24-MB03	₫38.00		Remove

PRODUCT TYPES - BUNDLE

Bundle Items 

Ship Bundle Items [global]

Together 

Add Option

< 1 of 1 >

  New Option 

Option Title *

Drop-down 

☒ Required

Is Default	Name	SKU	Default Quantity *	User Defined
 ☐	Joust Duffle Bag	24-MB01		☐
 ☐	Strive Shoulder Pack	24-MB04		☐

Add Products to Option

FULLPAGECACHING

- All customer related data is loaded via AJAX
- Pages can be cached by builtin or varnish cache

CHECKOUT

- Full AJAX (Asynchronous JavaScript and XML)
- API driven (headless! kind of...)
- only two steps

CHECKOUT

This is demo store. No orders will be fulfilled.

LUMA

[Sign In](#)

1 

2 

ShippingReview & Payments

Shipping Address

EMAIL ADDRESS 



You can create an account after checkout.

FIRST NAME 

LAST NAME 

COMPANY

STREET ADDRESS 

CITY 

STATE/PROVINCE

ZIP/POSTAL CODE 

COUNTRY 

PHONE NUMBER 



Order Summary

1 Item in Cart

Shipping Methods

☒ €5.00

☐ Fixed

☐ Flat Rate

Next

Copyright © 2016 Magento. All rights reserved.

CHECKOUT

This is demo store. No orders will be fulfilled.

 LUMA

Sign In

Shipping

Review & Payments

Payment Method:

CHECK / MONEY ORDER

☒ My billing and shipping address are the same

Testvorname Testnachname
Teststr. 12
Teststadt, Bremen 10000
Germany
[+49 111 11111111](#)

Place Order

Apply Discount Code

Order Summary

Cart Subtotal	€45.00
SHIPPING Flat Rate - Fixed	€5.00
Order Total	€50.00
1 Item in Cart	▼

Ship To:

Testvorname Testnachname
Teststr. 12
Teststadt, Bremen 10000
Germany
[+49 111 11111111](#)

Shipping Method:

Flat Rate - Fixed

Copyright © 2016 Magento. All rights reserved.

CHECKOUT

This is demo store. No orders will be fulfilled.

Default welcome msg! Sign In or Create an Account Default Store View ▾



Search entire store here... 



[Home](#) [Women](#) [Men](#) [Bags](#) [Blog](#) [Lookbook](#) [Contact](#)

Thank you for your purchase!

 You added Driven Backpack to your shopping cart.

Your order # is: 000000005.

We'll email you an order confirmation with details and tracking info.

[Continue Shopping](#)

You can track your order status by creating an account.

Email Address: testemail@fbtest.de

[Create an Account](#)

[Privacy and Cookie Policy](#)
[Search Terms](#)
[Orders and Returns](#)
[Advanced Search](#)
[Contact Us](#)

 Enter your email address [Subscribe](#)

Copyright © 2016 Magento. All rights reserved.

LET'S TALK BACKEND

RUN MODES

- Developer
- Default
- Production

DEVELOPER MODE

```
SetEnv MAGE_MODE developer  
bin/magento deploy:mode:set developer
```

- errors on screen
- missing subscribers trigger exception
- auto materialisation

DEFAULT MODE

```
SetEnv MAGE_MODE default  
bin/magento deploy:mode:set default
```

- no errors on screen
- auto materialisation

PRODUCTION MODE

```
SetEnv MAGE_MODE production  
bin/magento deploy:mode:set production
```

- no errors on screen
- no auto materialisation

Note: bin/magento triggers compilation and deployment!

MODE SETTINGS

can be found in env.php

```
return [  
    ...  
    'MAGE_MODE' => 'developer',  
    ...  
];
```

ONE SOURCE FITS ALL

<https://devdocs.magento.com/>

NAMING CONVENTIONS

- Vendor: Magento
- Component Catalog
[ucfirst, Module]
- Component blank
[strotolower, theme]

ALL IN ONE

- all **components** reside in their *single* directory now
- components are:
libraries, themes, languages, modules

FOLDER LOCATIONS

without composer you're working in:

- app/code/{Vendor}/{Module}
- app/design/frontend/{Vendor}/{theme}

COMPOSER INSTALLATION

projects can be fully installed by composer

```
composer create-project \  
    --repository=https://repo.magento.com/ \  
    magento/project-community-edition \  
    htdocs
```

COMPONENTS VIA COMPOSER

- component needs a `registration.php`
 - maps location to magento structure
- component needs a `composer.json`
 - enables installation via composer

REGISTRATION.PHP

```
use Magento\Framework\Component\ComponentRegistrar;  
  
ComponentRegistrar::register(  
    ComponentRegistrar::MODULE,  
    'Vendor_Module',  
    __DIR__  
);
```

Note: The 2nd argument has a different structure for other types

COMPOSER.JSON

```
{  
  "name": "magento/module-catalog",  
  "description": "N/A",  
  "require": {  
    "php": "~7.0.13|~7.1.0",  
    "magento/module-store": "100.2.*",  
    "magento/framework": "101.0.*",  
    "magento/module-ui": "101.0.*"  
  },  
  "type": "magento2-module",  
  "version": "102.0.7",  
  "autoload": {  
    "files": [  
      "registration.php"  
    ],  
    "psr-4": {  
      "Magento\\Catalog\\": ""  
    }  
  }  
}
```

DEPENDENCY INJECTION

DEPENDENCY INJECTION

Using Constructor Injection

```
public function __construct(  
    \Magento\Catalog\Model\ProductFactory $productFactory  
) {  
    $this->productFactory = $productFactory;  
}
```

INJECTABLES AND NON-INJECTABLES

INJECTABLES

any **shareable** instance:

- Sessions
- Services
- all stateless objects

NON INJECTABLES

any **non-shareable** instance:

- Products
- Categories
- Customers
- most stateful objects

OBJECTMANAGER JUST NO.

```
\Magento\Framework\App\ObjectManager::getInstance()  
→create( '\Magento\Catalog\Model\Product' )
```

REPOSITORIES

used to save/load injectables for non-injectables

Good example:

`\Magento\Cms\Model\BlockRepository`

CODE GENERATION

Magento **creates** php classes *on the fly*
stores them in `./generated/code`¹

¹see constructor injection!

FACTORIES

used to create injectables for non-injectables:

Product  Product**Factory**

Just append Factory!

PROXIES

used to create lazy instantiatables:

HeavyService ➡ HeavyService**Proxy**

Just append Proxy!

INTERCEPTORS

automatically generated if a *Plugin* is defined
for a class

MODULE DIRECTORY STRUCTURE

PREDEFINED DIRECTORY NAMES

- Controller - Action Controllers
- i18n - localization
- Setup - install scripts
- etc - configuration
- view - frontend specific files

DIRECTORY NAMES BY CONVENTION

- Api - Interfaces for **Services**
- Api/Data - Interfaces for **Data Objects**
- Console - bin/magento **Commands**

BY CONVENTION - OBVIOUS EDITION

- Plugin - Plugins
- Observer - Observer
- Model - Models
- Block - Blocks
- Test - Tests

REMAINING DIRECTORY NAMES

class directories can be arbitrary 

referenced by full qualified class names:

`\Buro71a\Core\Service\CustomerNameService`

CONFIG FILES IN .//ETC

XML VALIDATION

all xml files in magento contain
references to **xsd files** 🤘

Magento will *and* **PHPStorm can**
validate **xml files** *with these* **xsd files**

CONFIG FILES AREA SCOPING

extends and overwrites ./etc global config

- ./etc/frontend
- ./etc/adminhtml
- ./etc/webapi_rest
- ./etc/graphql

MODULE.XML

module loading sequence and
setup version for install scripts

```
<?xml version="1.0"?>
<config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:noNamespaceSchemaLocation="urn:magento:framework:Module/etc/module.xsd">
    <module name="Magento_Catalog" setup_version="2.2.6">
        <sequence>
            <module name="Magento_Eav" />
        </sequence>
    </module>
</config>
```

MODULE.XML

with declarative schema > 2.3 no setup_version

```
<?xml version="1.0"?>
<config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:noNamespaceSchemaLocation="urn:magento:framework:Module/etc/module.xsd">
    <module name="Magento_Catalog">
        <sequence>
            <module name="Magento_Eav" />
        </sequence>
    </module>
</config>
```

DB_SCHEMA.XML

```
<?xml version="1.0"?>
<schema xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="urn:magento:framework:Setup/Declaration/Schema/etc/schema.xsd">
  <table name="adminnotification_inbox" resource="default" engine="innodb" comment="Adminnotification Inbox">
    <column xsi:type="int" name="notification_id" padding="10" unsigned="true" nullable="false" identity="true"
      comment="Notification id"/>
    <column xsi:type="smallint" name="severity" padding="5" unsigned="true" nullable="false" identity="false"
      default="0" comment="Problem type"/>
    <constraint xsi:type="primary" referenceId="PRIMARY">
      <column name="notification_id"/>
    </constraint>
    <index referenceId="ADMINNOTIFICATION_INBOX_SEVERITY" indexType="btree">
      <column name="severity"/>
    </index>
  </table>
</schema>
```

DB_SCHEMA.XML

```
<?xml version="1.0"?>
<schema xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="urn:magento:framework:Setup/Declaration/Schema/etc/schema.xsd">
  <table name="adminnotification_inbox" resource="default" engine="innodb" comment="Adminnotification Inbox">
    <column xsi:type="int" name="notification_id" padding="10" unsigned="true" nullable="false" identity="true"
      comment="Notification id"/>
    <column xsi:type="smallint" name="severity" padding="5" unsigned="true" nullable="false" identity="false"
      default="0" comment="Problem type"/>
    <constraint xsi:type="primary" referenceId="PRIMARY">
      <column name="notification_id"/>
    </constraint>
    <index referenceId="ADMINNOTIFICATION_INBOX_SEVERITY" indexType="btree">
      <column name="severity"/>
    </index>
  </table>
</schema>
```

DB_SCHEMA.XML

```
<?xml version="1.0"?>
<schema xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="urn:magento:framework:Setup/Declaration/Schema/etc/schema.xsd">
  <table name="adminnotification_inbox" resource="default" engine="innodb" comment="Adminnotification Inbox">
    <column xsi:type="int" name="notification_id" padding="10" unsigned="true" nullable="false" identity="true"
      comment="Notification id"/>
    <column xsi:type="smallint" name="severity" padding="5" unsigned="true" nullable="false" identity="false"
      default="0" comment="Problem type"/>
    <constraint xsi:type="primary" referenceId="PRIMARY">
      <column name="notification_id"/>
    </constraint>
    <index referenceId="ADMINNOTIFICATION_INBOX_SEVERITY" indexType="btree">
      <column name="severity"/>
    </index>
  </table>
</schema>
```

DB_SCHEMA.XML

```
<?xml version="1.0"?>
<schema xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="urn:magento:framework:Setup/Declaration/Schema/etc/schema.xsd">
  <table name="adminnotification_inbox" resource="default" engine="innodb" comment="Adminnotification Inbox">
    <column xsi:type="int" name="notification_id" padding="10" unsigned="true" nullable="false" identity="true"
      comment="Notification id"/>
    <column xsi:type="smallint" name="severity" padding="5" unsigned="true" nullable="false" identity="false"
      default="0" comment="Problem type"/>
    <constraint xsi:type="primary" referenceId="PRIMARY">
      <column name="notification_id"/>
    </constraint>
    <index referenceId="ADMINNOTIFICATION_INBOX_SEVERITY" indexType="btree">
      <column name="severity"/>
    </index>
  </table>
</schema>
```


There is a tool Schema Listener Tool to migrate
setup scripts to `db_schema.xml`

db_schema_whitelist.json

DI.XXML

DI.XML – PREFERENCES

```
<?xml version="1.0"?>
<config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:noNamespaceSchemaLocation="urn:magento:framework:ObjectManager/etc/config.xsd">
    <preference for="Magento\Catalog\Api\Data\ProductInterface" type="Magento\Catalog\Model\Product" />
</config>
```

defines which **classes** should be instantiated
if a specific **interface** is injected¹

¹see constructor injection!

DI.XML - VIRTUAL TYPES

```
<?xml version="1.0"?>
<config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:noNamespaceSchemaLocation="urn:magento:framework:ObjectManager/etc/config.xsd">
    <VirtualType name="Magento\Catalog\Model\Session\Storage" type="Magento\Framework\Session\Storage">
        <arguments>
            <argument name="namespace" xsi:type="string">catalog</argument>
        </arguments>
    </VirtualType>
</config>
```

defines a **fake**² class (name) using another class (type) with different argument values

²there's no php file! you can't use this class *anywhere else* but in **constructor injection**!

DI.XML - CONSTRUCTOR ARGUMENTS

```
<?xml version="1.0"?>
<config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:noNamespaceSchemaLocation="urn:magento:framework:ObjectManager/etc/config.xsd">
    <type name="Magento\Catalog\Model\Config\Source\ListPerPage">
        <arguments>
            <argument name="options" xsi:type="string">5,10,15,20,25</argument>
        </arguments>
    </type>
</config>
```

defines default __construct argument values

DI.XML - SHARED

```
<config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="urn:magento:framework:ObjectManager/etc/config.xsd">
    <type name="Magento\CatalogGraphQL\Model\ProductInterfaceTypeResolverComposite"
shared="false">
    </type>
</config>
```

PLUGINS

extend or replace existing methods of classes

- before
- after
- around

can run in a specific sortOrder

PLUGIN EXCEPTION

FINAL METHODS AND CLASSES

PLUGIN EXCEPTION NON-PUBLIC METHODS

PLUGIN EXCEPTION

_construct

PLUGIN EXCEPTION VIRTUAL TYPES

PLUGIN EXCEPTION

ANYTHING BEFORE \Magento\Framework\Interception

PLUGIN CONFIGURATION DI.XML

```
<?xml version="1.0"?>
<config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:noNamespaceSchemaLocation="urn:magento:framework:ObjectManager/etc/config.xsd">
    <type name="Magento\Theme\Block\Html\Topmenu">
        <plugin name="catalogTopmenu" type="Magento\Catalog\Plugin\Block\Topmenu" sortOrder="10" />
    </type>
</config>
```

defines a class that wraps another class

PLUGIN BEFORE

- create the class from `<plugin .. type="" />`

```
public function before{OrigMethod}($subject, $arg1, $arg2 ...){  
    return;  
    //or  
    return [$arg1, $arg2, ...]  
}
```

PLUGIN BEFORE

If it **returns null**, Magento passes *original arguments* to next plugin,
otherwise **return an array** with *as many items as function arguments*.

PLUGIN AFTER

- create the class from `<plugin .. type="" />`

```
public function after{OrigMethod}($subject, $result, $arg1, $arg2 ...){  
    return $result;  
}
```

PLUGIN AFTER

Only **returns the result**,
but has *access to original arguments*.

PLUGIN AROUND

```
public function around{OrigMethod}($subject, $proceed, $arg1, $arg2 ...){  
    $this→doThisBefore();  
    $result = $proceed();  
    $this→doThisAfter();  
    return $result;  
}
```

NOTE: Using around plugins is *very* expensive

PLUGIN AROUND

It must call `$proceed()` and return its result

If `$proceed()` **isn't called**, no other of this method's plugins *and* the original method is processed.

EVENTS

EVENTS CONFIGURATION

events.xml in ./etc and area scoped as well

```
<?xml version="1.0"?>
<config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:noNamespaceSchemaLocation="urn:magento:framework:Event/etc/events.xsd">
    <event name="magento_catalog_api_data_productinterface_save_before">
        <observer name="legacy_model_save"
                instance="Buro71\Training\Observer\BeforeEntitySave" />
    </event>
</config>
```

EVENTS CREATION

- create the class

```
from <observer ... instance="" />
```

```
namespace Buro71a\Training\Observer;
```

```
use Magento\Framework\Event\ObserverInterface;
```

```
class BeforeEntitySave implements ObserverInterface{}
```

REQUESTFLOW

REQUESTFLOW

./pub/index.php 

\Magento\Framework\App\Http 

\Magento\Framework\App\FrontController 

\Magento\Framework\App Router\Base 

\Vendor\Module\Controller\Action\Path\Actionname

ROUTER MATCHING

1. `\Magento\Robots\Controller\Router`
2. `\Magento\UrlRewrite\Controller\Router`
3. `\Magento\Framework\App\Router\Base`
4. `\Magento\Cms\Controller\Router`
5. `\Magento\Framework\App\Router\DefaultRouter`

BASE ROUTER

default URL Scheme matching

`https://shop.com/frontname/actionpath/action`

URL MATCHING

FRONTNAME

defined in

`./etc/frontend/routes.xml`

or

`./etc/adminhtml/routes.xml`

ROUTES.XML

`https://shop.com/frontname/actionpath/action`

```
<?xml version="1.0"?>
<config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="urn:magento:framework:App/etc/routes.xsd">
  <router id="standard">
    <route id="catalog" frontName="catalog">
      <module name="Magento_Catalog" />
    </route>
  </router>
</config>
```

route id is being used in full action name

ADMIN ROUTES

route id and **frontName**
must match in *adminhtml* area

ACTION PATH

[https://shop.com/frontname/*actionpath*/action](https://shop.com/frontname/actionpath/action)

just a directory structure in your module

app/code/

Vendor/

Module/

Controller/

Actionpath/

ACTION

`https://shop.com/frontname/actionpath/action`

a file (php class) in your *action path* directory

`Action.php`

ACTION CLASSES

have to **implement**

`\Magento\Framework\App\ActionInterface`

or **extend** (*easier and faster*)

`\Magento\Framework\App\Action\Action`

They should return an *result object!*

RESULTOBJECTS

have to **implement**

`\Magento\Framework\Controller\ResultInterface`

a bunch are already there, use them!

PAGERESULT

\Magento\Framework\View\Result\Page

most common object,

triggers *layout xml* rendering

into html

```
use Magento\Framework\App\Action\Context;
use Magento\Framework\View\Result\PageFactory;

class Page extends \Magento\Framework\App\Action\Action
{
    private $pageFactory;

    public function __construct(Context $context, PageFactory $pageFactory)
    {
        parent::__construct($context);
        $this->pageFactory = $pageFactory;
    }

    public function execute()
    {
        return $this->pageFactory->create();
    }
}
```

JSONRESULT

`\Magento\Framework\Controller\Result\Json`

returns a *json encoded* representation
of an *array with data*

Already sets the **content-type** header for you!

```
use Magento\Framework\App\Action\Context;
use Magento\Framework\Controller\Result\JsonFactory;

class Json extends \Magento\Framework\App\Action\Action
{
    private $jsonFactory;

    public function __construct(Context $context, JsonFactory $jsonFactory)
    {
        parent::__construct($context);
        $this->jsonFactory = $jsonFactory;
    }

    public function execute()
    {
        return $this->jsonFactory->create()->setData(['hello' => 'world']);
    }
}
```

RAWRESULT

`\Magento\Framework\Controller\Result\Raw`

returns the data as it's been set.

Doesn't set the **content-type** header!


```

use Magento\Framework\App\Action\Context;
use Magento\Framework\Controller\Result\RawFactory;

class Raw extends \Magento\Framework\App\Action\Action
{
    /**
     * @var RawFactory
     */
    private $rawFactory;

    public function __construct(Context $context, RawFactory $rawFactory)
    {
        parent::__construct($context);
        $this->rawFactory = $rawFactory;
    }

    public function execute()
    {
        return $this->rawFactory->create()->setContents('This is the content')
            ->setHeader('content-type', 'text/plain', true);
    }
}

```

REDIRECTRESULT

`\Magento\Framework\Controller\Result\Redirect`

Creates an external redirect,
which the browser follows and
requests a new url.


```
use Magento\Framework\App\Action\Context;
use Magento\Framework\Controller\Result\RedirectFactory;

class Redirect extends \Magento\Framework\App\Action\Action
{
    private $redirectResultFactory;

    public function __construct(Context $context, RedirectFactory $redirectResultFactory)
    {
        parent::__construct($context);
        $this->redirectResultFactory = $redirectResultFactory;
    }

    public function execute()
    {
        return $this->redirectResultFactory->create()->setPath('*/*/*raw');
        // alternative for external page:
        // return $this->redirectResultFactory->create()->setUrl('http://www.google.com/');
    }
}
```

FORWARDRESULT

`\Magento\Framework\Controller\Result\Forward`

Internally calls the `execute` method of another action class and doesn't trigger a new request from the browser.

The URL stays the same.

```
use Magento\Framework\App\Action\Context;
use Magento\Framework\Controller\Result\FowardFactory;

class Forward extends \Magento\Framework\App\Action\Action
{
    private $forwardFactory;

    public function __construct(Context $context, FowardFactory $forwardFactory)
    {
        parent::__construct($context);
        $this->forwardFactory = $forwardFactory;
    }

    public function execute()
    {
        return $this->forwardFactory->create()->forward('json');
    }
}
```

SYSTEMCONFIG IN ADMIN

ADMINHTML/SYSTEM.XML

```
<?xml version="1.0"?>
<config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="urn:magento:module:Magento_Config:etc/system_file.xsd">
    <system>
        <tab id="catalog" translate="label" sortOrder="200">
            <label>Catalog</label>
        </tab>
    </system>
</config>
```

ADMINHTML/SYSTEM.XML

```
<?xml version="1.0"?>
<config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="urn:magento:module:Magento_Config:etc/system_file.xsd">
    <system>
        <section id="catalog" translate="label" type="text" sortOrder="40" showInDefault="1" showInWebsite="1" showInStore="1">
            <class>separator-top</class>
            <label>Catalog</label>
            <tab>catalog</tab>
            <resource>Magento_Catalog::config_catalog</resource>
            <group id="frontend" translate="label" type="text" sortOrder="100" showInDefault="1" showInWebsite="1" showInStore="1">
                <label>Storefront</label>
                <field
id="flat_catalog_category" translate="label" type="select"
sortOrder="100" showInDefault="1" showInWebsite="0" showInStore="0" canRestore="1">
                    <label>Use Flat Catalog Category</label>
                    <backend_model>Magento\Catalog\Model\Indexer\Category\Flat\System\Config\Mode</backend_model>
                    <source_model>Magento\Config\Model\Config\Source\Yesno</source_model>
[closing tags]
```


CONFIG.XML

```
<?xml version="1.0"?>
<config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="urn:magento:module:Magento_Store:etc/config.xsd">
    <default>
        <system>
            <adminnotification>
                <feed_url>notifications.magentocommerce.com/magento2/community/notifications.rss</feed_url>
                <popup_url>widgets.magentocommerce.com/notificationPopup</popup_url>
                <severity_icons_url>widgets.magentocommerce.com/%s/%s.gif</severity_icons_url>
            </adminnotification>
        </system>
    </default>
</config>
```

WIDGETS

WIDGETS - WIDGET CLASS

```
<?xml version="1.0" ?>
<widgets xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="urn:magento:module:Magento_Widget:etc/widget.xsd">
    <widget class="Company\CustomWidget\Block\Widget\Headline" id="company_customwidget_headline">
        <label>Headline</label>
        <description>Add a headline</description>
        <parameters>
            <parameter name="headline" sort_order="10" visible="true" xsi:type="text">
                <label>Title</label>
            </parameter>
        </parameters>
    </widget>
</widgets>
```

WIDGETS - WIDGET CLASS

```
<?xml version="1.0" ?>
<widgets xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="urn:magento:module:Magento_Widget:etc/widget.xsd">
    <widget class="Company\CustomWidget\Block\Widget\Headline" id="company_customwidget_headline">
        <label>Headline</label>
        <description>Add a headline</description>
        <parameters>
            <parameter name="headline" sort_order="10" visible="true" xsi:type="text">
                <label>Title</label>
            </parameter>
        </parameters>
    </widget>
</widgets>
```

WIDGETS - WIDGET CLASS

```
<?xml version="1.0" ?>
<widgets xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="urn:magento:module:Magento_Widget:etc/widget.xsd">
    <widget class="Company\CustomWidget\Block\Widget\Headline" id="company_customwidget_headline">
        <label>Headline</label>
        <description>Add a headline</description>
        <parameters>
            <parameter name="headline" sort_order="10" visible="true" xsi:type="text">
                <label>Title</label>
            </parameter>
        </parameters>
    </widget>
</widgets>
```

WIDGETS - WIDGET CLASS

```
<?php
namespace Company\CustomWidget\Block\Widget;
use Magento\Framework\View\Element\Template;
use Magento\Widget\Block\BlockInterface;
class HeadLineAndText extends Template implements BlockInterface
{
    protected $_template = "widget/headLineAndText.phtml";
}
```

WIDGETS - WIDGET TEMPLATE

Company/CustomWidget/view/frontend/
templates/widget/headline.phtml

```
<?php if($block→getData('headline')): ?>
    <h2 class="title"><?php echo $block→getData('headline'); ?></h2>
<?php endif; ?>
```

<http://theslidesareonline.de/>

К Т Н Х В А И

