Magento 2 Useful Code Snippets and Tips Cheat Sheet

by Bhavi via cheatography.com/198595/cs/42120/

Useful Tips for becoming Quality magento developer

Use dependency injection:	Inject dependencies into your classes for better testability and maintainability
Follow Magento's coding standards:	Adhere to coding conventions for consistency and readability
Test your code thoroughly:	Write unit and integration tests to ensure code quality and prevent regressions
Use Magento's built-in features:	Take advantage of Magento's features like layout XML, UI components, and API endpoints before creating custom extensions
Utilize the Magento community:	Engage with the active Magento community for help, support, and resources

Getting the current customer session

```
$objectManager = \Magento\Framework\App\ObjectManager::getInstance();
$custo mer Session = $objec tMa nag er- >ge t( \Mag ent o\C ust ome r\M ode l\S ess ion ::c lass);
$custo merId = $custo mer Ses sio n-> get Cus tom erId();
```

Retrieving a product by ID

```
$objectManager = \Magento\Framework\App\ObjectManager::getInstance();
$produ ctR epo sitory = $objec tMa nag er- >ge t( \Mag ent o\C ata log \Ap i\P rod uct Rep osi tor yIn -
ter fac e:: class);
$product = $produ ctR epo sit ory ->g etB yId ($p rod uctId);
```

Sending an email programmatically

```
$objectManager = \Magento\Framework\App\ObjectManager::getInstance();
$trans por tBu ilder = $objec tMa nag er- >ge t( \Mag ent o\F ram ewo rk \Mai l\T emp lat e\T ran spo rtB -
uil der ::c lass);
$transport = $trans por tBu ild er- >se tTe mpl ate Ide nti fie r(' my_ ema il_ tem plate')
        ->s etT emp lat eOp tio ns( ['area' => 'front end', 'store' => $storeId])
        ->s etT emp lat eVa rs( ['data' => $myData])
        ->s etF rom ('s end er@ exa mpl e.com')
        ->a ddT o(' rec ipi ent @ex amp le.c om')
        ->g etT ran spo rt();
$trans por t-> sen dMe ssa ge();
```

Adding a custom layout block

```
<referenceContainer name="content">
        <block class= " Mag ent o\F ram ewo rk \Vie w\E lem ent \Te mpl ate " name="m y.c ust om.b lo ck"
templa te= " Mag ent o_T hem e:: my- cus tom -bl ock.ph tml " />
</r efe ren ceC ont ain er>
```

Creating a custom UI component

```
define([
        'ui Com ponent'
], function (Compo nent) {
       'use strict';
```

Creating a custom UI component (cont)

```
>    return Component.extend({
       // Your component logic here
   });
});
```

Using a custom module's configuration

```
$scopeConfig = $objectManager->get(\Magento\Framework\App\Config\ScopeConfigInterface::class);
$myMod ule Con fig Value = $scope Con fig ->g etV alu e(' my/ mod ule /co nfi g/p ath', \Magen to \Sto -
re \Mod el \Sco peI nte rfa ce: :SC OPE _ST ORE);
```

Logging messages for debugging

```
$logger = $objectManager->get(\Psr\Log\LoggerInterface::class);
$logge r-> deb ug('My debug message');
$logge r-> inf o('My info message');
$logge r-> err or('My error message');
```

Creating a custom admin grid

```
<listing xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:no Nam esp ace Sch ema Loc ati on= " urn :ma gen to: mod ule :Ma gen to_ Ui: etc /ui _co nfi gur ati -
on.x sd ">
</l ist ing>
```

Creating a custom API endpoint

```
// In your module's webapi.xml
<route url="/V 1/m y-e ndp oin t" method ="GE T">
        <se rvice class= " My \Mod ule \Ap i\M yEn dpo int Int erf ace " method ="ge tLi st"/>
        <re sou rce s>
                <re source ref="an ony mou s"/>
        </r eso urc es>
</r out e>
```