



Demonstrate how to use dependency injection

Magento 2 uses constructor injection, where all injected objects or factories are provided when the object is constructed.

Magento loads di.xml files and merges them all together from the following stages:

- Initial (app/etc/di.xml)
- Global ({moduleDir}/etc/di.xml)
- Area-specific ({moduleDir}/etc/{area}/di.xml)

Object manager

Under the hood, the [ObjectManager](#) will use PHP reflection features to look at a class's `__construct` type hints/parameters, automatically instantiate the object for us, and then pass it into the constructor as an argument.

[AbstractFactory](#) > [\Magento\Framework\ObjectManager\FactoryInterface](#) and their implementations use to resolve arguments and create new objects.

By default, all objects created via automatic constructor dependency injection are singleton objects, because they created via `ObjectManager::get()` method.

```
if ($isShared) {
    $argument = $this->objectManager->get($argumentType);
} else {
    $argument = $this->objectManager->create($argumentType);
}
```

[\Magento\Framework\ObjectManager\Factory::resolveArgument\(\)](#)

Arguments

```
<argument xsi:type="object">{typeName}</argument>
<argument xsi:type="object" shared="{shared}">{typeName}
</argument>

<argument xsi:type="string">{strValue}</argument>
<argument xsi:type="string" translate="true">{strValue}</argument>

<argument xsi:type="boolean">{boolValue}</argument>
<argument xsi:type="number">{numericValue}</argument>
<argument xsi:type="init_parameter">{Constant::NAME}</argument>
<argument xsi:type="const">{Constant::NAME}</argument>
<argument xsi:type="null"/>

<argument xsi:type="array">
    <item name="someKey" xsi:type="<type>">someVal</item>
</argument>
```

Links

- [Dependency injection](#)
- [The di.xml file](#)
- [ObjectManager](#)
- [Proxies](#)
- [Factories](#)
- [Alan Storm, The Magento 2 Object System](#)

- [Alan Kent, Magento 2 Dependency injection](#)