**Quick Overview Of Object Manager In Magento 2**

## **Mechanism**

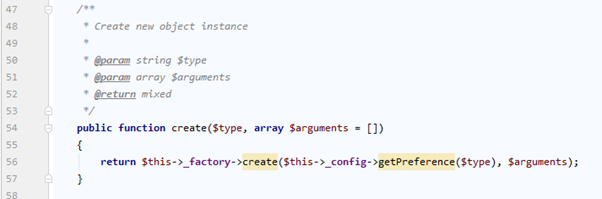
The Object Manager is mainly responsible for the instantiation and configuration of an Object through two main methods: GET, CREATE.

* The GET method returns a singleton object (an instance of the class shared among components when running Magento).
* **vendor/magento/framework/ObjectManager/ObjectManager.php::get()**



* The CREATE method returns an entirely new object (a new class instance).

**vendor/magento/framework/ObjectManager/ObjectManager.php::create()**



Therefore, if you call the GET method from 2 places, the same result will be generated. Otherwise, you will get a new object using the CREATE method.

## **Object Manager configuration**

The di.xml file configures the object manager and tells it how to handle dependency injection

This file specifies the preferred implementation class the object manager creates for the interface declared in a constructor class. The file also determines whether the object manager will create an object for every request or the object is a singleton.

## **Object Manager’s goal**

* Use the object manager to instantiate and insert the declaration class in the constructor.
* public function \_\_construct(
* \Magento\Framework\ObjectManagerInterface $objectManager
* ) {
* $this->\_objectManager = $objectManager;
* }
* Implement the singleton pattern (learn more at https://en.wikipedia.org/wiki/Singleton\_pattern)
* Manage dependencies
* Automatically initialize parameters

According to Magento’s core group, you should not use Object Manager in modules because it makes the class lose dependency injection.

## **You can use Object Manager in the following exceptions:**

* Use Object manager in static magic methods such as \_\_wakeup (), \_\_sleep (), …
* Use Object manager to maintain backward compatibility for a constructor
* The Object manager can depend on classes used to create objects like factories or proxies.

# <https://magefan.com/blog/magento-2-object-manager>

# Singleton pattern

 the **singleton pattern** is a [software design pattern](https://en.wikipedia.org/wiki/Software_design_pattern) that restricts the [instantiation](https://en.wikipedia.org/wiki/Instantiation_(computer_science)) of a [class](https://en.wikipedia.org/wiki/Class_(computer_programming)) to one "single" instance. This is useful when exactly one object is needed to coordinate actions across the system.

# Dependency Injection in Magento 2

In Magento 2 Dependency Injection is one of the most useful design pattern.

<https://www.sparsh-technologies.com/blog/dependency-injection-in-magento-2>

### Factory and proxies

Factory and Proxies are some kinds of exceptions for the direct call of Object manager because they need Object manager to generate new objects. As an example, you can overview any kind of DTO factory.

<https://www.atwix.com/magento/design-patterns-in-magento-2-object-manager/>

<https://www.sparsh-technologies.com/blog/dependency-injection-in-magento-2>

## Magento 2 Clean Cache vs. Flush:

You can run both the commands to remove played out items from your website: clean and flush Magento 2 cache. However, it is important that you learn the difference:

### Magento 2 Clean cache:

Since it deletes all enabled cache types, disabled cache types are not cleaned from Magento 2

### Flush Cache

Flushing a cache type purges the cache storage, which might affect other processes applications that are using the same storage. e.g. other websites. Flush cache types if you’ve already tried cleaning the cache and you’re still having issues that you cannot isolate.

* **By Default create() and get() method is responsible for generate new object and retrieve obejct in magento 2.**

<https://magento.stackexchange.com/questions/176018/how-magento2-creates-object-of-factory-classes>

**Object Manager:**

object manager that is responsible for the creation of all the objects.Object Manager resides here  
“Magento\Framework\ObjectManager\ObjectManager”  
it has three methods two of them are responsible for the creation of the objects get() and create(), get is like Mage::getSingleton(“ClassName”) and create is like Mage::getModel(“ClassName”) , **get creates sharable object and create will create new objects** .

<https://webkul.com/blog/magento2-code-generation-and-factory-design-pattern/>

* you must use Factory classes to inject non-injectable objects.

<https://symphisys.com/blog-patterns>

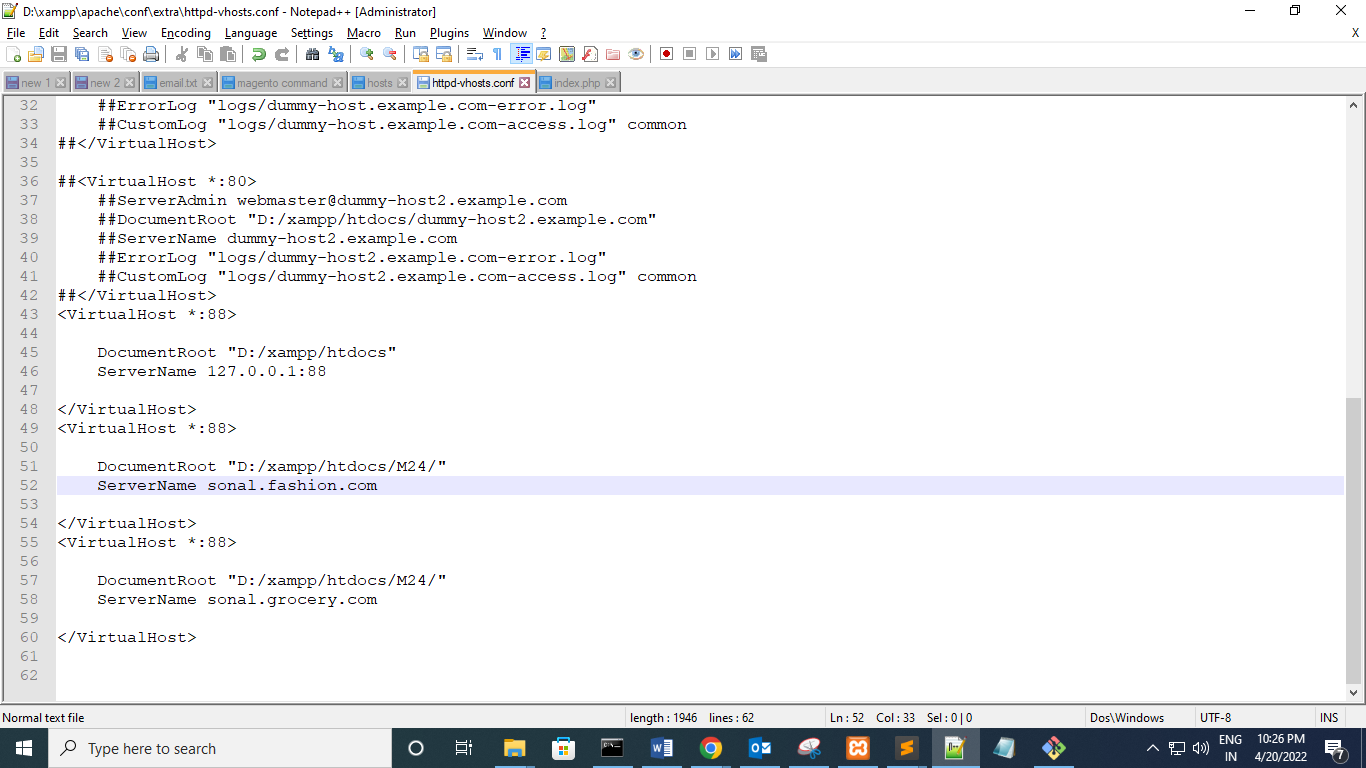
**create multiple website and multiple store:**

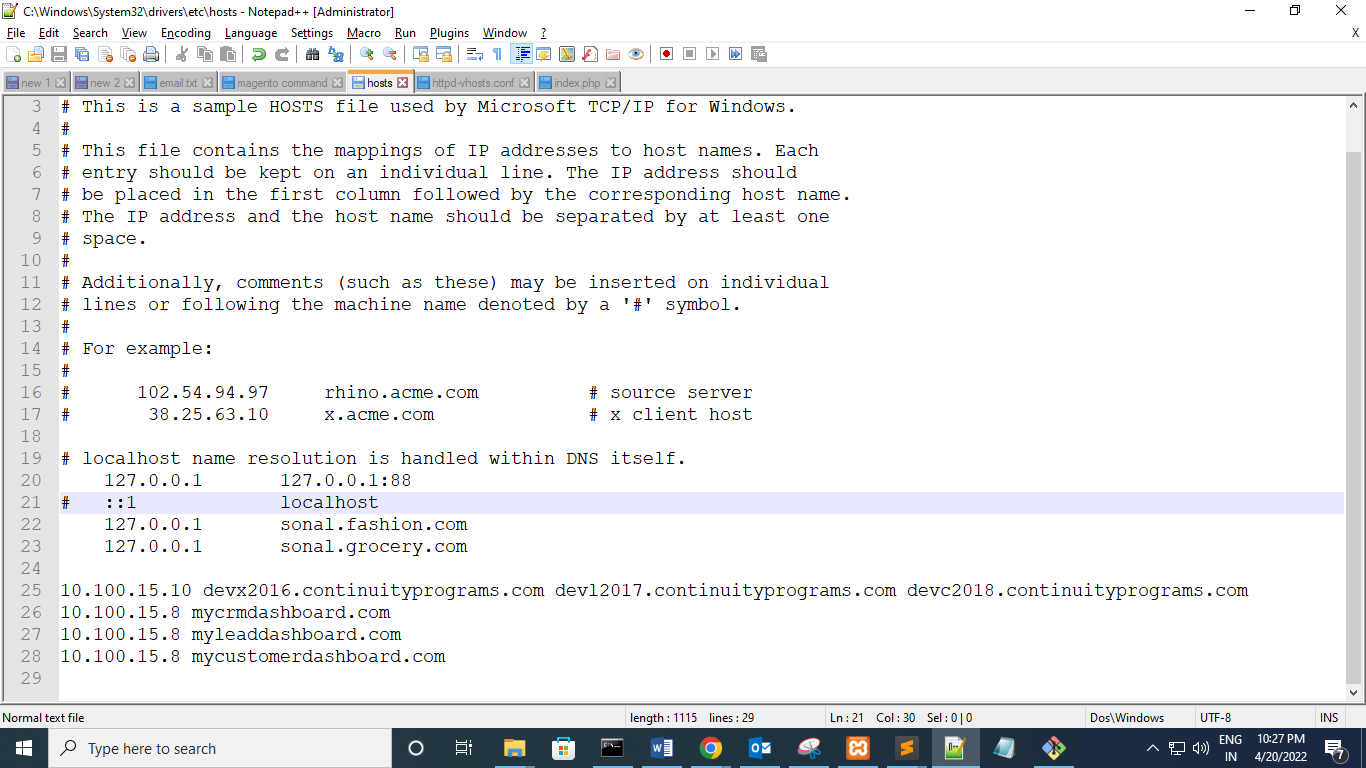
[**https://www.thecoachsmb.com/multiple-website-and-multistore-in-magento-2-on-localhost-xampp-windows/**](https://www.thecoachsmb.com/multiple-website-and-multistore-in-magento-2-on-localhost-xampp-windows/)

**Create virtual Host on local**

**step1:>** D:\xampp\apache\conf\extra\httpd-vhosts.conf

**step2:>** C:\Windows\System32\drivers\etc\hosts





* When we delete default store view

1. Edit store table
2. Store\_group
3. Store\_website