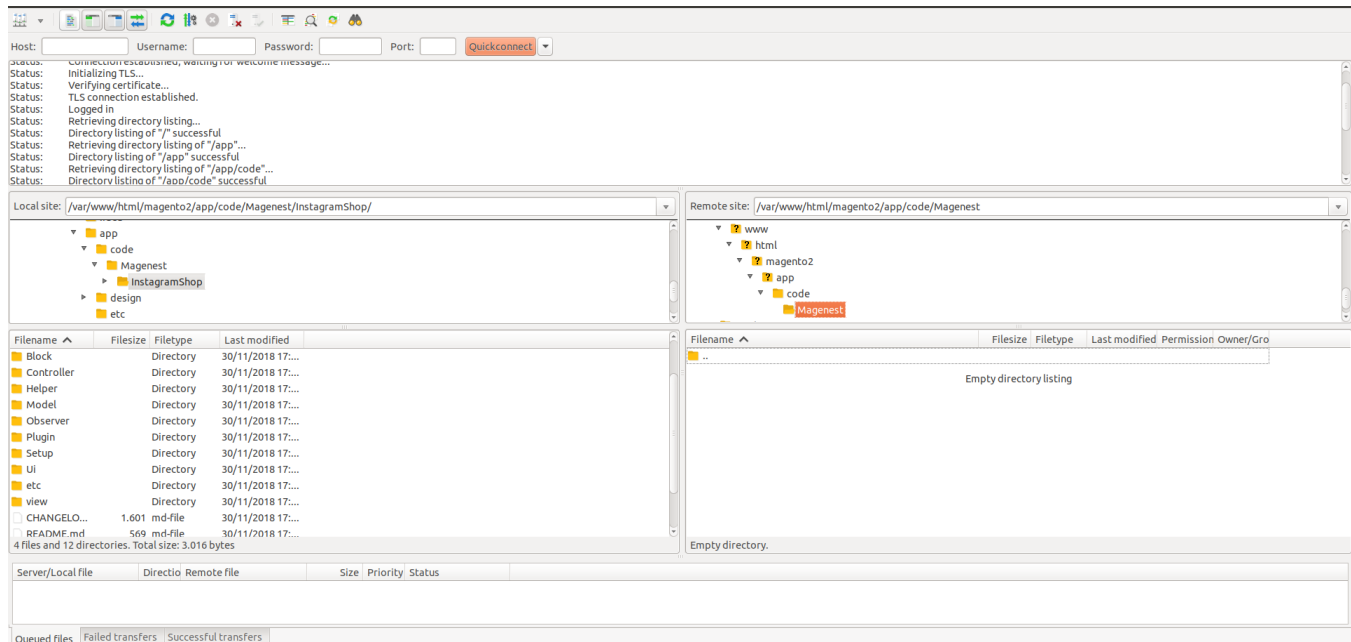


1. Instagram Integration Installation Guides

Here we will guide you through the installation process of our extension. First off, open FileZilla, log in to your host as follows:



In the above example we switched the local site to the extension directory, and the remote site to the code directory. Now on the remote site window, inside app/code directory, create a folder named **Magenest** and go inside it. Now on your local site window, right click on Wrapper extension and click **Upload**. The extension will be uploaded onto your host in a few minutes. Correct folder structure would be **app/code/Magenest/[Extension Name]**

The second method will require you to compress the extension into .zip or .tar.gz format, then execute the following command in your Command Line /Terminal:

```
scp <path to your extension> <ssh server>:<path to Magenest directory>
```

Then log into your ssh server, go to Magenest directory and execute the following command to decompress the module:

```
unzip <.zip compressed extension>
```

Or

```
tar -xzf <.tar.gz compressed extension>
```

Now go to your **Magento root directory**, say it's named **magento2**, and execute the following commands:

```
sudo chmod -R 755 .
php bin/magento setup:upgrade
php bin/magento setup:di:compile
php bin/magento setup:static-content:deploy
sudo chmod -R 777 var pub
```

After running all these, your Command line/Terminal window might look like this:

```

Module 'Magento_BundleSampleData':
Module 'Magento_Rss':
Module 'Magento_DownloadableSampleData':
Module 'Magento_Captcha':
Module 'Magento_OfflineShippingSampleData':
Module 'Magento_ConfigurableSampleData':
Module 'Magento_SalesSampleData':
Module 'Magento_ThemeSampleData':
Module 'Magento_ProductLinksSampleData':
Module 'Magento_ReviewSampleData':
Module 'Magento_SendFriend':
Module 'Magento_Ui':
Module 'Magento_Sitemap':
Module 'Magento_Authorizenet':
Module 'Magento_Swagger':
Module 'Magento_Swatches':
Module 'Magento_SwatchesSampleData':
Module 'Magento_CatalogRuleSampleData':
Module 'Magento_TaxImportExport':
Module 'Magento_TaxSampleData':
Module 'Magento_CatalogRuleConfigurable':
Module 'Magento_CmsSampleData':
Module 'Magento_Translation':
Module 'Magento_Shipping':
Module 'Magento_Ups':
Module 'Magento_UrlRewrite':
Module 'Magento_GroupedProductSampleData':
Module 'Magento_Usps':
Module 'Magento_Variable':
Module 'Magento_Version':
Module 'Magento_Webapi':
Module 'Magento_SalesRuleSampleData':
Module 'Magento_CatalogWidget':
Module 'Magento_WidgetSampleData':
Module 'Magento_Wishlist':
Module 'Magento_WishlistSampleData':
Please re-run Magento compile command
[root@centos-2gb-sgp1-01 magento2]# sudo chmod -R 777 var pub
[root@centos-2gb-sgp1-01 magento2]#

```

Finally, you need to flush the cache by either type this into the Command Line/Terminal:

```
php bin/magento cache:flush
```

Or go to **System > Cache Management** and click **Flush Magento Cache**

The screenshot shows the Magento Admin interface. On the left, the 'System' menu is open, and 'Cache Management' is selected. In the top right corner, the 'Flush Magento Cache' button is highlighted with a red box. The main content area displays a table of cache types. All cache types listed are currently 'DISABLED'.

	Description	Tags	Status
☐	Various XML configurations that were collected across modules and merged.	CONFIG	DISABLED
☐	Layout building instructions.	LAYOUT_GENERAL_CACHE_TAG	DISABLED
☐	Page blocks HTML.	BLOCK_HTML	DISABLED
☐	Collection data files.	COLLECTION_DATA	DISABLED
☐	API interfaces reflection data.	REFLECTION	DISABLED
☐	Results of DDL queries, such as describing tables or indexes.	DB_DDL	DISABLED
☐	Entity types declaration cache.	EAV	DISABLED
☐	Integration configuration file.	INTEGRATION	DISABLED
☐	Integrations API Configuration	INTEGRATION_API_CONFIG	DISABLED
☐	Page Cache	FPC	DISABLED
☐	Translations	TRANSLATE	DISABLED

Now you can go to your store and check if the extension is working as expected.