

Files needed inorder to create a custom module

module_name.info.yml file

Meta details inside info.yml file are

```
name: Learn DB
type: module
description: Demonstrates DB Insertion, updation, deletion module for drupal 8
package: Learn
core: 8.x
```

Only module_name.info.yml is enough to create a custom module. We can go and enable the module

Other files needed are

```
module_name.module
Module_name.routing.yml
Moudle_name.services.yml
```

Form

src/Form/FormName.php file

Controller

src/Controller/ControllerName.php file

Services and Dependency Injection in Drupal 8

Very Imporant link

<https://www.valuebound.com/resources/blog/how-to-define-your-own-services-drupal-8>

Service is a PHP class with some code that provides a single specific functionality throughout the application. So you can easily access each service and use its functionality wherever you need it. Because of that it is easy to test and configure in your application. This is called service-oriented architecture which is not unique to Symfony or even PHP.

The Services and Dependency Injection Container concepts have been adopted by Drupal from the Symfony framework. Accessing the database, sending email, or translating user interface are examples for the services in Drupal 8.

Please ref to this link for example on services :

<https://www.valuebound.com/resources/blog/how-to-define-your-own-services-drupal-8>

Dependency Injection

As of Drupal 8, *dependency injection* is the preferred method for accessing and using services and should be used whenever possible. Rather than calling out to the global services container, services are instead passed as arguments to a constructor or injected via setter methods. This allows for a loose coupling of components, which makes things easier to test, and easier to replace with alternative implementations.

<https://drupalize.me/topic/dependency-injection#:~:text=As%20of%20Drupal%208%2C%20dependency,or%20injected%20via%20setter%20methods.>

Event Subscribers

Event Subscribers in Drupal 8 are similar to hooks in Drupal 7. They help you hook into an event happening in a Drupal site and execute the appropriate code. So if you are a Drupal developer and would like to execute code when an event occurs it will be worthwhile to know about Event Subscribers