

**Start: M2 Module Using Repository.**  
Repositories use Service Contract Design Pattern  
(Our module namespace is Vendor\_Test)

**Create InstallSchema script**  
(Vendor/Test/Setup/InstallSchema.php)

**Create Model, ResourceModel, Collection, Grid Collection**  
(Vendor/Test/Model/Custom,  
Vendor/Test/Model/ResourceModel/Custom.php  
Vendor/Test/Model/ResourceModel/Custom/Collection.php  
Vendor/Test/Model/ResourceModel/Grid/Custom/Collection.php)

**Create Data Interface, Data Model**  
(Vendor/Test/Api/Data/CustomInterface.php  
Vendor/Test/Api/Model/Data/Custom.php)

**Create Repository Interface. This is our Service Contract.**  
(Vendor/Test/Api/CustomRepositoryInterface.php)

**Create SearchResultInterface**  
(Vendor/Test/Api/Data/CustomSearchResultInterface.php  
Why? In M2 we need to define one interface for each data entity for type hinting purposes.  
That is, the `getItems()` function in the `CustomSearchResultsInterface` which returns an array of `CustomInterface` data entities)

**Create CustomRepository (Concrete Repository)**  
(Vendor/Test/Model/CustomRepository.php  
Why? In M2 remember, each repository implements its own interface. Through constructor DI, we pass in the factory, resource model, collection factory, `SearchResultsInterfaceFactory` for our custom module. Regarding the CRUD method in this repository, due to `CustomRepositoryInterface`, we *must* pass in a parameter of `\Vendor\Test\Api\Data\CustomInterface`. Our module's `di.xml` has a preference to replace an interface of this type with an entity model. The entity model gets passed into the Resource Model and is saved.)

**Set preference in global di.xml**  
(Vendor/Test/etc/di.xml

```
<preference for="Vendor\Test\Api\Data\CustomInterface"
            type="Vendor\Test\Model\Data\Custom" />
<preference for="Vendor\Test\Api\CustomRepositoryInterface"
            type="Vendor\Test\Model\CustomRepository" />
<preference for="Vendor\Test\Api\Data\CustomSearchResultsInterface"
            type="Magento\Framework\Api\SearchResults" />
```