In Magento 2 we can instantiate a model using **Object Manager**. Yes, you heard it correctly, ObjectManager is a replacement for getModel and getSingleton Method.

There are two approaches you can invoke ObjectManager in Magento 2 to instantiate a model or to get the model collection.

**Approach 1: Using ObjectManager Directly anywhere**

Let’s say we want to get product collection in our Magento 2 custom module phtml file. You can get this using below code.

|  |
| --- |
| <?php   $objectManager = \Magento\Framework\App\ObjectManager::getInstance();   $productCollection = $objectManager->create('Magento\Catalog\Model\ResourceModel\Product\CollectionFactory');   $productCollection->addAttributeToSelect('\*');   $productCollection->load();   foreach ($productCollection as $product) {      echo "<pre>";      print\_r($product->getData());    } |

**Approach 2: Inject model into Constructor and instantiate**

Second and recommended approach of instantiating a model in Magento 2 is using a constructor. To get product collection in your custom phtml file, You need to create a block file inject the product model into the constructor of that block file.  
   
**Step 1: Create Block.php file**  
For example our block file Index.php is under app/code/Codextblog/Pcollection/Block/Index

|  |
| --- |
| <?php  namespace Codextblog\Pcollection\Block\Index;  class Index extends \Magento\Framework\View\Element\Template {    protected $\_productCollectionFactory;    public function \_\_construct(      \Magento\Backend\Block\Template\Context $context,      \Magento\Catalog\Model\ResourceModel\Product\CollectionFactory $productCollectionFactory,      array $data = []  )     {      $this->\_productCollectionFactory = $productCollectionFactory;      parent::\_\_construct($context, $data);     }     public function getProductCollection()      {          $collection = $this->\_productCollectionFactory->create();          $collection->addAttributeToSelect('\*');          return $collection;      }  } |

**Step 2: Call getProductCollection in your phtml file**  
For example our phtml file Index.phtml is under app/code/Codextblog/Pcollection/view/frontend/templates

|  |
| --- |
| <?php  $pCollection = $block->getProductCollection();  foreach ($pCollection as $product) {      echo "<pre>";      print\_r($product->getData());    } |

This is how you can get product collection in your custom phtml file.