

Import Statement

```
use
Magento\Catalog\Api\ProductRepositoryInterface;
use Magento\Catalog\Api\Data\ProductInterfaceFactory;
```

Constructor

```
/**
 * @var ProductRepositoryInterface
 */
protected $productRepositoryInterface;
/**
 * @var ProductInterfaceFactory
 */
protected $productInterfaceFactory;
/**
 * @param ProductRepositoryInterface $productRepositoryInterface
 *
 * public function __construct(
 *     ProductRepositoryInterface $productRepositoryInterface,
 *     ProductInterfaceFactory $productInterfaceFactory,
 * ) {
 *     $this->productRepositoryInterface = $productRepositoryInterface;
 *     $this->productInterfaceFactory = $productInterfaceFactory;
 * }
```

Product Types

```
\Magento\Catalog\Model\Product\Type::TYPE_SIMPLE
\Magento\Catalog\Model\Product\Type::TYPE_BUNDLE
\Magento\Catalog\Model\Product\Type::TYPE_VIRTUAL
\Magento\Catalog\Model\Product\Type::TYPE_DOWNLOADABLE
\Magento\Catalog\Model\Product\Type::TYPE_CONFIGURABLE
\Magento\Catalog\Model\Product\Type::TYPE_CODE
\Magento\Catalog\Model\Product\Type::TYPE_CODE
```

Get info about product by product SKU

```
$product = $this->productRepositoryInterface->get($sku);
```

```
/**
 * @param string $sku
```

Delete by SKU

```
$this->productRepositoryInterface->deleteById($sku);

/**
 * @param string $sku
 * @return bool Will return True if deleted
 * @throws \Magento\Framework\Exception\NoSuchEntityException
 * @throws \Magento\Framework\Exception\StateException
 */
```

Create New Product

```
$product = $this->productInterfaceFactory->create(
    $sku);

// other required attributes
$product = $this->productRepositoryInterface->save($product);
```

Update Existing Product

```
$product = $this->productRepositoryInterface->get('SKU');
$product->setAttribute('sku', 'SKU');
$product->save();

$updated = $this->productRepositoryInterface->save($product);
```

Advanced Search Filter Group

```

* @param bool $editMode
* @param int|null $storeId
* @param bool $forceReload
* @return \Magento\Catalog\Api\Data\ProductInterface
* @throws \Magento\Framework\Exception\NoSuchEntityException
*/

```

Get info about product by product id

```

$product = $this->productRepositoryInterface->
    getById($productId);

```

```

/**
 * @param int $productId
 * @param bool $editMode
 * @param int|null $storeId
 * @param bool $forceReload
 * @return \Magento\Catalog\Api\Data\ProductInterface
 * @throws \Magento\Framework\Exception\NoSuchEntityException
 */

```

```

/**
 * @var \Magento\Catalog\Api\Data\ProductInterface
 * repository
 */
private $productRepository;

/**
 * @var \Magento\Framework\Api\SearchCriteriaBuilder
 */
private $searchCriteriaBuilder;

/**
 * @var \Magento\Framework\Api\FilterBuilder
 */
private $filterBuilder;

/**
 * @var \Magento\Framework\Api\SearchCriteriaBuilder
 */
private $filterGroupBuilder;

/**
 * @var \Magento\Framework\Api\SortOrderFactory
 */
private $sortOrder;

/**
 * @param \Magento\Catalog\Api\Data\ProductInterface
 * repository
 * @param \Magento\Framework\Api\SearchCriteriaBuilder
 * $searchCriteriaBuilder
 * @param \Magento\Framework\Api\FilterBuilder
 * $filterBuilder
 * @param \Magento\Framework\Api\SearchCriteriaBuilder
 * $filterGroupBuilder
 * @param \Magento\Framework\Api\SortOrderFactory
 * $sortOrder
 */

```



By **DominicXigen**

cheatography.com/dominicxigen/

Published 2nd April, 2020.

Last updated 2nd May, 2020.

Page 1 of 6.

Sponsored by **ApolloPad.com**

Everyone has a novel in them. Finish

Yours!

<https://apollopadd.com>

Advanced Search Filter Group (cont)

```
> public function __construct(
    \Magento\Catalog\Api\ProductRepositoryInterface $productRepository,
    \Magento\Framework\Api\SearchCriteriaBuilder $searchCriteriaBuilder,
    \Magento\Framework\Api\FilterBuilder $filterBuilder,
    \Magento\Framework\Api\Search\FilterGroupBuilder $filterGroupBuilder,
    \Magento\Framework\Api\SortOrderFactory $sortOrderFactory
){
    $this->productRepository = $productRepository;
    $this->searchCriteriaBuilder = $searchCriteriaBuilder;
    $this->filterBuilder = $filterBuilder;
    $this->filterGroupBuilder = $filterGroupBuilder;
    $this->sortOrder = $sortOrder;
}
[...]
```

`$filter1 = $this->filterBuilder`

```
->setField("url")
->setValue("%magento.com")
->setConditionType("like")
->create();
```

`$filter2 = $this->filterBuilder`

```
->setField("store_id")
->setValue("1")
->setConditionType("eq")
->create();
```

`$filterGroup1 = $this->filterGroupBuilder`

```
->setFilters([$filter1, $filter2])
->create();
```

`$filter3 = $this->filterBuilder`

```
->setField("url_type")
->setValue(1)
->setConditionType("eq")
->create();
```

`$filterGroup2 = $this->filterGroupBuilder`

```
->setFilters([$filter3])
->create();
```

`$searchCriteria = $this->searchCriteriaBuilder->create();`

`$searchCriteria->setFilterGroups([$filterGroup1, $filterGroup2]);`

`$this->productRepository->getList($searchCriteria->getItems());`

The code above creates a Search Criteria with the Filters put together in the following way: (url like %magento.com OR store_id eq 1) AND (url_type eq 1)

Search Criteria

```
/**
 * @var ProductRepositoryInterface
 */
private $productRepositoryInterface;
/**
 * @var SearchCriteriaBuilder
 */
private $searchCriteriaBuilder;
/**
 * @param ProductRepositoryInterface $productRepositoryInterface
 * @param SearchCriteriaBuilder $searchCriteriaBuilder
 */
public function __construct(
    ProductRepositoryInterface $productRepositoryInterface,
    SearchCriteriaBuilder $searchCriteriaBuilder
) {
    $this->productRepository = $productRepositoryInterface;
    $this->searchCriteriaBuilder = $searchCriteriaBuilder;
}
/**
 * Get products by ids
 * @param array $productIds
 * @return ProductInterface[]
 */
public function getProducts(array $productIds): array
{
    $this->searchCriteriaBuilder->addFilter('entity_id', $productIds, 'in');
    $searchCriteria = $this->searchCriteriaBuilder->create();
    $products = $this->productRepositoryInterface->getList($searchCriteria->getItems());
    $count = $products->getTotalCount();
    return $products;
}
```

Useful constants

```
\Magento\Catalog\Api\Data\ProductInterface::SKU
\Magento\Catalog\Api\Data\ProductInterface::NAME
\Magento\Catalog\Api\Data\ProductInterface::PRICE
\Magento\Catalog\Api\Data\ProductInterface::WEIGHT
\Magento\Catalog\Api\Data\ProductInterface::STATUS
\Magento\Catalog\Api\Data\ProductInterface::VISIBILITY
\Magento\Catalog\Api\Data\ProductInterface::ATTRIBUTES_SET_ID
\Magento\Catalog\Api\Data\ProductInterface::TYPE_ID
\Magento\Catalog\Api\Data\ProductInterface::CREATED_AT
\Magento\Catalog\Api\Data\ProductInterface::UPDATED_AT
```



By **DominicXigen**

cheatography.com/dominicxigen/

Published 2nd April, 2020.

Last updated 2nd May, 2020.

Page 2 of 6.

Sponsored by **ApolloPad.com**

Everyone has a novel in them. Finish Yours!

<https://apollopadd.com>

Useful constants (cont)

```
> \Magento\Catalog\Api\Data\ProductInterface::MEDIA_GALLERY
\Magento\Catalog\Api\Data\ProductInterface::TIER_PRICE
```

Advanced Search

```
/**
 * @var \Magento\Catalog\Api\ProductRepositoryInterface
 */
private $productRepository;

/**
 * @var \Magento\Framework\Api\SearchCriteriaBuilder
 */
private $searchCriteriaBuilder;

/**
 * @var \Magento\Framework\Api\FilterBuilder
 */
private $filterBuilder;

/**
 * @var \Magento\Framework\Api\SortOrderFactory
 */
private $sortOrder;

/**
 * @param \Magento\Catalog\Api\ProductRepositoryInterface $productRepository
 * @param \Magento\Framework\Api\SearchCriteriaBuilder $searchCriteriaBuilder
 * @param \Magento\Framework\Api\FilterBuilder $filterBuilder
 * @param \Magento\Framework\Api\SortOrderFactory $sortOrder
 */
public function __construct(
    \Magento\Catalog\Api\ProductRepositoryInterface $productRepository,
    \Magento\Framework\Api\SearchCriteriaBuilder $searchCriteriaBuilder,
    \Magento\Framework\Api\FilterBuilder $filterBuilder,
    \Magento\Framework\Api\SortOrderFactory $sortOrder
) {
    $this->productRepository = $productRepository;
    $this->searchCriteriaBuilder = $searchCriteriaBuilder;
    $this->filterBuilder = $filterBuilder;
}
```

Advanced Search (cont)

```
> * @throws \Magento\Framework\Exception\InputException
*/

public function retrieve(
    array $excludedIds,
    array $filterIds,
    int $page = 1,
    int $pageSize = 100
): array {
    $this->searchCriteriaBuilder->addFilter('type_id', [
        \Magento\Catalog\Model\Product\Type::TYPE_SIMPLE,
        \Magento\Downloadable\Model\Product\Type::TYPE_DOWNLOADABLE
    ], 'in');
    if (count($filterIds)) {
        $this->searchCriteriaBuilder->addFilter('entity_id', $filterIds, 'in');
    }
    if (count($excludedIds)) {
        $this->searchCriteriaBuilder->addFilter('entity_id', $excludedIds, 'nin');
    }
    $sortOrder = $this->sortOrder->setField('updated_at')
        ->setDirection(\Magento\Framework\Api\SortOrder::SORT_DESC);
    $this->searchCriteriaBuilder->setSortOrders([$sortOrder]);
    $this->searchCriteriaBuilder->setCurrentPage($page);
    $this->searchCriteriaBuilder->setPageSize($pageSize);
    $searchCriteria = $this->searchCriteriaBuilder->create();
    return $this->productRepository->getList($searchCriteria)->getItems();
}
```

Useful Constants

```
\Magento\Framework\Api\SortOrder::SORT_ASC
\Magento\Framework\Api\SortOrder::SORT_DESC
```

Print Query String

```
$products = $this->productRepositoryInterface->getList($searchCriteria)
echo $products->getSqlText()->assemble();
```

```
ilder;  
    $th is- >so rtOrder = $sortO rder;  
}  
/**  
 * Retrieves products to be sent  
 * @param array $exclu dedIds  
 * @param array $filterIds  
 * @param int $page  
 * @param int $pageSize  
 * @return Produc tIn ter face[]
```



By **DominicXigen**

cheatography.com/dominicxigen/

Published 2nd April, 2020.

Last updated 2nd May, 2020.

Page 3 of 6.

Sponsored by **ApolloPad.com**

Everyone has a novel in them. Finish
Yours!

<https://apollopad.com>