2020 - 2021

EpiServer CMS

.NET FRAMEWORK (TESTED ON EPISERVER 11)

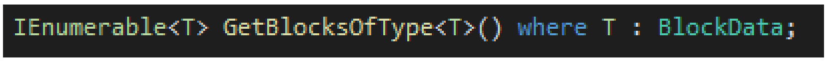
SONYA NADESAN

SONYA. | https://github.com/SonyaNadesan | https://www.linkedin.com/in/sonya-nadesan/

Table of Contents

BlockRepository: IBlockRepository

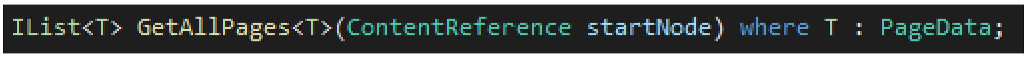
Retrieve all blocks of the specified type



```
IEnumerable<T> GetBlocksOfType<T>() where T : BlockData;
```

PageRepository: IPageRepository

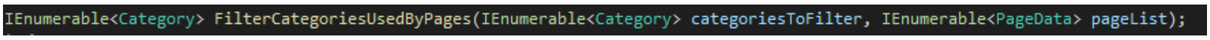
Retrieve all pages of the specified type



```
IList<T> GetAllPages<T>(ContentReference startNode) where T : PageData;
```

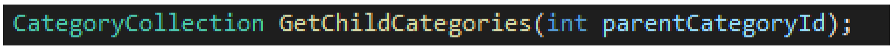
CategoryService: ICategoryService

Retrieve categories that are in use (i.e. those that have been assigned to at least one of the pages)



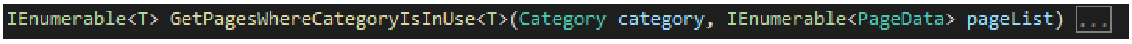
```
IEnumerable<Category> FilterCategoriesUsedByPages(IEnumerable<Category> categoriesToFilter, IEnumerable<PageData> pageList);
```

Retrieves child categories of a given category



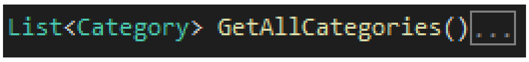
```
CategoryCollection GetChildCategories(int parentCategoryId);
```

Retrieves pages that have been assigned the given category



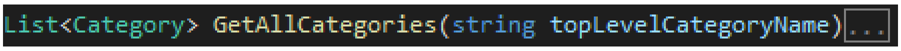
```
IEnumerable<T> GetPagesWhereCategoryIsInUse<T>(Category category, IEnumerable<PageData> pageList) ...
```

Retrieve all categories from hierarchy as a flat structure



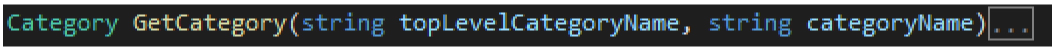
```
List<Category> GetAllCategories()...
```

Retrieve all categories under specified category from hierarchy as a flat structure



```
List<Category> GetAllCategories(string topLevelCategoryName)...
```

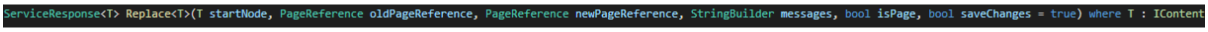
Finds the top-level category that matches the given name and loops through all categories under it; category is returned when the name matches the second argument



```
Category GetCategory(string topLevelCategoryName, string categoryName)...
```

PageReferenceReplacementService: IPageReferenceReplacementService

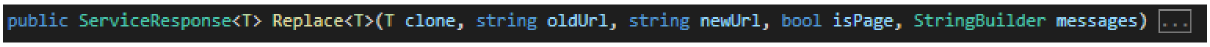
Loops through the properties of a given content (e.g. block/page), finds and replaces all references to a given page with another; references may be found in page reference fields, XHtmlString fields (rich text), URL fields or Link Item Collection Fields and will be handled by this method. This removes the need of implementing redirects, which can be better for performance.



```
ServiceResponse<T> Replace<T>(T startNode, PageReference oldPageReference, PageReference newPageReference, StringBuilder messages, bool isPage, bool saveChanges = true) where T : IContent
```

UrlReferenceReplacementService : IUrlReferenceReplacementService

Loops through the properties of a given content (e.g. block/page), finds and replaces all references to a given URL with another; references may be found in string fields (text), XHtmlString fields (rich text), URL fields or Link Item Collection Fields and will be handled by this method. This removes the need of implementing redirects, which can be better for performance.

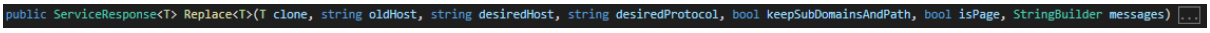


```
public ServiceResponse<T> Replace<T>(T clone, string oldUrl, string newUrl, bool isPage, StringBuilder messages) ...
```

Loops through the properties of a given content (e.g. block/page), finds and modifies all references to URLs with the specified host with another.

The protocol is also changed if necessary (e.g. from http to https); subdomains and paths are either kept or removed depending on the argument passed in.

References may be found in string fields (text), XHtmlString fields (rich text), URL fields or Link Item Collection Fields and will be handled by this method. This removes the need of implementing redirects, which can be better for performance.



```
public ServiceResponse<T> Replace<T>(T clone, string oldHost, string desiredHost, string desiredProtocol, bool keepSubDomainsAndPath, bool isPage, StringBuilder messages) ...
```