

View Base Class

`@inherits KarbonView`

General base class for views and partials.

`@inherits KarbonView<TCurrentPageType>`

Base class for views / partials with a strongly typed current page.

`@inherits KarbonView<TCurrentPageType, THomePageType>`

Base class for views / partials with a strongly typed current page and home page.

View Properties

`@Model.CurrentPage`

The `IContent` model for the current page.

`@Model.HomePage`

The `IContent` model for the home page.

Content Properties

`.Name:String`

The name of the page parsed from the folder name. Can be overridden in the content file.

`.Slug:String`

The URL slug for the content.

`.TypeName:String`

The type name for the content as parsed from the content file name.

`.SortOrder:Int`

The sort order for the content.

`.Depth:Int`

The depth of the content.

`.RelativePath:String`

The path of the content relative to the content store root.

`.RelativeUrl:String`

The URL of the content relative to the application root.

`.Created:DateTimeOffset`

The creation date of the content.

`.Modified:DateTimeOffset`

The last modified date of the content.

`.Data:IDictionary<String, String>`

The key value dictionary of parsed values from the content file. If a strongly typed model is used, values will be mapped to model properties and the Data collection will then contain any un-mapped values.

Content Helper Methods

`.Url():String`

Get the absolute URL for the content.

`.IsVisible():Bool`

Gets a flag indicating whether the content is visible in the navigation.

`.IsOpen():Bool`

Gets a flag indicating whether the content is open in the navigation.

`.IsHomePage():Bool`

Gets a flag indicating whether the content is the home page.

`.IsChildOf(IContent content):Bool`

Gets a flag indicating whether the content is a child of the supplied content.

`.IsAncestorOf(IContent content):Bool`

Gets a flag indicating whether the content is an ancestor of the supplied content.

`.IsDescendantOf(IContent content):Bool`

Gets a flag indicating whether the content is a descendant of the supplied content.

`.HasPrev([Func<IContent, Bool> filter]):Bool`

`.HasPrev<TContentType>([Func<TContentType, Bool> filter]):Bool`

Gets a flag indicating whether the content has a previous sibling optionally filter by the type or filter function parameter.

`.HasNext([Func<IContent, Bool> filter]):Bool`

`.HasNext<TContentType>([Func<TContentType, Bool> filter]):Bool`

Gets a flag indicating whether the content has a next sibling optionally filter by the type or filter function parameter.

Content Traversal

`.Parent():IContent`

`.Parent<TContentType>():TContentType`

Gets the parent content optionally cast to the supplied type parameter

`.Parents([Func<IContent, Bool>`

`filter]):IEnumerable<IContent>`

`.Parents<TContentType>([Func<TContentType, Bool>`

`filter]):IEnumerable<TContentType>`

Gets the ancestor content optionally filtered by the type and filter function parameter.

`.Closest(Func<IContent, Bool> filter):IContent`

`.Closest<TContentType>([Func<TContentType, Bool>`

`filter]):TContentType`

Gets the closest ancestor content item filtered by the type and / or filter function parameter.

`.Children([Func<IContent, Bool>`

`filter]):IEnumerable<IContent>`

`.Children<TContentType>([Func<TContentType, Bool>`

`filter]):IEnumerable<TContentType>`

Gets the child content optionally filtered by the type and filter function parameter.

`.Siblings([Func<IContent, Bool>`

`filter]):IEnumerable<IContent>`

`.Siblings<TContentType>([Func<TContentType, Bool>`

`filter]):IEnumerable<TContentType>`

Gets the sibling content optionally filtered by the type and filter function parameter.

`.Prev([Func<IContent, Bool> filter]): IContent`

`.Prev<TContentType>([Func<TContentType, Bool>`

`filter]):TContentType`

Gets the previous content item optionally filtered by the type and filter function parameter.

`.Next([Func<IContent, Bool> filter]): IContent`

`.Next<TContentType>([Func<TContentType, Bool>`

`filter]):TContentType`

Gets the previous content item optionally filtered by the type and filter function parameter.

`.Find(Func<IContent, Bool> filter):IEnumerable<IContent>`

`.Find<TContentType>([Func<TContentType, Bool>`

`filter]):IEnumerable<TContentType>`

Gets the descendant content optionally filtered by the type and filter function parameter.

Content Media Access

`.Files([Func<IFile, Bool> filter]):IEnumerable<IFile>`
`.Files<TFileType>([Func<TFileType, Bool> filter]):IEnumerable<TFileType>`

Gets all files optionally filtered by the type and filter function parameter.

`.Images([Func<IImageFile, Bool> filter]):IEnumerable<IImageFile>`
`.Images<TFileType>([Func<TFileType, Bool> filter]):IEnumerable<TFileType>`

Gets all image files optionally filtered by the type and filter function parameter. Image files must have one of the following file extensions:

- Jpg	- Jpeg	- Gif
- Png	- Bmp	- Tif
- Tiff		

`.Videos([Func<IVideoFile, Bool> filter]):IEnumerable<IVideoFile>`
`.Videos<TFileType>([Func<TFileType, Bool> filter]):IEnumerable<TFileType>`

Gets all video files optionally filtered by the type and filter function parameter. Video files must have one of the following file extensions:

- Ogg	- Ogv	- Webm
- Mp4	- Mov	- Avi
- Flv	- Swf	

`.Sounds([Func<ISoundFile, Bool> filter]):IEnumerable<ISoundFile>`
`.Sounds<TFileType>([Func<TFileType, Bool> filter]):IEnumerable<TFileType>`

Gets all sound files optionally filtered by the type and filter function parameter. Sound files must have one of the following file extensions:

- Mp3	- Wav	- Wma
- Mid	- Ra	- Ram
- Rm		

`.Documents([Func<IDocumentFile, Bool> filter]):IEnumerable<IDocumentFile>`
`.Documents<TFileType>([Func<TFileType, Bool> filter]):IEnumerable<TFileType>`

Gets all sound files optionally filtered by the type and filter function parameter. Document files must have one of the following file extensions:

- Pdf	- Doc	- Docx
- Xls	- Xlsx	- Ppt
- Pptx	- Rtf	

File Properties

`.Name:String`

The name of the file parsed from the file name. Can be overridden in the meta data file.

`.Slug:String`

The URL slug for the file.

`.Extension:String`

The file extension of the file.

`.Size:Long`

The size of the file in bytes.

`.TypeName:String`

The type name for the meta data as parsed from the meta data file name.

`.SortOrder:Int`

The sort order for the file.

`.RelativePath:String`

The path of the file relative to the content store root.

`.RelativeUrl:String`

The URL of the file relative to the application root.

`.ContentRelativeUrl:String`

The relative URL of the associated content.

`.Created:DateTimeOffset`

The creation date of the file.

`.Modified:DateTimeOffset`

The last modified date of the file.

`.Data:IDictionary<String, String>`

The key value dictionary of parsed values from the meta data file. If a strongly typed model is used, values will be mapped to model properties and the Data collection will then contain any un-mapped values.

File Helper Methods

`.Url():String`

Gets the absolute URL for the content.

`.NiceSize():String`

Gets the size of the file formatted in a human readable format.

`.MimeType():String`

Gets the mime type of the file.

`.IsImage():Bool`

Gets a flag indicating whether the file is an image.

`.IsVideo():Bool`

Gets a flag indicating whether the file is a video.

`.IsSound():Bool`

Gets a flag indicating whether the file is a sound.

`.IsDocument():Bool`

Gets a flag indicating whether the file is a document.

`.HasPrev([Func<IFile, Bool> filter]):Bool`

`.HasPrev<TFileType>([Func<TFileType, Bool> filter]):Bool`
 Gets a flag indicating whether the file has a previous sibling optionally filter by the type or filter function parameter.

`.HasNext([Func<IFile, Bool> filter]):Bool`

`.HasNext<TFileType>([Func<TFileType, Bool> filter]):Bool`
 Gets a flag indicating whether the file has a next sibling optionally filter by the type or filter function parameter.

`.Prev([Func<IFile, Bool> filter]): IFile`

`.Prev<TFileType>([Func<TFileType, Bool> filter]):TFileType`
 Gets the previous file optionally filtered by the type and filter function parameter.

`.Next([Func<IFile, Bool> filter]): IFile`

`.Next<TFileType>([Func<TFileType, Bool> filter]):TFileType`
 Gets the previous file optionally filtered by the type and filter function parameter.

Image File Properties

In addition to the standard file properties, image files have the following additional properties:

`.Width:Int`

The width of the image.

`.Height:Int`

The height of the image.

Image File Helper Methods

The following "Fit" methods all make use of the ImageResizing.net library. See <http://imageresizing.net/docs/basics> for details of advanced options.

All "Fit" methods return an IFilteredImage object to allow you to daisy chain additional commands together. To get the final URL, simply output the object itself or explicitly call .Url().

.FitWidth(int width):IFilteredImage

Gets the URL of the image resized to fit within the supplied width parameter.

.FitHeight(int height):IFilteredImage

Gets the URL of the image resized to fit within the supplied height parameter.

.Fit(int maxWidthHeight, [FitMode fitMode, ScaleMode scaleMode, AlignMode alignMode, ImageFormat format, int quality, int colors, string bgColor]):IFilteredImage
.Fit(int width, int height, [FitMode fitMode, ScaleMode scaleMode, AlignMode alignMode, ImageFormat format, int quality, int colors, string bgColor]):IFilteredImage
 Gets the URL of the image resized to fit within the supplied width / height parameter, optionally constrained by the supplied parameters.

Data Helper Methods

Most Karbon entities will expose a .Data property. Whilst you can access the values directly as you would with any other dictionary, the following helper methods also exist to provide you with built in error checking and fallback values.

.GetValue(String key, [String defaultValue]):String
.GetValue<TValueType>(String key, [TValueType defaultValue]):TValueType

.GetValue<TValueType, TConverterType>(String key, [TValueType defaultValue]):TValueType

Gets the value for the given key. If no key exists, or the value is empty, the optional defaultValue will be returned.

.TryGetValue(String key, out String value):Bool
.TryGetValue<TValueType>(String key, out TValueType value):Bool
.TryGetValue<TValueType, TConverterType>(String key, out TValueType value):Bool

Gets a flag indicating whether a value for the given key can be found.

Controller Base Class

KarbonController

General base class for custom controllers.

KarbonController<TCurrentPageType>

Base class for custom controllers with a strongly typed current page.

KarbonController<TCurrentPageType, THomePageType>

Base class for controllers with a strongly typed current page and home page.

Controller Properties

.CurrentPage

The IContent model for the current page.

.HomePage

The IContent model for the home page.

Html Helper Methods

@Html.Markdown(String markdown)

@Html.Md(String markdown)

Parses and outputs the supplied markdown formatted string parameter.

Cheat Sheet v1.0

Please go to <http://karboncms.com/docs> to get the full documentation.

<http://karboncms.com>

<http://twitter.com/karboncms>