/sitecore/content/Motorola/Settings/Motorola Settings - Apps/32x32/Setting.png

Get Sitecore Items in MVC

Direct Access in View

```
@{
    var item = Sitecore.Context.Database.GetItem("/sitecore/content/home");
    // OR Pass by Id
    //var item = Sitecore.Context.Database.GetItem("{110D559F-DEA5-42EA-9C1C-
8A5DF7E70EF9}");
}

<span>Way 1 to get Title: </span> <span> @item["Title"] </span>
<br>
<span>Way 2 to get Title: </span> <span> @item.Fields["Title"].Value </span>
<br>
<span>Way 3 to get Title: </span> <h4>@Html.Sitecore().Field("Title", item)</h4>
<br>
<span>Way 1 to get Image:</span> <h4>@Html.Sitecore().Field("MainImage", item)</h4>
<br>
<span>Way 2 to get Image:</span> <h4>@Html.Sitecore().Field("MainImage", item, new {
@class = "img-responsive" })</h4>
<br>
<span>Way 3 to get Image:</span>
       @{Sitecore.Data.Fields.ImageField imageField = item.Fields["WebsiteLogo"];}
<img src=@(Sitecore.Resources.Media.MediaManager.GetMediaUrl(imageField.MediaItem))
class="img-responsive" />

<br>
<div id="footer">
     @Html.Sitecore().Field("Copyright Text",
Sitecore.Context.Database.GetItem("/sitecore/Content/Global/Configurations"))
</div>
```

Controller – Model Item - View

```
    public class BlogController : Controller
    {
        public ActionResult ShowArticle()
        {
            var item = Sitecore.Context.Database.GetItem("/sitecore/content/home");
            string title = item.Fields["Title"].Value;
            return View(item);
        }
    }
```

View and Model Item

```
@model Sitecore.Data.Items.Item

<div>
    @Model.DisplayName

    <span>Title: </span> <span> @Model.Fields["Title"].Value </span>
</div>
```

Controller – Model - View

Model

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

namespace Sitecore.Feature.Header.Models
{
    public class BlogHeader
    {
        public string Heading { set; get; }
        public string SubHeading { set; get; }
    }
}
```

Controller

```
using Sitecore.Feature.Header.Models;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web.Mvc;
using Sitecore.Mvc;
using Sitecore.Data.Fields;

namespace Sitecore.Feature.Header.Controllers
{
    public class HeaderController : Controller
    {
        public ActionResult HeaderBlogSection()
        {
            var item = Sitecore.Context.Database.GetItem(new Data.ID("{6114EA40-687B-
43C5-9774-AB02C199D262}"));

            BlogHeader blogHeader = new BlogHeader { Heading =
item.Fields["Heading"].Value, SubHeading = item.Fields["SubHeading"].Value
            };

            return View(blogHeader);
        }
    }
}
```

View

```
@model Sitecore.Feature.Header.Models.BlogHeader

<div class="jumbotron text-white bg-primary">
    <h1 class="display-3">@Model.Heading</h1>
    <p class="lead">@Html.Raw(Model.SubHeading)</p>
</div>
```

Layout with CSS and JS File

```
D:\inetpub\wwwroot\xptensc.dev.local\assets\js\myscripts.js
```

```
function myFunction() {
  alert("Hello Sitecore!!!");
}
```

```
D:\inetpub\wwwroot\xptensc.dev.local\assets\css\blogsite.css
```

```
  .main-search {
    width: 50%;
    margin: 50px auto;
}

.has-search .form-control {
    padding-left: 2.375rem;
}

.has-search .form-control-feedback {
    position: absolute;
    z-index: 2;
    display: block;
    width: 2.375rem;
    height: 2.375rem;
    line-height: 2.375rem;
    text-align: center;
    pointer-events: none;
    color: #aaa;
}

.contentTitle {
       margin-top: 50px;
       font-weight:400;
       font-size: 36px;
}
```

Layout

```
@using Sitecore.Mvc
@using Sitecore.Mvc.Analytics.Extensions
@{
    Layout = null;
}
<!DOCTYPE html>
<html lang="en" xmlns="http://www.w3.org/1999/xhtml">
    <head>
        <title>@Html.Sitecore().Field("title", new { DisableWebEdit = true })</title>
        @Html.Sitecore().VisitorIdentification()
              <link rel="stylesheet" href="/assets/css/blogsite.css">
              <script src="/assets/js/myscripts.js"></script>
    </head>
    <body>

        <div>
            @Html.Sitecore().Placeholder("phMain")
        </div>
    </body>
</html>
```

Header Component

```
<div>
@{
       var item = Sitecore.Context.Database.GetItem("{BBCF2B98-68A6-4A38-8BC2-
0679AF79F845}");

  }

       <span>Way 1 to get Field: </span> <span class="contentTitle">
@item["WebsiteTitle"] </span>

</div>

<div style="background-color: #3333;" onclick="myFunction()">
       This is Header component
</div>
```

```
@Html.Sitecore().Field(“imageField”, new {w=160, h=160})
@Html.Sitecore().Field("link", new { @class = "red-link" })
@Html.Sitecore().Field("post date", new { format = "MMM dd,yyyy" })
@Html.Sitecore().Field("<field name>", new { DisableWebEdit = true })
foreach(var item in Model.ListOfItems)
{
     <h1>
          @Html.Sitecore().Field(“title”, item)
     </h1>
}
```

```
using Sitecore.Web.UI.WebControls;

public HtmlString MyAwesomeField { get; set; }

model.MyAwesomeField = new HtmlString(FieldRenderer.Render(item, "My Awesome Field"));
```

To disable content editing for an item field,

```
        @Html.Sitecore().Field("<field name>", new { DisableWebEdit = true })
```

How to get Item , Field, database in Sitecore

How to get Item , Field, database in Sitecore

To get current context item in sitecore use following code.

```
Sitecore.Data.Items.Item _currentItem = Sitecore.Context.Item
```

To get current context database in sitecore

```
Sitecore.Data.Database db = Sitecore.Context.Database;
```

To get an item from current context database

```
Sitecore.Data.Items.Item item = Sitecore.Context.Database.GetItem("Enter GUID or
Path here");
```

To get any database in sitecore

```
Sitecore.Data.Database masterdb =
Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Database webdb =
Sitecore.Configuration.Factory.GetDatabase("web");
```

To get single line text/ multi line text/ rich text field value in sitecore

```
string Title =  item.Fields["Title"].Value;
```

```
We also can access field value using GUID of field
```

```
string Title =  item.Fields["GUID here"].Value;
```

To get Droplink field value in Sitecore

```
Sitecore.Data.Fields.ReferenceField droplinkField = Sitecore.Context.Item.Fields["Title"];
if(droplinkField !=null && droplinkField.TargetItem != null)
 {
 Sitecore.Data.Items.Item targetItem = droplinkField.TargetItem;
// here targetietm is the value in the Droplink field "Title"
 }
```

To get Multilist field value in Sitecore

```
Sitecore.Data.Fields.MultilistField multilistField =

Sitecore.Context.Item.Fields["CityNames"];


if(multilistField != null)

 {

 foreach(Sitecore.Data.Items.Item city in multilistField.GetItems())

 {

 //your code is here

 }

 }
```

To get link field value in sitecore

```
Sitecore.Data.Fields.LinkField lf = Sitecore.Context.Item.Fields["PageUrl"];

string url =string.Empty;

if (lf != null)

 {

url = lf.GetFriendlyUrl(); // just get friendly url. if you are not happy

with this then use //below switch case.

 switch (lf.LinkType.ToLower())

 {

 case "internal":

 // link is internal to sitecore item then use LinkMananger

 url = lf.TargetItem != null ? Sitecore.Links.LinkManager.GetItemUrl(lf.TargetItem) :

string.Empty;

 break;

 case "media":
```

```
 // Use MediaManager for media link
 url = lf.TargetItem != null ?
Sitecore.Resources.Media.MediaManager.GetMediaUrl(lf.TargetItem）：string.Empty；
 break;
 case "external":
 // for external links
 url = lf.Url;
 break;
 case "anchor":
 // Prefix anchor link with #
 url = !string.IsNullOrEmpty(lf.Anchor) ? "#" + lf.Anchor : string.Empty;
 break;
 case "mailto":
url = lf.Url;
 break;
 case "javascript":
url = lf.Url;
 break;
 default:
 // If all fails
 url = lf.Url;
 break;
 }
 }
```