

radley yeldar.

Umbraco Practices!

Anthony Dang - Technical Director

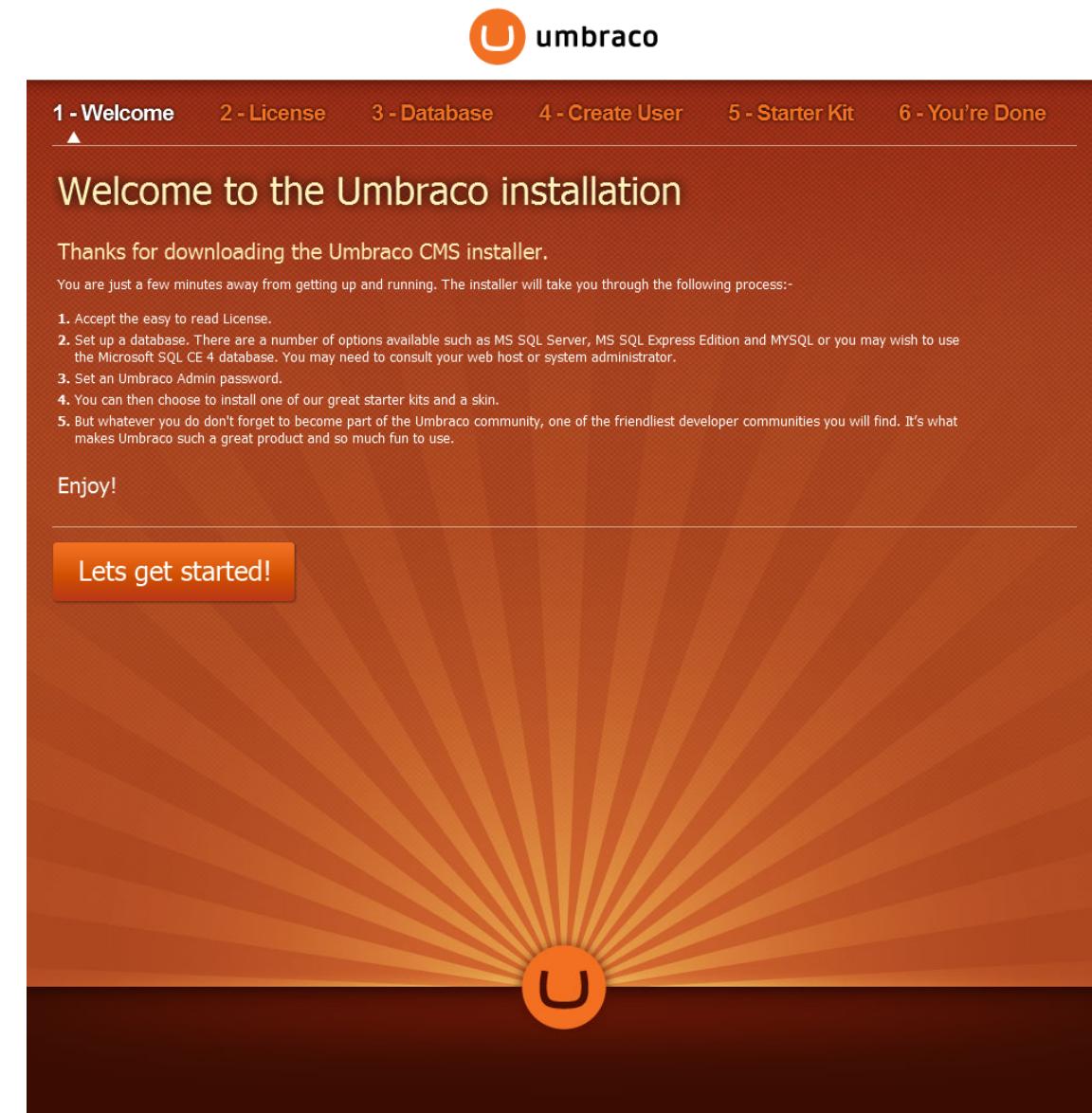
@anthonydotnet

radley yeldar.

A bit about me

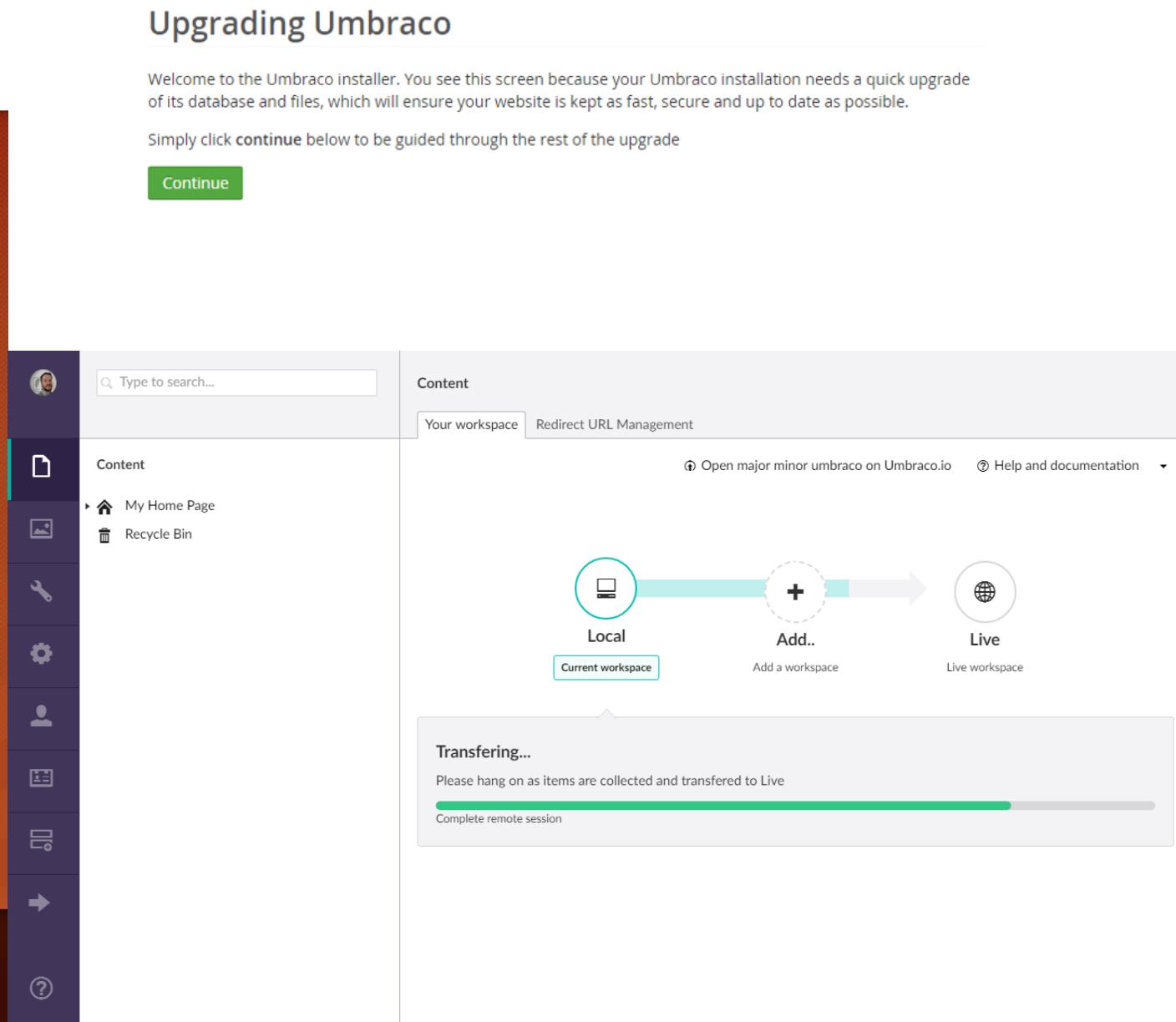


UMBRACO MVP



The Umbraco CMS installer welcome screen features a large orange 'U' logo at the bottom center. The main title 'Welcome to the Umbraco installation' is displayed prominently. Below it, a message says 'Thanks for downloading the Umbraco CMS installer.' A list of steps follows: 'You are just a few minutes away from getting up and running. The installer will take you through the following process:-
1. Accept the easy to read License.
2. Set up a database. There are a number of options available such as MS SQL Server, MS SQL Express Edition and MYSQL or you may wish to use the Microsoft SQL CE 4 database. You may need to consult your web host or system administrator.
3. Set an Umbraco Admin password.
4. You can then choose to install one of our great starter kits and a skin.
5. But whatever you do don't forget to become part of the Umbraco community, one of the friendliest developer communities you will find. It's what makes Umbraco such a great product and so much fun to use.'

Below the steps, a button labeled 'Enjoy!' is followed by a large orange button labeled 'Let's get started!'



Upgrading Umbraco

Welcome to the Umbraco installer. You see this screen because your Umbraco installation needs a quick upgrade of its database and files, which will ensure your website is kept as fast, secure and up to date as possible.

Simply click **continue** below to be guided through the rest of the upgrade

Continue

The dashboard shows a sidebar with icons for Content, Media, Tools, Settings, Members, and Reports. The main area has a search bar and tabs for 'Content', 'Your workspace', and 'Redirect URL Management'. At the bottom, there's a workspace management section with three circles: 'Local' (current workspace), 'Add..', and 'Live'. A progress bar indicates 'Transferring...' items from Local to Live. Buttons for 'Open major minor umbraco on Umbraco.io' and 'Help and documentation' are also present.

- Less code/keep it simple
- C# purists with dogmantic design patterns



- ~~Virtual Machine~~
 - ~~RDP (Ghetto Deploy)~~
 - Azure App Services / AWS Elastic Beanstalk / Umbraco Cloud
 - Teamcity + Octopus Deploy
 - AppVeyour
 - Azure DevOps
-

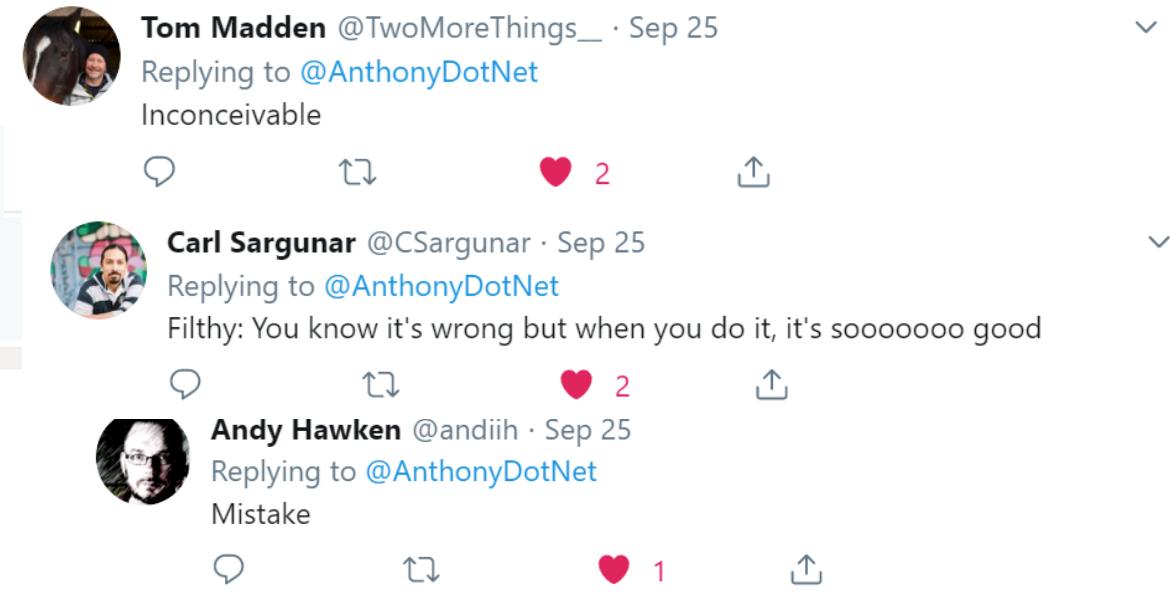
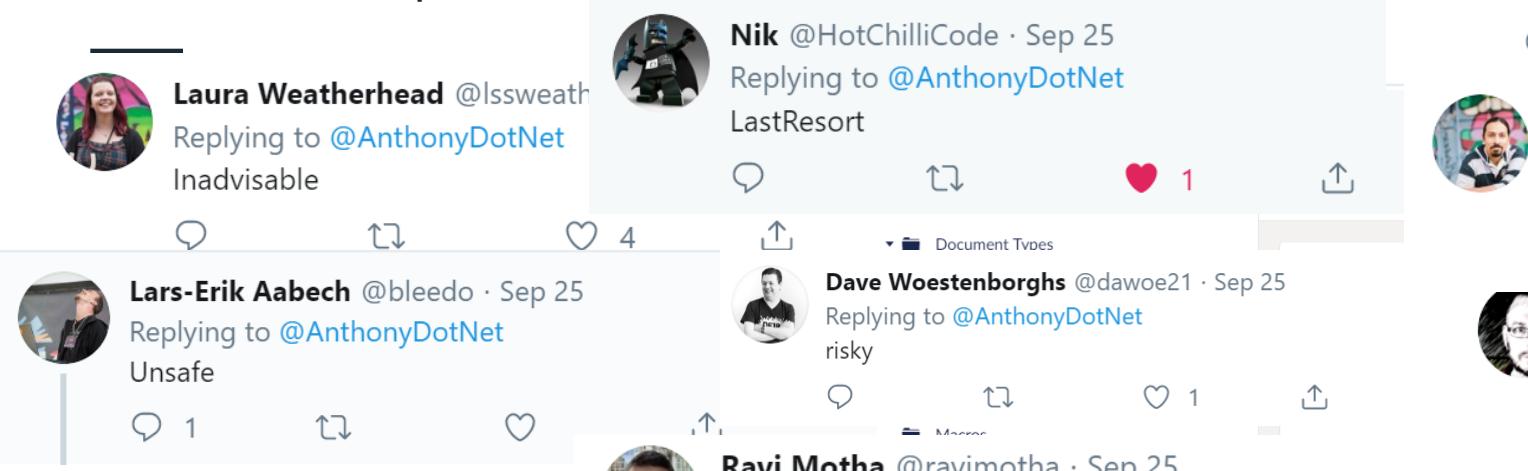
Google Cloud?

- Corrupt examine indexes
- Azure App Services
- Old Lucene - Facet support?
- Azure Search?

radley yeldar.

Backoffice...

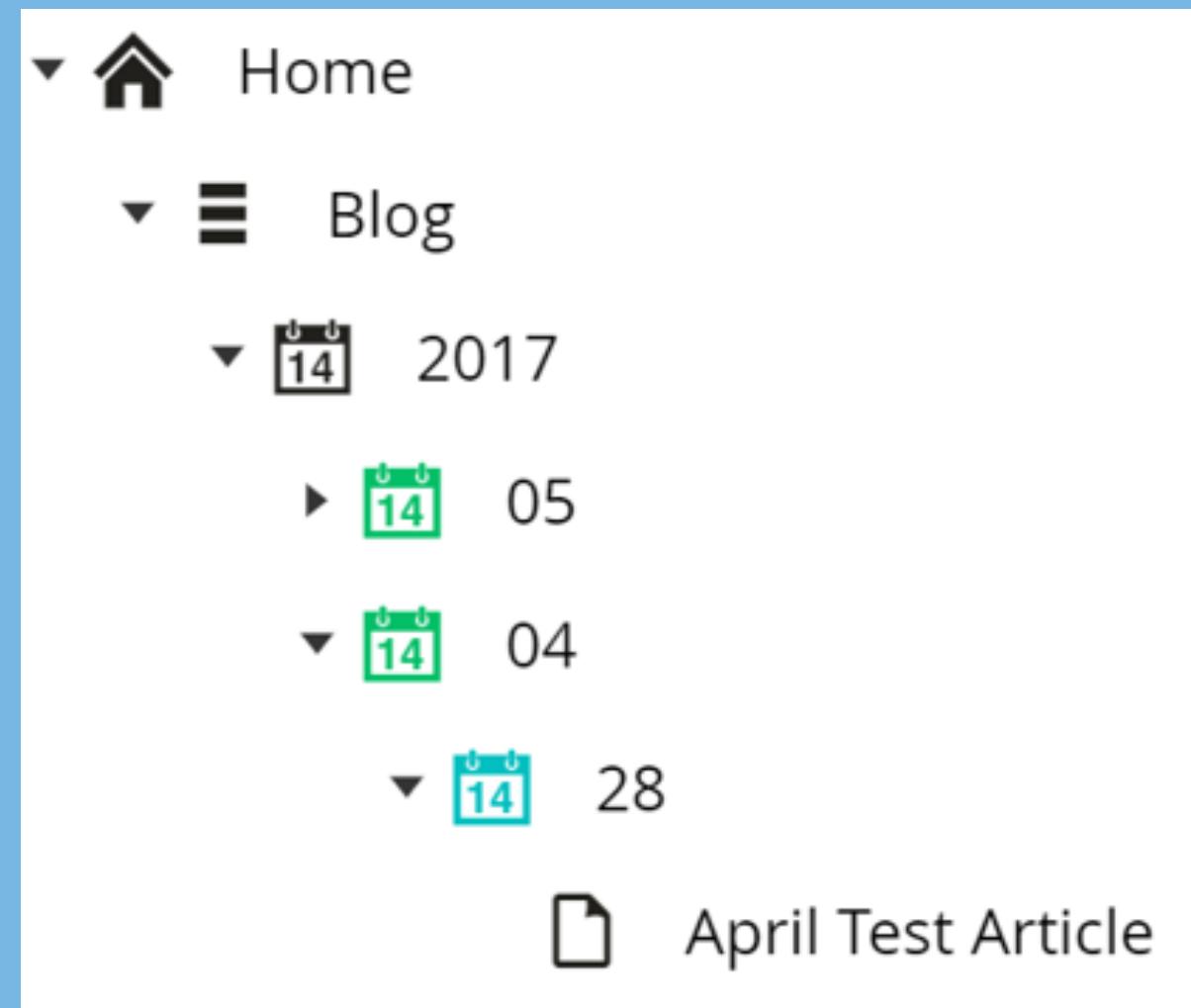
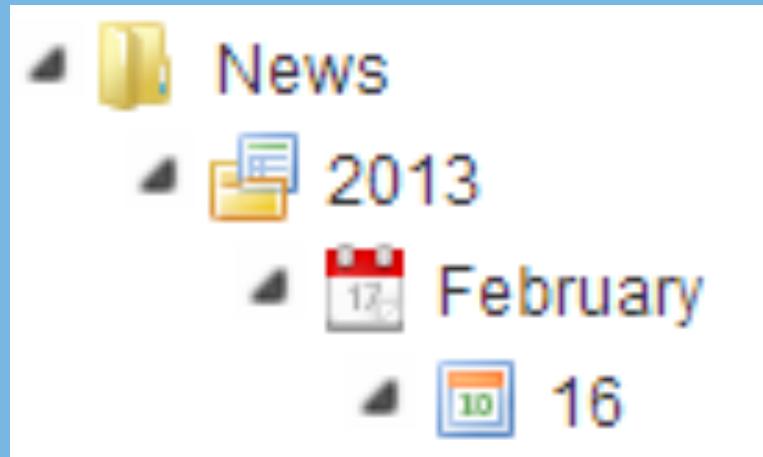
Backoffice Template Editor



- Source control
- Overwrite on deploy

What about Umbraco Cloud?

Legacy - Date Folders



Christmas songs					Actions ▾
	Child items	Properties			
Create Song			☰	Q Type to search...	
NAME +	ARTIST	RELEASED	GENRE	IMAGE	
♪ Do They Know it's Christmas	Band Aid	1984		1794	
♪ Rockin' Around the Christmas Tree	Brenda Lee	1958	Christmas,country,...	1799	
♪ The Christmas Song	Nat King Cole	1946	Christmas,Jazz,Pop	1795	
♪ White Christmas	Bing Crosby	1946	Jazz,Holiday,Easy lis...	1801	

- Archetype
- Grid
- Doctype Grid Editor
- Nested Content
- Stacked Content (not ready)
- Custom Property Editor



Headline	Article	50 - 50 Row	33-33-33 Row	Hero Banner
HTML Editor	Large CTA Blocks	Small CTA Blocks	CTA Related Links	Latest News Blocks
Share Ticker	1 Col Row	Graphic Block	CTA Roll Over Link	Sortable Blogs Block

_UmbracoNaviHide

Main Navigation

Footer Navigation

Sub Navigation

Sitemap XML

Sitemap page

Tab: Generic Properties

[Edit "Hide in navigation?"](#)

Name

Hide in navigation?

Alias

umbracoNaviHide

Type

True/false

Tab

Generic Properties

Mandatory



Validation

Type to search...

Home Content Navigation Site Properties Actions ▾

Content

- Home
 - Learn
 - The starter kit
 - Basics
 - Masterclasses
 - Explore
 - Our Umbraco
 - Codegarden
 - Meetup groups
 - Extend
 - Umbraco Forms
 - Package repository
 - Blog
 - Contact
 - Recycle Bin

Main Navigation

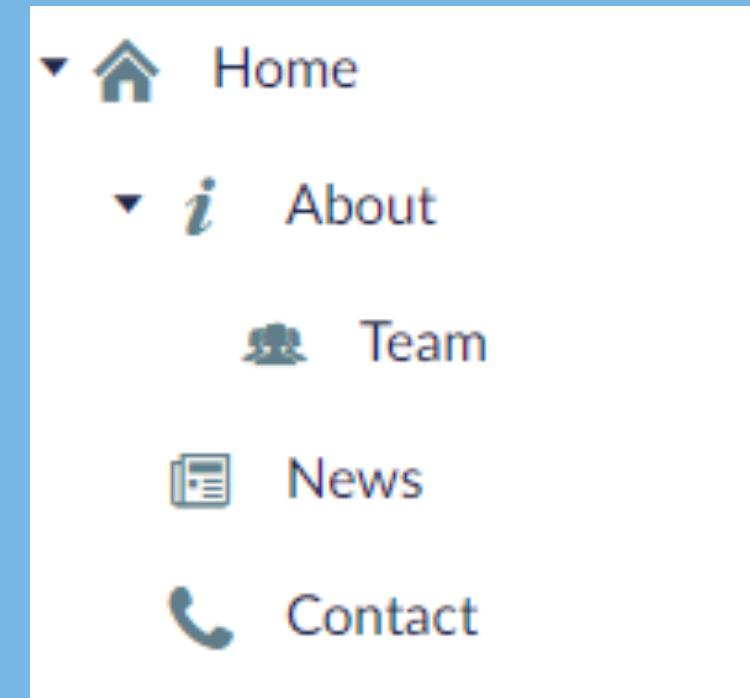
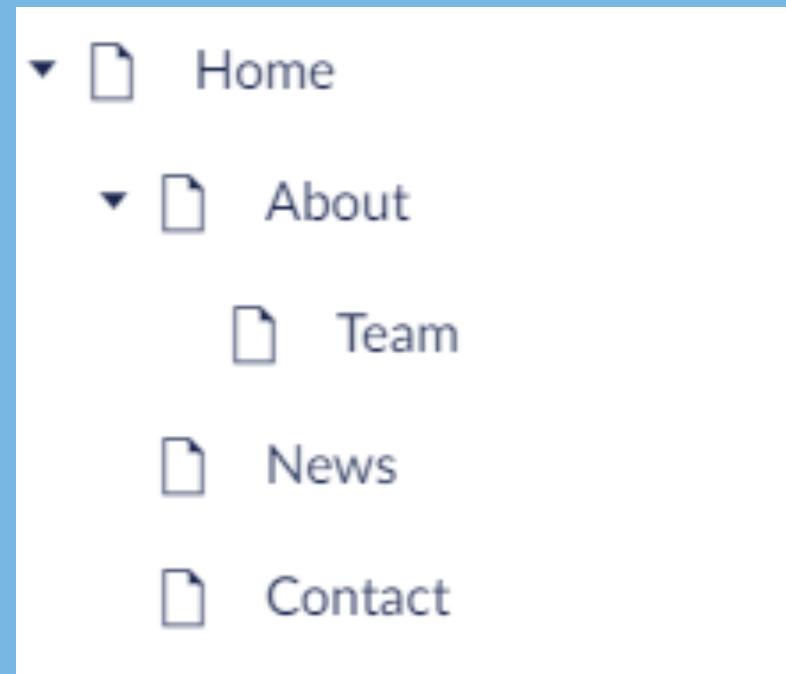
- Home / Edit Delete
- Explore /explore/ Edit Delete
- Extend /extend/ Edit Delete
- Contact /contact/ Edit Delete
- Blog https://blog.example.com/ Edit Delete
- Categories https://blog.example.com/categories/ Edit Delete

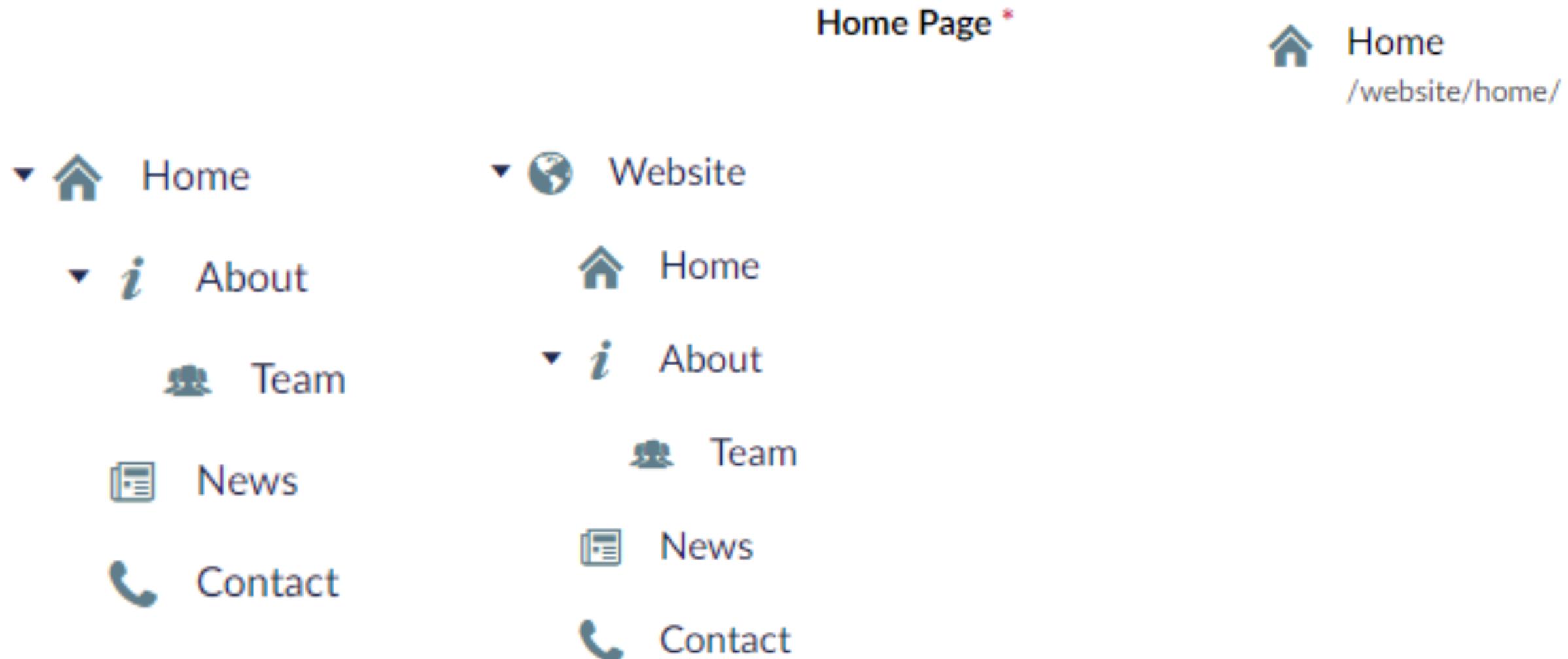
Add

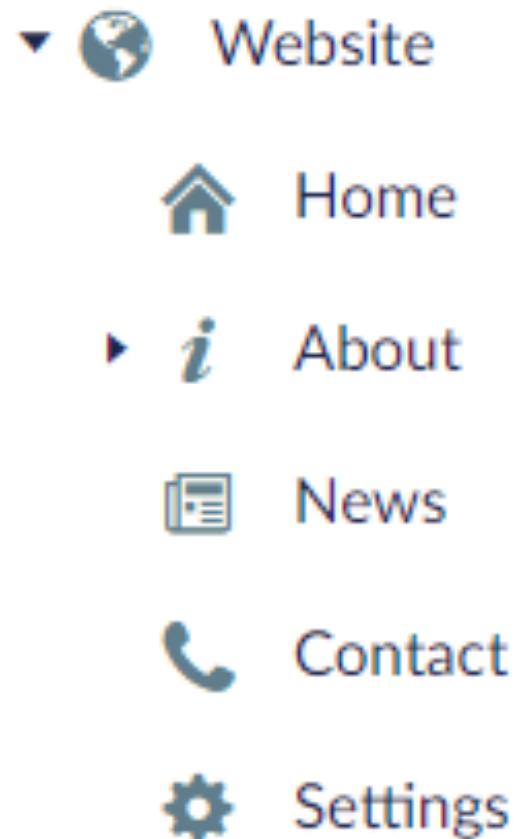
Preview Save and publish ▾

@anthonydotnet

Content Tree - Icons

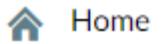






Site Settings

Home Page *



Home

/website/home/

SEO

Robots txt

Disallow Robots



User-agent: *

Disallow: /

Scripts

Tags

Categories

Authors

List views

Use Multinode Tree Picker

▼  Data Repositories

▼  Taxonomies

 Categories

 Tags

 Authors

Content Tree - Modern Node Structure?

Site Container

Root has settings

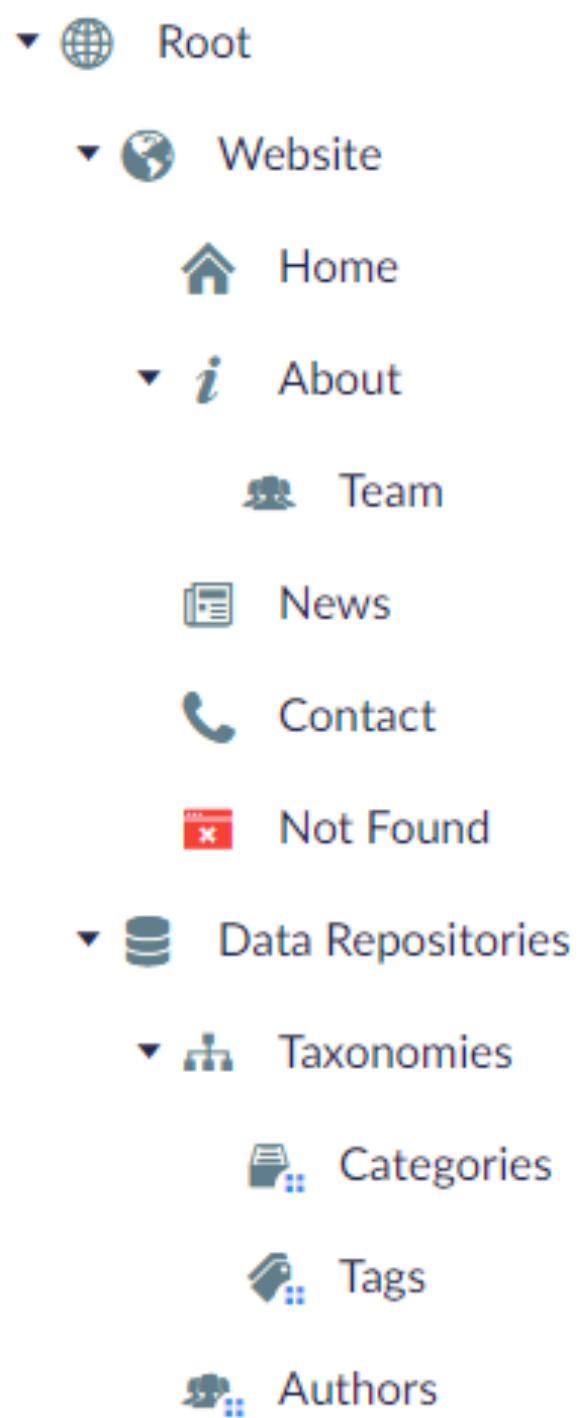
Home is child of root

Home is selectable

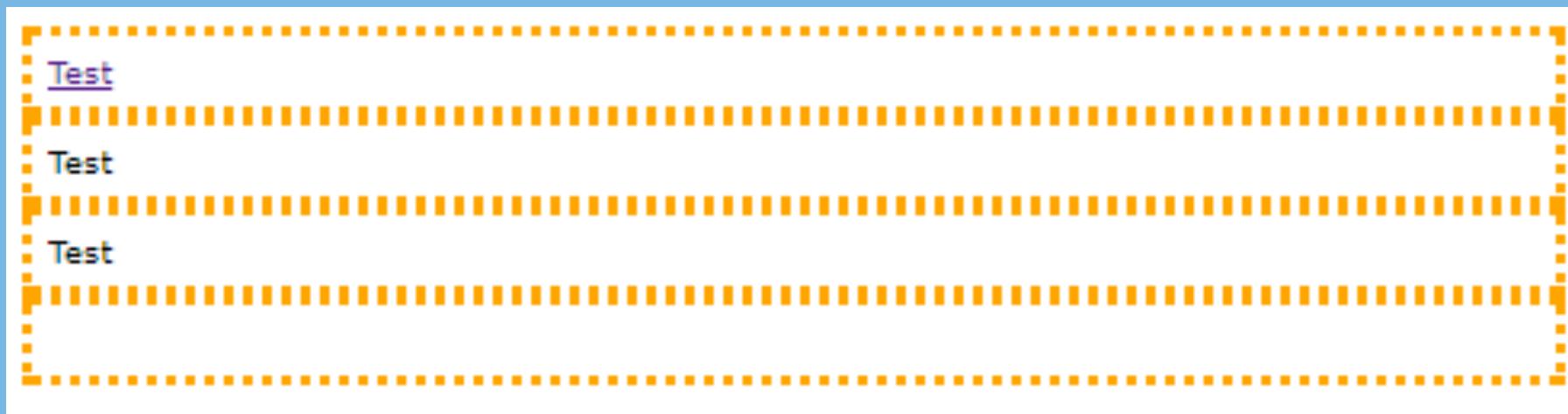
News is list view

Data Repositories

Tags, Categories, Authors are list views



- Only for rich text editor
- Tempting for frontend devs



Doctypes

- Inheritance vs compositions
- Error 404 has SEO Meta data?
- Compositions are the future!

▼ └ └ Document Types	
▼ └ └ Shared	
▼ └ └ Compositions	
grid Module Composition	
grid Page Modules Composition	
grid Page SEO Settings Composition	
grid Script Settings Composition	
grid Site Composition	
grid Site SEO Settings Composition	
▼ └ └ Elements	
▶ └ └ Modules	
grid Script Settings Item	
▼ └ └ Pages	
grid Error 404 Page	

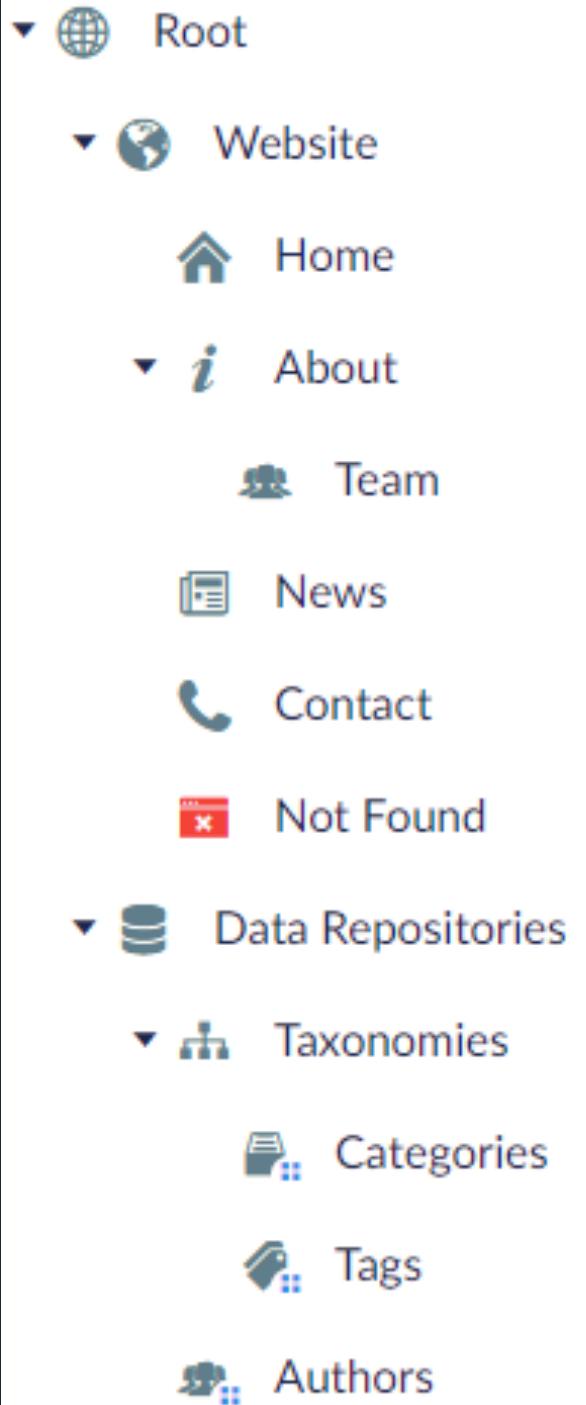
Tree Crawling & Cache



Tree Traversal

- Descendants
- Ancestors
- Children

```
var contactPage = currentPage
    .AncestorsOrSelf(1)
    .Descendants("ContactUs")
    .First()
```

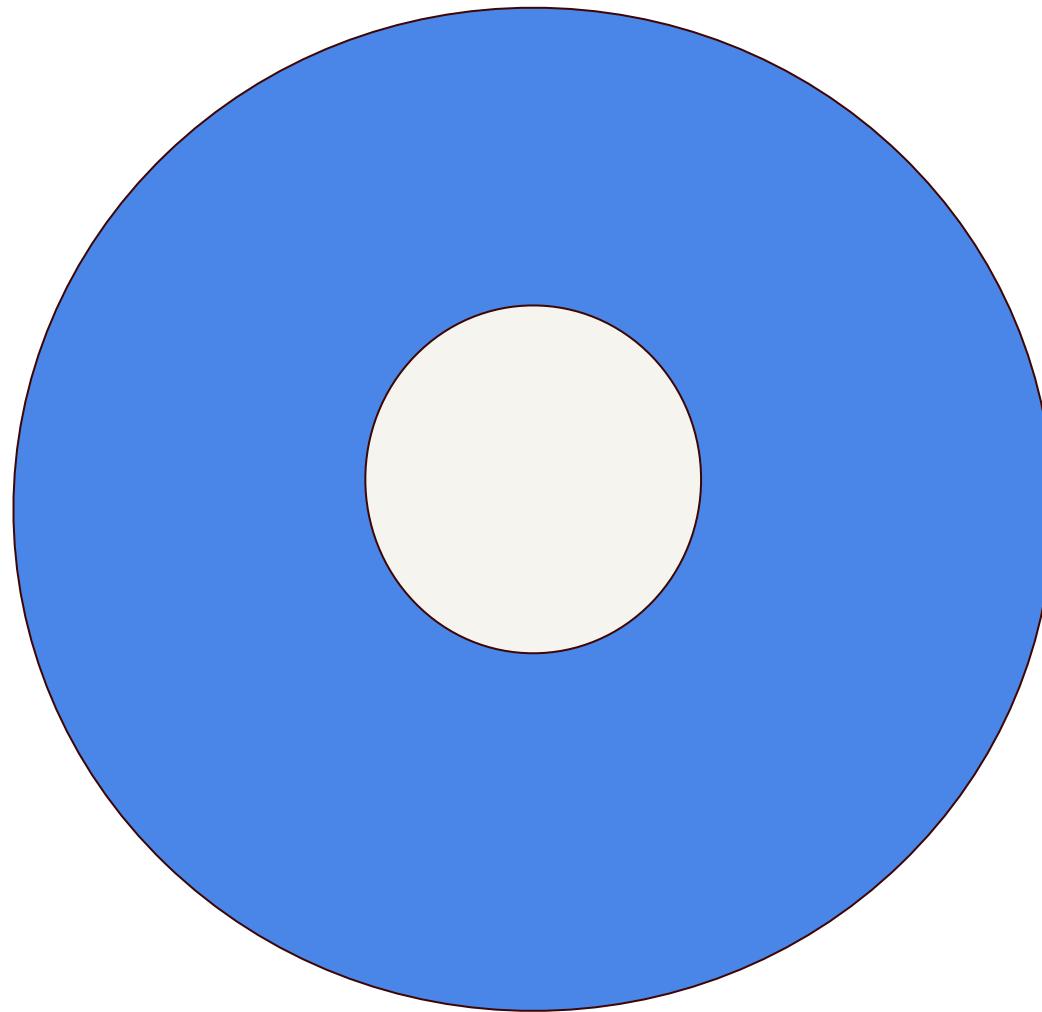


```
@Html.CachedPartial("MyPartialName", new MyModel(), 3600)
```

- Hardcoded cache time
 - How to clear cache in code?
 - How to turn off cache?
-

```
@Html.CachedPartial("MyPartialName", new MyModel(), 3600)
```

```
@Html.Action("Login", "Account", true)
```



excludeFromParentCache



```
@Html.Action("Login", "Account", true)
```

```
[DonutOutputCache(Duration=60)]  
public ActionResult Index()
```

excludeFromParentCache

```
[DonutOutputCache(CacheProfile="TwoMins")]  
public ActionResult Index()
```

```
[DonutOutputCache(Duration=60, VaryByCustom="whatever")]  
public ActionResult Index()
```

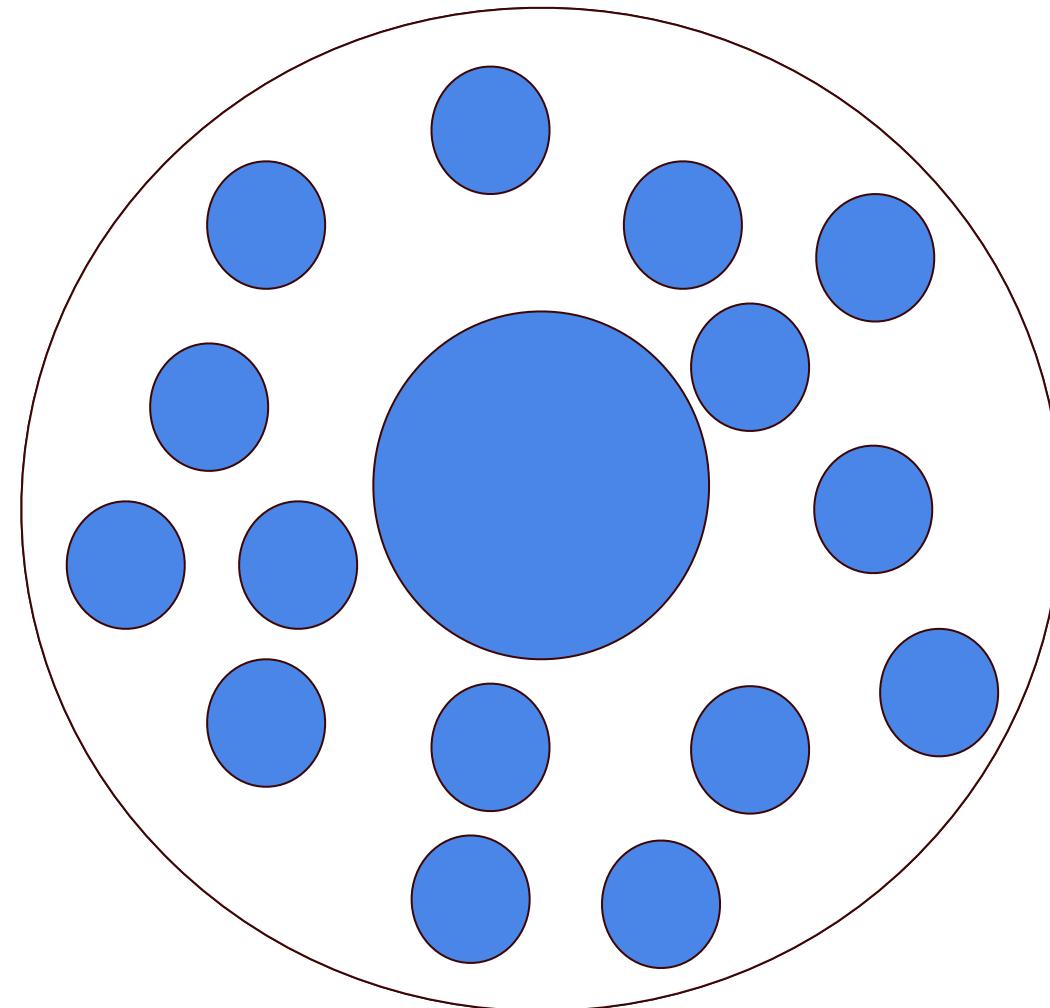
```
[DonutOutputCache(Duration=60, VaryByParam="something;that")]  
public ActionResult Index(string something)
```



- Remember to punch holes
 - Cache not close to the problem?
 - Harder to diagnose performance problems
-



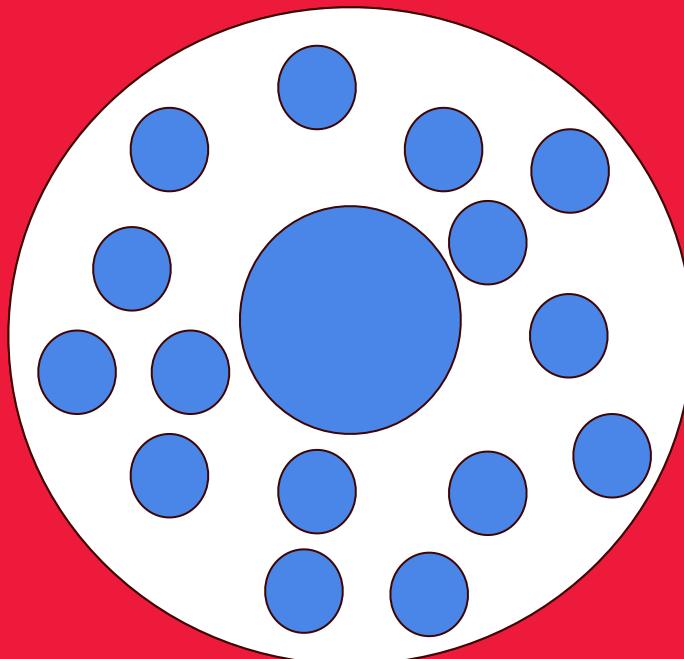
```
<div>@Html.Action("CategoriesList")</div>
```



```
<div>@Html.Action("CategoriesList")</div>
```

```
[ChildActionOnly]
[OutputCache(Duration=60)]
public ActionResult CategoriesList()
{
    // Get categories list from the database and
    // pass it to the child view
    ViewBag.Categories = Model.GetCategories();
    return View();
}
```

- Cache not close to the problem?
- Harder to diagnose performance problems



```
public HomePage GetHomeNode(int nodeId)
{
    var siteNode = GetSiteNode(nodeId);
    var node = siteNode.Children<HomePage>().FirstOrDefault();

    return node;
}
```

```
public HomePage GetHomeNode_Cached(int nodeId)
{
    var cacheKey = "HomePage";

    var node = HttpRuntime.Cache[cacheKey] as HomePage;

    if (node == null)
    {
        var siteNode = GetSiteNode(nodeId);
        node = siteNode.Children<HomePage>().FirstOrDefault();

        HttpRuntime.Cache.Add(cacheKey, node, null, DateTime.Now.AddHours(1),
            Cache.NoSlidingExpiration, CacheItemPriority.Default, null);
    }

    return node;
}
```

- Not single responsibility
- Repeated code... everywhere!
- Encourages sprinkling (like mozzarella)
- Hard to see where cache is being used
- How to turn it off?

```
public HomePage GetHomeNode_Cached(int nodeId)
{
    var cacheKey = "HomePage";

    var node = HttpRuntime.Cache[cacheKey] as HomePage;

    if (node == null)
    {
        var siteNode = GetSiteNode(nodeId);
        node = siteNode.Children<HomePage>().FirstOrDefault();

        HttpRuntime.Cache.Add(cacheKey, node, null, DateTime.Now.AddHours(1),
            Cache.NoSlidingExpiration, CacheItemPriority.Default, null);
    }
}
```

```
public class CmsService : ICmsService  
public class CmsServiceCachedProxy : ICmsService
```

```
public interface ICmsService  
{  
    4 references  
    SiteContainer GetSiteContainer(int nodeId);  
  
    13 references  
    SiteRoot GetSiteNode(int nodeId);  
  
    3 references  
    HomePage GetHomeNode(int nodeId);  
  
    3 references  
    Error404 GetError404Node(int nodeId);  
  
    4 references  
    DataRepositories GetDataRepositoriesNode(int nodeId);  
  
    4 references  
    Taxonomies GetTaxonomiesNode(int nodeId);  
  
    3 references  
    Tags GetTagsNode(int nodeId);  
}
```

```
private static void RegisterCachedServices(Composition composition)
{
    if (ConfigurationHelper.ServiceCacheEnabled())
    {
        composition.Register(typeof(DangEasy.Interfaces.Caching.ICache), typeof(Cache));

        composition.Register(typeof(CmsService), typeof(CmsService));
        composition.Register(typeof(ICmsService), typeof(CmsServiceCachedProxy));
    }
    else
    {
        composition.Register(typeof(ICmsService), typeof(CmsService));
    }
}
```

```
<add key="ServiceCache:Enabled" value="true" />
```

```
public HomePage GetHomeNode(int nodeId)
{
    var siteNode = GetSiteNode(nodeId);
    var node = siteNode.Children<HomePage>().FirstOrDefault();

    return node;
}

public HomePage GetHomeNode(int nodeId)
{
    var siteNode = GetSiteNode(nodeId); // use site node id in cache key (for multi-site instances)
    var cacheKey = CacheKey.Build<cmsServiceCachedProxy, HomePage>(siteNode.Id.ToString());

    return _cache.Get(cacheKey, () => _cmsService.GetHomeNode(siteNode.Id));
}
```

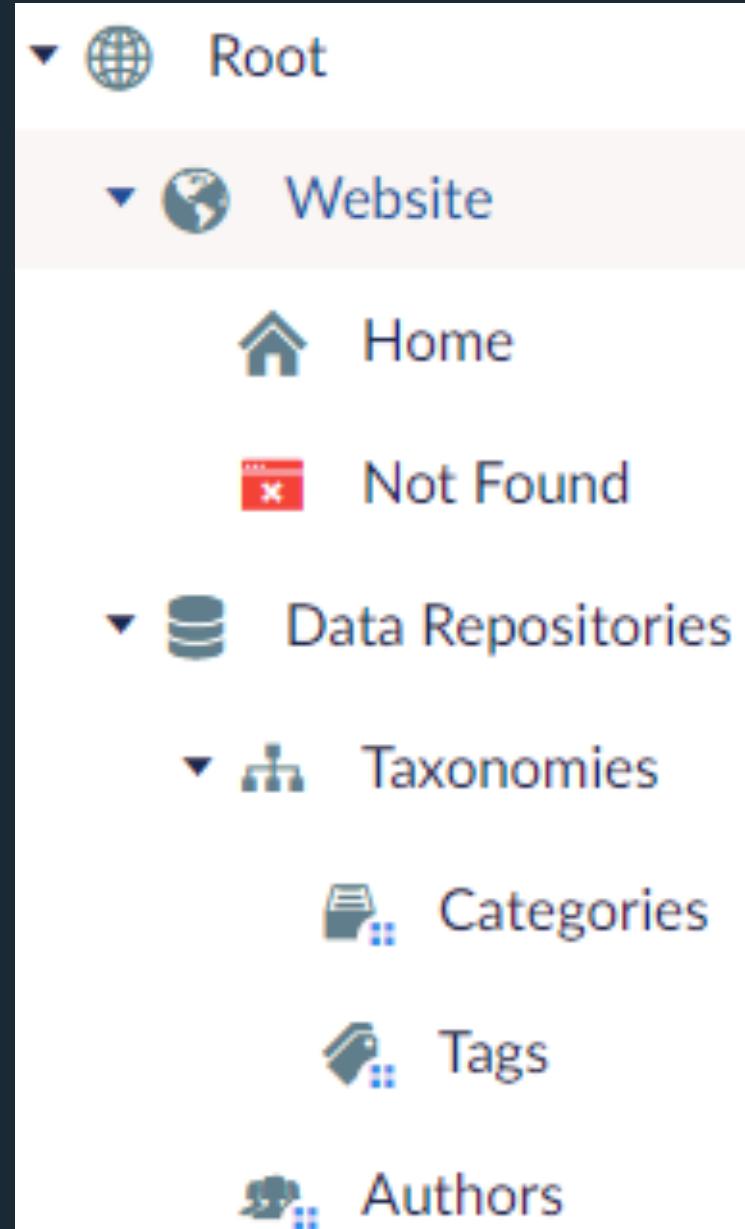
```
private void ContentCacheRefresher_CacheUpdated(ContentCacheRefresher sender,
| Umbraco.Core.Cache.CacheRefresherEventArgs e)
{
| Cache.Instance.RemoveByPrefix(typeof(CmsServiceCachedProxy).ToString());
}
```

0 references

```
public void Initialize()
{
| ContentCacheRefresher.CacheUpdated += ContentCacheRefresher_CacheUpdated;
}
```

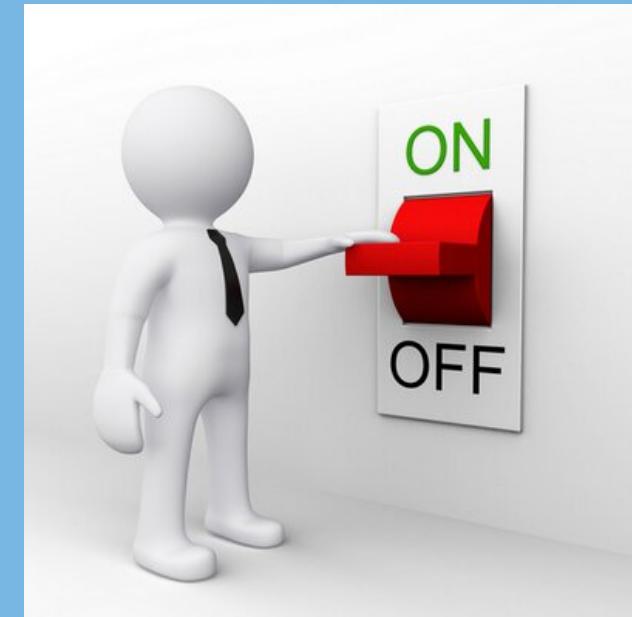
- Common good Umbraco Practices
- Not a starter kit
- Doctypes
- Patterns & Practices
- Cache IoC

github.com/anthonydotnet/ubase



- Keep it simple
- Avoid overuse of design patterns
- Avoid descendants / ancestors
- Cache only where needed
- Good practices change over time
- Ask & Share

github.com/anthonydotnet/code-guide



Thanks!

radley yeldar.

Cache me on Twitter
@anthonydotnet

Git Kraken: anthonyd19

@anthonyd19

Thanks!

radley yeldar.

Cache me on Twitter
@anthonydotnet

Git Kraken: anthonyd19

@anthonyd19.net