



Especial Barcelona
Miércoles 10 de Abril

Yet Another v8 Talk

Stéphane Gay

Repairman Of All Things Completely Broken™

Umbraco HQ

@zpqrtbnk





v8 is LIVE

... and it works ...

umbraco

The background of the entire image is a close-up, high-contrast photograph of flames. The fire is bright orange and yellow, with dark, swirling patterns of smoke and ash visible within the fire itself. The flames are rising and dancing, creating a sense of intense heat and movement. The overall effect is dramatic and somewhat ominous, especially given the text overlay.

8.365.112 SITES

Could be running Umbraco v8 in a distant future,
provided that all other CMSes give up the fight
and Umbraco becomes the only remaining CMS
(number may vary)

v8 hecho bien

Stéphane Gay

Repairman Of All Things Completely Broken™

Umbraco HQ

@zpqrtbnk



umbraco

Variants

101

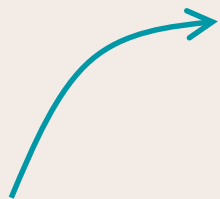
One Document – Several Cultures

Document

FR: "bonjour" → /bonjour

EN: "hello" → /hello

example.com/fr → FR
example.com/en → EN



Document

FR: "bonjour" → /fr/bonjour

EN: "hello" → /en/hello

```
var title = Model.Value("title");
```

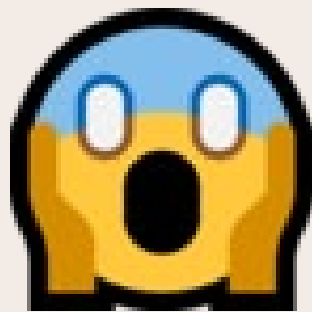
```
var title = Model.Value("title", "fr-FR");
```

umbraco

Composition

~

Dependency Injection



Composer

Where Everything Begins

```
public class MyComposer : IComposer
{
    public void Compose(Composition composition)
    {
        // ...
    }
}
```

```
// set the last chance content finder  
composition.SetContentLastChanceFinder<MyLastChanceFinder>();  
  
// append a finder to the pipeline  
composition.ContentFinders().Append<MyContentFinder>();
```

```
// register  
composition.Register<ISomething, Something>();
```



```
// register  
composition.RegisterFor<ISomething, Something>();
```



Predictable Composition

Less Magic

```
[RuntimeLevel(MinLevel = RuntimeLevel.Install)]  
[ComposeAfter(typeof(MyOtherComposer))]  
[Disable(typeof(OtherComposer))]  
public class MyComposer : IUserComposer  
{  
    public void Compose(Composition composition)  
    {  
        // ...  
    }  
}
```

Component

Bring Your Own Code

```
public class MyComponent : IComponent
{
    public void Initialize()
    {
        // ...
    }

    public void Terminate()
    {
        // ...
    }
}
```

```
public class MyComponent : IComponent
{
    private readonly IContentService _contentService;

    public MyComponent(IContentService contentService)
    {
        _contentService = contentService;
    }
}
```

```
public void Compose(Composition composition)
{
    composition.Components().Append<MyComponent>();
}
```



umbraco



umbraco

Context

Is Everything

```
private readonly IUmbracoContextFactory _contextFactory;

public MyComponent(IUmbracoContextFactory contextFactory)
{
    _contextFactory = contextFactory;
}

public void Initialize()
{
    using (var contextRef = _contextFactory.EnsureUmbracoContext())
    {
        var context = contextRef.UmbracoContext;

        // ...
        var item = context.ContentCache.GetById(1234);
    }
}
```



umbraco

The Database

Where ?!

umbraco

Use The Scope

Luke

```
private readonly IScopeProvider _scopeProvider;

public MyComponent(IScopeProvider scopeProvider)
{
    _scopeProvider = scopeProvider;
}

public void Initialize()
{
    using (var scope = _scopeProvider.CreateScope())
    {
        scope.Database.Execute("DELETE * FROM umbracoNode;");

        scope.Complete();
    }
}
```

umbraco

DI.Die.Die.Die

I Don't Wanna DI


```
public void Whatever()  
{  
    Current.Logger.Debug<MyClass>("Hello!");  
}
```

```
public void Whatever()  
{  
    var service = Current.Factory.GetInstance<IMyService>();  
}
```

umbraco

Published Elements

Lightweight Published Content

umbraco

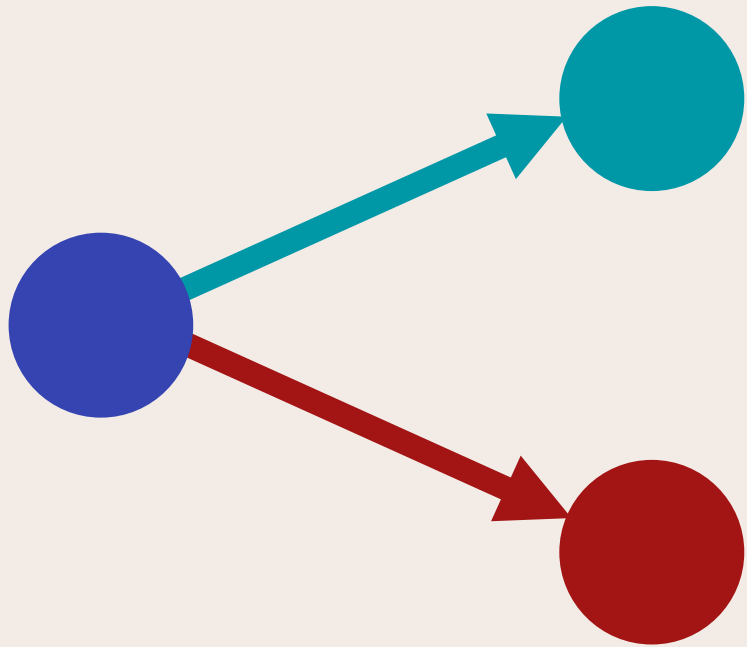
Migrations

Plan

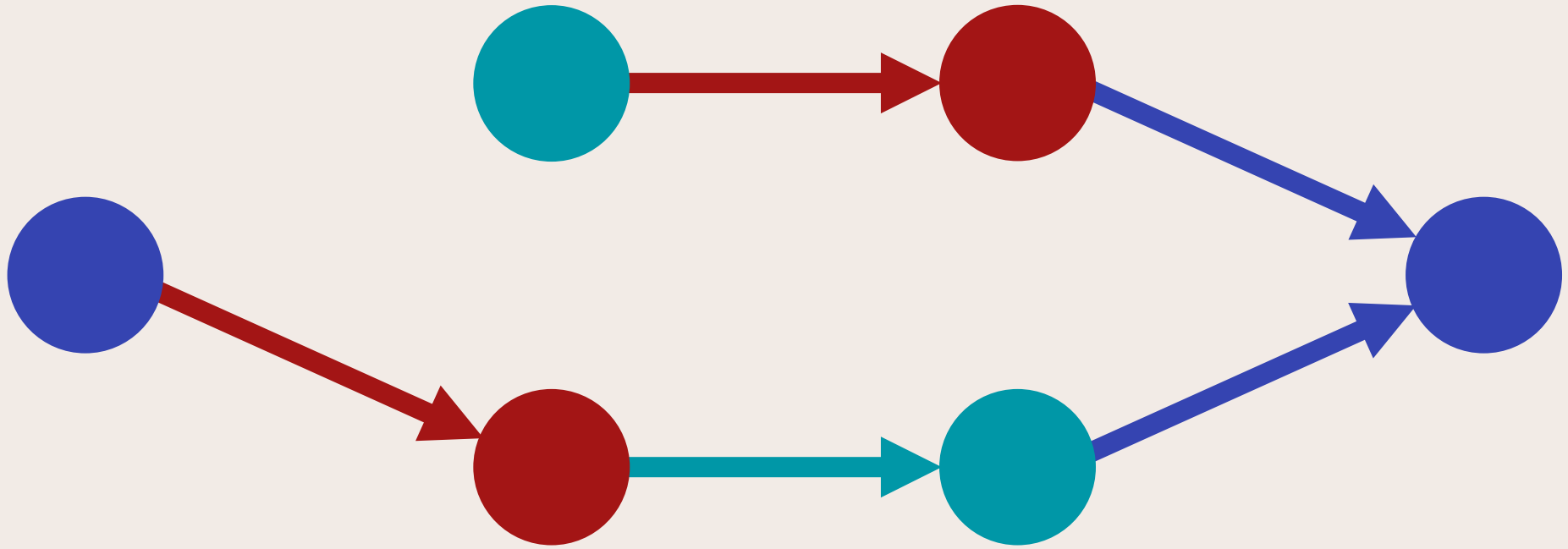


umbraco

```
public class MyPlan : MigrationPlan
{
    public MyPlan()
        : base("myPlan")
    {
        From("")
            .To<MyMigration1>("state-1")
            .To<MyMigration2>("state-2");
    }
}
```



umbraco



umbraco


```
...  
    .To<MyMigration2>("state-2")  
    .Merge()  
        .To<Dev1Migration>("dev-1-state")  
    .With()  
        .To<Dev2Migration>("dev-2-state")  
    .As("state-3");
```

```
var plan = new MyPlan();  
var upgrader = new Upgrader(plan);  
upgrader.Upgrade(...);
```

umbraco

Migrating From v7

Have You Found The Way?

umbraco

One More Thing

What Is Next?

```
var title = Model.Value("title");
```

```
var title = Model.Value("title", "fr-FR");
```

```
// use the property  
var children = Model.Children;  
  
// use the method  
var children = Model.Children();  
  
// use the method, with a culture  
var children = Model.Children("fr-FR");
```

```
// current culture
var name = Model.Name;

// FR culture
var nameFR = Model.GetCulture("fr-FR").Name;

// Can I has this?
var nameFR = Model.Name("fr-FR");
```

Fixing Is Breaking



umbraco

The Friendliest CMS Community On The Planet

Make sure to voice your opinion
Issues are monitored

Get what you want!





Thank you



Especial Barcelona
Miércoles 10 de Abril