ARLANET

DeninaSharp for Umbraco

LENNARD FONTEIJN

Senior .NET Developer at Arlanet

Umbraco since version 4
Little bit of Episerver here and there

LE NARD FONTAINE

Senior .NET Developer at Arlanet

Umbraco since version 4
Little bit of Episerver here and there

LENNARD FONTEIJN

IT Lecturer at the Amsterdam University of Applied Scienes

Raising a new generation of developers

LENNARD FONTEIJN

Game Developer

Semi-professional in whatever time is left

LENNARD FONTEIJN

Mad scientist

Because I apparantly also find the time to do weird stuff like this.

LENNARD FONTEIJN

Mad scientist

Because I apparantly also find the time to do weird stuff like this.

With all due respect, Deane ;)

ME AND DENINASHARP

- Met Deane at last years CodeGarden

- Intrigued by his passion

- Wanted to collaborate

WARNING: THIS IS NOT A DRILL!

I will actually be showing code, which might not be for the faint hearted.

WHAT DID I MAKE?

- Took Deane's script-based version…

- … and slapped an UI on it!

- Because that's how we do that the Umbraco-way.

ONE WORD

- Reflection!
  - Because we can.

- In all seriousness…
  - … it was actually already there.
  - Because Deane is equally mad as me apparantly.

```
[Filter("Extract", "Retieves the valud at a defined \"path\" into the JSON
object.")]
[ArgumentMeta("path", true, "A valid JSON path, as accepted by the Newtonsoft
JObject.SelectToken method (see:
http://www.newtonsoft.com/json/help/html/SelectToken.htm).")]
[CodeSample("resource:json-extract-input.json", "Json.Extract -path:person.name",
"James Bond")]
[Requires("Newtonsoft.Json.Linq.JObject, Newtonsoft.Json", "This is in the JSON.NET
Nuget package.")]
[Requires("Newtonsoft.Json.Linq.JToken, Newtonsoft.Json", "This is in the JSON.NET
Nuget package.")]
public static string ExtractFromJson(string input, PipelineCommand command,
ExecutionLog log)
{
    var path = command.GetArgument(key: "path");

    var jobject = (JToken)JObject.Parse(json: input);

    return (string)jobject.SelectToken(path: path);
}
```

SO WHY DO WE WANT THIS?

- Because it's fun, mostly.

- But also, because the possibilities are infinite™!

- Imagine…

SMALL RECAP

- Lots of possibilities for extension
  - Re-usable command-blocks
  - Configurable permissions in Umbraco
  - New commands/filters

- Good showcase of supporting both U7 and U8
  - Also nice example of using as much out-of-the-box Umbraco-stuff as possible.

ROADMAP

- Now
    - Wrap up loose ends
        - Grid in Umbraco 8 is untested.
    - NuGet packages
    - Umbraco Package page

- Next
    - Separation of existing filters from DeninaSharp Core
    - Better exception-handling for editorial purposes

- Later
    - DeninaSharp to .NET Core / Standard
        - Actually about 99% is already .NET Core 2.0 compatible.
    - Determine where to take things…

WE NEED YOU!

- Everything you saw is available right now!

- See for yourself at https://github.com/ lennardf1989/DeninaSharp.Umbraco

  - Or: https://tinyurl.com/deninasharp

WHAT DO WE NEED?

- Get to know where people want to use this for. Share stories!

- Report issues, make PR's.

- Discuss where we should take things to make this an unmissable tool for both developers and editors.

THANK YOU FOR YOUR ATTENTION!

Questions?