Templating in Vapor

The boring part

— Nathan / @zeitschlag 🤓

— By day: iOS Engineer @ T-Systems MMS 👨‍💻

— By night: Shredlog, Einkaufszettl, bullenscheisse.de 👨‍💻, Batman

Agenda

1. Templating in General
2. Templating in Vapor
3. Leaf
    1. Installation
    2. Basics
    3. Tags
4. Other Resources

Templating in General

- Replace certain keywords in a template with values
- Replacement follows certain rules
- Web App: Template + Data = HTML
- There's a language for templating:
  - Mustache, Stencil, Django Template Language...

Templating in Vapor

- Template + Data = HTML
- Brings his own Templating Language called Leaf
- But: BYOT[1]

[1] Bring your own Templating Engine, if you write your own provider

Leaf

> *Leaf's goal is to be a simple templating language that can make generating views easier. [...] The goals of Leaf are as follows:*
> *- Small set of strictly enforced rules*
> *- Consistency*
> *- Parser first mentality*
> *- Extensibility*
>
> — Leaf Documentation

Installation

- Business as usual:
  - Add LeafProvider-package as dependency to `Package.swift`
  - Add Provider.
  - Configure Droplet
  - Done.

Basics

- Leaf-files are in a certain folder: 'Resources/Views'
- You write the template files (name.leaf) using tags

```
return try drop.view.make("name")
return try drop.view.make("name", context) // context is
a NodeRepresentable
```

Tags

- Parts to be replaced
- Syntax: `# Name (Parameter-List) {optional Body}`[2]
- ##Chaining
- Several built-in tags like #(variable), #if() ##else , #loop(collection, "name"), #raw(), #()… [3]
- Layout tags: #embed, #extend, #import, #export

[2] There needs to be a space between ')' and '{'

[3] Leaf Documentation

Thank you!

Questions[4]?

[4] (No, there's no Syntax Highlighting for Xcode)

Other Resources

- slides/code on github
- Leaf Documentation
- Ray Wenderlich - Server Side Swift with Vapor: Templating with Leaf