Angular Scaffolding

James Kleeh

Grails Core Team Member

github.com/jameskleeh

OCI | HOME TO GRAILS

Goals of This Talk

What is Scaffolding Core?
What can I do with Scaffolding Core?
What does the Angular scaffolding provide?
How can I customize the output?

Scaffolding Core

The Angular scaffolding is an extension of scaffolding core.

What is scaffolding core?

Scaffolding core is a library that provides a set of services to render markup based on a domain class and its properties.

https://github.com/grails/scaffolding

Scaffolding Core

Renders markup based on a domain class definition.

Markup is designed to be used by a further post processor (GSP, Handlebars, Angular, etc.)

Scaffolding Core - Definitions

Definitions:

- Input: Markup for capturing user input. Examples `<input />, <select />, <textarea />`
- Output: Markup/Text for displaying data in the markup. Examples `${data}, {{data}}`
- Context: Markup that surrounds inputs and outputs.

Scaffolding Core - Definitions

Practical Examples of the definitions: Input

```
<form class="form-inline">
    <div class="form-group">
        <label for="exampleName2">Name</label>
        <input type="text" class="form-control" id="exampleName2">
    </div>
    <div class="form-group">
        <label for="exampleEmail2">Email</label>
        <input type="email" class="form-control" id="exampleEmail2">
    </div>
    <fieldset>
        <legend>Address</legend>
        <div class="form-group">
            <label for="city">City</label>
            <input type="text" class="form-control" id="city">
        </div>
    </fieldset>
    <button type="submit" class="btn btn-default">Send invitation</button>
</form>
```

Scaffolding Core - Definitions

Domain Input Context

```
<form class="form-inline">
    <div class="form-group">
        <label for="exampleName2">Name</label>
        <input type="text" class="form-control" id="exampleName2">
    </div>
    <div class="form-group">
        <label for="exampleEmail2">Email</label>
        <input type="email" class="form-control" id="exampleEmail2">
    </div>
    <fieldset>
        <legend>Address</legend>
        <div class="form-group">
            <label for="city">City</label>
            <input type="text" class="form-control" id="city">
        </div>
    </fieldset>
    <button type="submit" class="btn btn-default">Send invitation</button>
</form>
```

Scaffolding Core - Definitions

Property Input Context

```
<form class="form-inline">
    <div class="form-group">
        <label for="exampleName2">Name</label>
        <input type="text" class="form-control" id="exampleName2">
    </div>
    <div class="form-group">
        <label for="exampleEmail2">Email</label>
        <input type="email" class="form-control" id="exampleEmail2">
    </div>
    <fieldset>
        <legend>Address</legend>
        <div class="form-group">
            <label for="city">City</label>
            <input type="text" class="form-control" id="city">
        </div>
    </fieldset>
    <button type="submit" class="btn btn-default">Send invitation</button>
</form>
```

Scaffolding Core - Definitions

Embedded Input Context

```
<form class="form-inline">
    <div class="form-group">
        <label for="exampleName2">Name</label>
        <input type="text" class="form-control" id="exampleName2">
    </div>
    <div class="form-group">
        <label for="exampleEmail2">Email</label>
        <input type="email" class="form-control" id="exampleEmail2">
    </div>
    <fieldset>
        <legend>Address</legend>
        <div class="form-group">
            <label for="city">City</label>
            <input type="text" class="form-control" id="city">
        </div>
    </fieldset>
    <button type="submit" class="btn btn-default">Send invitation</button>
</form>
```

Scaffolding Core - Definitions

Input

```
<form class="form-inline">
    <div class="form-group">
        <label for="exampleName2">Name</label>
        <input type="text" class="form-control" id="exampleName2">
    </div>
    <div class="form-group">
        <label for="exampleEmail2">Email</label>
        <input type="email" class="form-control" id="exampleEmail2">
    </div>
    <fieldset>
        <legend>Address</legend>
        <div class="form-group">
            <label for="city">City</label>
            <input type="text" class="form-control" id="city">
        </div>
    </fieldset>
    <button type="submit" class="btn btn-default">Send invitation</button>
</form>
```

Scaffolding Core

Practical Examples of the definitions: Output

```
<div class="container">
    <div class="row">
        <div class="col-sm-2">Name</div>
        <div class="col-sm-10">
            ${user.name}
        </div>
        <div class="col-sm-2">Age</div>
        <div class="col-sm-10">
            ${user.age}
        </div>
        <div class="col-sm-2">Address</div>
        <div class="col-sm-10">
            <div class="row-fluid">
                <div class="col-sm-2">City</div>
                <div class="col-sm-10">
                    ${user.address.city}
                </div>
            </div>
        </div>
    </div>
</div>
```

Scaffolding Core

Domain Output Context

```
<div class="container">
    <div class="row">
        <div class="col-sm-2">Name</div>
        <div class="col-sm-10">
            ${user.name}
        </div>
        <div class="col-sm-2">Age</div>
        <div class="col-sm-10">
            ${user.age}
        </div>
        <div class="col-sm-2">Address</div>
        <div class="col-sm-10">
            <div class="row-fluid">
                <div class="col-sm-2">City</div>
                <div class="col-sm-10">
                    ${user.address.city}
                </div>
            </div>
        </div>
    </div>
</div>
```

Scaffolding Core

Property Output Context

```
<div class="container">
    <div class="row">
        <div class="col-sm-2">Name</div>
        <div class="col-sm-10">
            ${user.name}
        </div>
        <div class="col-sm-2">Age</div>
        <div class="col-sm-10">
            ${user.age}
        </div>
        <div class="col-sm-2">Address</div>
        <div class="col-sm-10">
            <div class="row-fluid">
                <div class="col-sm-2">City</div>
                <div class="col-sm-10">
                    ${user.address.city}
                </div>
            </div>
        </div>
    </div>
</div>
```

Scaffolding Core

Embedded Output Context

```
<div class="container">
    <div class="row">
        <div class="col-sm-2">Name</div>
        <div class="col-sm-10">
            ${user.name}
        </div>
        <div class="col-sm-2">Age</div>
        <div class="col-sm-10">
            ${user.age}
        </div>
        <div class="col-sm-2">Address</div>
        <div class="col-sm-10">
            <div class="row-fluid">
                <div class="col-sm-2">City</div>
                <div class="col-sm-10">
                    ${user.address.city}
                </div>
            </div>
        </div>
    </div>
</div>
```

Scaffolding Core

Output

```
<div class="container">
    <div class="row">
        <div class="col-sm-2">Name</div>
        <div class="col-sm-10">
            ${user.name}
        </div>
        <div class="col-sm-2">Age</div>
        <div class="col-sm-10">
            ${user.age}
        </div>
        <div class="col-sm-2">Address</div>
        <div class="col-sm-10">
            <div class="row-fluid">
                <div class="col-sm-2">City</div>
                <div class="col-sm-10">
                    ${user.address.city}
                </div>
            </div>
        </div>
    </div>
</div>
```

Scaffolding Core

List Output

```
<table>
    <thead>
        <tr>
            <th>Name</th>
            <th>Age</th>
            <th>City</th>
    </thead>
    <tbody>
        <g:each var="user" in="${userList}">
            <tr>
                <td>${user.name}</td>
                <td>${user.age}</td>
                <td>${user.address.city}</td>
            </tr>
        </g:each>
    </tbody>
</table>
```

Scaffolding Core - The API

DomainModelService: Defines which domain properties should be rendered

- Inspects constraints for display: false
- Sensible defaults for excluded properties
- Returns instances of DomainProperty

Scaffolding Core - The API

DomainProperty: Combines the PersistentEntity API and the Constraints API.

Provides access to:

- Domain class
- Constraints
- Associated entity (If it's an association)
- Helper methods for labels, etc

Scaffolding Core - The API

ContextMarkupRenderer: Responsible for rendering the different contexts.

PropertyMarkupRenderer: Renders domain properties by looking up input/output renderers in a registry.

DomainMarkupRenderer: The front facing API that calls everything else to render the full markup.

Scaffolding Core - The API

Input / Output Registries: Store instances of DomainInputRenderer and DomainOutput renderers.

DomainInputRenderer: Renders a domain input based on a DomainProperty and a map of default attributes.

DomainOutputRenderer: Renders a domain output based on a DomainProperty.

Angular Scaffolding

Generates:

- Templates
- One or many modules
- Controllers
- Domain (`$resource`)
- Module Dependencies
- Associated Modules

Angular Scaffolding

Expects:

- Asset Pipeline & Plugins
  - Angular Annotate
  - Angular Template
  - Closure Wrap
- Angular
- Angular Resource
- UI-Router

Angular Scaffolding

Supports:

- Associations
  - One To Many
  - One To One
- Currency
- TimeZone
- byte[]
- “Simple” types (String, Number, Boolean, Date, etc)

Angular Scaffolding

Demo

Angular Scaffolding

Customization Demo

Angular Scaffolding

Q&A

Thank You!

OCI | HOME TO GRAILS