WEBINAR

Single Page Apps with Grails & Vue.js

Zachary Klein, Senior Software Engineer

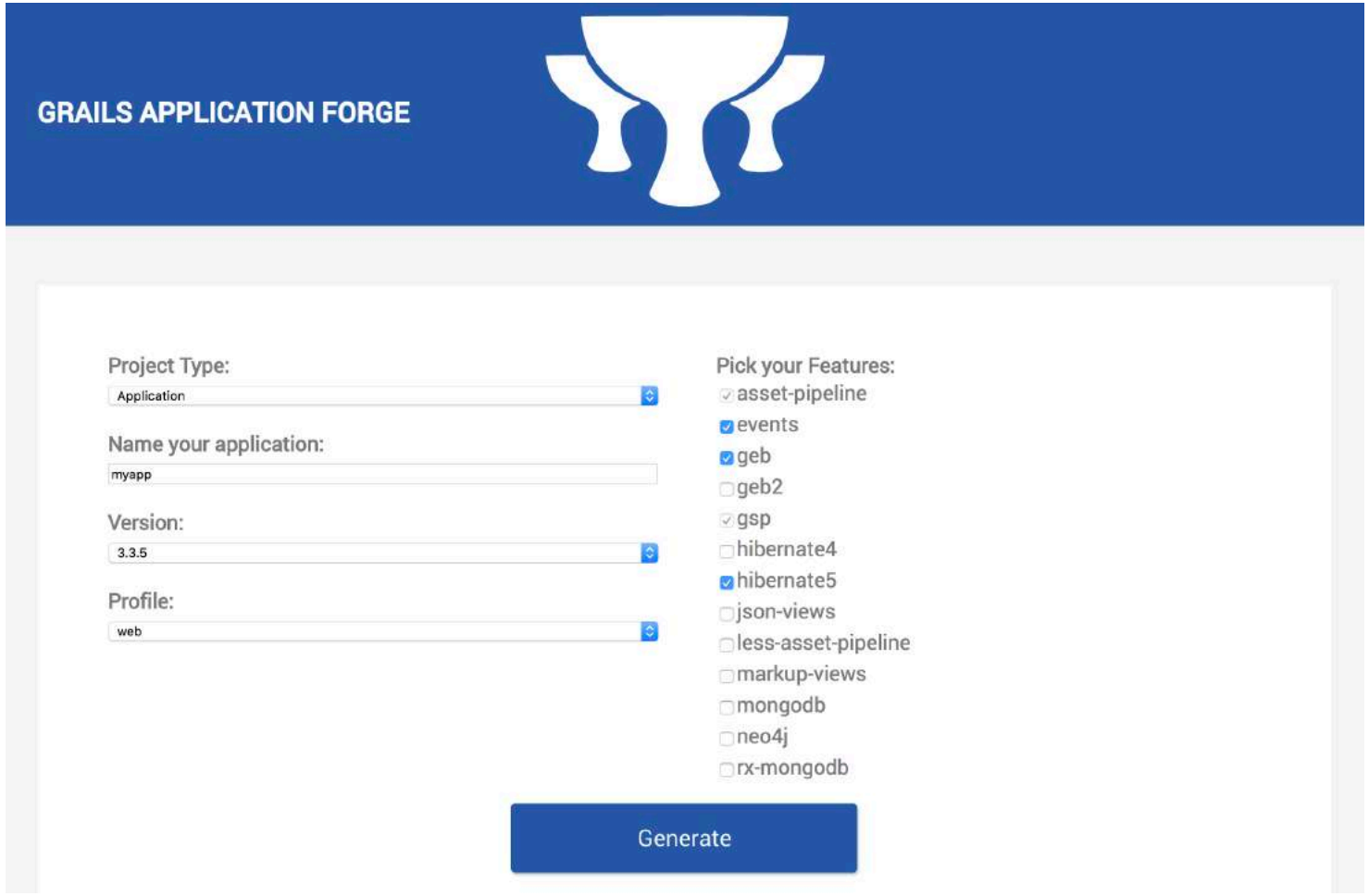
© 2019, Object Computing, Inc. (OCI). All rights reserved. No part of these notes may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior, written permission of Object Computing, Inc. (OCI)

Speaker

Zachary Klein is a Senior Software Engineer at OCI. He has been practicing web development since 2010 and frontend development since 2015. He's a contributor to both the Grails and Micronaut frameworks, a conference speaker and an instructor in OCI's training practice. Zachary's home base is in St Louis, MO, along with his wife, Beth, and their three children.

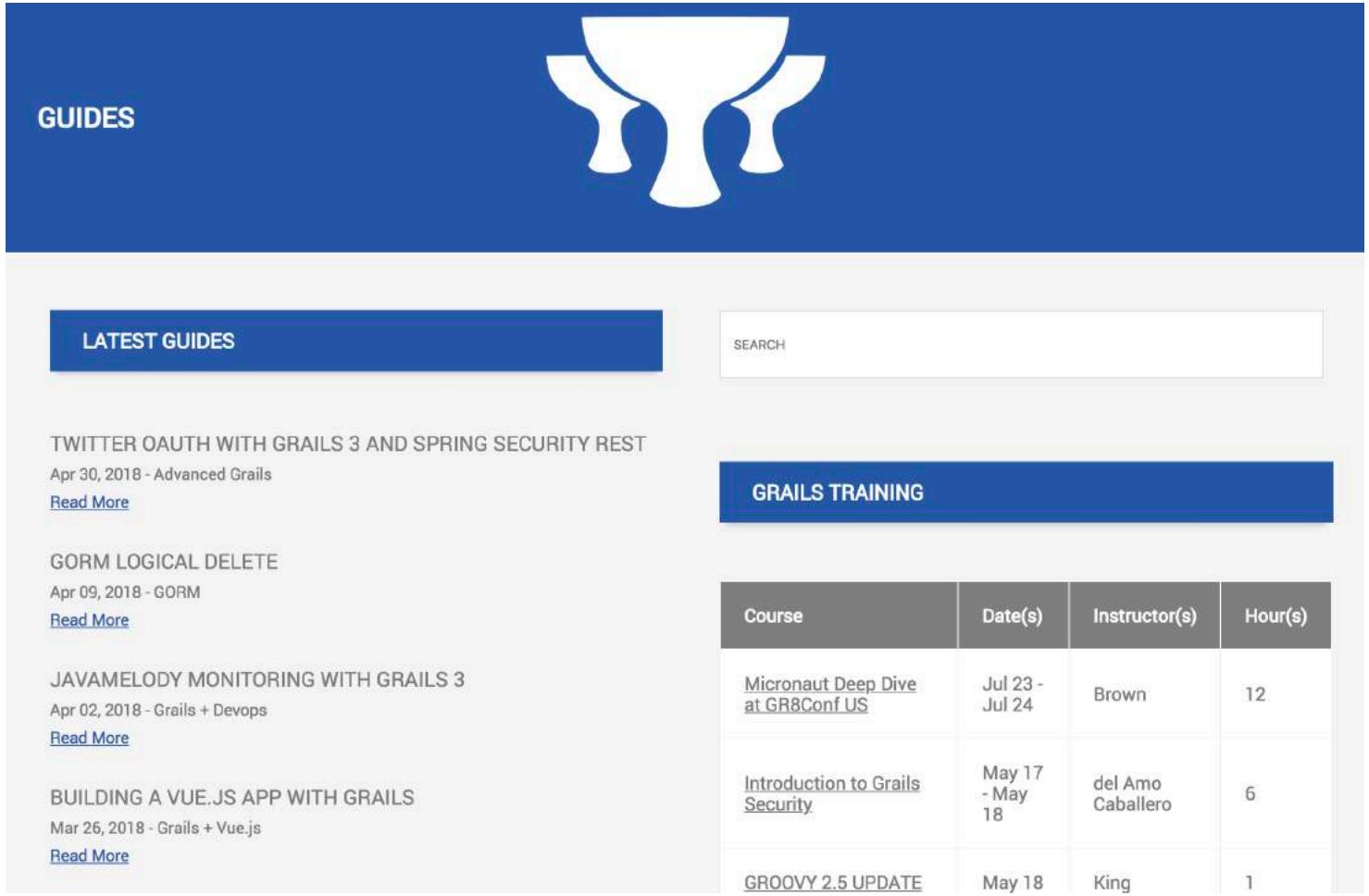
© 2019, Object Computing, Inc. (OCI). All rights reserved.

Grails Application Forge: http://start.grails.org



© 2019, Object Computing, Inc. (OCI). All rights reserved.

Grails Guides: http://guides.grails.org



© 2019, Object Computing, Inc. (OCI). All rights reserved.

Introduction to Vue.js

© 2019, Object Computing, Inc. (OCI). All rights reserved.

What is Vue?

OBJECT COMPUTING

- Small, performant view library
- Mature ecosystem
- Powerful devtools
- Active community
- vs Angular vs React…

© 2019, Object Computing, Inc. (OCI). All rights reserved.

Getting Started with Vue

OBJECT COMPUTING

- Install vue-cli (v3):
  1. `npm install -g @vue/cli`
  2. vue create my-project

© 2019, Object Computing, Inc. (OCI). All rights reserved.

Getting Started with Vue

```
<script src="https://unpkg.com/vue"></script>

<div id="app">
  {{ text }}
</div>

<script>

  var app = new Vue({
    el: '#app',
    data: {
      text: 'Hello World!'
    }
  })

</script>
```

© 2019, Object Computing, Inc. (OCI). All rights reserved.

The Vue Instance

- Ground zero for a Vue.js app
- Accepts an object containing the Vue **Instance definition**
- Most definitions shared between instance and **components**

Vue Instance

```
new Vue({
  el: '#app',
  data: {...},
  computed: {...},
  methods: {...},
  template: '<App/>'
})
```

Vue Component

```
export default {
  name: 'component',
  data () { return {...} },
  methods: {...}
}
```

© 2019, Object Computing, Inc. (OCI). All rights reserved.

The Vue Instance

- `el` - Specifies the DOM element where the app will be rendered

```
new Vue({
  el: '#app'
})
```

```
<div id="app"></div>
```

© 2019, Object Computing, Inc. (OCI). All rights reserved.

The Vue Instance

- `data` - An object describing the state of the app/component
  - An object describing the state of the app/component
  - Properties are subject to reactive rendering within the app/component (one-way & two-way)
  - New properties can be added, but will not be subject to reactive behavior
  - Components must use a **data()** function that returns state

© 2019, Object Computing, Inc. (OCI). All rights reserved.

The Vue Instance

- `data` - An object for standalone Vue instances, function for components

Vue Instance

```
new Vue({
  data: {
    myValue: 3,
    myObject: {
      prop: 'abc'
    }
  }
})
```

Vue Component

```
export default {
  data () {
    return {
      myValue: 3,
      myObject: {
        prop: 'abc'
      }
    }
  }
}
```

© 2019, Object Computing, Inc. (OCI). All rights reserved.

The Vue Instance

- `methods` - Arbitrary functions that can access/manipulate data, and be called from templates or other methods.
- `computed` - Functions that return dynamic (cacheable) values and are accessed as data variables.
- **Lifecycle Methods -** Functions that are called at specific points in the app/component lifecycle (e.g, `created`)

© 2019, Object Computing, Inc. (OCI). All rights reserved.

Templates

```
<template>
  <div class="form">
    <div class="title cell">
      <label>Title</label>
      <input ref="bookTitle" v-model="book.title" type="text"/>
    </div>
    <div class="pages cell">
      <label>Pages</label>
      <input v-model="book.pages" type="text"/>
    </div>
    <div class="author cell">
      <label>Author</label>

      <select v-model="book.author">
        <option disabled selected value="">Choose Author</option>
        <option v-if="author !== null"
                v-bind:value="{ id: author.id }"
                v-for="author in authors">{{author.name}}</option>

      </select>
    </div>
    <div class="save cell">
      <button @click="submitNewBook()" >Add Book</button>
    </div>
  </div>
</template>
```

© 2019, Object Computing, Inc. (OCI). All rights reserved.

Components (*.vue files)

- Nested component hierarchy
- Each component renders either a template or returns createElement() calls (JSX is supported)
- Components typically defined as **Single File Component** with <**template**>, <**script**>, and <**style**> sections

HELLOWORLD.VUE

```
<template>
  <div class="hello">
    <h1>{{ msg }}</h1>
  </div>
</template>
```

Template Section

```
<script>
export default {
  name: 'HelloWorld',
  data () {
    return {
      msg: 'Welcome to Your Vue.js App'
    }
  }
}
</script>
```

Script Section

```
<style scoped>
h1 {
  font-color: blue;
}
</style>
```

Style Section

© 2019, Object Computing, Inc. (OCI). All rights reserved.

Demo

© 2019, Object Computing, Inc. (OCI). All rights reserved.

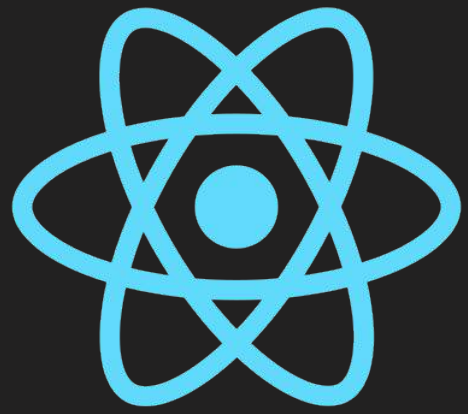
Vue vs Angular

- A cleaner, simpler way to build apps
- Similar concepts/syntax: directives, templates, v-model
- More understandable, readable code
- A rich, featureful API that you can grow into
- Not a framework - you'll still need to add packages

© 2019, Object Computing, Inc. (OCI). All rights reserved.

Vue vs React

- Adds a bit more magic
- Similar conceptually: components, virtual DOM, lifecycle methods
- Supports JSX, createElement API, render functions
- Less code… sometimes
- Favors pragmatism over “pure” simplicity



© 2019, Object Computing, Inc. (OCI). All rights reserved.

Vue Profile for Grails

© 2019, Object Computing, Inc. (OCI). All rights reserved.

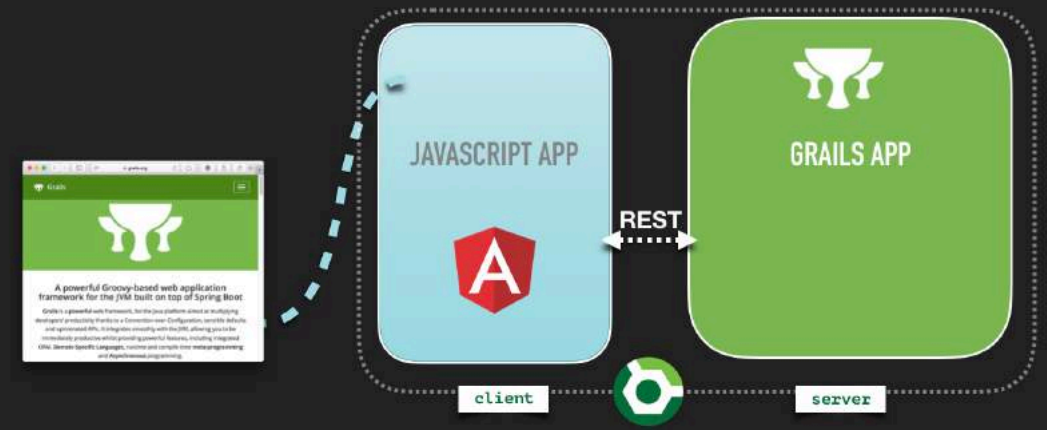
Why Use Vue (and other JavaScript frameworks) with Grails?

- Convention over Configuration
- Spring Boot compatibility
- Profiles & Plugins
- Gradle
- RESTful controllers & URL mappings
- JSON Views
- GORM (including GraphQL)

© 2019, Object Computing, Inc. (OCI). All rights reserved.

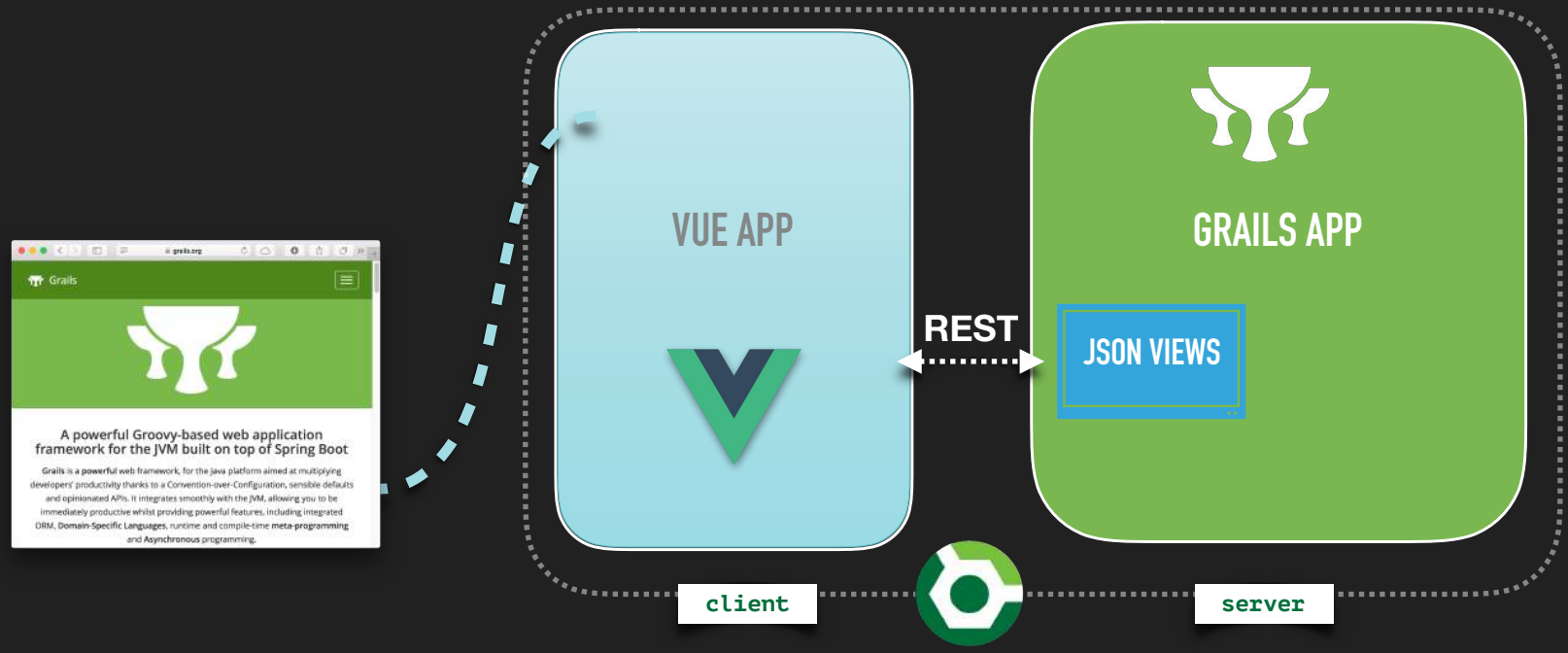
Approaches to using JavaScript Frameworks with Grails

Asset Pipeline **Hybrid Webapp**



© 2019, Object Computing, Inc. (OCI). All rights reserved.

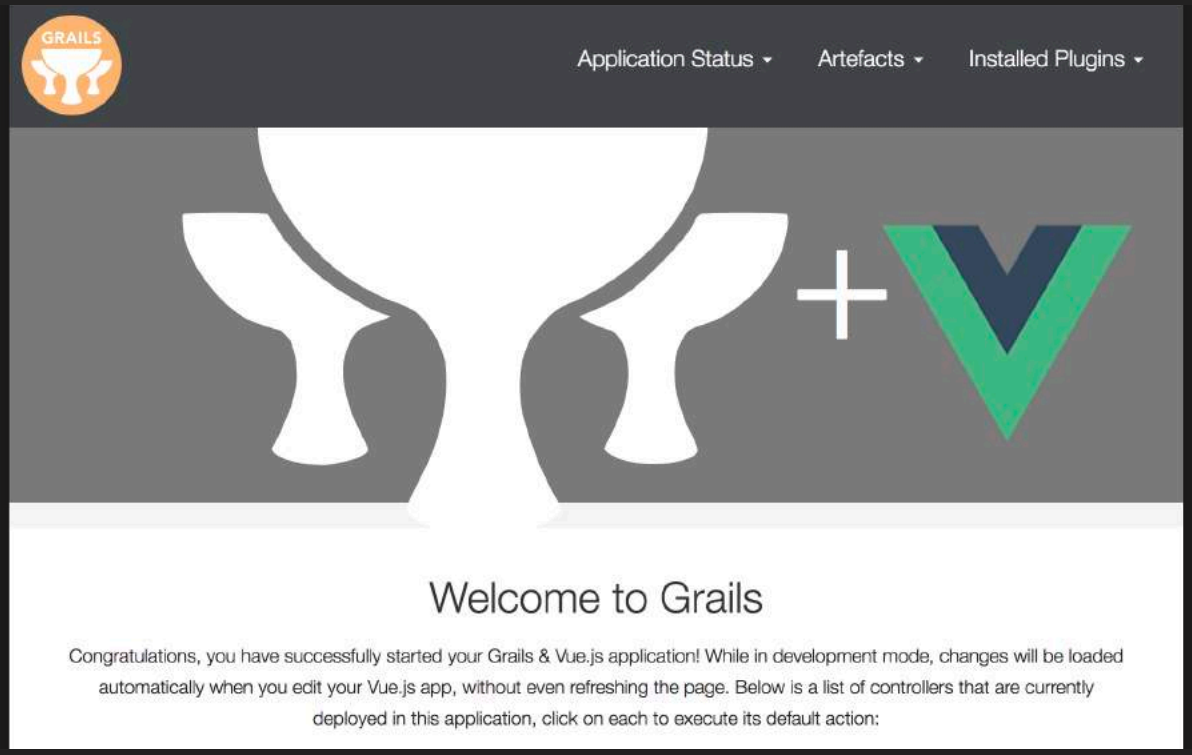
Multi-project Build Approach



© 2019, Object Computing, Inc. (OCI). All rights reserved.

Vue Profile for Grails

https://grails-profiles.github.io/vue/latest/guide/index.html

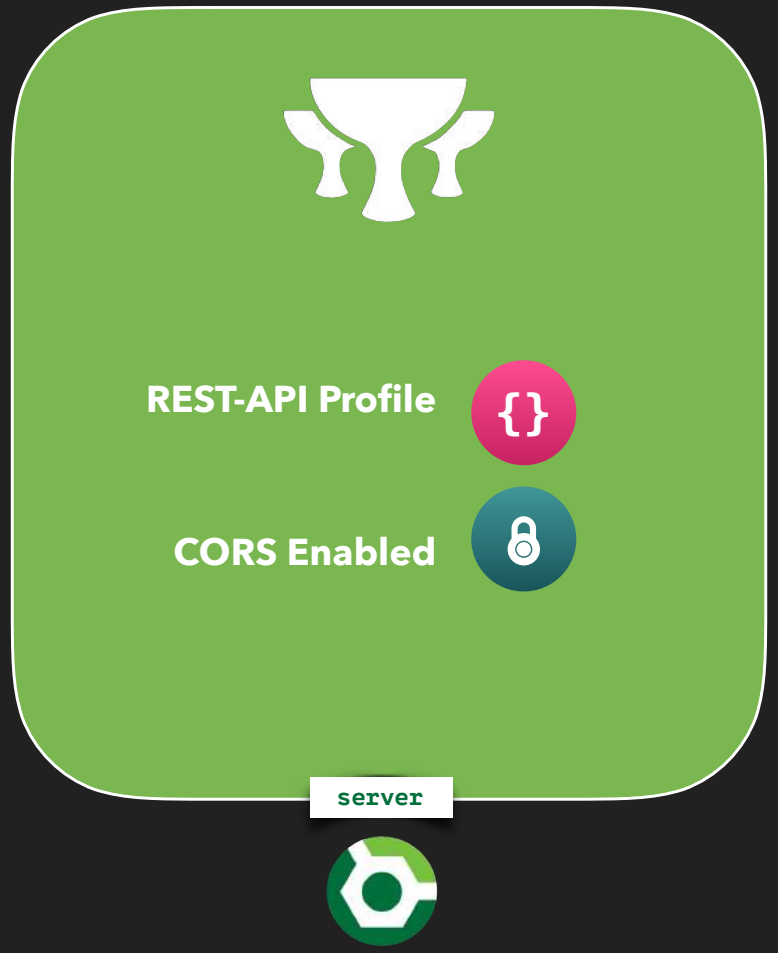


> grails create-app myapp -profile vue

> curl -O start.grails.org/myapp.zip -d profile=vue

© 2019, Object Computing, Inc. (OCI). All rights reserved.

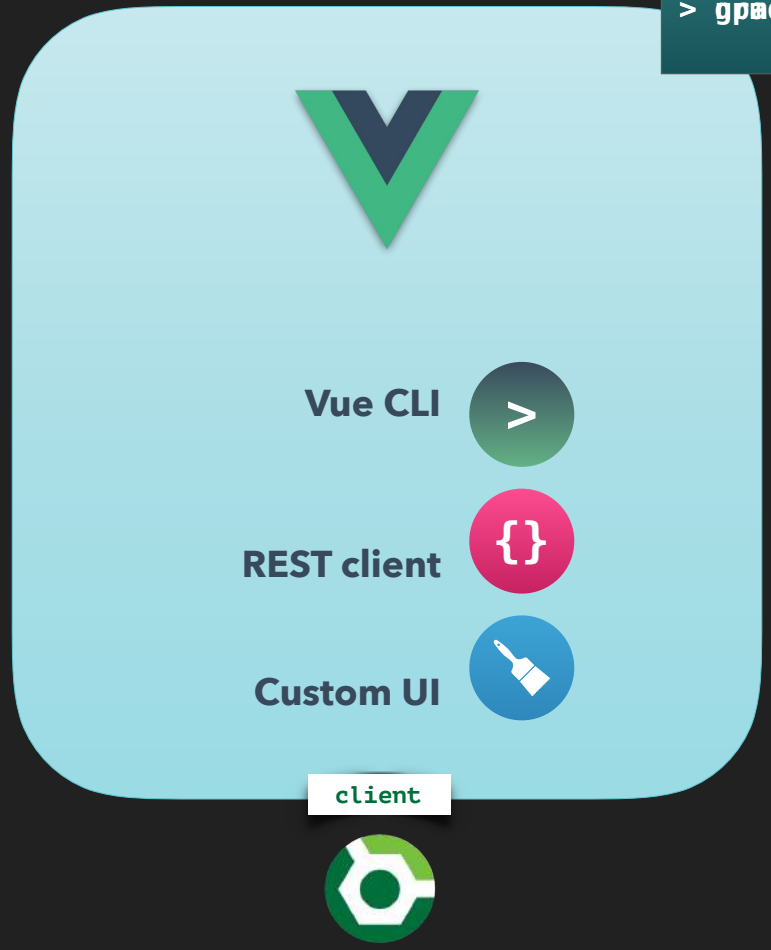
Vue Profile for Grails



© 2019, Object Computing, Inc. (OCI). All rights reserved.

Vue Profile for Grails

```
> gradle client:start    dev server
> gradle client:test     tests
> gradle client:build    dist.
```



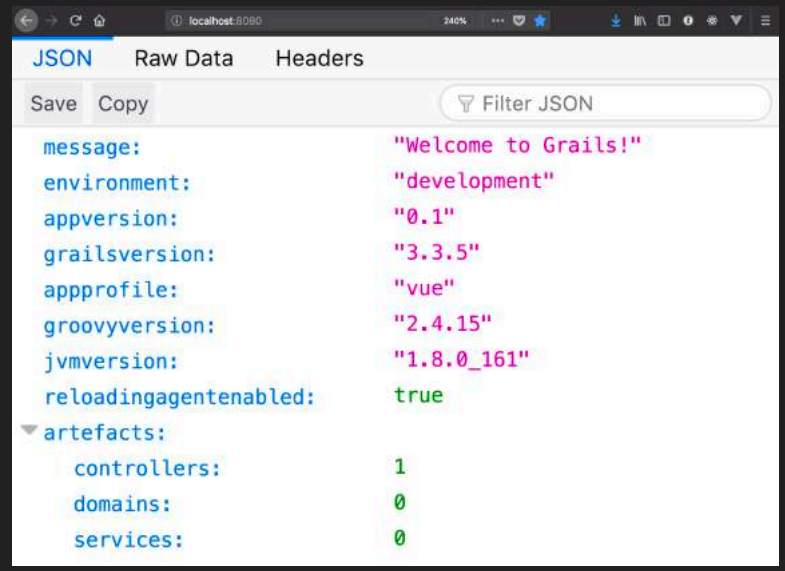
- Standardize Node versions
- Project-local installation
- Handles npm/yarn tasks

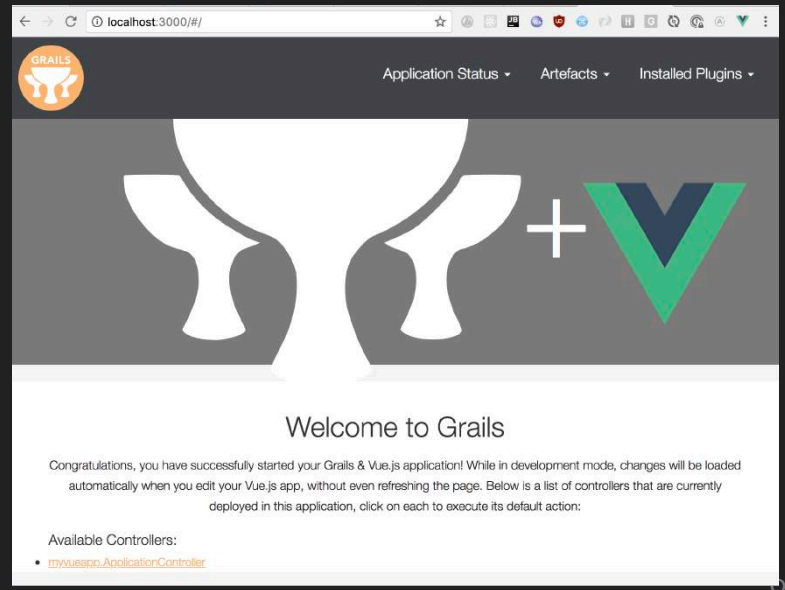
© 2019, Object Computing, Inc. (OCI). All rights reserved.

Vue Profile for Grails

```
> gradle bootRun//or grails run-app

Grails application
running at http://localhost:8080
```

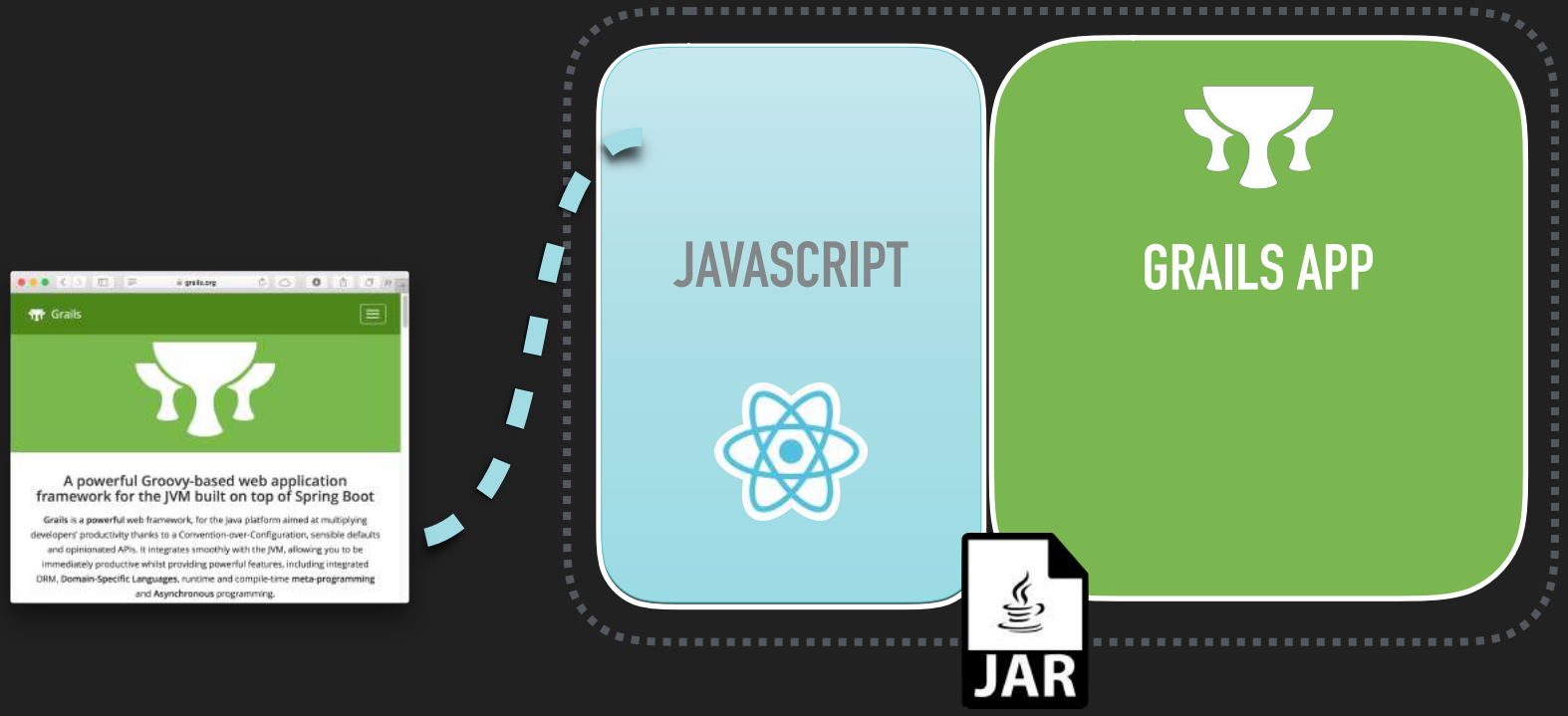
```
> gradle start //or npm start

Your application is running here:
http://localhost:3000
```





© 2019, Object Computing, Inc. (OCI). All rights reserved.

Combined JAR Deployment




```
java -jar myapp.jar
```

© 2019, Object Computing, Inc. (OCI). All rights reserved.

Vue with JWT (Spring) Security

© 2019, Object Computing, Inc. (OCI). All rights reserved.

Spring Security REST

OBJECT COMPUTING

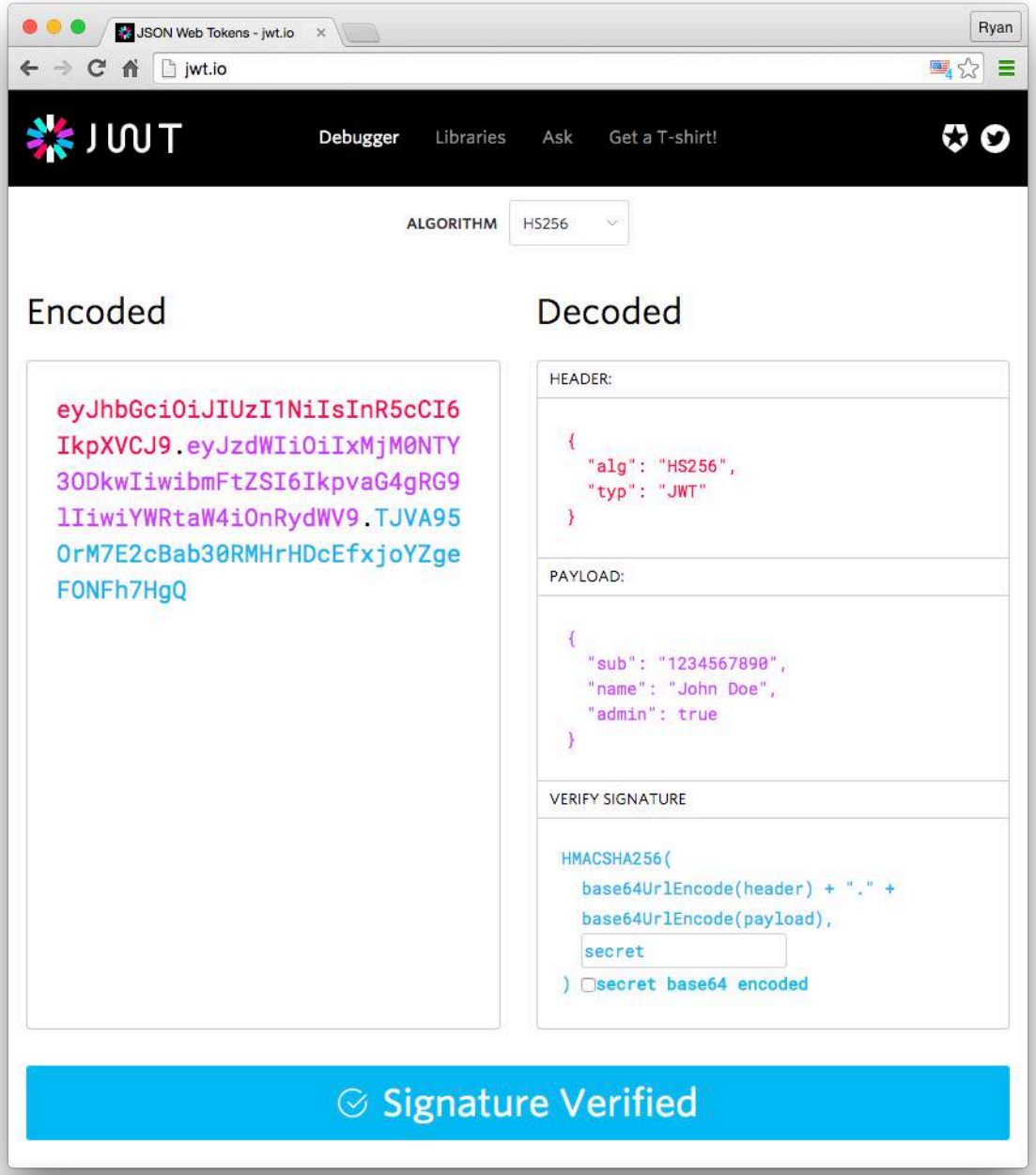
- Extends Spring Security Core
- OCI-supported
  - Adds support for stateless token-based authentication
  - Designed to secure restful APIs
  - Supports multiple token types/storage: JWT, GORM, Redis, etc

```
compile 'org.grails.plugins:spring-security-rest:3.0.0.RC1'
```

© 2019, Object Computing, Inc. (OCI). All rights reserved.

JSON Web Token

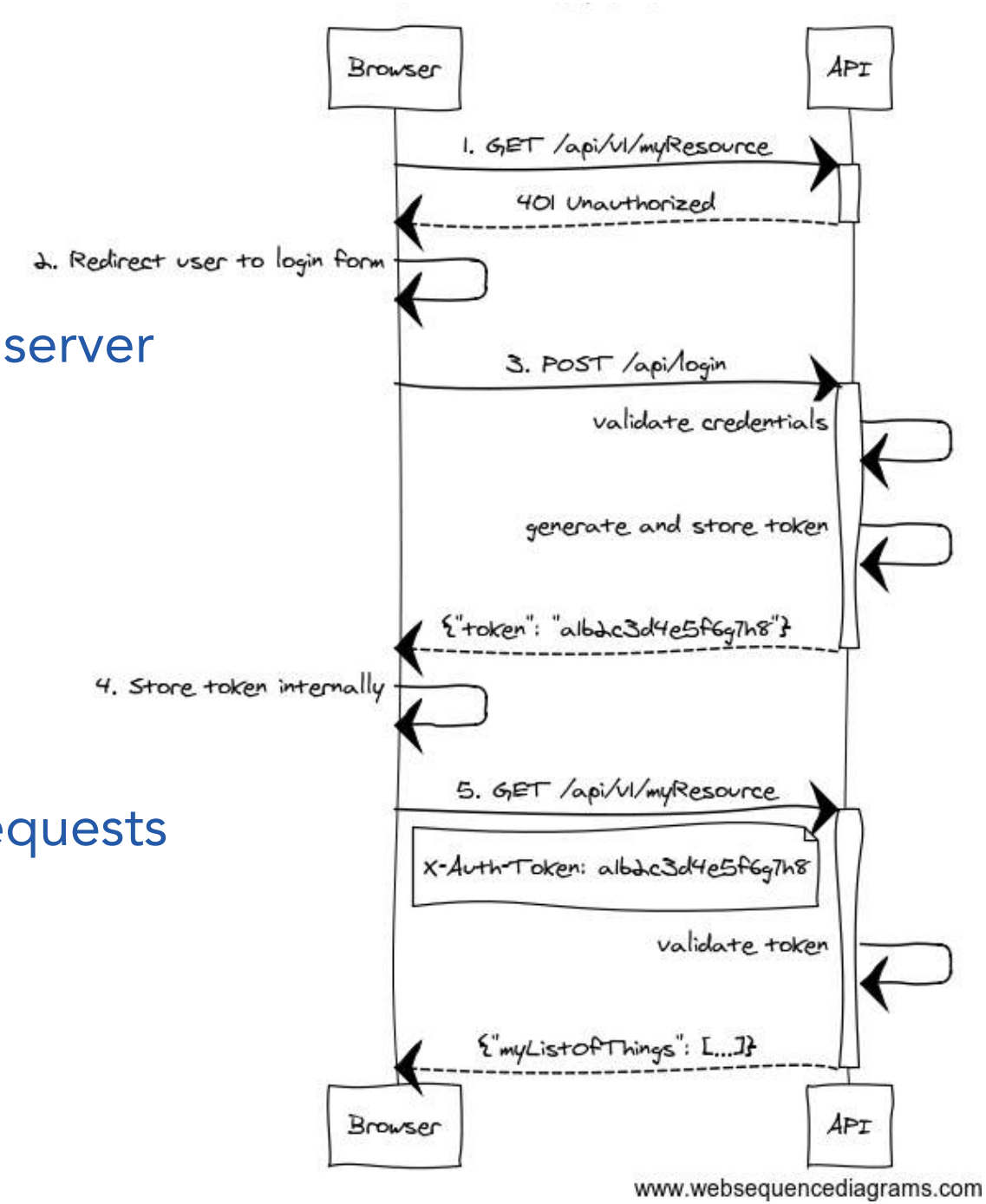
- https://jwt.io
- Open, industry-standard method for representing claims securely between two parties
- Consist of a header, payload, and signature



© 2019, Object Computing, Inc. (OCI). All rights reserved.

JWT Authentication

- Client makes unauthorized request to server
- Server responds with 401
- Client POSTs to /login endpoint
- Server responds with token
- Client includes token in subsequent requests
- Server responds with resource



© 2019, Object Computing, Inc. (OCI). All rights reserved.

Demo

© 2019, Object Computing, Inc. (OCI). All rights reserved.

Grails & Vue Links

- Vue Profile docs: https://grails-profiles.github.io/vue/latest/guide/
- Building a Vue App: http://guides.grails.org/building-a-vue-app/guide
- Combined Build: https://guides.grails.org/grails-vue-combined/guide
- Using the Vue Profile: http://guides.grails.org/using-the-vue-profile/guide

© 2019, Object Computing, Inc. (OCI). All rights reserved.

Thank you!

© 2019, Object Computing, Inc. (OCI). All rights reserved.

LEARN MORE ABOUT OCI EVENTS AND TRAINING

Events:

- objectcomputing.com/events

Training:

- objectcomputing.com/training
- grailstraining.com
- micronauttraining.com

Or email info@ocitraining.com to schedule a custom training program for your team online, on site, or in our state-of-the-art, Midwest training lab.

© 2019, Object Computing, Inc. (OCI). All rights reserved.

OBJECT COMPUTING

CONNECT WITH US

1+ (314) 579-0066

@objectcomputing

objectcomputing.com

© 2019, Object Computing, Inc. (OCI). All rights reserved.