

Moving from Server-side to Client-side with Angular



Strong grasp on how to **construct** a **single** feature in Angular

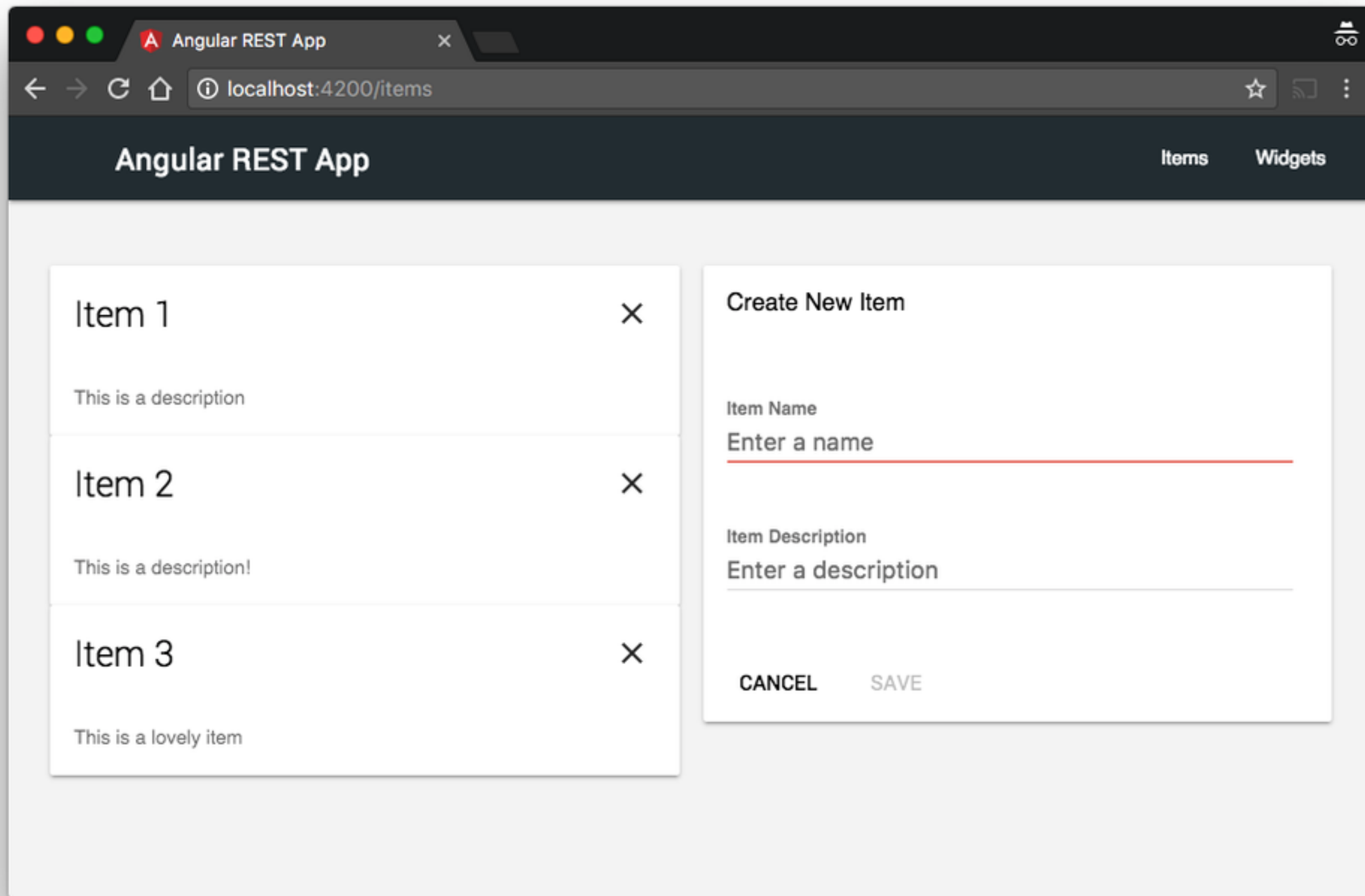
Agenda

- **Getting Started**
- **Hello Angular**
- **The Angular CLI**
- **Components**
- **Templates**
- **Services**
- **Component Driven Architecture**
- **Server Communication**
- **BONUS MODULE!**

Getting Started



<https://github.com/onehungrymind/angular-server-to-client>



The Demo Application

- A simple RESTful master-detail application built using Angular and the Angular CLI
- We will be building out a new **widgets** feature
- Feel free to use the existing **items** feature as a reference point
- Please explore! Don't be afraid to try new things!

Challenges

- Make sure you can run the application

The Big Picture

Challenges

- Identify the major Angular pieces in the sample application
- Add a new property to the **Items** component and bind to it in the view
- Add a new property to the **ItemsService** and consume it in a component

The Angular CLI

Challenges

- Use the Angular CLI to generate a **candidates** component

Component Fundamentals

Challenges

- Check out the **00-start** branch
- Create the file structure for a new **widgets** feature
- Create the ES6 class for the **widgets** component
- Import the appropriate modules into the **widgets** component
- Decorate the **widgets** component to use the **widgets** template
- Display the **widgets** component in the **app** component
- **BONUS** Create a simple route to view the **widgets** component by itself

Template Fundamentals

Challenges

- Create a **widgets** collection in the **widgets** component with mock objects
- Display the **widgets** collection in the template using **ngFor**
- Create a **selectedWidget** property in the widget component
- Use an **event binding** to set a selected widget
- Display the **widgets** properties using **property binding** and **interpolation binding**
- Use **ngIf else** to show an alternate message if no widget is selected

ACTION ITEM! Go to <http://bit.ly/workshop-snippets> to save on typing

Services

Challenges

- Extract the **widgets collection** to a **widgets service**
- Add the **widgets service** to the **application module** so that it can be consumed
- Inject that **widgets service** into the **widgets component**
- Consume and display the new **widgets collection**

Component Driven Architecture

Challenges

- Create a presentational **widgets-list** and **widget-details** component using **@Input** and **@Output**
- Pass the **widgets** collection to the **widgets-list** component
- Capture a **selected** output event from the **widgets-list** component
- Create a **delete** output event in the **widgets-list** component
- Display the **selectedWidget** in the **widget-details** component
- Create a **save** output event in the **widget-details** component
- Create a **cancel** output event in the **widget-details** component

Template Driven Forms

Challenges

- Create a form to display the currently selected **widget**
- Use a **lifecycle hook** or **setter** to isolate the **widget** mutation
- Create a button to **save** the edited **widget** to the parent component
- Create a button to **cancel** editing the **widget** to the parent component
- Using **ngForm**, add in some validation for editing the **widget** component

Server Communication

Challenges

- Replace the local **widgets** collection with a call to the **widgets** endpoint
- Update the **widgets** component to handle the async call
- Flesh out the rest of the **CRUD** functionality using **ItemsService** as reference

I  YOU!



@simpulton



<https://venturplex.com/>



Thanks!