Tour of Heroes

App & Tutorial

Tour of Heroes

App & Tutorial

- Install Node.js
- npm install -g @angular/cli
- ng new angular-tour-of-heroes
- cd angular-tour-of-heroes
- ng serve

1. The Hero Editor

- You used the CLI to create a second **HeroesComponent**.
- You displayed the **HeroesComponent** by adding it to the **AppComponent** shell.
- You applied the **UppercasePipe** to format the name.
- You used **two-way** data binding with the **ngModel** directive.
- You learned about the **AppModule**.
- You imported the **FormsModule** in the **AppModule** so that Angular would recognize and apply the **ngModel** directive.
- You learned the importance of declaring components in the **AppModule** and appreciated that the **CLI** declared it for you.

2. Displaying a List

- The Tour of Heroes app displays a list of heroes in a **Master/Detail view**.
- The user can select a hero and see that hero's details.
- You used ***ngFor** to display a list.
- You used ***ngIf** to conditionally include or exclude a block of HTML.
- You can toggle a CSS style class with a **class binding**.

3. Master/Detail Components

- You created a separate, reusable **HeroDetailComponent**.
- You used a property binding to give the parent **HeroesComponent** control over the child **HeroDetailComponent**.
- You used the **@Input** decorator to make the hero property available for binding by the external **HeroesComponent**.

4. Services

- You refactored data access to the HeroService class.
- You registered the HeroService as the provider of its service at the root level so that it can be injected anywhere in the app.
- You used Angular Dependency Injection to inject it into a component.
- You gave the HeroService get data method an asynchronous signature.
- You discovered Observable and the RxJS Observable library.
- You used RxJS of() to return an observable of mock heroes (Observable<Hero[]>).
- The component's ngOnInit lifecycle hook calls the HeroService method, not the constructor.
- You created a MessageService for loosely-coupled communication between classes.
- The HeroService injected into a component is created with another injected service, MessageService.

5. Routing

- You added the Angular router to navigate among different components.
- You turned the AppComponent into a navigation shell with <a> links and a <router-outlet>.
- You configured the router in an AppRoutingModule
- You defined simple routes, a redirect route, and a parameterized route.
- You used the routerLink directive in anchor elements.
- You refactored a tightly-coupled master/detail view into a routed detail view.
- You used router link parameters to navigate to the detail view of a user-selected hero.
- You shared the HeroService among multiple components.

6. HTTP

- You added the necessary dependencies to use HTTP in the app.
- You refactored HeroService to load heroes from a web API.
- You extended HeroService to support post(), put(), and delete() methods.
- You updated the components to allow adding, editing, and deleting of heroes.
- You configured an in-memory web API.
- You learned how to use observables.

Tour of Heroes

Rails API + Angular Frontend