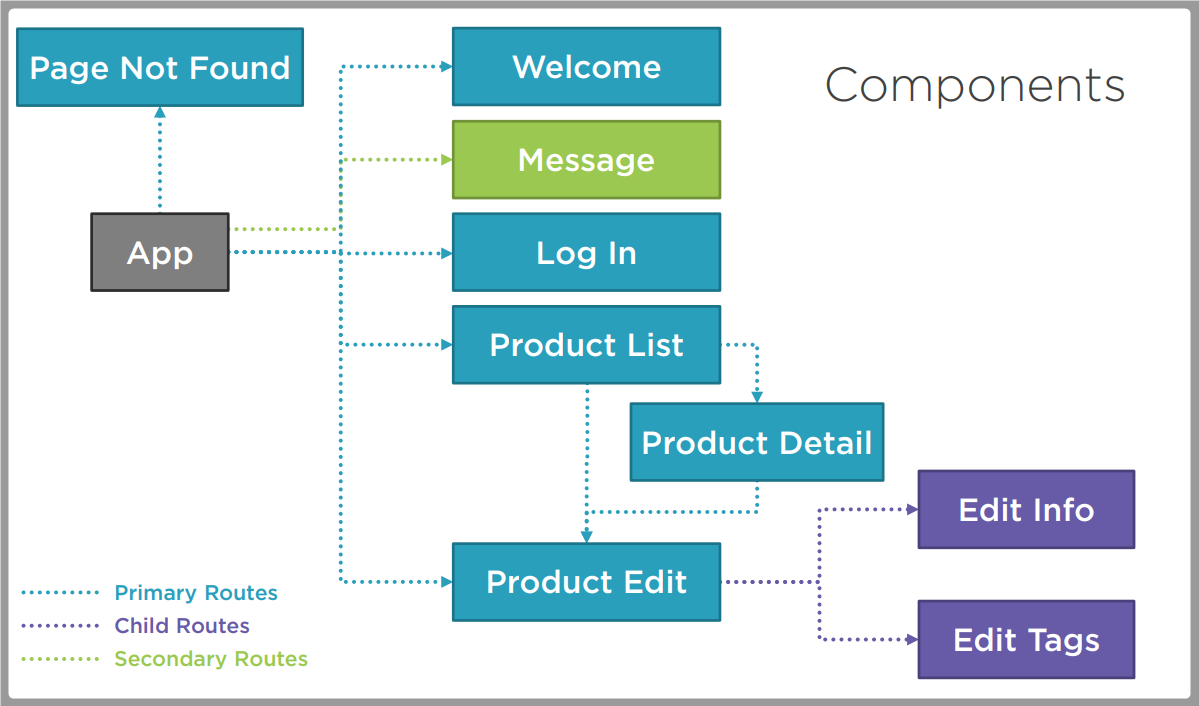
**Angular Routing**

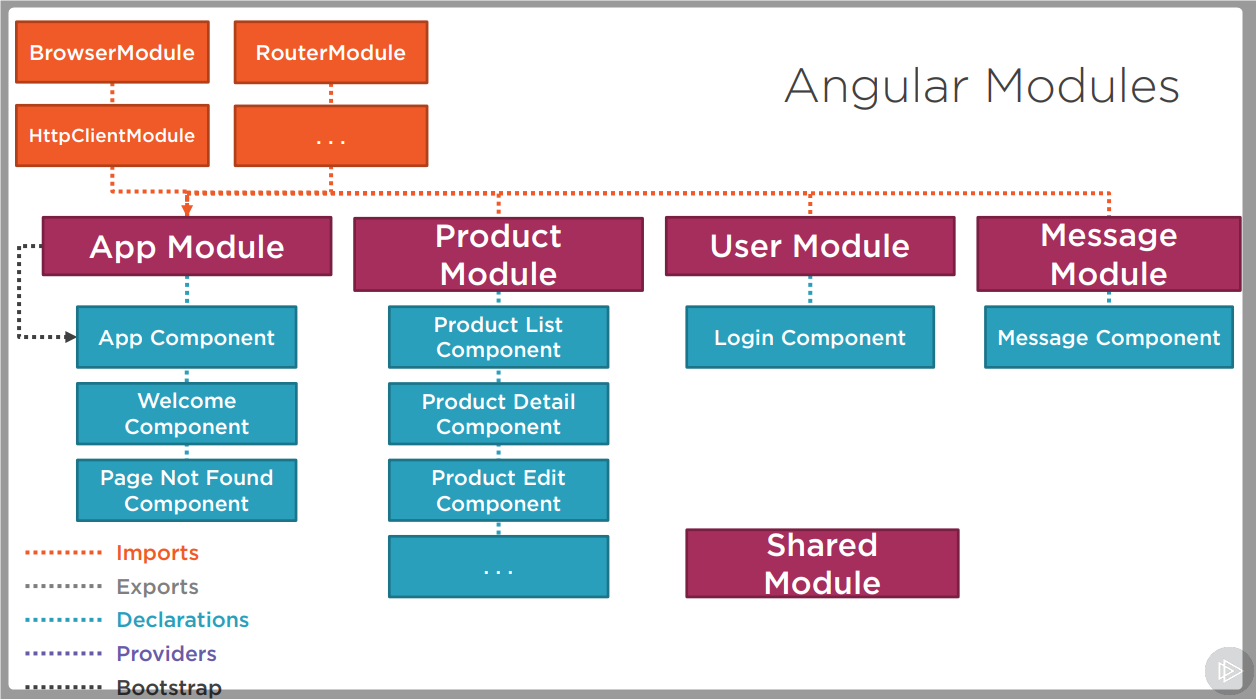
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Here we are working on a sample application especially on routing concepts in angular.

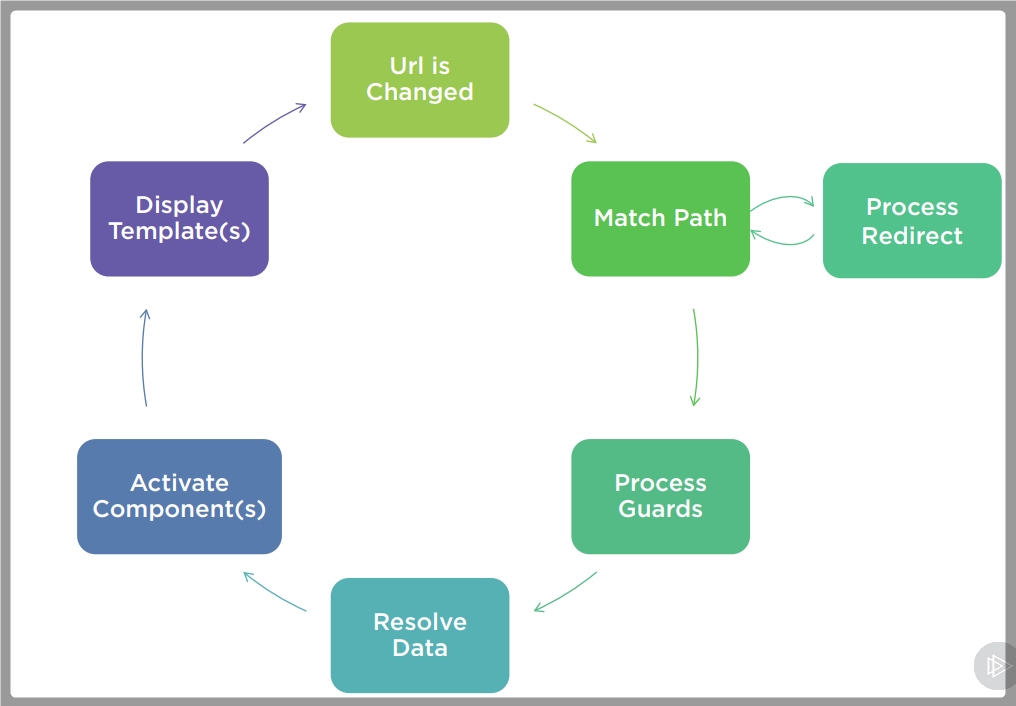
Here is the sample application **components** configured in this demo.



Here is the sample application modules configured in this demo.



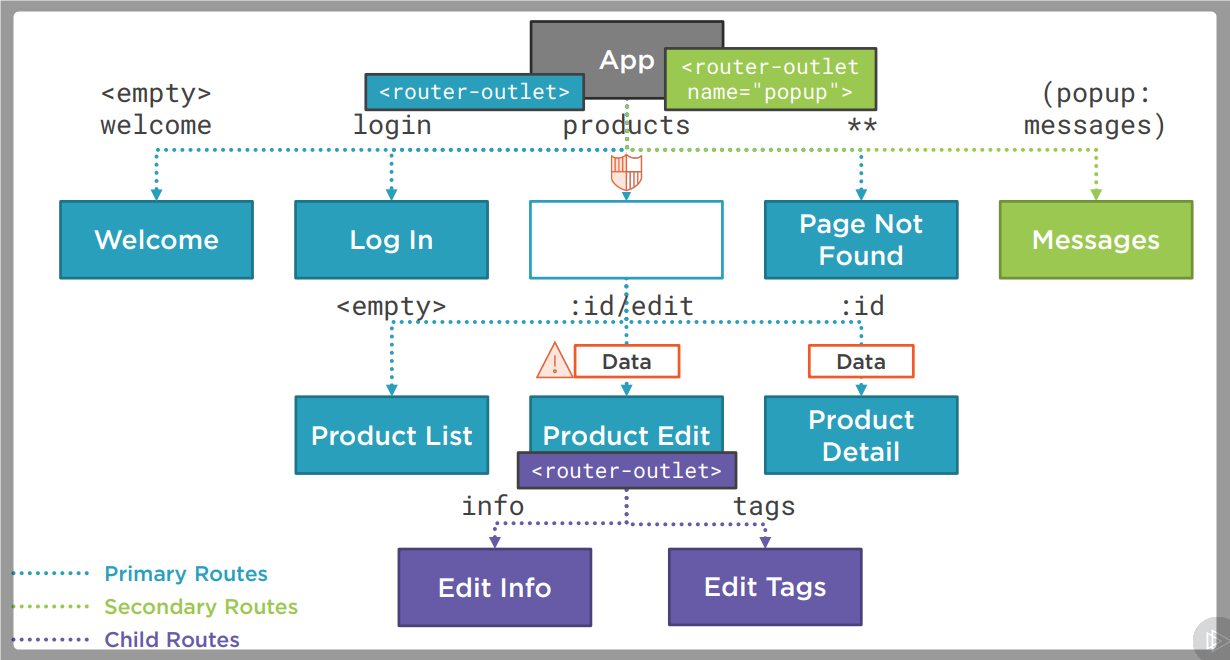
Routing works as follows?



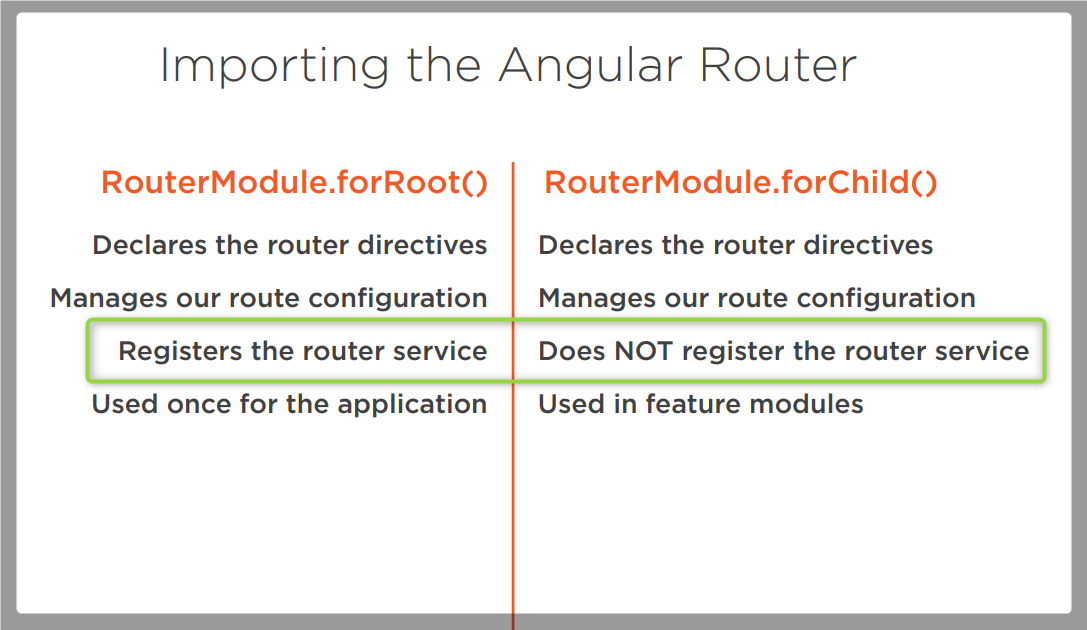
Course Outline =>



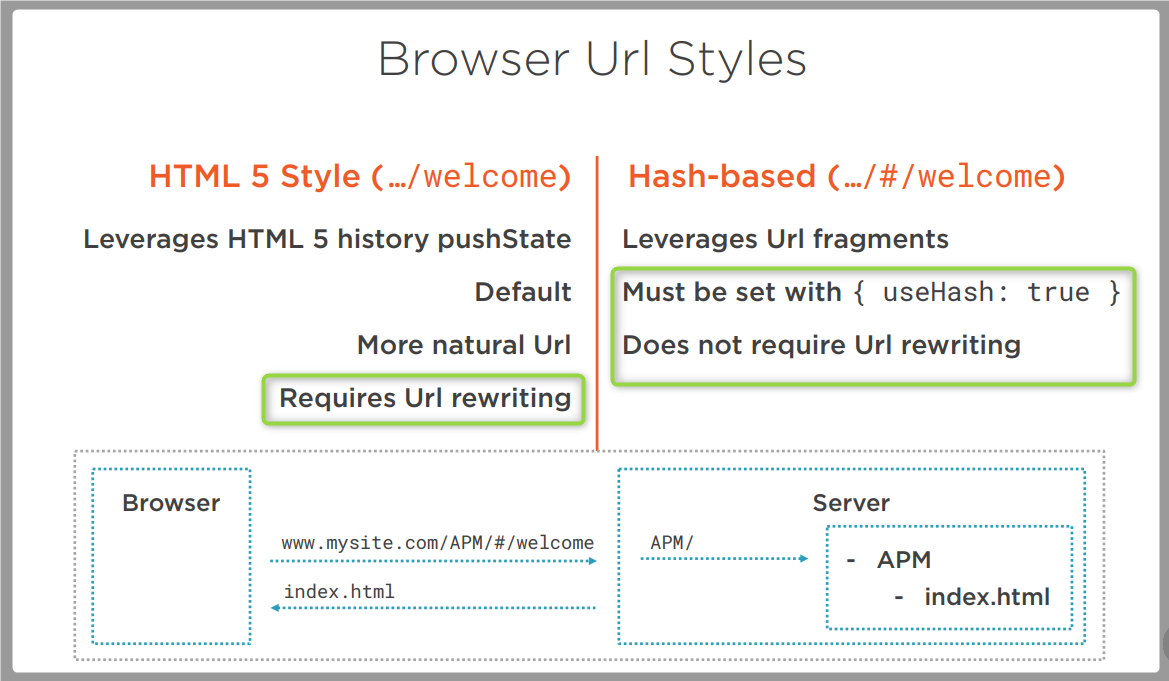
Our final application routing

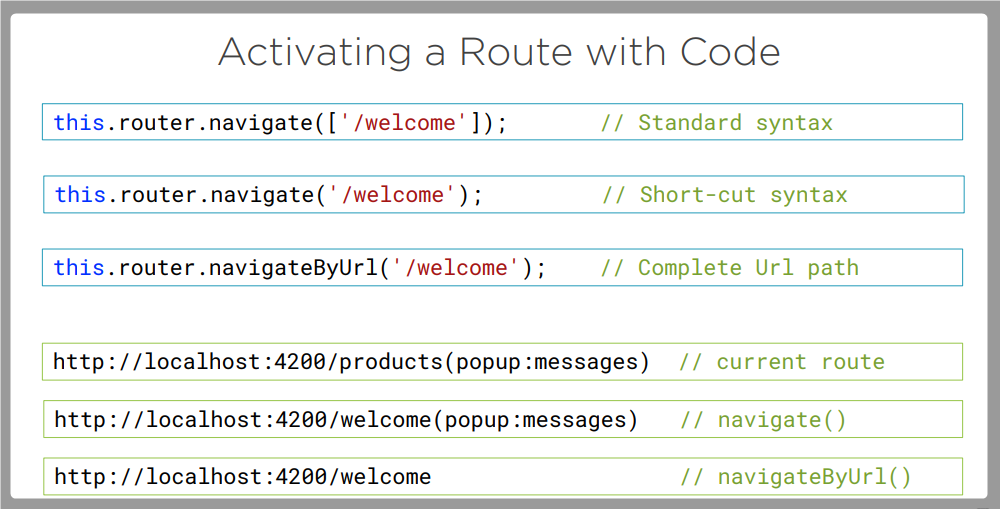


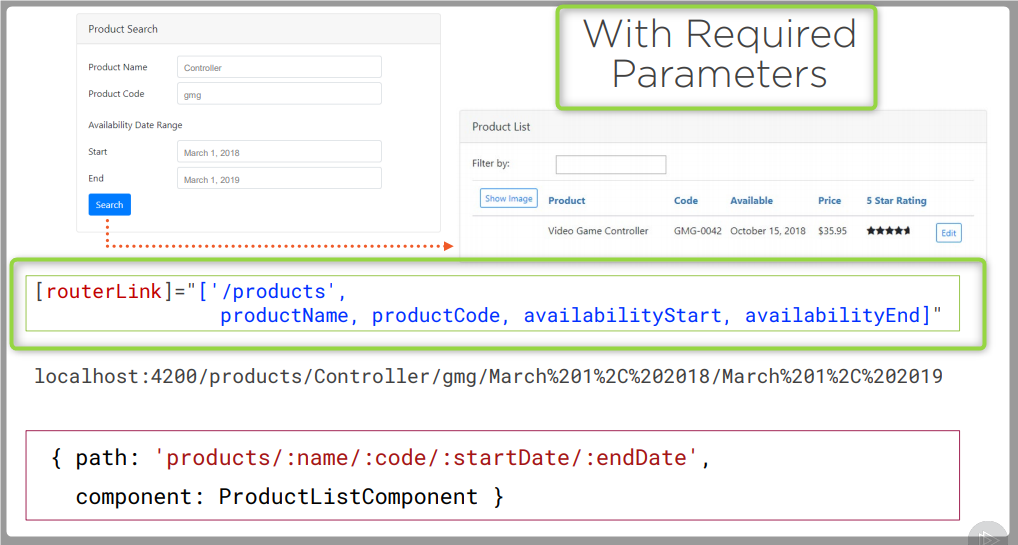
Import Angular Router

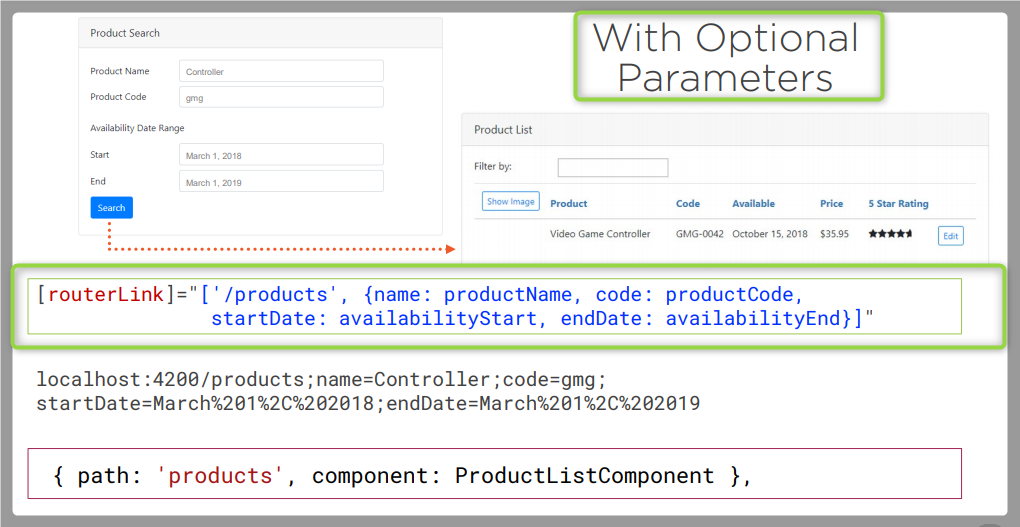


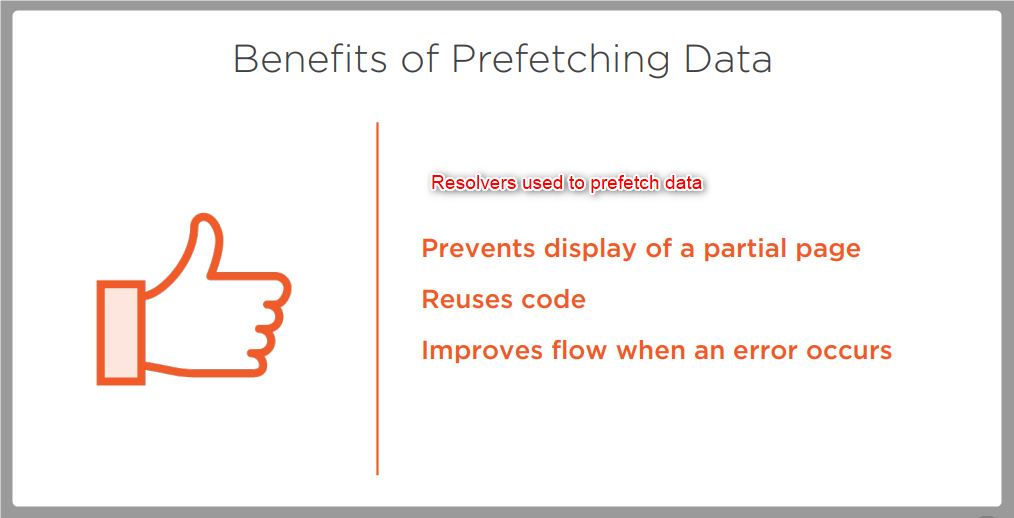
**HTML 5 vs Hashbased Routing**



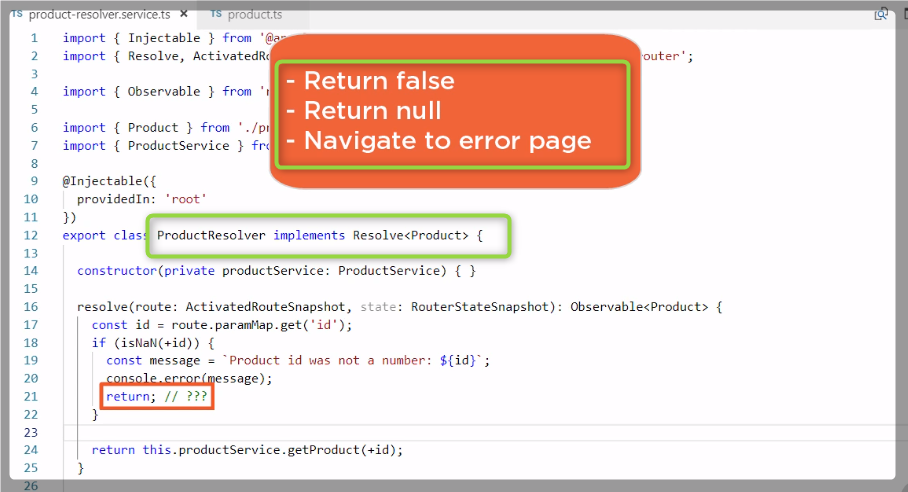


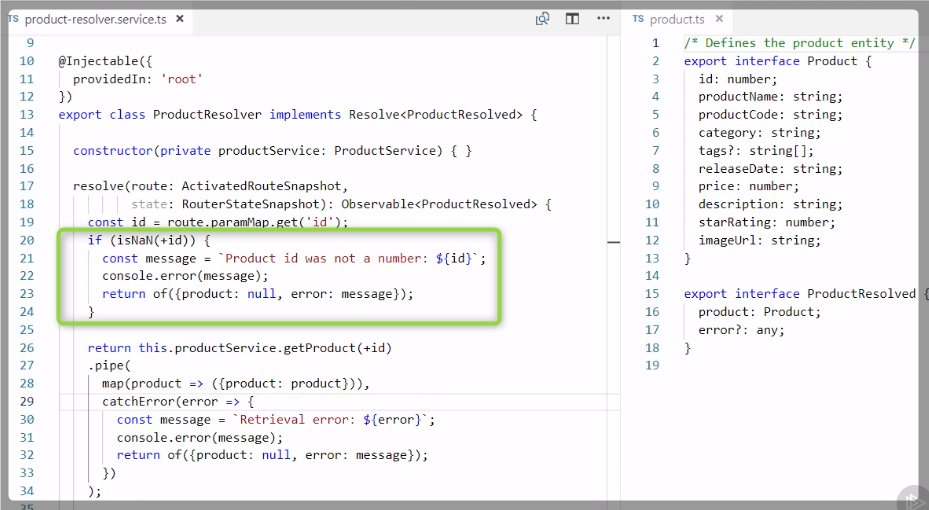


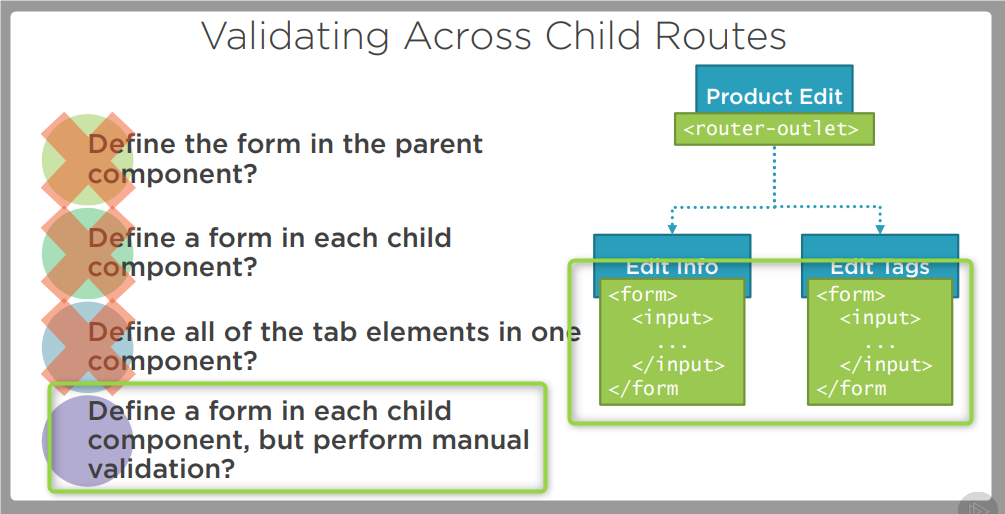




Resolver Error Handling:



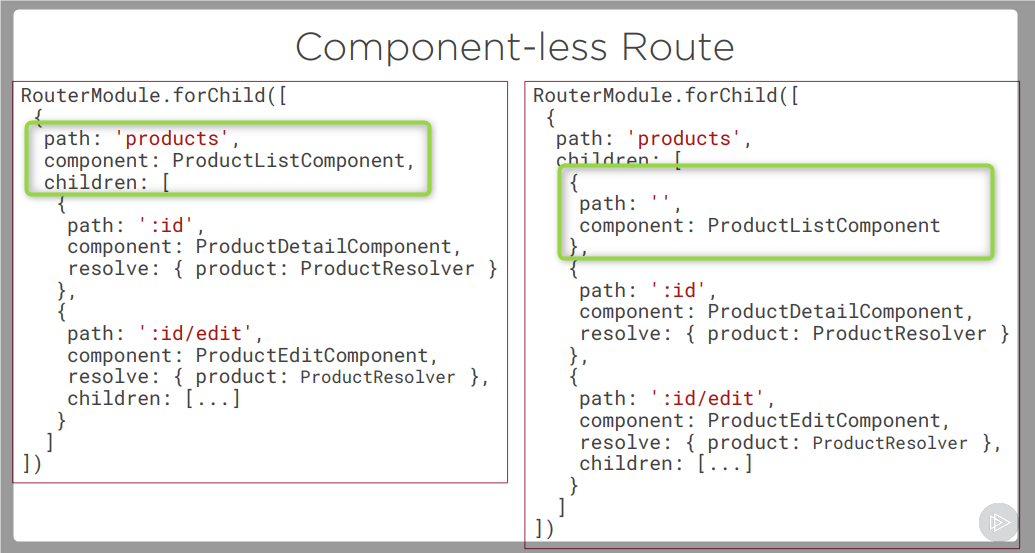




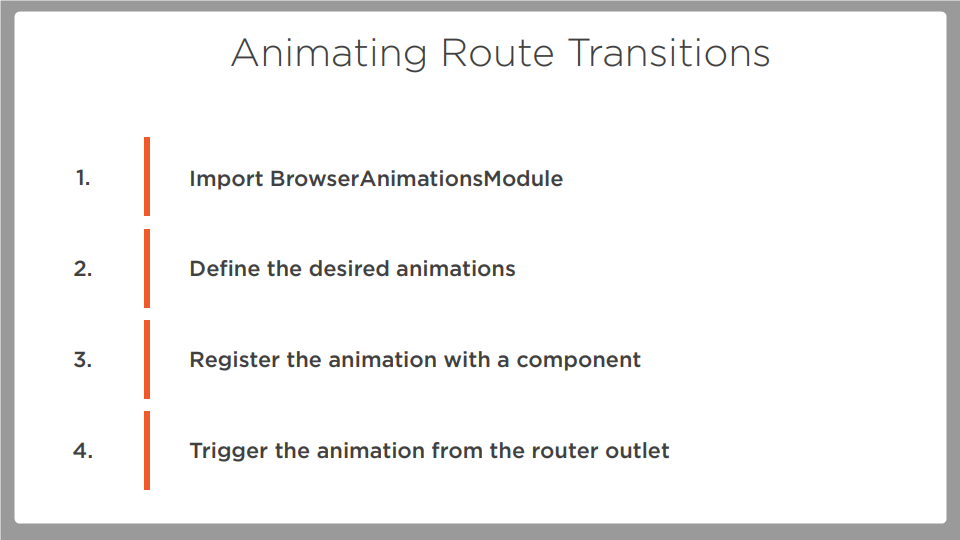
**Routes Grouping**

****

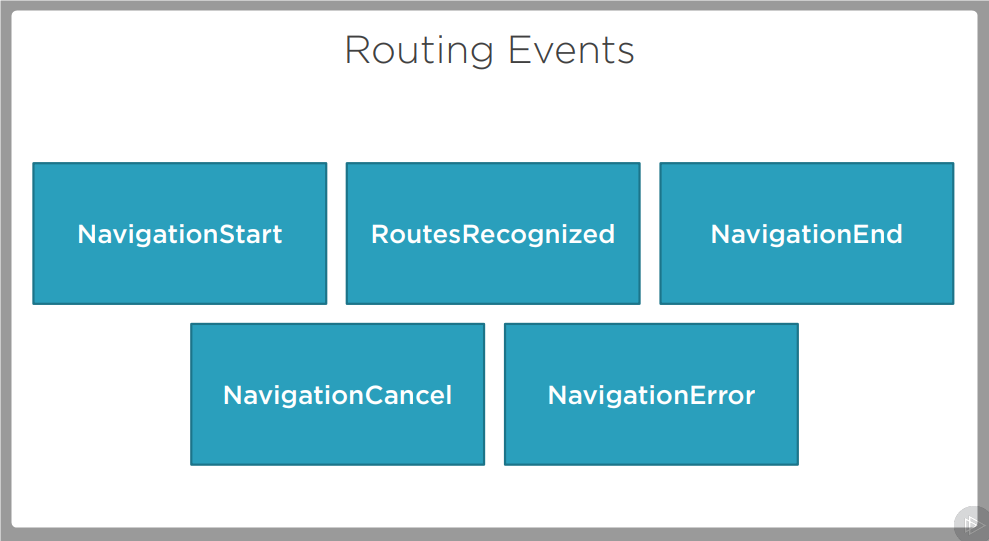
**Component-less routing**

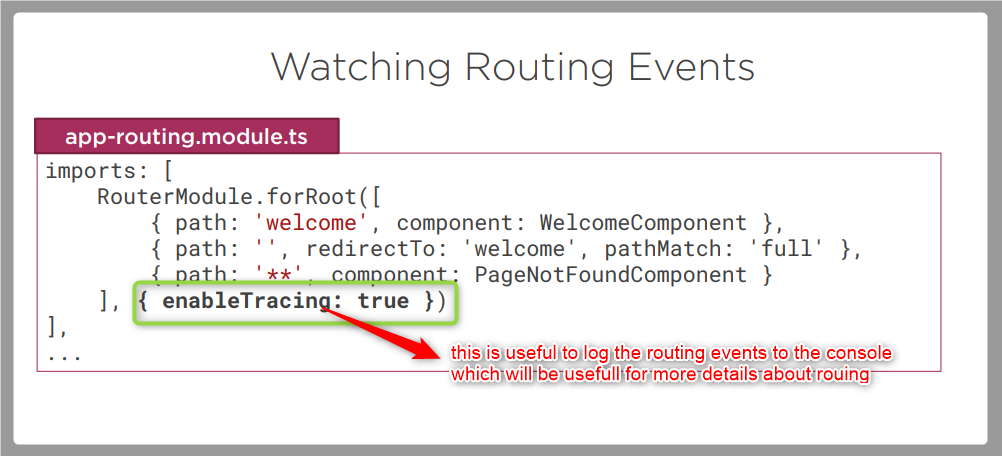


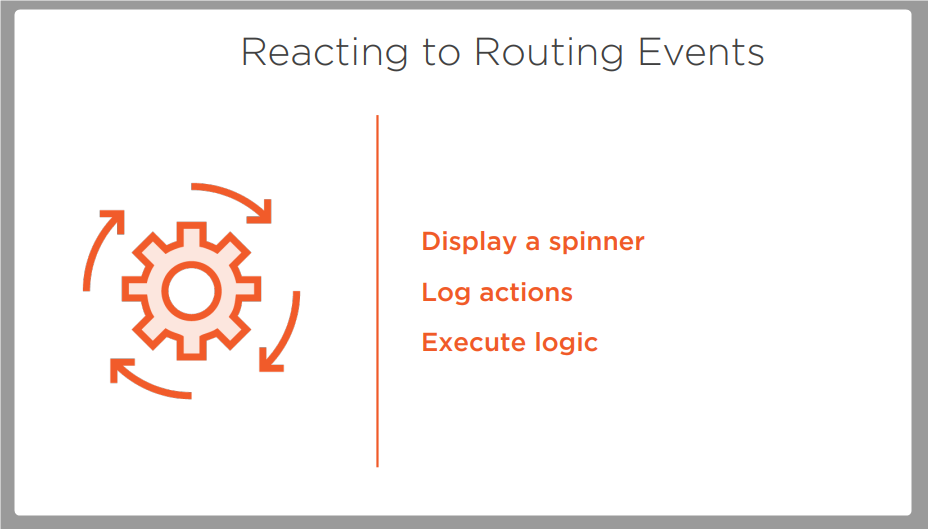
**Animating Route Transitions**



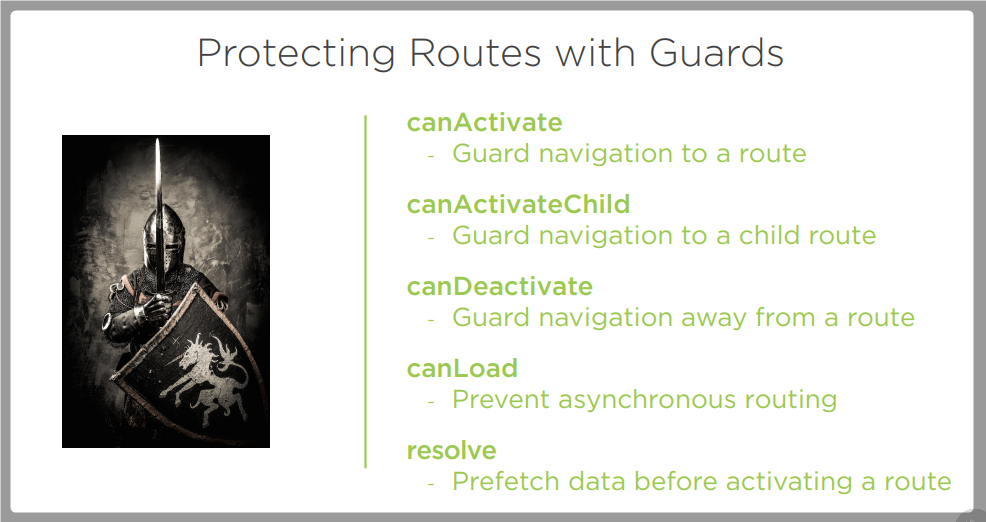
**Routing Events:**

****

****

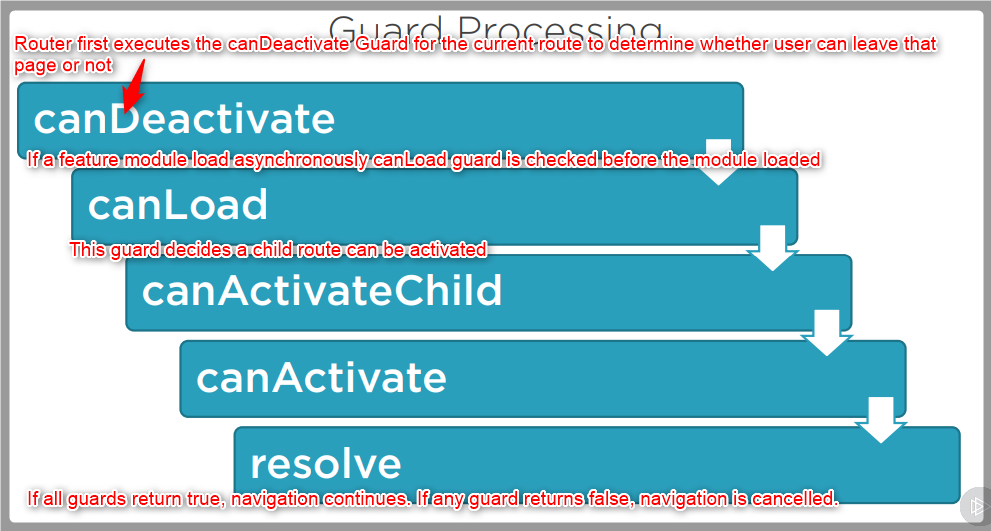
****

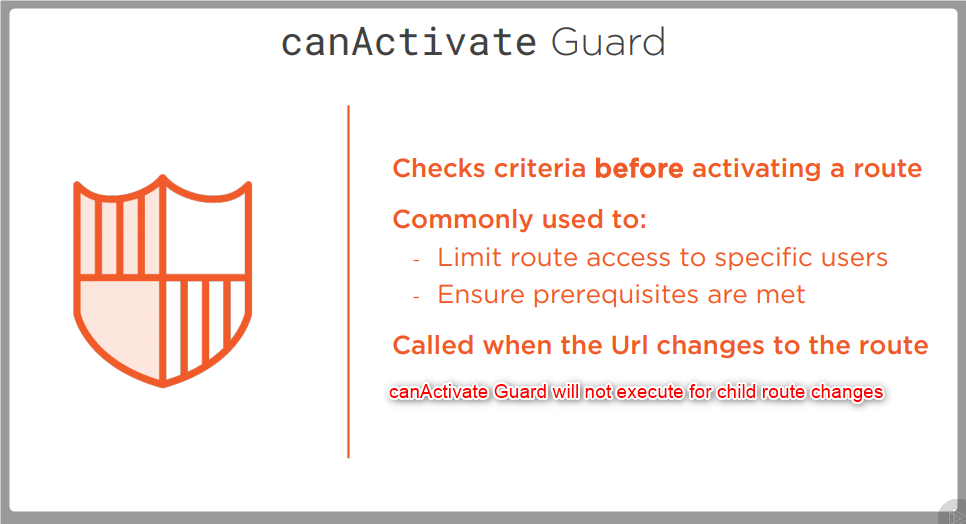
**Route Guards:**



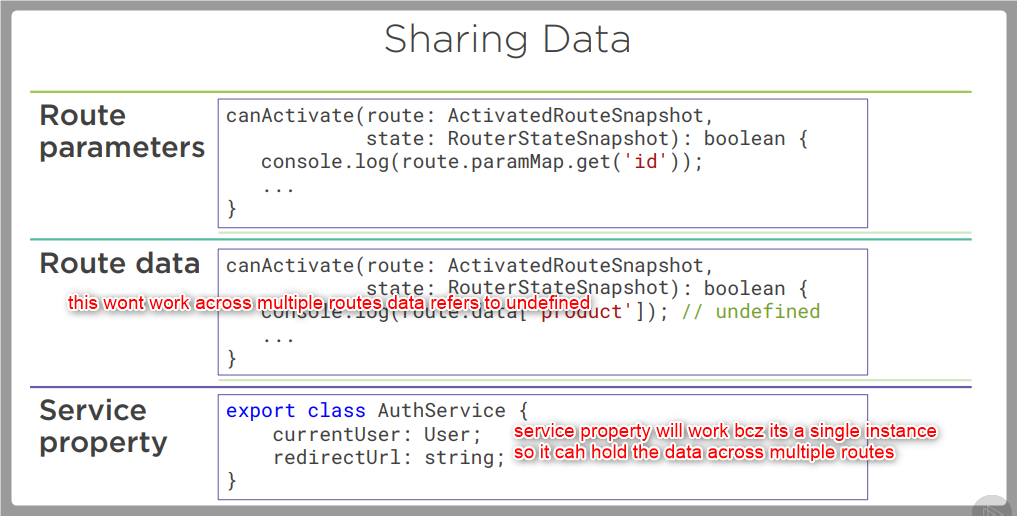
Realtime benefits using Route Guards.

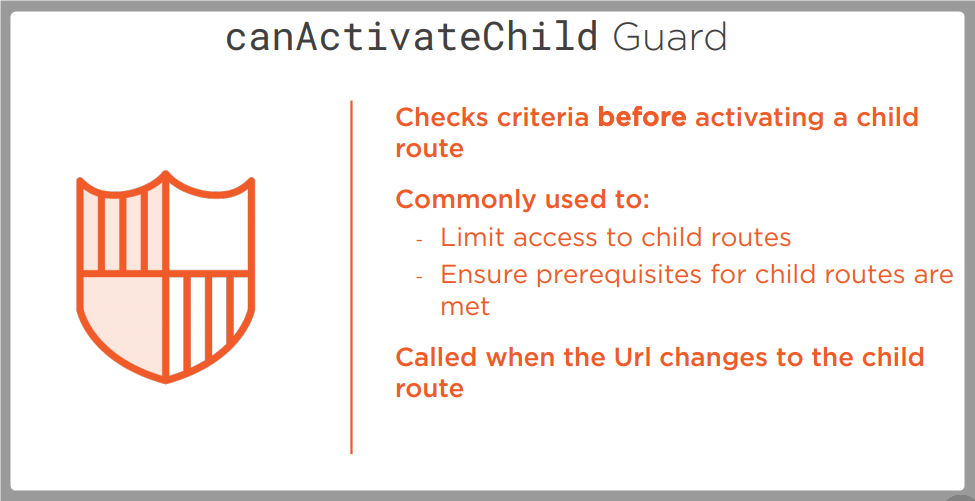


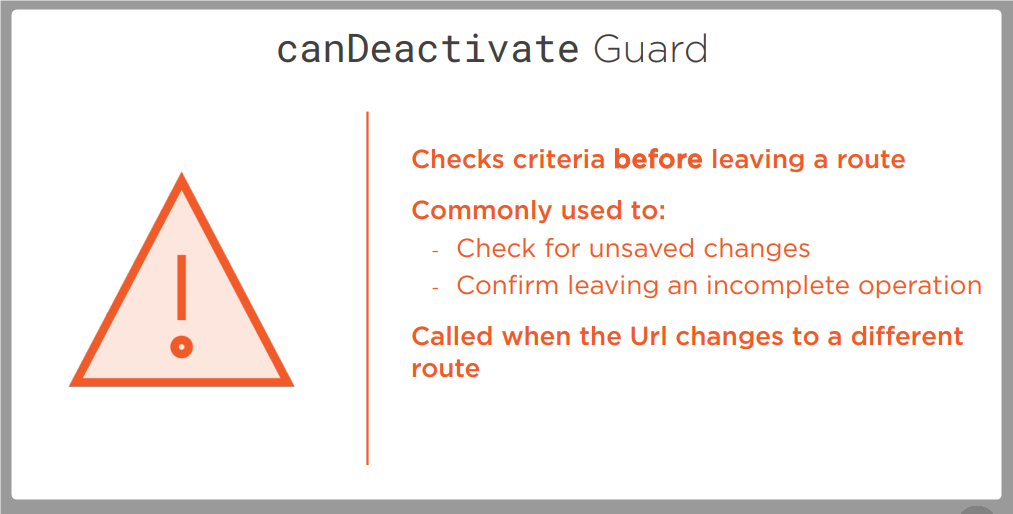


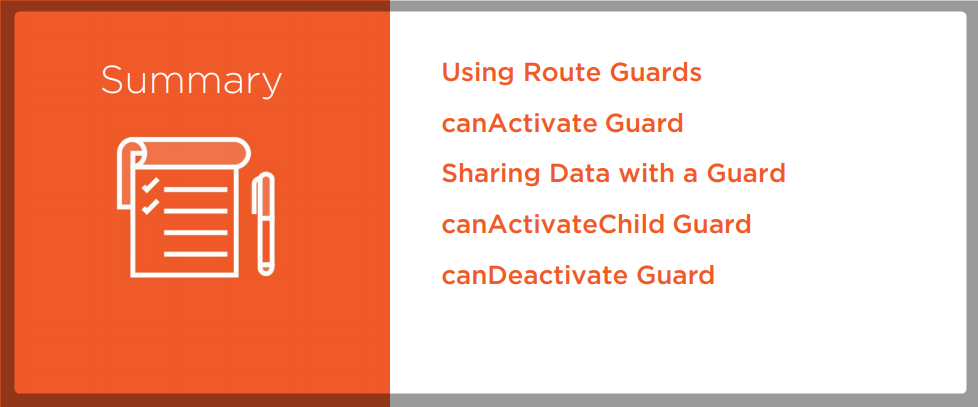


**Sharing Data across routes**

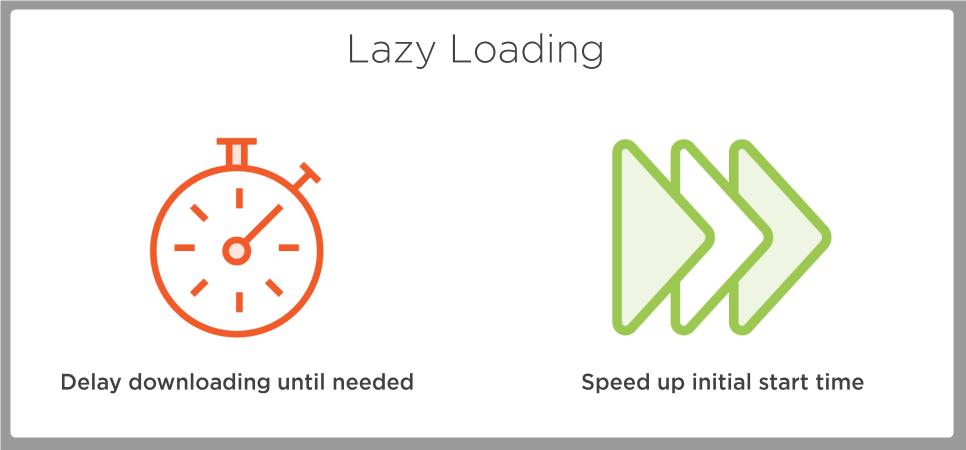




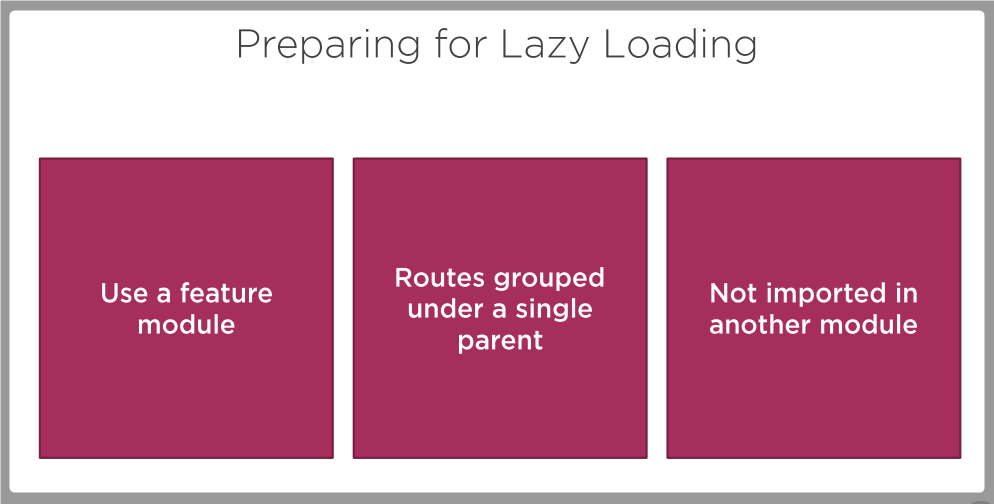




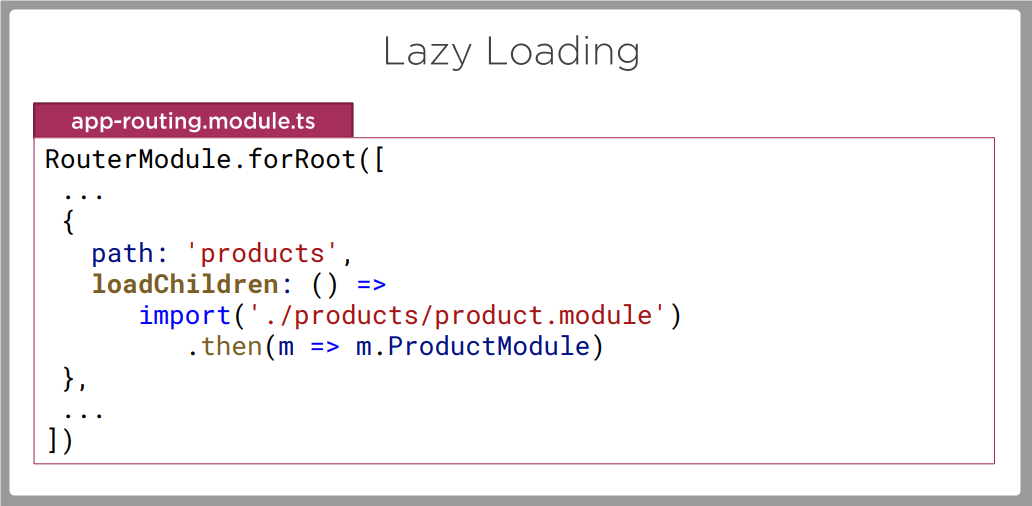
**Lazy Loading: benefits**



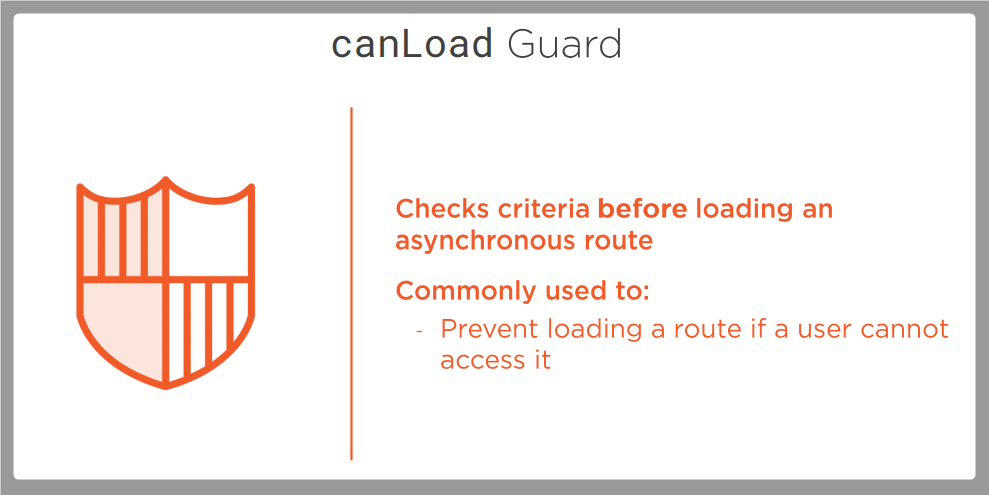
**Pre-Requisites for Lazy Loading**



**Syntax:**

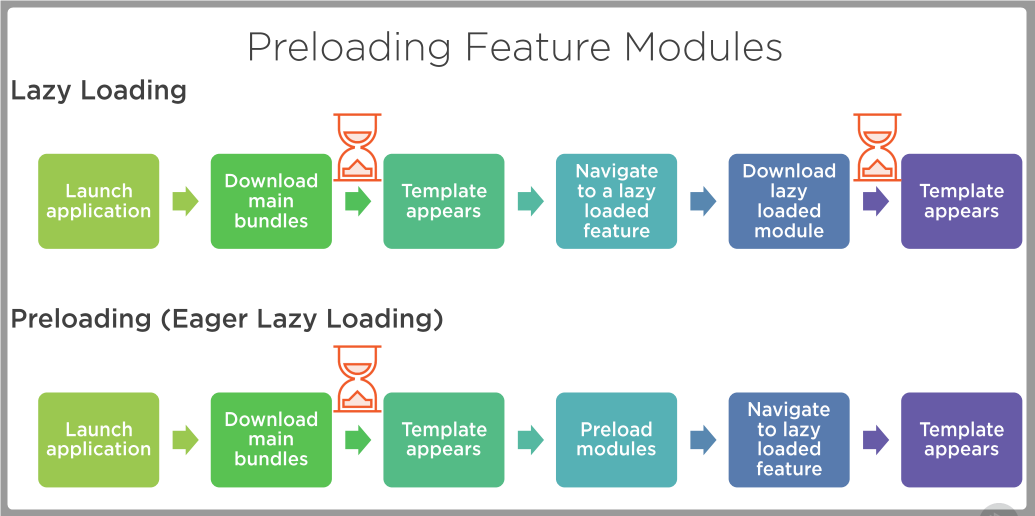


**CanLoad Guard => Commonly used for preventing a route accessing if user not authorized to it.**

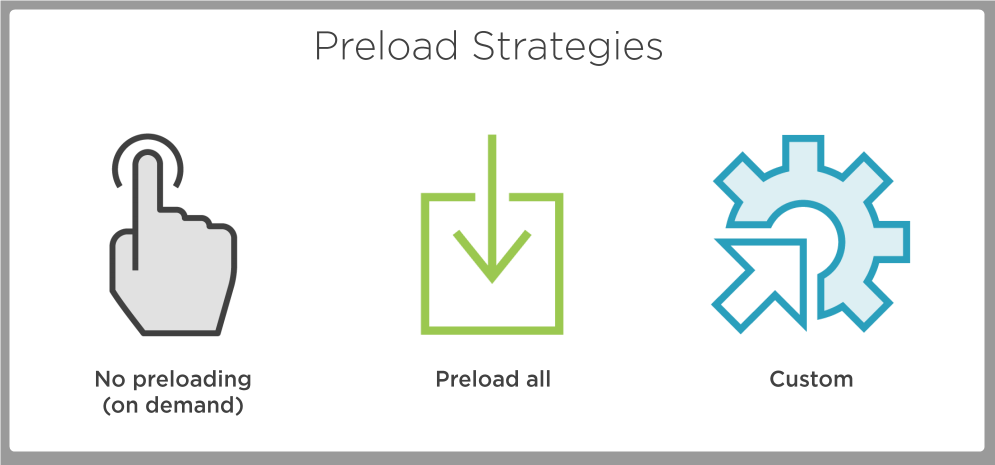


**Lazy loading has a drawback depending on the users download-speed it may cause noticeable delay in downloading of feature module or noticeable delay in navigating to the first route of feature module.**

**Preload or Eager Lazy loading**

****

**Preload Strategies**

****

**CanLoad guard blocks the pre-load strategies.**

****