There are times in our applications when we just need to prevent users from taking certain actions. It could be something as simple as viewing a page, leaving a page with unsaved data, to needing to load data before the new component is displayed on screen. “Angular routing plays an important role in a real-world application which is responsible for loading different components which may have sensitive data as well. (Jolly.exe, 2019)” This is where route guards come into play.

The big question is, what is a route guard. “Angular’s route guards are interfaces which can tell the router whether or not it should allow navigation to a requested route. (Chenkie, 2019)” Angular’s route guard calls are expected to return a true or false response.

Angular route guards are accessed through the route guard interface. Angular has five different route guard interfaces: CanActivate, CanActivateChild, CanDeactivate, CanLoad, Resolve.

Resolve is a route guard interface that is executed before a component is loaded. It is a way to retrieve data before activating a given component.

CanActivate is a route guard interface that is called to authenticate whether or not a user has permission to view the content of the given path. It will run its authentication function then return true or false based upon the results.

So when you need to restrict data access or warn the user that they have not saved the form they just spent time filling in, turn to Angular’s route guards to help you with your Angular needs.

References:

Chenkie, R. (2019, July 9). *Angular Authentication: Using Route Guards*. Retrieved January 15, 2020, from https://medium.com/@ryanchenkie\_40935/angular-authentication-using-route-guards-bf7a4ca13ae3

Jolly.exe. (2019, March 7). *Angular 7/6: Use Auth Guards CanActivate and Resolve in Angular Routing Quick Example*. Retrieved January 15, 2020, from https://www.freakyjolly.com/angular-7-6-use-auth-guards-canactivate-and-resolve-in-angular-routing-quick-example/