William Watlington

November 3, 2022

Discussion 3.1

WEB425

In Angular, sometimes we might want to navigate users differently depending on certain variables such whether or not the user is logged in. In this example, a conditional statement could be placed on the individual pages to change navigation depending on certain variables, but this method can quickly get out of hand and become unmanageable on large applications. Instead, the correct option is the use of Angular route guards. Angular route guards are built-in Angular interfaces that allow for routing based on conditions. There are five types of built-in route guards: CanActivate, CanActivateChild, CanLoad, CanDeactivate, and Resolve.

The CanActivate route guard is used to handle allowing or disallowing navigation based on a Boolean. A common example of this is checking is a user is logged in before showing a page, and redirecting to the login screen if the user is not logged in. This can be accomplished by created a method that checks the status of the user’s session and returns a Boolean.

The CanActivateChild route guard is very similar to the CanActivate route guard. The major difference here that CanActivateChild, when utilized on a parent route, automatically allows for user access to all of that route’s child routes.

The CanLoad, similar to CanActivate, limits a user’s access to pages when unauthorized. Where CanActivate controls the routing/display of modules, CanLoad affects lazy loaded modules. The CanLoad route guard can stop lazy loaded modules from appearing if not authorized.

Next is CanDeactivate. The CanDeactivate Angular route guard does almost the opposite of the above route guards. It stops users from navigating *away* from a route. This is often used in situations where you may be stopping a user from navigating away when they would lose unsaved input or need to complete a task to continue.

The final route guard is the resolve guard. The resolve route takes data and resolves it based on a resolve method and then passes that data to the component.

Reference:

Sharma, P. (2019, December 1). *Understanding Angular Guards*. Codeburst. https://codeburst.io/understanding-angular-guards-347b452e1892