RouterOutlet is essentially a placeholder that “Angular fills based on the current router state.” (*Angular*, n.d.-b) An outlet can be named with the name value which is an optional attribute although if it is named it can also have secondary routes. (*Angular*, n.d.-b) An example of code for RouterOutlet looks like this:

<router-outlet></router-outlet>

//Or it can look like this:

<router-outlet name=”left”></router-outlet>

If you use a named outlet and use secondary routes with it, you are also able to get a few outlets within the same RouterLink. (*Angular*, n.d.-b) Where the router will have a record of the “branches” you are using for the outlets then creates a “representation of that tree in the URL.” (*Angular*, n.d.-b) What the website is talking about the syntax for a primary or secondary route. (*Angular*, n.d.-b)

The Router Outlet is also able to activate or deactivate events depending on whether there was component added or even removed. (*Angular*, n.d.-b) These events start with the RouteReuseStrategy.

An example from angular.io:

<router-outlet

(activate)='onActivate($event)'

(deactivate)='onDeactivate($event)'

(attach)='onAttach($event)'

(detach)='onDetach($event)'></router-outlet>

(*Angular*, n.d.-b)

A Router Oulet can be used during a single use application or a SPA. A SPA is when the page is to be reloaded however other components on the web page might change. (Fain & Moiseev, 2019) When we are to add navigation to the web application and by using a Router Outlet it would “change the content area of the page based on the user’s action.” (Fain & Moiseev, 2019) Basically the Router is what is behind the “view state” of the web pages. (Fain & Moiseev, 2019)

References:

Fain, Y., & Moiseev, A. (2019). *Angular Development with TypeScript* (2nd ed.). Manning.

*Angular*. (n.d.-b). Retrieved May 29, 2023, from https://angular.io/api/router/RouterOutlet#description

*Angular*. (n.d.-c). angular.io. Retrieved May 29, 2023, from https://angular.io/guide/router