Angular Dynamic Routing

Here’s a technique for implementing dynamic routing, where we want a common page or component to serve multiple route paths.

If you’re interested in passing query parameters through Angular routes, see this: Angular: Add Query Parameters without Router

Implementation

The technique, below, is from here: Angular Basics: Dynamic Activated Route Snapshots

We will use an Activated Route, in the target component, so it can recover path fragments from its route.

First. We will create a route to our component in the routing class. But, we will include a variable in the route string.

Here’s what ours will look like:

```
const routes: Routes = [
  {
    path: 'home',
    component: HomePageComponent,
  },
  // Below, is our dynamic route, with a dynamic suffix of 'dbname'.
  // Any calls to 'liveview/***', will open the LiveViewComponent.
  {
    path: 'liveview/:dbname',
    component: LiveViewComponent,
  },
  // redirect to `home` if there is no path
  {
    path: '',
    redirectTo: 'home',
    pathMatch: 'full',
```

```
  },
];
```

In the above route list, the `LiveViewComponent` will be called for any path that begins with ‘liveview’.

Second. We will update `LiveViewComponent`, to retrieve the ‘dbname’ portion of the route, so it can display the appropriate data.

To do this, we add ActivatedRoute to the constructor of LiveViewComponent, like this:

```
export class DashbLayout4Component {
  constructor(private route: ActivatedRoute)
  {
    ...
  }
}
```

With the ActivatedRoute in the constructor, our component can now retrieve the ‘dbname’ fragment from the url route, like this:

```
let dbname = this.route.snapshot.params['dbname'] ?? "";
```

If we add the above line to the ngOnInit or ngAfterContentInit, it will be able to load any special content, based on the ‘dbname’ value.

Revision #2
Created 22 March 2025 22:44:28 by glwhite
Updated 22 March 2025 22:48:58 by glwhite