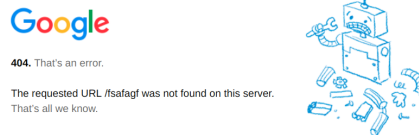


Tuesday: 404 Pages

404 Pages

We have all experienced times when we are web browsing and the browser responds with a 404 error of not finding the resource we are looking for which looks similar to this:



This is google's 404 page and different applications have their own customized 404 pages. It's very informative to a user as opposed to having a blank page. As much as it might be disappointing to the user, it improves the user experience. In severe cases, it keeps our application from breaking when a user requests for a resource that is generated by the app but isn't present and the occurrence of that error isn't mitigated by the app. Let's create a component for error pages in our application with this command on our terminal:

```
$ ng generate component not-found
```

In the template file of this component, let's include this code:

src/app/not-found/not-found.component.html

```
<div class="alert alert-danger" role="alert">
  <h1 class="text-center"><b>Sorry, we couldn't find what you're looking for</b></h1>
  <hr>
  <p class="text-center">Please check your URL!</p>
</div>
```

We have used a bootstrap alert component in the template file. We now need to inform our router to direct users to this component if they look for a resource that isn't found in the app.

src/app/app-routing.module.ts

```
...
import { NotFoundComponent } from './not-found/not-found.component';

const routes: Routes = [
  { path: 'goals', component: GoalComponent},
  { path: 'about', component: AboutComponent},
  { path: '**', component: NotFoundComponent},
];

...
```

At the top, we have imported our `NotFoundComponent`. We have then defined a route whose path has wildcards and directed this path to the `NotFoundComponent`. The wildcards define a route that is not present in our routes array. So in the case a user tries to look up a route that does not exist in the routes of our app, they will be taken to the not-found component. Let's fire up our servers and try requesting a route in the address bar that doesn't exist in our app to see our apps error page working.

Redirect

None of our routes caters for an empty path in our app. This is why when we load the path <http://localhost:4200> (<http://localhost:4200/>), only the navbar shows on our app. We can implement a redirect in our app so that when a user loads an empty path, they don't have an empty screen but instead see the goals. Let's configure a route for empty paths.

src/app/app-routing.module.ts

```
...
import { NotFoundComponent } from './not-found/not-found.component';

const routes: Routes = [
  { path: 'goals', component: GoalComponent},
  { path: 'about', component: AboutComponent},
  { path: '**', component: NotFoundComponent},

  { path: '', redirectTo: "/goals", pathMatch: "full"},
];

...
```

We have created an empty path with the `redirectTo` property that points to the path of the `GoalComponent`. A redirect requires the `pathMatch` property which tells the router how to match the URL to with the path given. In our case, the router will only redirect to the `/goals` path if the URL given is empty `' '`. If our server is still running and we load our application, we are redirected to the `GoalComponent` instantly.