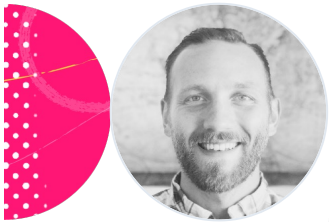


Implementing Lazy Loading for Performance



Zachary Bennett

Lead Software Developer

@z_bennett_ | <https://www.linkedin.com/in/zbennett10>



Why Lazy Loading with Routes?



Large apps can still load fast



It works well with a micro-frontend architecture



Application responsiveness is enhanced



It helps you scale!



```
export const routes: Routes = [  
  { path: '', redirectTo: 'home', pathMatch: 'full' },  
  {  
    path: 'home',  
    loadComponent: () => import('./home.component').then((m) => m.HomeComponent),  
  },  
  
  <snip>  
];
```

Loading Standalone Components

You can use the `loadComponent` field in the routes configuration to dynamically load a standalone component directly.



**Large apps can meet
scaling needs with the use
of Angular Guards!**



```
<snip>

{
  path: 'heavy-feature',
  canActivate: [canLoadHeavyGuard], // Apply the guard
  loadComponent: () => import('./heavy.component').then(m => m.HeavyComponent)
},

<snip>
```

Conditionally Load Components at Scale

Guards can help large applications determine whether or not they want to load a heavier, more resource intensive component/module or a more lightweight alternative.





Dynamic Imports with Components





Optimizing Dynamic Loading



**What about that delay that
can happen when I
dynamically load a large
component?**



Preloading Strategies

You can elect to preload certain components or even use custom strategies

routes.ts

```
{
  path: 'dashboard',
  loadComponent: () => import('./home.component').then(m => m.HomeComponent),
  data: { preload: true } // Mark for preloading
},
```



```
@Injectable({ providedIn: 'root' })  
export class CustomPreloadingStrategy implements PreloadingStrategy {  
  
    preload(route: Route, load: () => Observable<any>): {  
        <snip>  
    }  
  
    <snip>  
  
}
```

Custom Preloading Strategy

The above code demonstrates the scaffolding required when creating a custom preloading strategy.





Preloading and Debugging Bundle Sizes



Thank you for your time!

@z_bennett_ | <https://www.linkedin.com/in/zbennett10>

