

Route Guards



Deborah Kurata

CONSULTANT | SPEAKER | AUTHOR | MVP | GDE

@deborahkurata | blogs.msmvps.com/deborahk/





Protecting Routes with Guards



canActivate

- Guard navigation to a route

canActivateChild

- Guard navigation to a child route

canDeactivate

- Guard navigation away from a route

canLoad

- Prevent asynchronous routing

resolve

- Prefetch data before activating a route

Module Overview



Using Route Guards

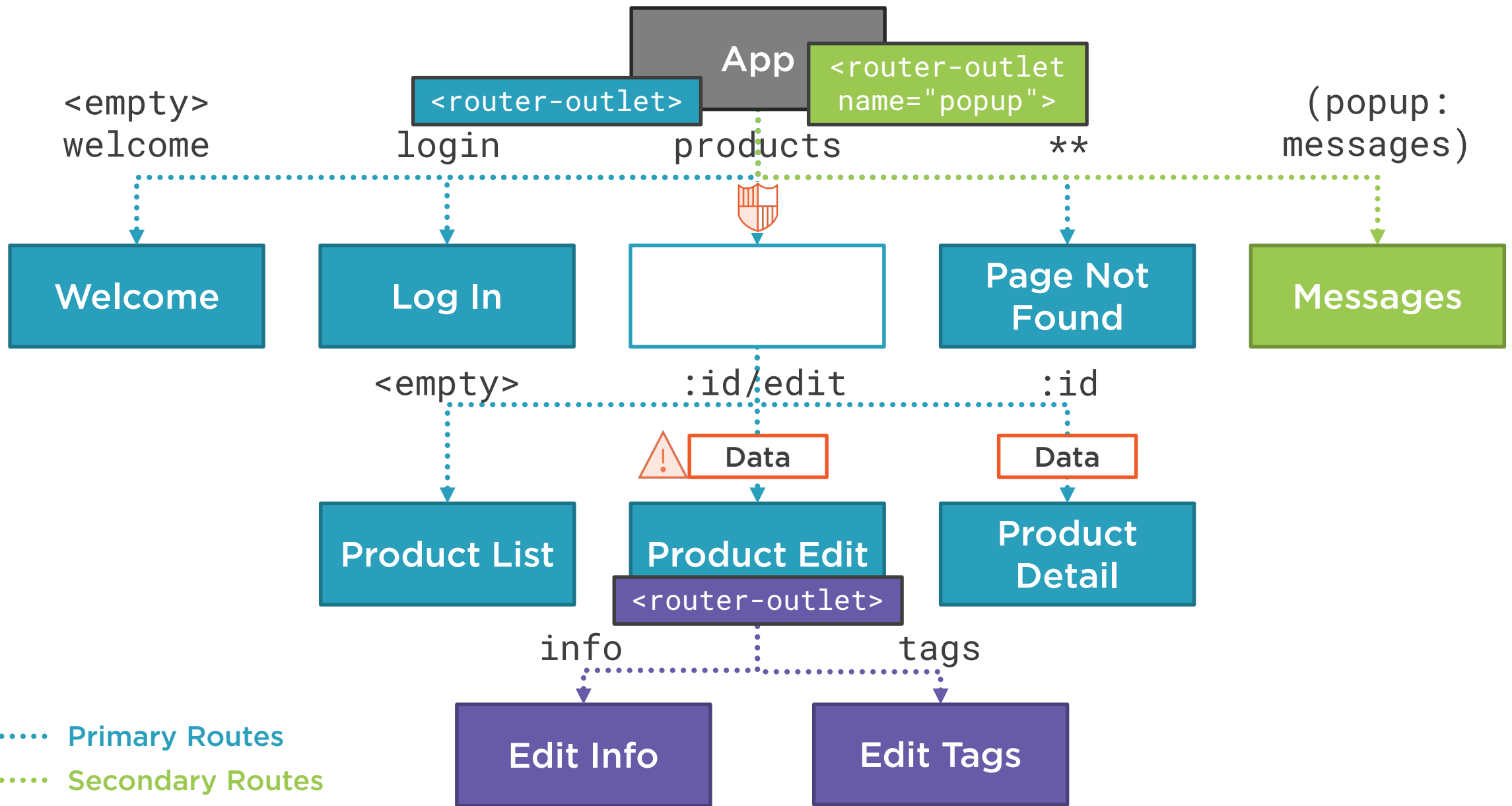
`canActivate` Guard

Sharing Data with a Guard

`canActivateChild` Guard

`canDeactivate` Guard

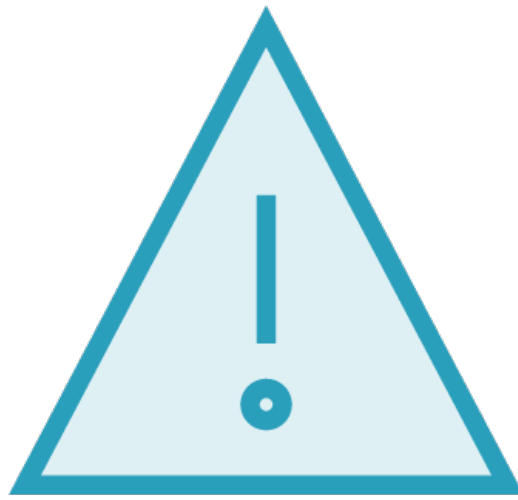




Using Route Guards



Limit access to a route



Warn before leaving a route



Retrieve data before accessing a route

Guard Processing

canDeactivate

canLoad

canActivateChild

canActivate

resolve



Building a Guard Service

auth-guard.service.ts

```
import { Injectable } from '@angular/core';
import { CanActivate } from '@angular/router';

@Injectable()
export class AuthGuard implements CanActivate {

    canActivate(): boolean {
        ...
    }
}
```



Registering a Guard

Angular Module

```
...  
import { AuthGuard } from './auth-guard.service';  
  
@NgModule({  
  imports: [...],  
  declarations: [...],  
  providers: [ AuthGuard, ... ]  
})  
export class UserModule { }
```



Guarding a Route

Angular Module

```
...  
import { AuthGuard } from './auth-guard.service';  
...  
  
{  
    path: ' :id',  
    component: ProductDetailComponent,  
    resolve: { product: ProductResolver },  
    canActivate: [ AuthGuard ]  
}
```



Sharing Guards

```
RouterModule.forChild([
  {
    path: 'products',
    children: [
      {
        path: '',
        component: ProductListComponent
      },
      {
        path: ':id',
        component: ProductDetailComponent,
        resolve: { product: ProductResolver }
      },
      {
        path: ':id/edit',
        component: ProductEditComponent,
        resolve: { product: ProductResolver },
        children: [...]
      }
    ]
  }
])
```



Guarding a Route



**Build a guard
service**

**Register the
service in an
Angular module**

**Add the guard
to a route**



canActivate Guard



Checks criteria **before activating a route**

Commonly used to:

- Limit route access to specific users
- Ensure prerequisites are met

Called when the Url changes to the route

Sharing Data

Route parameters

```
canActivate(route: ActivatedRouteSnapshot,  
            state: RouterStateSnapshot): boolean {  
    console.log(route.params['id']);  
    ...  
}
```

Route data

```
canActivate(route: ActivatedRouteSnapshot,  
            state: RouterStateSnapshot): boolean {  
    console.log(route.data['product']); // undefined  
    ...  
}
```

Service property

```
export class AuthService {  
    currentUser: IUser;  
    redirectUrl: string;  
}
```



canActivateChild Guard

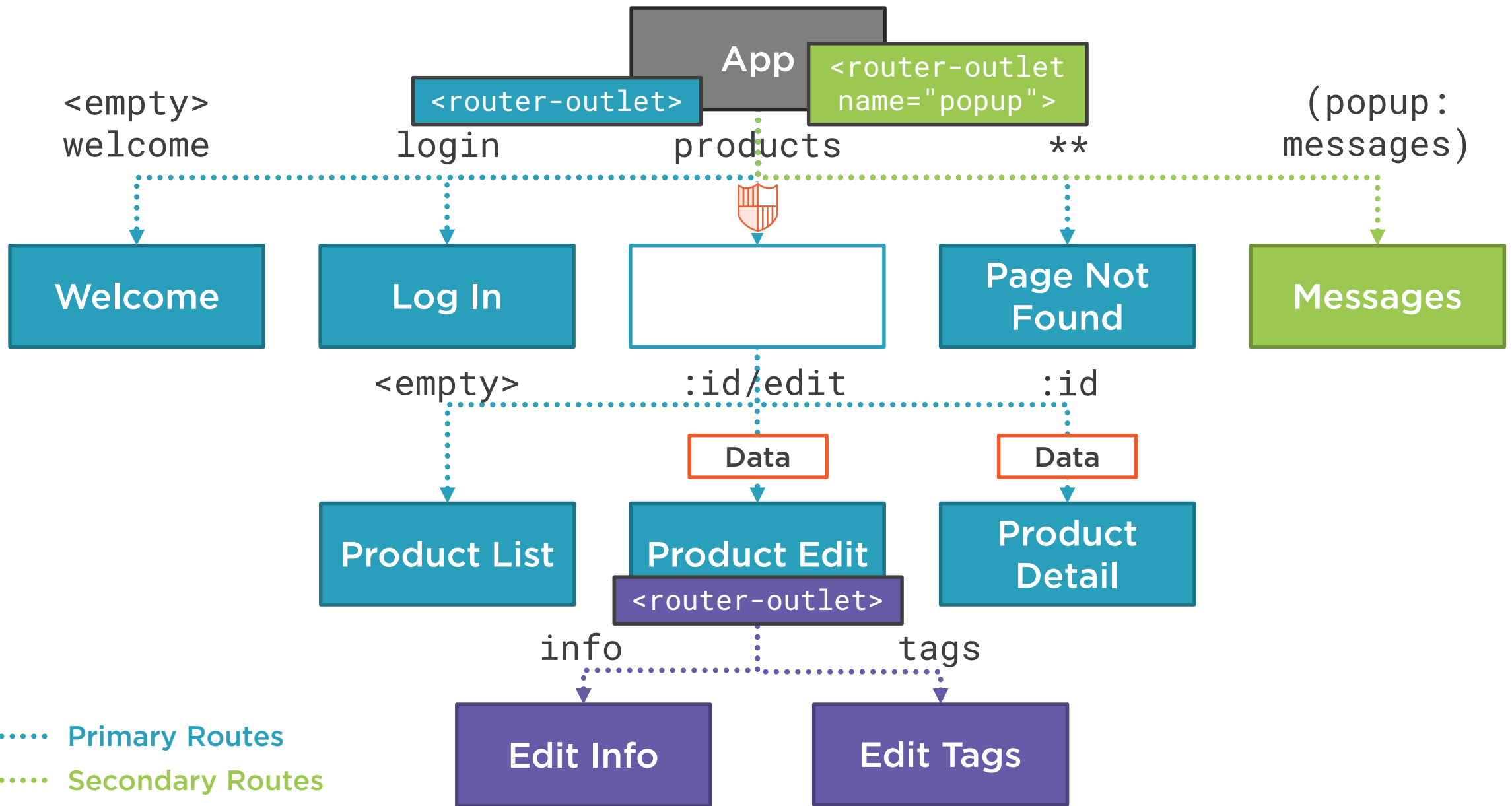


Checks criteria **before activating a child route**

Commonly used to:

- Limit access to child routes
- Ensure prerequisites for child routes are met

Called when the Url changes to the child route



canDeactivate Guard



Checks criteria **before leaving a route**

Commonly used to:

- Check for unsaved changes
- Confirm leaving an incomplete operation

Called when the Url changes to a different route



Route Guards Checklist: Building







Build a service

Implement the guard type

Create the associated method

Register the service provider

```
import { Injectable } from '@angular/core';
import { CanActivate } from '@angular/router';
import { AuthService } from './auth.service';

@Injectable({
  providedIn: 'root'
})
export class AuthGuard implements CanActivate {
  constructor(private authService: AuthService) {}

  canActivate(): boolean {
    return this.authService.isLoggedIn();
  }
}
```



Summary



Using Route Guards

`canActivate` Guard

Sharing Data with a Guard

`canActivateChild` Guard

`canDeactivate` Guard



