

Lazy Loading



Deborah Kurata

CONSULTANT | SPEAKER | AUTHOR

@deborahkurata | blogs.msmvps.com/deborahk/







Web Browser

Web
Server

URL Request (www.mysite.com)

Response

index.html

App Files





Web Browser

Web
Server

URL Request (www.mysite.com)

Response

index.html

App Files

Lazy
Loaded
Files



Module Overview



Building and Serving Our Files

Preparing for Lazy Loading

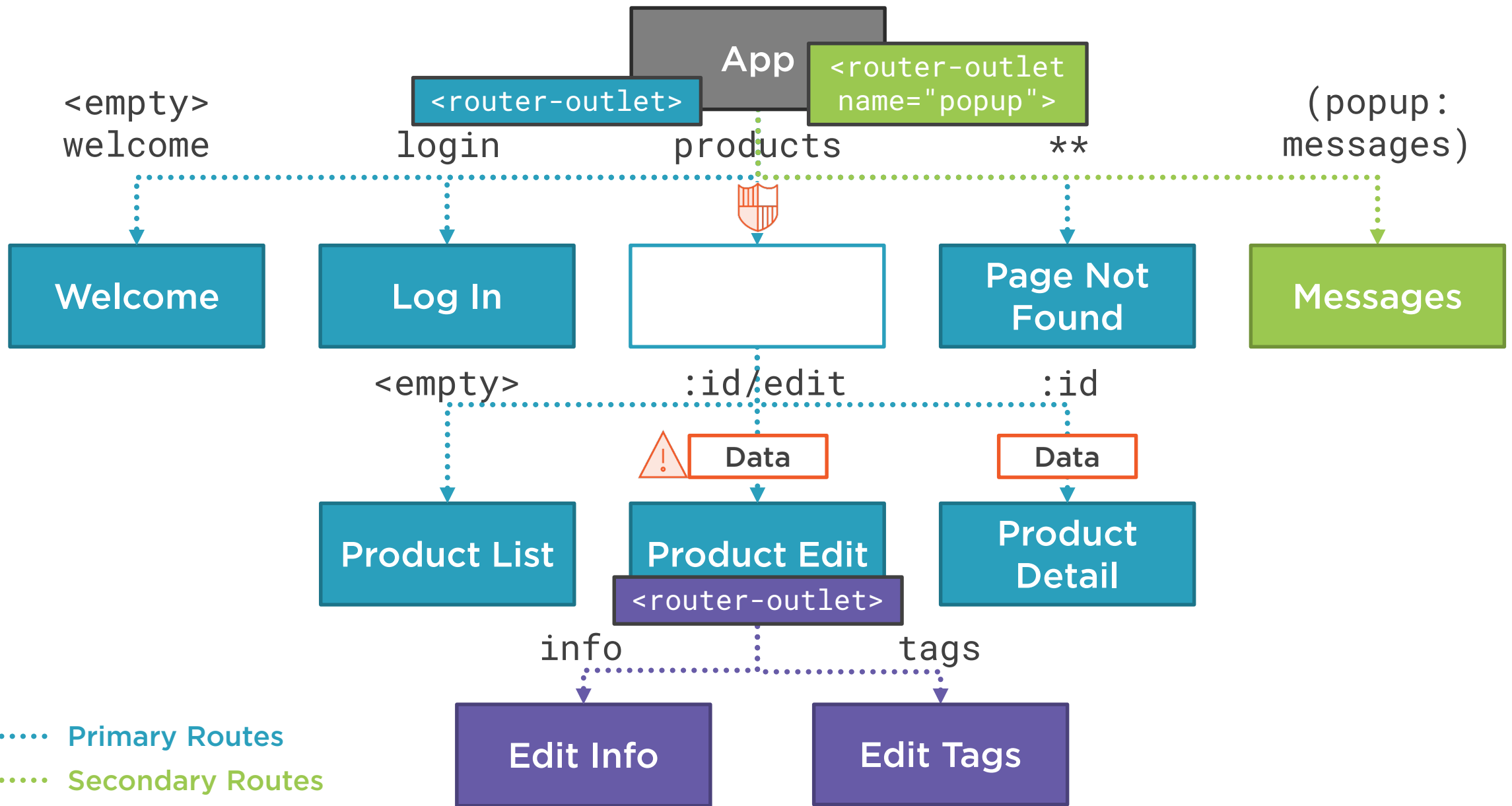
Lazy Loading

canLoad Guard

Preloading Feature Modules

Custom Loading Strategies





- Primary Routes
- Secondary Routes
- Child Routes



Demo



Building and Serving Our Files



Preparing for Lazy Loading

Use a feature
module

Routes grouped
under a single
parent

Not imported in
another module



Lazy Loading

app-routing.module.ts

```
RouterModule.forRoot([  
  ...  
  {  
    path: 'products',  
    loadChildren: './products/product.module#ProductModule'  
  },  
  ...  
])
```



Lazy Loading



Delay downloading until needed



Speed up initial start time



canLoad Guard



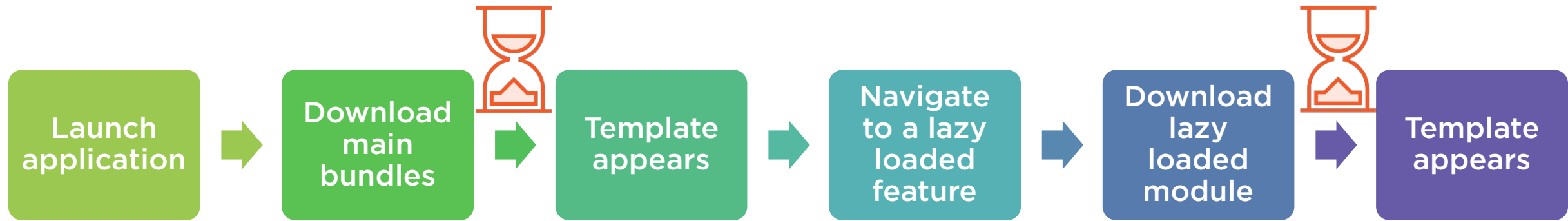
Checks criteria **before loading an asynchronous route**

Commonly used to:

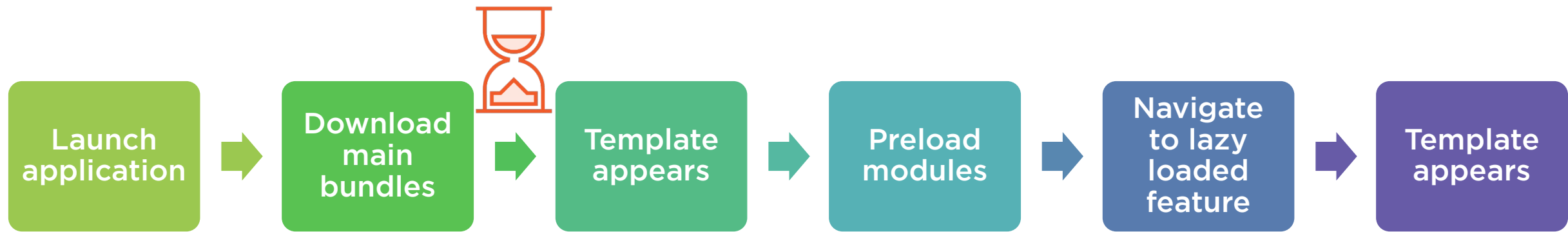
- Prevent loading a route if a user cannot access it

Preloading Feature Modules

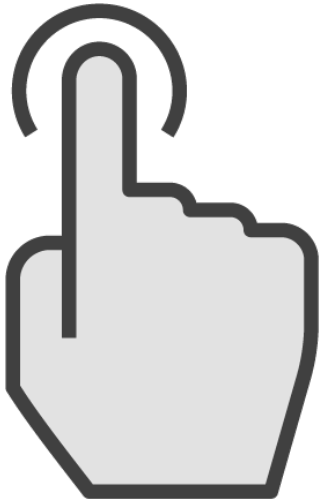
Lazy Loading



Preloading (Eager Lazy Loading)



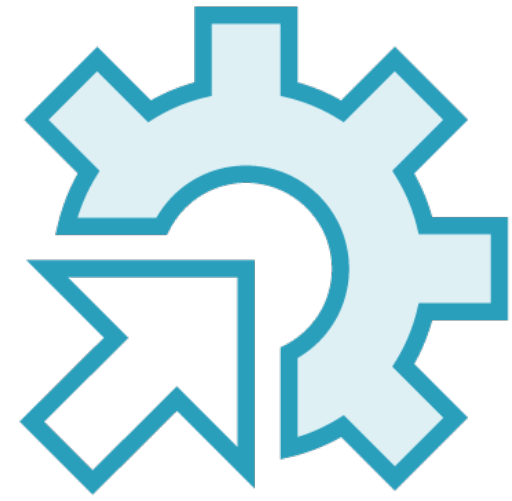
Preload Strategies



No preloading
(on demand)



Preload all



Custom

Preload All Strategy

app-routing.module.ts

```
import { RouterModule, PreloadAllModules } from '@angular/router';
...

RouterModule.forRoot([
  { path: 'welcome', component: WelcomeComponent },
  {
    path: 'products',
    loadChildren: './products/product.module#ProductModule'
  },
  ...
], { preloadingStrategy: PreloadAllModules })
```



Demo



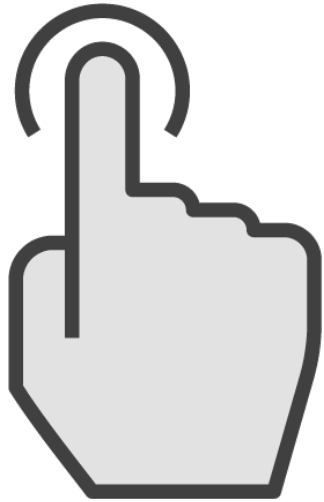
Preloading Feature Modules



canLoad guard
blocks
preloading



Preload Strategies



No preloading
(on demand)



Preload all



Custom

Custom Preloading Strategy

1. **Build a preloading strategy service**
2. **Set the preloading strategy routing option**



Building a Preload Strategy Service

selective-strategy.service.ts

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

@Injectable({
  providedIn: 'root'
})
export class SelectiveStrategy implements PreloadingStrategy {

  preload(route: Route, load: Function): Observable<any> {
    ...
  }
}
```



Enabling a Strategy

app-routing.module.ts

```
...
import { SelectiveStrategy } from './selective-strategy.service';

@NgModule({
  imports: [
    RouterModule.forRoot([
      { path: 'welcome', component: WelcomeComponent },
      {
        path: 'products',
        loadChildren: './products/product.module#ProductModule'
      }, ...
    ]), { preloadingStrategy: SelectiveStrategy } ),
])
})
export class AppModule { }
```



Lazy Loading Checklist: Preparing







Use a feature module

Group routes under a single parent

Don't import feature module



Lazy Loading Checklist: Configuring







Add the loadChildren property

Set it to:

- The full path to the module
- A hash
- And the module's class name

```
{  
  path: 'products',  
  loadChildren: './products/product.module#ProductModule'  
}
```



Lazy Loading Checklist: Preloading







Preload all modules with
`PreloadAllModules`

Or build a custom preloading strategy
service

```
RouterModule.forRoot([
  {
    path: 'products',
    loadChildren: './products/product.module#ProductModule'
  },
  ...
], { preloadingStrategy: PreloadAllModules })
```

Summary



Building and Serving Our Files

Preparing for Lazy Loading

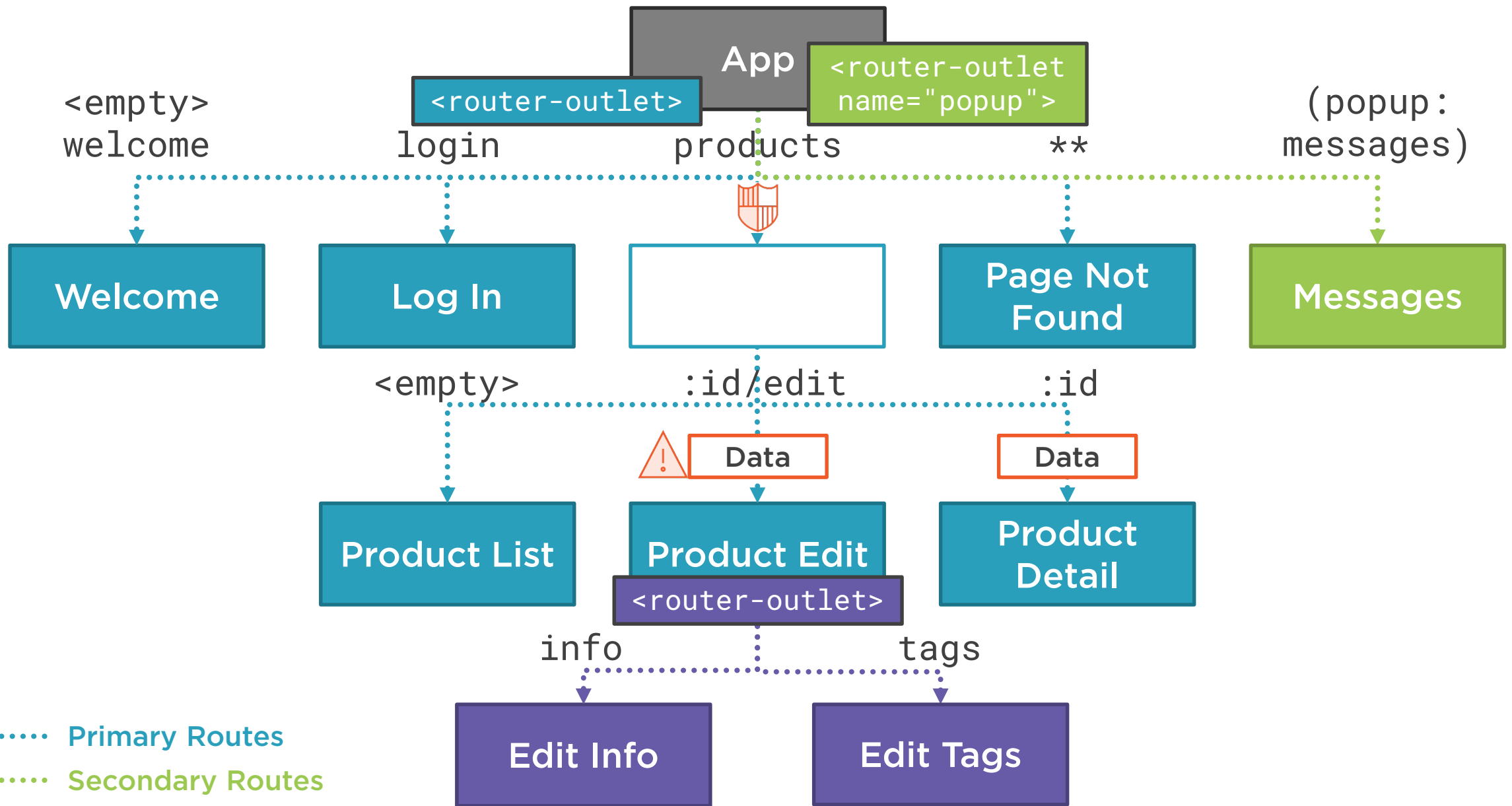
Lazy Loading

canLoad Guard

Preloading Feature Modules

Custom Loading Strategies





- Primary Routes
- Secondary Routes
- Child Routes

