



# Dynamic configuration

Vojtech Mašek

Head of engineering

@ flow<sup>up</sup>



/ vmasek



/ VojtechMasek



/ @vmasek



# Build time configuration

flow<sup>up</sup>

- App has to be re-build for each config
- Built in (file replacement, DI, imports)
- Easy process.env integration *with extra webpack config*
- No runtime “slowdown”



# Dynamic configuration

- Angular has its own mechanics called APP\_INITIALIZER
- We can also provide static provider when bootstrapping
- Simple way to load some data before your app bootstraps
- It is blocking the app from being rendered

- Really useful in some situations
  - ◆ Fetching app configuration
- Harmful in some others
  - ◆ Waiting for too long or too many requests
  - ◆ Doing what could easily be managed during runtime



# How to provide APP\_INITIALIZER <sup>flow<sup>up</sup></sup>

```
1 @NgModule({
2   /* ... */
3   providers: [
4     {
5       provide: APP_INITIALIZER,
6       useFactory: (init: InitService) => () => init.load(),
7       multi: true,
8       deps: [InitService],
9     },
10  ],
11 })
12 export class AppModule {}
```



# Initialisation using the config

flow<sup>up</sup>

```
1 @Injectable({ providedIn: 'root' })
2 export class InitService {
3
4   constructor(private store: Store<State>, private http: HttpClient,) {}
5
6   async load(): Promise<Config> {
7     return this.http.get<Config>('/config').pipe(
8       tap(config => this.initApolloClient(config)),
9       tap(config => this.store.dispatch(storeConfigAction(config))),
10    ).toPromise());
11  }
12
13  async initApolloClient(config: Config): Promise<void> {
14    this.apollo.setClient(new AWSAppSyncClient(config));
15    return client.hydrated();
16  }
17 }
```



# Providing fetched config before bootstrap

flow<sup>up</sup>

```
1 // main.ts
2 fetch('/assets/config.json').then(async (response: Response) => {
3   const config: Config = await response.json();
4   // environment.config = config;
5   return platformBrowserDynamic([
6     {
7       provide: USE_DOMAIN,
8       useValue: config.domain,
9     },
10  ])
11    .bootstrapModule(AppModule).catch(err => console.error(err));
12 });
```



# Choose the right setup

Don't wait with bootstrap if not necessary



# Q&A

Vojtech Mašek

Head of engineering

@ flow<sup>up</sup>



/ vmasek



/ VojtechMasek



/ @vmasek