Inject a service with constructor in the main.ts

Asked 7 months ago Active 7 months ago Viewed 331 times

0

In my Angular project, I am using AWS Amplify to handle my authentication. Now, I want to configure it to manage tokens with cookies.

After reading the documentation I found out that I need to provide my own storage to Amplify in the `main.ts` .

So I installed `ngx-cookie` and started my mapping between the library methods and the AWS Amplify signatures like this :

main.ts

```
class MyStorage {

  constructor(private cookieService: CookieService) { }

  setItem(key: string, value: string) {
    this.cookieService.put(key, value);
  }

  getItem(key: string): string {
    return this.cookieService.get(key);
  }

}

Amplify.configure({
  ...amplify,
  Auth: {
    oauth: oauth,
    cookieStorage: {
      domain: 'localhost'
    },
    storage: new MyStorage(), <--- expect a cookieService here
  }
});
```

Problem: `new MyStorage()` expect a cookieService in argument and it seems not possible to add in constructor in the `main.ts` .

Question: How to pass a service in the `main.ts` ?

Thanks for your help.

angular amazon-cognito aws-amplify

asked Dec 10 '18 at 21:59

Johan Rin
987 1 9 28

1 Answer

1

✓

Well, the `main.ts` is the entry point for the Angular application. Here, we're not able to query instances of services or get them injected because we're outside of the Angular application. It looks like that `Storage` object is completely decoupled from the Angular application as well, so any dependency injection you're after you'd have to do yourself.

Also of note: while I can't find any explicit documentation on this, Amazon's example of this Storage class has all of its properties as static, so I'm not sure if the configuration is expecting an instance of a class at all.

One thought I had would be to hijack `APP_INITIALIZER` in order to get DI tokens when the app is first bootstrapping. However, I am not familiar with Amplify, so I don't know *when* it needs to be configured, so this may or may not work:

App.module.ts

```
providers: [
  {
    provide: APP_INITIALIZER,
    useFactory: amplifyConfigFactory,
    deps: [CookieService],
    multi: true
  }
]
```

amplifyConfigFactory.factory.ts

```
export function amplifyConfigFactory(cookieService: CookieService) {
  Amplify.configure({
    // ....
    storage: new MyStorage(cookieService)
  });
  return () => Promise.resolve(); // factories must return a Promise
}
```

so, above our `deps` entries get instantiated and injected into the factory in order, and we can create a `MyStorage` with the `cookieService` from the Injector.

Again, if the properties on MyStorage must be static, you're going to need a different solution here. It seems to me like `MyStorage` should not be something known to the Angular application, but depending on whether or not Amblify needs to be configured before the `APP_INITIALIZER` hook, this could work.

answered Dec 10 '18 at 22:32

joh04667
4,237 10 26