

# Dependency Injector

## 4. DI Decorators - Fine tuning the DI search algorithm

# Fine tuning the DI Search algorithm

- We use decorators to further control how the DI search algorithm will work
- For example:

```
constructor(@SomeDIDecorator() private _http: HttpClient) {}
```

# @Optional()

- If you didn't find the requested item in all the injectors and in all the trees, just assign null to the variable (the default behaviour is to throw exception)

```
constructor(@SomeDIDecorator() private _http: HttpClient) {}
```

# @Self() / @SkipSelf

- @Self - Will look for the data only in the component / directive element injector and will not traverse the tree
- @SkipSelf- will start the search from my parent

```
constructor(@Self() private _http: HttpClient) {}
```

```
constructor(@SkipSelf() private _http: HttpClient) {}
```

- If I'm a projected component (ng-content) will limit the search up to my host
- Or if I'm a directive placed on a component, will limit the search up to the component host

```
constructor(@Host() private _http: HttpClient) {}
```

# Thank You

**Next Lesson: 5. Providers**