Models & Injection

Want to easily handle your raw data?

Or simply, why we want models in angular?

The model in an application is responsible for modeling data used in the view .

Let’s see the need of models: Nowadays handling raw JSON objects is hard to maintain can be done using Models.

Models are extensible. Good reason for using models is that we're working with Typescript. We want to know the type of things we use and can’t just define everything as any. In combination with a good IDE this makes work easier.

We can easily understand models with an example:

This is a User model having firstName and lastName.

If we are handling the raw JSON we'll have to print out the full name of your user within our templates like this:

If customer tells to switch the order of firstname and lastname. If our user is a User object then we need not to go through every template and switch the expressions. we can implement a function to print the fullname:

we can now call this function in our template.

Now let’s move to Dependency injection

What is Dependency injection?

Why we Need Dependency injection?

Dependency injection is the ability to add the functionality of components at runtime.

Dependency injection is an application design pattern.

Angular has its own DI framework, which is used in the design of Angular applications to increase their efficiency and modularity.

In Angular, the DI framework provides declared dependencies to a class when that class is instantiated

Example: injector is basically like a container of all the dependencies like engines, tires dependency A, B and so on and if we want a car we can ask for a car and the injector will provide a car for us the DI framework will manage all the dependencies so that we don't have to keep track of it.

The DI framework makes the developers job much easier.