**106. Creating a Logging Service**

* -: As mentioned at the end of the last lecture, I wanna start by building our first Service here.
* *A Service which takes over this logging task here.*
* So a Service where we can log data.
* Now, how do we create a new Service? How does a Service look like in Angular? Creating it is super simple.

1. First we have to choose a fitting place; Now, since this Service will do some generic task I will place it in the App folder. You could also create a shared sub folder there.

* I will again place it right in the App main folder, and I will simply name it "logging.
* " And the file name should be "logging.service.ts.
* " To follow this convention of being pretty clear about what's in a file.
* So here, this file will hold a Service.
* So that's why I'll name it like this.

b) There, I will export a TypeScript class, and I will name this class "LoggingService" too; again, to be explicit about what this class does.

* Now, a Component becomes a Component because we attach this "@Component" decorator.
* A Directive becomes a Directive because we attach "@Directive.
* " You could think that we now simply add "@Service," right? But we don't do that.
* A Service is just a normal TypeScript class.
* Like this! This can already be used as a Service.

Text

Description automatically generated with medium confidence

* So here, we could create a helper method "logToConsole," and simply pass the message.
* But, since we know that we simply want to log the change of, uh, account status here, I will simply say: "logStatusChange" like this, and take the new status which will be a string as an argument.
* Here, I can then simply copy the code I do have in one of my components right now, this "console log" code here, and copy it into my "loggingService" here.
* Of course, "accountStatus" now is just "status" here.
* This argument we're getting.
* So with that, remove the colo.
* .
* .
* semicolon, with that I centralized this code.

c) Now, we need to use this Service in our our Components.

* So everywhere where we "console log.
* " Now, you could think that we simply use it by importing it at the top of our file.
* So here, we could import the "loggingService" from, and now, let's move up to the main folder, and then here from the "logging.
* service.
* ts" file, and then, we could simply create our Service like this: "new LoggingService.
* " And on our Service here, we could call: "logStatusChange," and pass the new "accountStatus.
* " And actually, if I get rid of this code here, this will work! Let it recompile, and if I now create a new Service here, you see I do log it to the console.

Text

Description automatically generated

Graphical user interface, text

Description automatically generated

* *So this would work, and still, this is wrong! This is not, and this is super important, this is not how you use a Service in Angular.*
* *The reason for this will become more clearer throughout this module, but generally Angular provides a much better way of getting access to your Services, and therefore, you should not create the instances manually.*
* So let's get rid of that import, and of that instantiation, and let's learn in the next video which tool Angular offers us to get access to our Services.