# Lab S3: Dependency Injection

## Objectives

In this lab, you will configure a dependency injection between the ITGuru bean and a new bean called TechGuide which implements a Manual interface.

## Lab Setup

This lab builds on the previous lab, no additional setup is required. You can start coding from the final state of the code from the last lab.

## Modifying the Java Components

1. In the previous labs, the Consultant objects just returned a string. In this lab you will implement a TechGuide type that the ITGuru objects do a lookup on to produce their advice.
2. There will be a constuctor injection used to create a link between the two objects. This will be indicated by an @Autowired annotation.
3. In terms of lifecycles, each ITGuru bean is a reference to a unique instance, but the TechGuide is a singleton object which means that all the TechGuru objects use the same instance of TechGuide

### Step 1: Add the Manual interface

Graphical user interface, text, application

Description automatically generatedAs shown below

### Step 2: Add the TechGuide implementation class

Graphical user interface, text, application

Description automatically generatedAs shown below.

### Step 3: Modify the ITGuru class

1. There are three changes to be made to the class
2. The first is to add a private instance variable of type Manual and a constructor that takes a reference to a Manual and initializes the variable to the reference passed.
3. The second is to change the giveAdvice() method to return the results of lookup() on the Manual.
4. Graphical user interface, text, application

   Description automatically generatedThe third is to add the @Autowired annotation to the constructor to tell Spring to resolve the dependency when the ITGuru bean is created

### Step 4: Modify the App

1. Create two ITGuru beans and show that the getAdvice() method works on both of them.

Graphical user interface, text, application

Description automatically generated

### Step 5 (OPTIONAL)

1. In the project, the ITGuru class asked for a bean that implemented the Manual interface. Spring knew to create the TechGuide bean because it was the only class that implemented the Manual interface.
2. But what if there are two classes that implement the Manual interface?
3. Copy the TechGuide file and rename the copy UserGuide. Now there are two implementations of the Manual class.
4. Run the code again and see what happens. Can you explain the results?
5. Change the constructor in ITGuru and the private instance variable to be of type TechGuide and then rerun the code. Can you explain the results?