



# Lab 7.1

## Using Conditionals for different environments

On the existing BeanDefinition that returns StudentServiceImpl only, add a `@ConditionalOnExpression` where an environment variable “env” equals “prod”



On the StudentServiceProxy BeanDefinition, add a `@ConditionalOnExpression` where an environment variable “env” equals “test”



In your application.properties file add a key value pair of;  
env=test

- Execute your JUnit; you should see log statements along the lines of
  - INFO: Calling Service getAllStudents()
- 

Change the properties file value for the entry “env” to “prod”, rerun your test, no logs this time

## Lab 7.2

### Custom Conditional

In the lab setup directly for this lab are two classes, `com.student.config.StudentConditional_Prod` and `com.student.config.StudentConditional_Test.java`. Copy these beans into the respective package in your project

In `ApplicationConfig`, change your conditionals to use these custom conditional beans. Use the `@Conditional` annotation with the desired class

Rerun your test for “test” and “prod” scenarios. Again, one run has logs and one does not