# Spring Boot Techniques

## Overview



In this lab you’ll enhances your “online retailer” application from the previous lab, to make use of application properties.

## IntelliJ starter project

If you’re happy to continue where you left off in the previous lab, use the following project:

* **student\student-online-retailer**

If you’d prefer a fresh start, use the solution project from the previous lab instead:

* **solutions\solution-configuration-classes**

## IntelliJ solution project

The solution project for this lab is located here:

* **solutions\solution-spring-boot-techniques**

## Roadmap

There are 4 exercises in this lab, of which the last exercise is “if time permits”. Here is a brief summary of the tasks you will perform in each exercise; more detailed instructions follow later:

1. Adding properties in **application.properties**
2. Adding properties in **application.yml**
3. Adding some profile-specific properties
4. (If time permits) Using Spring Actuator

## Exercise 1: Adding properties in application.properties

Define and use some simple properties in **application.properties**. For example:

* Define a property named **contactEmail**, to hold the company's email address.
* Inject this value into your **CartServiceImpl** class.
* Write some client code to verify **CartServiceImpl** picks up the value correctly.

## Exercise 2: Adding properties in application.yml

Define and use some complex properties in **application.yml**. Here are the properties we’d like you to define:

**Name of property Value Description**

onlineRetailer:salesTaxRate 0.20 Sales tax rate.

onlineRetailer:deliveryCharge:normal 2.50 Delivery charge (GBP) on normal deliveries.

onlineRetailer:deliveryCharge:threshold 3000 Threshold cart value (GBP), for free delivery.

## Inject these values into your CartServiceImpl class. Then write some client code to verify CartServiceImpl picks up the values correctly.

## Exercise 3: Adding some profile-specific properties

Define some properties that have different values depending on the current profile. For example:

**Name of property Value if "development" profile Value if "production" profile**

resources.db H2 Oracle

resources.logs C:\temp\logs\ //PROD\_SERVER/logs/

resources.secure false true

Define a component class named **ResourcesBean** and inject these properties into it. Then run the application, first in "development" profile and then in "production" profile, and verify the correct values are injected into the **ResourcesBean**.

## Exercise 4 (If time permits): Using Spring Actuator

Add support for Spring Actuator in your application and explore the information made available via Spring Actuator (e.g., health, metrics, beans, etc.).