# Spring Boot class

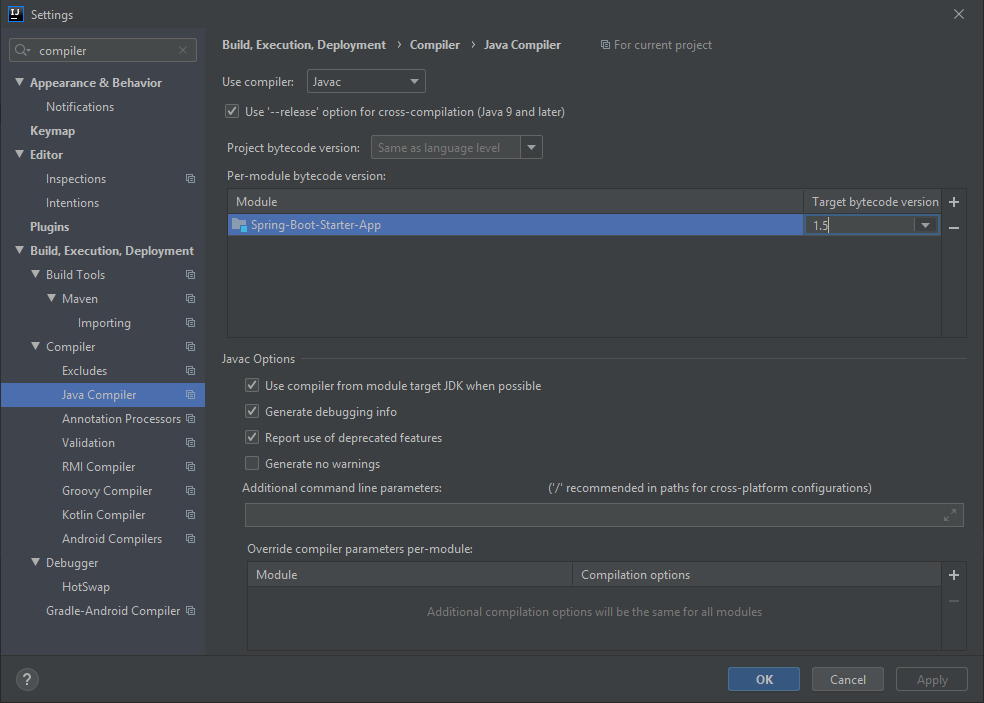
Week One



* Intellij recommended

1 - Change the maven settings file to the maven settings from git

2- Change project version to Java 8



3- Add data storage service in a database using a repository.

4- Add relevant values to item repository

5- Create a person repository to hold information related to the user that’s buying the items.

6 – Create a new ” CreateUserRequestDto” and “CreateUserResponseDto” with relevant properties to create and save users that buy items. Using Lombok, to avoid have to manually create constructors, getters and setters, with the relevant annotations.

7 – Make RestcontrollerAdvice return properly handled errors with buildErrorMessageResponseEntity() method like: “public ResponseEntity<ErrorMessage>”;

8 – Add a new Logger to all classes using the lib “org.slf4j”.

9 – Log relevant information using distinct logger levels.

10 – Update the “com.training.springbootbuyitem.service.ICrudService”, add possibility to load and update list of items.

11 – Remove unused imports from the classes.

12 – Add session info and create a new protect rout for logged users. Block access to anonymous customers.

13 – Add relevant postman requests to test the solution.

**Week two:**

14 – Add all the needed configs to access the H2 interface on the browser

15 – Add persistence database config to save the data using a local database.

16 – Implement a CustomSession Manager to manage the data on the session to avoid load of SQL info at every application start.

17 – Implement the Builder and fix the commented method in com.training.springbootbuyitem.controller.RestControllerAdvice#buildErrorMessageResponseEntity to return custom error messages.

18 – Add filter to avoid null parameters and to control the accepted values of variable using custom validators: ex: a DateValidator that accepts only one date format.

public class DateValidator extends …<BuyDate> {  
  
 private final DateFormat DATE\_FORMAT = new SimpleDateFormat("dd-MM-yyyy");  
  
…  
 @Override  
 protected boolean doIsValid(String value, ConstraintValidatorContext context) {  
..

DATE\_FORMAT.parse(value);  
…

}

19 – Use methods that requires and accept only authenticated users requests using the annotation - @RequiresAuthentication.

20 – Implement a Role based session manager.

21 – Implement an entry point that clear all the sessions for all the clients, this should be accessible only by admin roles.

22 – Implement an entry point to clear the session for the authenticated user. It should return forbidden for invalid users access.

23 – Implement a cache system using hazelcast.