**Module 1: Spring Foundation**

This Document proves task execution supplying with screenshots of given results.

**Task 1 - Spring Configuration**

Prerequisites:

1. Spring Boot Application has entity Address and CRUD repository (Spring Data JPA) configured for H2 in-memory local database.
2. Unit Test tries ‘SAVE’ method providing Address entity generated as test data.
3. Having extracted all data from database, unit test verifies that just one record selected and compares that with initial test data to check correctness.

Address Repository ‘SAVE’ Test execution:

A screenshot of a computer

Description automatically generated

**Task 2 - Conditional Configuration**

In frames of the task application.yaml file has been updated with properties namespace ‘custom’ with stands for custom datasource switching on and off . The datasource differs from standard one with ‘url’ address.

A screenshot of a computer

Description automatically generated

To handle the configurations there are two data source building configurations created. Property condition “custom.datasource.enabled” controls with configuration builder to invoke.

Also, value injection is applied to bring custom datasource url value used for custom datasource bean builder.

Here below are screenshots from verification of running application custom datasource on and off states.

1. Custom Datasource is enabled: (url ends with ‘custom’ )

A screenshot of a computer

Description automatically generated

1. Custom Datasource is disabled:

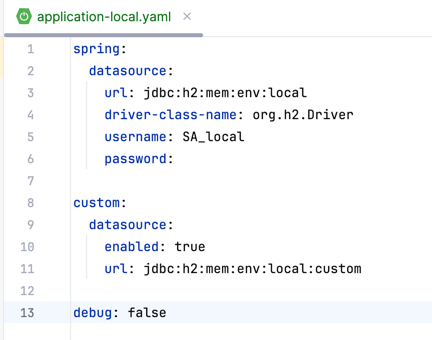
A screenshot of a computer

Description automatically generated

**Task 3 - Spring Profiles**

There were two profiles added to application:

1. local
2. dev

 A screenshot of a computer

Description automatically generated

Once gradle is used it’s simple to enhance bootRun task to accept spring active profile.

Using correspondent argument in gradle bootRun command we can easily switch among profiles:

A screenshot of a computer

Description automatically generated

After run with provided argument we get the following log at Spring Boot Application start:

A screenshot of a computer

Description automatically generated

Having change active profile from local to dev:

A screenshot of a computer

Description automatically generated After execute bootRun command with pointed dev profile we get the following log:

A screenshot of a computer

Description automatically generated

Having supplied tests class with own profile ‘test’, we can run tests with pointing of active profile and database connection will be utilized from the profile:

A screenshot of a computer

Description automatically generated