**Task 02 || Spring Boot, ORM (JPA, Hibernate)**

1. **Duration:** 3 working days
2. **Description:**

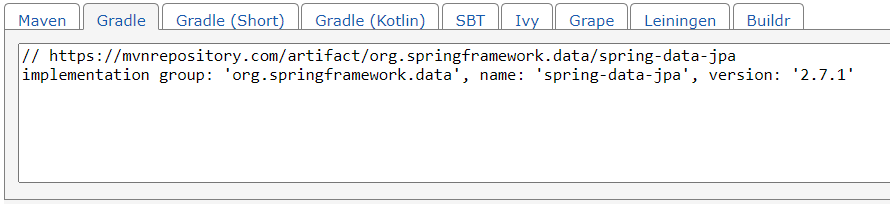
* You have seen how CRUD operation works and can be build using Spring Boot, data accessing using Java Persistence API(JPA) in task 01 and how to bootstrap a ready to deploy spring boot project from spring initializer which is also available in Intellij Idea.
* In task 01 you have seen Maven was used as build tool and dependency manager. In this course we will use Gradle. Don’t worry, Gradle is a build automation tool like Maven and easy to use. Just think you will use build.gradle file in replace of pom.xml for now.

Ref:

1. <https://docs.gradle.org/current/userguide/what_is_gradle.html>

2. <https://www.youtube.com/watch?v=-dtcEMLNmn0&ab_channel=TomGregory>

* Along with Maven, Gradle dependencies are always available in MavenRepository. Ex:



* In this task we will start our spring boot project 01 for this course. It will be a simple book info management tool.
* Get the project 01 template which is a ready to run spring boot project with gradle

1. clone project 01 repository to local:

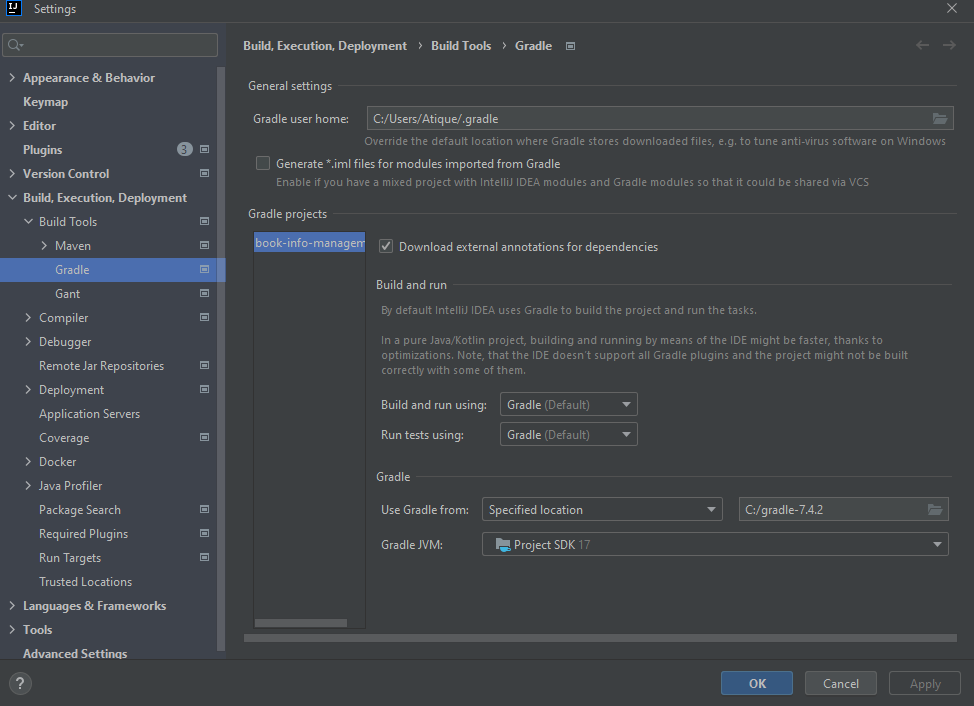
For SSH*: git clone git@github.com:atique7465/book-info-management.git*

For HTTP*: git clone https://github.com/atique7465/book-info-management.git*

2. fetch branches: *git fetch*

3. checkout develop branch: *git checkout bim/develop*

4. Set Gradle and JDK in idea as follows



5. Sync gradle

6. create your own release branch: *git checkout -b bim/atique-202015/release-1.0*

*[Note: use your nick name and emp\_id in place of “atique-202015”]*

7. create a directory named "documents" under "book-info-management" root directory like:

*book-info-management/documents*

8. keep your task 01 doc file into this documents directory like:

*/documents/atique\_202015\_task\_01.docx*

9. add your doc in git in the release branch: *git add .*

10. commit your changes: *git commit -m "[bim] task 01 doc upload"*

11. push you changes in release branch: *git push*

12. create feature branch for task 02 from release branch:

*git checkout -b feature/bim/atique-202015/task-02*

*[Note: use your nick name and emp\_id in place of “atique-202015”]*

* You will write simple CRUD operation to keep book info in your feature branch.
* CRUD Description:

1. Create Book:

|  |  |
| --- | --- |
| URI | /api/v1/book |
| Desc | POST 'localhost:8084/book-info-manager-1.0/api/v1/book'  Req:  {  "id" : 1,  "bookName" : "Kobi",  "bookType" : "NOVEL",  "author" : "Humaun Ahmed"  }  Res:  {  "id" : 1,  "bookName" : "Kobi",  "bookType" : "NOVEL",  "author" : "Humaun Ahmed"  }  *[Note: id (Long), bookName (String), bookType (Enum), author (String)]* |

1. Get Book:

|  |  |
| --- | --- |
| URI | /api/v1/book/{id} |
| Desc | GET 'localhost:8084/book-info-manager-1.0/api/v1/book/1'  Res:  {  "id" : 1,  "bookName" : "Kobi",  "bookType" : "NOVEL",  "author" : "Humaun Ahmed"  } |

1. Update Book:

|  |  |
| --- | --- |
| URI | /api/v1/book/{id} |
| Desc | PUT 'localhost:8084/book-info-manager-1.0/api/v1/book'  Req:  {  "id" : 1,  "bookName" : "Himu",  "bookType" : "NOVEL",  "author" : "Humaun Ahmed"  }  Res:  {  "id" : 1,  "bookName" : "Himu",  "bookType" : "NOVEL",  "author" : "Humaun Ahmed"  } |

1. Delete Book:

|  |  |
| --- | --- |
| URI | /api/v1/book/{id} |
| Desc | DELETE 'localhost:8084/book-info-manager-1.0/api/v1/book/1' |

* Create a directory named: controller

under *src/main/java/com.example.bookinfomanagement* for RestControllers

* Create a directory named: service

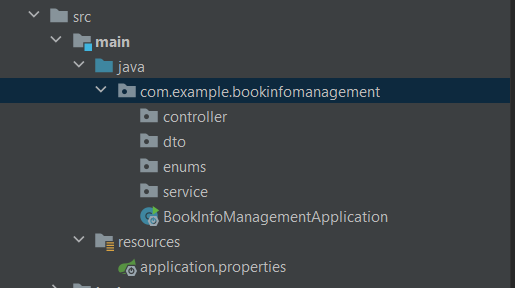
under *src/main/java/com.example.bookinfomanagement* for Services

* Create a directory named: dto

under *src/main/java/com.example.bookinfomanagement* for DTO’s

* Create a directory named: enums

under *src/main/java/com.example.bookinfomanagement* for Enums



* You can use a list/map variable in your service object to store the book info for this task. No need to connect any DBMS.

1. **Helper Project:**

* This is an already done project to help you.
* A simple CRUD operation to manage student info is done in this project.

1. clone helper project repository to local:

For SSH*: git clone git@github.com:atique7465/spring-boot-helper-project.git*

For HTTPS*: git clone https://github.com/atique7465/spring-boot-helper-project.git*

2. fetch branches: *git fetch*

3. checkout branch: *git checkout "hp/student-info-manage"*

4. Description can be found in README.md file.

5. Postman collection kept in documents folder.

* You can see the project structure and browse the code.

1. **What to submit**

* Prepare a doc/pptx with the answer of these questions:

1. How DispatcherServlet work?

ref: <https://www.baeldung.com/spring-dispatcherservlet>

1. How Spring MVC work?

Ref:<https://docs.spring.io/spring-framework/docs/3.2.x/spring-framework-reference/html/mvc.html>

1. How singleton pattern work?
2. How java bean work?
3. Write about the dependencies we used in *build.gradle* file
4. Why we used Lombok?
5. Write about the properties we used in application.properties file
6. Write about the annotations we used:

@SpringBootApplication, @RestController, @RequestMapping, @Autowired, @PostMapping, @GetMapping, @PutMapping, @DeleteMapping, @Service, @AllArgsConstructor, @NoArgsConstructor, @Data, @NotNull, @NotEmpty

* Submit your code for task 02 in GitHub. See how to submit section.

1. **How to submit**

* Name your doc/pptx as: <nick name>\_<emp\_id>\_<task id>.

Ex: *atique\_202015\_task\_02.docx*

* Upload your doc/pptx in documents folder
* commit your changes in your task 02 feature branch:

*git commit -m "[bim] task 02"*

* Push the changes to your feature branch: *git push*
* Create a merge req in GitHub [your feature branch → your release branch]

Ex: *feature/bim/atique-202015/task-02 → bim/atique-202015/release-1.0*

* Follow up instructors’ feedback on GitHub merge request.