1. Create a JDBC In-Memory Data project using the H2 Driver and the Snippet Code Manager source.

* Project Information:
* **Group**: io.pivotal.workshop
* **ArtifactId**: code-snippet-manager-jdbc
* **Dependencies**: web, jdbc, h2, devtools
* To create the project you can use the Spring Initializr or the STS IDE (***File → New → Spring Starter Project***).
* Notice the dependencies added to the pom.xml file:
* ***h2***
* ***spring-boot-starter-jdbc***

**<?**xml version="1.0" encoding="UTF-8"**?>**

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>io.pivotal.workshop</groupId>

<artifactId>code-snippet-manager-jdbc</artifactId>

<version>0.0.1-SNAPSHOT</version>

<packaging>jar</packaging>

<name>code-snippet-manager-jdbc</name>

<description>Demo project for Spring Boot</description>

<parent>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-parent</artifactId>

<version>1.4.0.RELEASE</version>

<relativePath/> *<!-- lookup parent from repository -->*

</parent>

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

<project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>

<java.version>1.8</java.version>

</properties>

<dependencies>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-devtools</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-jdbc</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

<dependency>

<groupId>com.h2database</groupId>

<artifactId>h2</artifactId>

<scope>runtime</scope>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-test</artifactId>

<scope>test</scope>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-maven-plugin</artifactId>

</plugin>

</plugins>

</build>

</project>

* Copy from the previous lab or create the following classes:
* ***src/main/java/io/pivotal/workshop/domain/Snippet.java***
* ***src/main/java/io/pivotal/workshop/controller/SnippetController.java***

***Snippet.java***

**package** io.pivotal.workshop.domain;

**import** java.util.Date;

**public** **class** **Snippet** {

**private** String id;

**private** String title;

**private** String code;

**private** Date created;

**private** Date modified;

**public** **Snippet**() {}

**public** **Snippet**() {

**this**.id = java.util.UUID.randomUUID().toString();

**this**.title = title;

**this**.code = code;

**this**.created = **new** Date();

**this**.modified = **new** Date();

}

**public** String **getId**() {

**return** id;

}

**public** String **getTitle**() {

**return** title;

}

**public** String **getCode**() {

**return** code;

}

**public** Date **getCreated**() {

**return** created;

}

**public** Date **getModified**() {

**return** modified;

}

**public** **void** **setId**(String id) {

**this**.id = id;

}

**public** **void** **setTitle**(String title) {

**this**.title = title;

}

**public** **void** **setCode**(String code) {

**this**.code = code;

}

**public** **void** **setCreated**(java.sql.Date created) {

**this**.created = created;

}

**public** **void** **setModified**(java.sql.Date modified) {

**this**.modified = modified;

}

}

***SnippetController.java***

**package** io.pivotal.workshop.controller;

**import** io.pivotal.workshop.domain.Snippet;

**import** io.pivotal.workshop.repository.SnippetRepository;

**import** org.springframework.beans.factory.annotation.Autowired;

**import** org.springframework.http.HttpHeaders;

**import** org.springframework.http.HttpStatus;

**import** org.springframework.http.ResponseEntity;

**import** org.springframework.web.bind.annotation.\*;

**import** org.springframework.web.servlet.support.ServletUriComponentsBuilder;

**import** java.util.List;

**@RestController**

**public** **class** **SnippetController** {

**@Autowired**

SnippetRepository snippetRepository;

**@GetMapping**("/snippets")

**public** List<Snippet> **snippets**() {

**return** snippetRepository.findAll();

}

**@GetMapping**("/snippets/{id}")

**public** Snippet **snippet**(@PathVariable("id") String id) {

**return** snippetRepository.findOne(id);

}

**@PostMapping**("/snippets")

**public** ResponseEntity<?> add(**@RequestBody** Snippet snippet) {

Snippet \_snippet = snippetRepository.save(snippet);

**assert** \_snippet != **null**;

HttpHeaders httpHeaders = **new** HttpHeaders();

httpHeaders.setLocation(ServletUriComponentsBuilder

.fromCurrentRequest().path("/" + \_snippet.getId())

.buildAndExpand().toUri());

**return** **new** ResponseEntity<>(\_snippet, httpHeaders, HttpStatus.CREATED);

}

}

* Create a new repository class (***src/main/java/io/pivotal/workshop/repository/SnippetRepository.java***) that is constructed with a JdbcTemplate and has the same methods/signatures as the repository from the previous lab.

**package** io.pivotal.workshop.repository;

**import** io.pivotal.workshop.domain.Snippet;

**import** org.springframework.beans.factory.annotation.Autowired;

**import** org.springframework.jdbc.core.JdbcTemplate;

**import** org.springframework.jdbc.core.RowMapper;

**import** org.springframework.stereotype.Repository;

**import** java.sql.Date;

**import** java.sql.PreparedStatement;

**import** java.sql.ResultSet;

**import** java.util.List;

**@Repository**

**public** **class** **SnippetRepository** {

**private** **final** JdbcTemplate jdbcTemplate;

**private** **final** String SQL\_INSERT = "insert into snippet(id,title,code,created,modified) values(?,?,?,?,?)";

**private** **final** String SQL\_QUERY\_ALL = "select \* from snippet";

**private** **final** String SQL\_QUERY\_BY\_ID = "select \* from snippet where id=?";

**private** **final** RowMapper<Snippet> rowMapper = (ResultSet rs, **int** row) -> {

Snippet snippet = **new** Snippet();

snippet.setId(rs.getString("id"));

snippet.setTitle(rs.getString("title"));

snippet.setCode(rs.getString("code"));

snippet.setCreated(rs.getDate("created"));

snippet.setModified(rs.getDate("modified"));

**return** snippet;

};

**@Autowired**

**public** **SnippetRepository**(JdbcTemplate template) {

**this**.jdbcTemplate = template;

}

**public** Snippet **save**(Snippet snippet) {

**assert** snippet.getTitle() != **null**;

**assert** snippet.getCode() != **null**;

**this**.jdbcTemplate.update(connection -> {

PreparedStatement ps = connection.prepareStatement(SQL\_INSERT);

ps.setString(1, \_snippet.getId());

ps.setString(2, \_snippet.getTitle());

ps.setString(3, \_snippet.getCode());

ps.setDate(4, **new** Date(\_snippet.getCreated().getTime()));

ps.setDate(5, **new** Date(\_snippet.getModified().getTime()));

**return** ps;

});

**return** snippet;

}

**public** List<Snippet> **findAll**() {

**return** **this**.jdbcTemplate.query(SQL\_QUERY\_ALL, rowMapper);

}

**public** Snippet **findOne**(String id) {

**return** **this**.jdbcTemplate.queryForObject(SQL\_QUERY\_BY\_ID, **new** Object[]{id}, rowMapper);

}

}

* **Spring Boot** can initialize your database at start up by including specially named files in the ***src/main/resources*** directory. Create the following files with the content noted:
* ***src/main/resources/schema.sql*** that contains the SQL database creation statements
* ***src/main/resources/data.sql*** that contains the SQL statements for insert values in the table

***schema.sql***

**DROP** **TABLE** **IF** **EXISTS** snippet;

**CREATE** **TABLE** snippet

(

**id** **varchar**(36) **NOT** NULL,

title **varchar**(200) **NOT** NULL,

code **varchar**(500) **DEFAULT** NULL,

created **date** **NOT** NULL,

modified **date** **NOT** NULL,

PRIMARY **KEY** (**id**)

);

***data.sql***

**insert** **into** snippet(**id**,title,code,created,modified) **values**('66921076-ed1d-458b-9d7d-ce9a227d64a5','JavaScript: Hello World','console.log("Hello World!");','2016-07-31','2016-07-31');

**insert** **into** snippet(**id**,title,code,created,modified) **values**('4465afe0-4e8f-4779-90e3-e6b971c4cc7d','HTML: Hello World','<html><body><h1>Hello World</h1></body></html>','2016-07-31','2016-07-31');

**insert** **into** snippet(**id**,title,code,created,modified) **values**('842c4bd0-32a0-4c6e-8f7a-74c45c23ddf1','Bash: Hello World','echo "Hello World"','2016-07-31','2016-07-31');

**insert** **into** snippet(**id**,title,code,created,modified) **values**('7d1bce20-799b-49b0-8306-6f558aac4dd9','Python: Hello World','print "Hello World"','2016-07-31','2016-07-31');

* Run the application by using the ***Spring Boot Dashboard*** from the ***STS IDE***    
  or you can use the terminal an execute:$ ./mvnw spring-boot:run
* The H2 library comes with a embedded SQL browser that you can access at: <http://localhost:8080/h2-console>. The database by default is named: testdb and the complete JDBC URL is: jdbc:h2:mem:testdb You can connect and run a query.
* Open a terminal window and run the following command: $ curl -s localhost:8080/snippets | jq
* Use curl to add a new snippet code and review the results in the [h2-console](http://localhost:8080/h2-console) and via the /snippets endpoint.

**CONGRATULATIONS!** You just created an In-Memory Data application using H2 and the JdbcTemplate. All of the wiring code was provided by Spring Boot.