Build a Rest web service using spring boot and see them in behavior

**Prerequisites:**

* Spring Tool Suite 3/4
* Java 1.8
* MySQL 5.5
* MySQL Workbench 5.2
* Postman

**Description:**

Spring Tool Suite is an open-source, Eclipse-based IDE distribution that provides a superset of the Java EE distribution of Eclipse. It includes features that making working with Spring applications even easier.

**Business Scenario:**

An application has to be developed for management of contacts. The application will provide following services:

* Based on contact requirement, existing contacts will be displayed on user interface. In addition, user can log the new records for contact information entered in system
* Contact information can be viewed by the user for any existing record
* Contact information can be updated by the user
* Contact information can be removed from the system

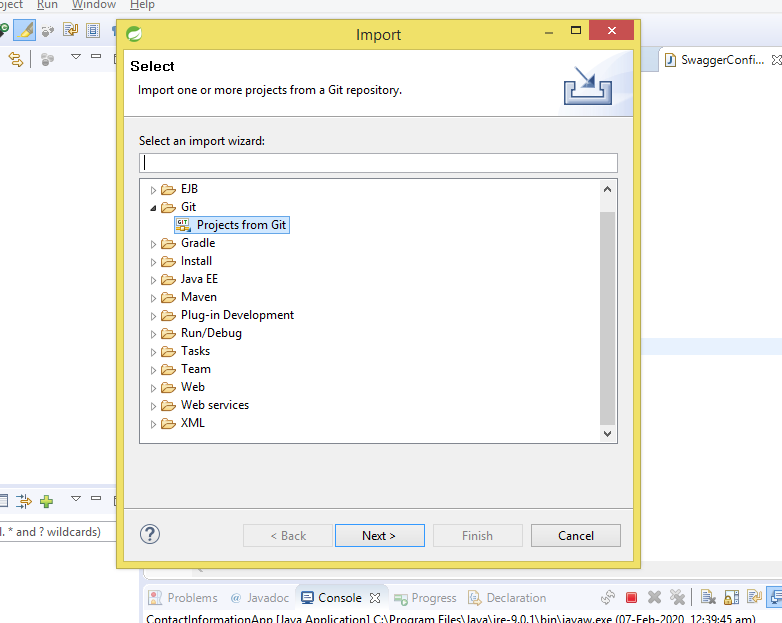
**Project codebase repository URL:**

* <https://github.com/ektauma/contact-details.git>

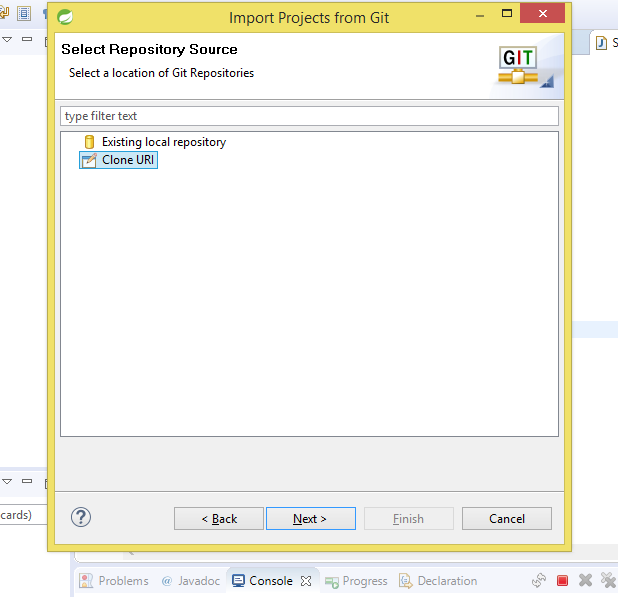
**How to import the project into Spring Tool Suite (STS):**

**Steps and screenshots with the flow:**

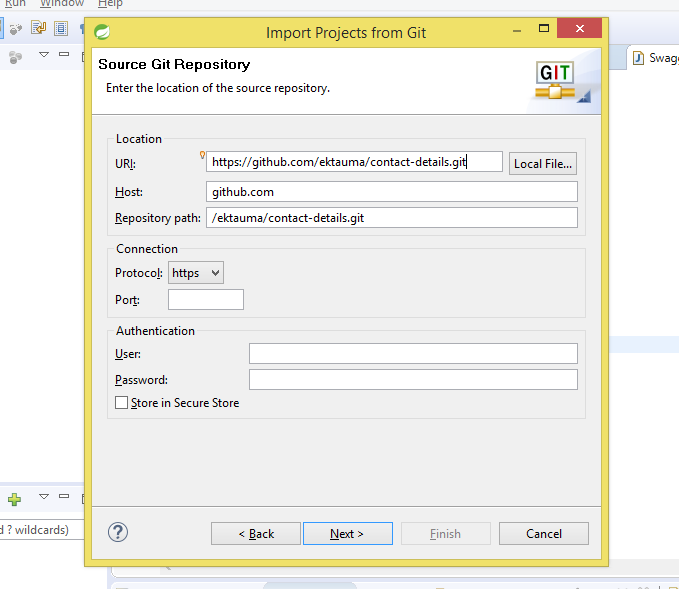
1. Launch STS IDE
2. Navigate through the path: **Click** -> **New** -> **Import** -> **Choose the project’s directory (from git**) and click Next



1. Select clone URI and click Next



1. Enter github URL:
2. Below github related fields need to be filled:
   1. URL: <https://github.com/ektauma/contact-details.git>
   2. Host: github.com
   3. Repository Path: /ektauma/contact-details.git
   4. Click next



1. Finish
2. Project setup will be completed
3. Update the dependencies: Right click on Project -> Select Maven -> Click Update Project
4. Run below SQL commands in MYSQL workbench

create database contactDb;

use contactDb;

CREATE TABLE `contact` (

`contact\_id` int(11) NOT NULL AUTO\_INCREMENT,

`email` varchar(255) DEFAULT NULL,

`first\_name` varchar(255) DEFAULT NULL,

`last\_name` varchar(255) DEFAULT NULL,

`phone\_number` int(11) NOT NULL,

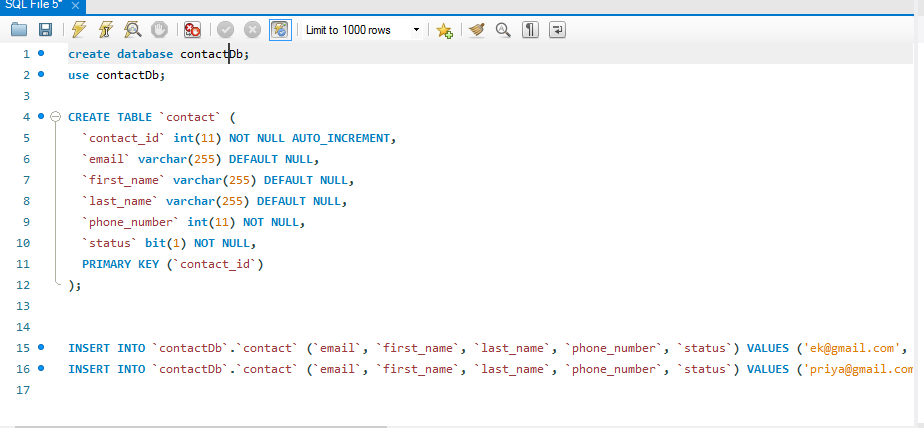
`status` bit(1) NOT NULL,

PRIMARY KEY (`contact\_id`)

);

INSERT INTO `contactDb`.`contact` (`email`, `first\_name`, `last\_name`, `phone\_number`, `status`) VALUES ('ek@gmail.com', 'Ekta', 'Sharma', '970986574', b'0');

INSERT INTO `contactDb`.`contact` (`email`, `first\_name`, `last\_name`, `phone\_number`, `status`) VALUES ('priya@gmail.com', 'Priya', 'Sharma', '76986574', b'1');



1. Right click on Project. Navigate through **Run As** -> **Java Application** -> select ContactInformationApp (if ask for) -> click ok

**API Request/Response Testing:** API can be tested via Postman

**Steps and screenshots with the flow:**

1. Get API call:
2. Select method GET,
3. Enter URL: <http://localhost:8089/api/contacts/>
4. Click Send
5. Response will be fetched in the body
6. Post API call:
7. Select method POST enter <http://localhost:8089/api/contacts/>
8. Go to tab Header select **Key** as **Content-Type** and **Value** as **application/json**
9. Go to tab Body and type below data in JSON format

{

"contactId": 2,

"firstName": "Roshni",

"lastName": "Sharma",

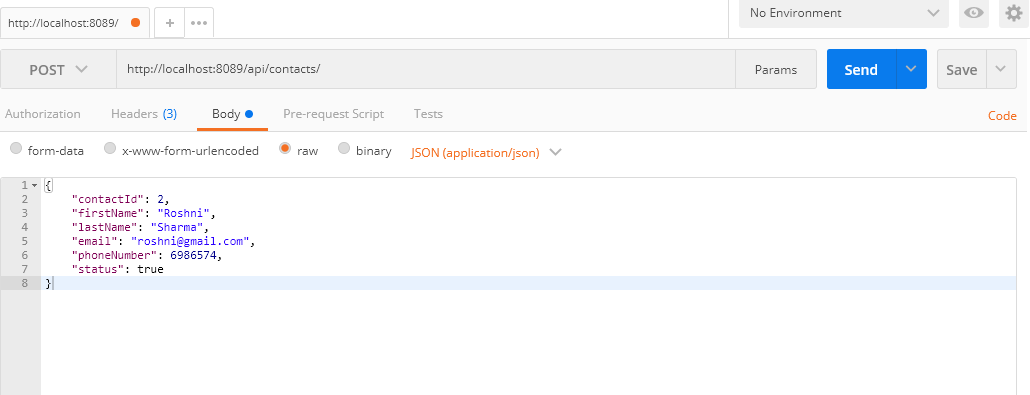
"email": "roshni@gmail.com",

"phoneNumber": 6986574,

"status": true

}

1. Click Send

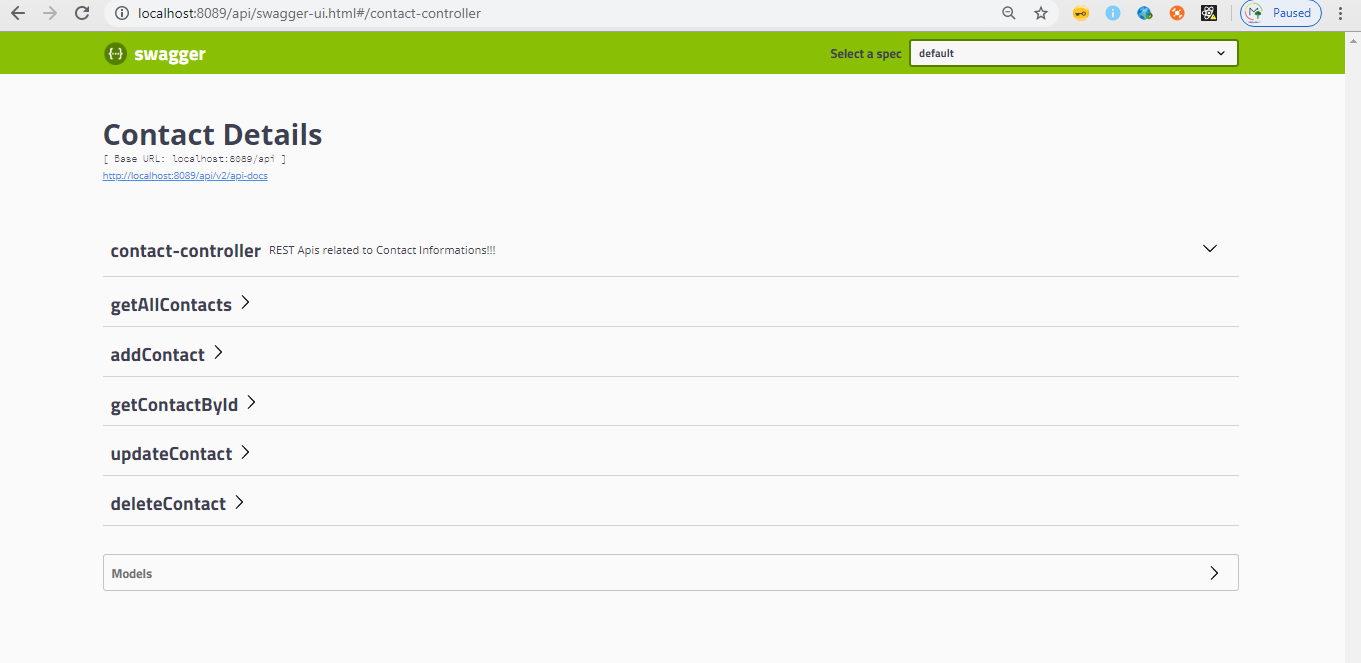


1. DELETE API call (Delete contact by **contactId**):
2. Select method DELETE,
3. Enter URL: <http://localhost:8089/api/contacts/1>
4. Click Send
5. If API fails (Because of non-existing contactId), response will be returning the failure information with corresponding message

**About Swagger:**

Swagger UI allows anyone — be it your development team or your end consumers — to visualize and interact with the API’s resources without having any of the implementation logic in place. It’s automatically generated from your OpenAPI (formerly known as Swagger) Specification, with the visual documentation making it easy for back end implementation and client side consumption.

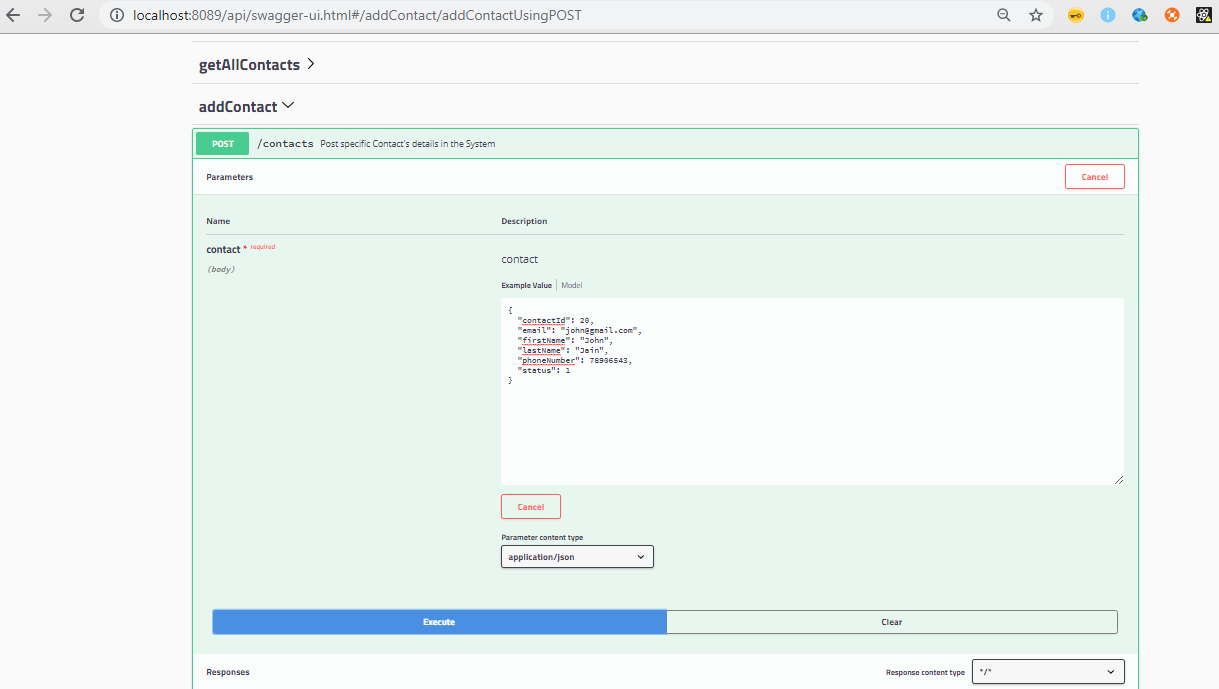
After launching the Swagger-UI (hit the URL <http://localhost:8089/api/swagger-ui.html#/> in the browser address bar), the screen will look like below:



API Call: I will demonstrate one POST API call (**addContact**) through swagger-ui which will be having request/response in JSON format.

**Steps and screenshots with the flow:**

1. Click **addContact** panel
2. Click **GET** button
3. Click **Try it out** button appearing on the right side of the panel
4. User information can be recorded here and this will work as a request parameter for this API



1. Click Execute
2. Response will be sent by the API as below:

