```
InMemory Authentication
============================pom.xml======================
<?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
    https://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>2.3.0.RELEASE</version>
        <relativePath/> <!-- lookup parent from repository -->
    </parent>
    <groupId>com.ameya</groupId>
    <artifactId>006-SpringSecurityBasicInMemory</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <name>006-SpringSecurityBasicInMemory</name>
    <description>Demo project for Spring Boot</description>
    <properties>
        <java.version>1.8</java.version>
    </properties>
    <dependencies>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-security</artifactId>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
        </dependency>

        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-test</artifactId>
            <scope>test</scope>
            <exclusions>
                <exclusion>
                    <groupId>org.junit.vintage</groupId>
                    <artifactId>junit-vintage-engine</artifactId>
                </exclusion>
            </exclusions>
        </dependency>
        <dependency>
            <groupId>org.springframework.security</groupId>
            <artifactId>spring-security-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>
============================application.properties================================

server.port=9001
logging.level.web=trace

==============================AppSecurityConfig.java======================
package com.ameya.configs;

import org.springframework.context.annotation.Configuration;
import
org.springframework.security.config.annotation.authentication.builders.AuthenticationM
anagerBuilder;
import org.springframework.security.config.annotation.web.builders.HttpSecurity;
```

```
import org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;
import org.springframework.security.config.annotation.web.configuration.WebSecurityConfigurerAdapter;

@Configuration
@EnableWebSecurity
public class AppSecurityConfig extends WebSecurityConfigurerAdapter {

    @Override
    protected void configure(AuthenticationManagerBuilder auth) throws Exception {
        auth.inMemoryAuthentication()
        .withUser("ameya").password("{noop}1234").roles("ADMIN")
        .and()
        .withUser("avani").password("{noop}1234").roles("USER");
    }

    @Override
    protected void configure(HttpSecurity http) throws Exception {
        http.authorizeRequests()
        .antMatchers("/").permitAll()
        .antMatchers("/rest/hello*").access("hasRole('ROLE_ADMIN')")
        .and().httpBasic();
        http.csrf().disable();
    }


}
```

==========HelloController==========

```
package com.ameya.controllers;

import org.springframework.security.core.context.SecurityContextHolder;
import org.springframework.security.core.userdetails.UserDetails;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
public class HelloController {

    @GetMapping(path="/")
    public String rootUrl() {
        return "This is permitted to ALL !!";
    }
    @GetMapping("/rest/hello")
    public String sayHello() {
        String userName=getUserName();
        return "Hi There - This is permitted only to Role Admin :: "+userName;
    }
    public String getUserName() {
        String userName=null;
        Object
        principal=SecurityContextHolder.getContext().getAuthentication().getPrincipal(
        );
        userName=((UserDetails)principal).getUsername();
        return userName;
    }
}
```

==========

JdbcAuthentication

==========

```
<?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
    https://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>2.3.0.RELEASE</version>
        <relativePath/> <!-- lookup parent from repository -->
```

```
    </parent>
    <groupId>com.ameya</groupId>
    <artifactId>006-SpringSecurityBasicJdbc</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <name>006-SpringSecurityBasicJdbc</name>
    <description>Demo project for Spring Boot</description>
    <properties>
        <java.version>1.8</java.version>
    </properties>
    <dependencies>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-jdbc</artifactId>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-security</artifactId>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
        </dependency>

        <dependency>
            <groupId>mysql</groupId>
            <artifactId>mysql-connector-java</artifactId>
            <scope>runtime</scope>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-test</artifactId>
            <scope>test</scope>
            <exclusions>
                <exclusion>
                    <groupId>org.junit.vintage</groupId>
                    <artifactId>junit-vintage-engine</artifactId>
                </exclusion>
            </exclusions>
        </dependency>
        <dependency>
            <groupId>org.springframework.security</groupId>
            <artifactId>spring-security-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>
==========================================application.properties=========================
========

server.port=9001
logging.level.web=trace

spring.datasource.url=jdbc:mysql://localhost:3306/nineleapsdb
spring.datasource.username=root
spring.datasource.password=root
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
#spring.datasource.initialization-mode=always
============================================schema.sql==============================
=====
CREATE TABLE users(username VARCHAR(45) NOT NULL,password VARCHAR(200) NOT
NULL,enabled TINYINT(4) NOT NULL DEFAULT '1',PRIMARY KEY (username));

CREATE TABLE user_roles (user_role_id INT(11) NOT NULL AUTO_INCREMENT,username
```

```
VARCHAR(45) NOT NULL,role VARCHAR(45) NOT NULL,PRIMARY KEY (user_role_id),UNIQUE INDEX uni_username_role (role, username),INDEX fk_username_idx (username),CONSTRAINT fk_username FOREIGN KEY (username) REFERENCES users (username));
==========================================data.sql==========================================
INSERT INTO users (username, password, enabled) VALUES('ameya', '1234', 1);
INSERT INTO users (username, password, enabled) VALUES('avani', '1234', 1);

INSERT INTO user_roles (user_role_id, username, role) VALUES(2, 'ameya', 'ROLE_ADMIN');
INSERT INTO user_roles (user_role_id, username, role) VALUES(1, 'ameya', 'ROLE_USER');
INSERT INTO user_roles (user_role_id, username, role) VALUES(3, 'avani', 'ROLE_USER');
==========================================SecurityConfiguration.java==========================
package com.ameya.configs;

import javax.sql.DataSource;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.security.config.annotation.authentication.builders.AuthenticationManagerBuilder;
import org.springframework.security.config.annotation.web.builders.HttpSecurity;
import org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;
import org.springframework.security.config.annotation.web.configuration.WebSecurityConfigurerAdapter;
import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;
import org.springframework.security.crypto.password.PasswordEncoder;

@Configuration
@EnableWebSecurity
public class SecurityConfiguration extends WebSecurityConfigurerAdapter{

    @Autowired
    private DataSource dataSource;
    @Bean
    public PasswordEncoder passwordEncoder() {
        return new BCryptPasswordEncoder();
    }
    @Override
    protected void configure(AuthenticationManagerBuilder auth) throws Exception {
        auth.jdbcAuthentication()
        .usersByUsernameQuery("select username,password,enabled from users where username=?")
        .authoritiesByUsernameQuery("select username,role from user_roles where username=?")
        .passwordEncoder(passwordEncoder()).dataSource(dataSource);

    }
    @Override
    protected void configure(HttpSecurity http) throws Exception {
        http
        .authorizeRequests()
        .antMatchers("/").permitAll()
        .antMatchers("/rest/hello*")
        .access("hasRole('ROLE_ADMIN')")
        .and().httpBasic();
        http.csrf().disable();

    }

}
================================HelloController.java================================
package com.ameya.controllers;

import org.springframework.security.core.context.SecurityContextHolder;
import org.springframework.security.core.userdetails.UserDetails;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;
```

```
266
267 @RestController
268 public class HelloController {
269
270     @GetMapping(path="/")
271     public String rootUrl() {
272         return "This is permitted to ALL !!";
273     }
274     @GetMapping("/rest/hello")
275     public String sayHello() {
276         String userName=getUserName();
277         return "Hi There - This is permitted only to Role Admin :: "+userName;
278     }
279     public String getUserName() {
280         String userName=null;
281         Object
        principal=SecurityContextHolder.getContext().getAuthentication().getPrincipal(
        );
282         userName=((UserDetails)principal).getUsername();
283         return userName;
284     }
285 }
286 ============================PasswordEncoder.java==========================
287 package com.ameya;
288
289 import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;
290
291 public class PasswordEncoder {
292
293     public static void main(String[] args) {
294         String encoded=new BCryptPasswordEncoder().encode("1234");
295         System.out.println(encoded);
296     }
297
298 }
299
300
```