Spring 2.0 security

```
package com.javatechie.spring.security.api.config;

import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import
org.springframework.security.config.annotation.authentication.builders.Authent
icationManagerBuilder;
import
org.springframework.security.config.annotation.web.builders.HttpSecurity;
import
org.springframework.security.config.annotation.web.configuration.WebSecurity
ConfigurerAdapter;
import org.springframework.security.crypto.password.NoOpPasswordEncoder;

@SuppressWarnings("deprecation")
@Configuration
public class SpringSecurityConfig extends WebSecurityConfigurerAdapter {
    @Override
    protected void configure(AuthenticationManagerBuilder auth) throws
Exception {
        auth.inMemoryAuthentication().withUser("Java
Techie").password("Password").roles("ADMIN");

auth.inMemoryAuthentication().withUser("Basant").password("Password2").rol
es("USER");
    }

    // security for all API

    /*
     * @Override protected void configure(HttpSecurity http) throws Exception
{
     * http.csrf().disable();
     * http.authorizeRequests().anyRequest().fullyAuthenticated().and().
     * httpBasic(); }
     */

    // security based on URL

    /*
     * @Override protected void configure(HttpSecurity http) throws Exception
{
     * http.csrf().disable();
     * http.authorizeRequests().antMatchers("/rest/
**").fullyAuthenticated().and
     * ().httpBasic(); }
     */

    // security based on ROLE
    @Override
    protected void configure(HttpSecurity http) throws Exception {
        http.csrf().disable();
        http.authorizeRequests().antMatchers("/rest/
**").hasAnyRole("ADMIN").anyRequest().fullyAuthenticated().and()
                                .httpBasic();
    }

    @Bean
    public static NoOpPasswordEncoder passwordEncoder() {
        return (NoOpPasswordEncoder)
NoOpPasswordEncoder.getInstance();
    }
}

// to use database


import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.context.annotation.Configuration;
import
org.springframework.security.config.annotation.authentication.builders.Authent
icationManagerBuilder;
import
org.springframework.security.config.annotation.web.configuration.EnableWebS
ecurity;
import org.springframework.security.core.userdetails.UserDetailsService;

@Configuration
@EnableWebSecurity
public class SecurityConfig extends WebSecurityConfigurerAdapter {

    @Autowired
    private UserDetailsService userDetailsService;

    @Override
    protected void configure(AuthenticationManagerBuilder auth) throws
Exception {
    auth.userDetailsService(userDetailsService);
    }
}
```