

SpringBoot Adding Spring Security Login, AddingRules and BcryptPassword

WebSecurityConfig.java

```
package org.studyeasy.SpringStarter.config;

import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.security.config.Customizer;
import org.springframework.security.config.annotation.method.configuration.EnableMethodSecurity;
import org.springframework.security.config.annotation.web.builders.HttpSecurity;
import org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;
import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;
import org.springframework.security.crypto.password.PasswordEncoder;
import org.springframework.security.web.SecurityFilterChain;

@Configuration
@EnableWebSecurity
@EnableMethodSecurity(prePostEnabled = true, securedEnabled = true)
public class WebSecurityConfig {
    private static final String[] WHITELIST = {
        "/",
        "/login",
        "/register",
        "/db-console/**",

        // Allow all the css and other resources
        "/css/**",
        "/fonts/**",
        "/images/**",
        "/js/**"
    };

    @Bean
    public static PasswordEncoder passwordEncoder() {
        return new BCryptPasswordEncoder();
    }
}
```

```

@Bean
public SecurityFilterChain filterChain(HttpSecurity http) throws
Exception {
    http
        .authorizeHttpRequests(auth -> auth
            .requestMatchers(WHITELIST)
            .permitAll()
            .anyRequest()
            .authenticated())

        .formLogin(formLogin -> formLogin
            .loginPage("/login")
            .loginProcessingUrl("/login")
            .usernameParameter("email")
            .passwordParameter("password")
            .defaultSuccessUrl("/", true)
            .failureUrl("/login?error")
            .permitAll())

        .logout(logout -> logout
            .logoutUrl("/logout")
            .logoutSuccessUrl("/logout?success")
            .permitAll())
        // Enable HTTP Basic authentication
        .httpBasic(Customizer.withDefaults());

        // TODO: remove these files after upgrading the DB from h2
        // infile DB

        http.csrf(csrf -> csrf.disable());
        // Disable frame options
        http.headers(headers -> headers.frameOptions(frameOptions ->
frameOptions.disable()));

        return http.build();
    }
}

```

AccountService.java -> Updated

```
public class AccountService {  
    @Autowired  
    private AccountRepository accountRepository;  
  
    @Autowired  
    private PasswordEncoder passwordEncoder;  
  
    public Account save(Account account){  
        account.setPassword(passwordEncoder.encode(account.getPassword()));  
        return accountRepository.save(account);  
    }  
}
```

Output:-

The screenshot shows a web browser window with the H2 Console interface. The address bar shows the URL: `localhost:8080/db-console/login.do?sessionId=a2c91632bde37fcdeb78bd2937f009`. The console displays the SQL statement `SELECT * FROM ACCOUNT` and its results. The results are shown in a table with columns `ID`, `EMAIL`, `FIRSTNAME`, and `PASSWORD`. There are two rows of data. The status bar at the bottom indicates the database is H2 2.2.224 (2023-09-17).

ID	EMAIL	FIRSTNAME	PASSWORD
1	account01@studyeasy.org	user01	\$2a\$10\$1mt02l9wLH18Sb96sq4u4XEhEgdtA119CpDvKd5czV1aVkbhO
2	account02@studyeasy.org	user02	\$2a\$10\$0cu6lAuJy4E7DqV0XnUegO2MpHuzeeuOuo76mYzohH6aotEQE9C6

(2 rows, 2 ms)