

# Prisijungimo sistema

---

# Parsisiųskime reikiamus paketus

---

```
<dependency>  
  <groupId>org.springframework.security</groupId>  
  <artifactId>spring-security-config</artifactId>  
  <version>5.0.8.RELEASE</version>  
</dependency>
```

```
<dependency>  
  <groupId>org.springframework.security</groupId>  
  <artifactId>spring-security-web</artifactId>  
  <version>5.0.8.RELEASE</version>  
</dependency>
```

```
<dependency>  
  <groupId>org.springframework.security</groupId>  
  <artifactId>spring-security-crypto</artifactId>  
  <version>5.0.8.RELEASE</version>  
</dependency>
```

# Kodavimo algoritmai esantys pakete security

Sutrumpinimas	Kodavimo algoritmas
bcrypt	BCryptPasswordEncoder (Also used for encoding)
ldap	LdapShaPasswordEncoder
MD4	Md4PasswordEncoder
MD5	new MessageDigestPasswordEncoder("MD5")
noop	NoOpPasswordEncoder
pbkdf2	Pbkdf2PasswordEncoder
scrypt	SCryptPasswordEncoder
SHA-1	new MessageDigestPasswordEncoder("SHA-1")
SHA-256	new MessageDigestPasswordEncoder("SHA-256")
sha256	StandardPasswordEncoder

Sutrumpinimas rašomas prieš užkoduotą slaptažodį. Pavyzdžiui: (noop)slaptazodis

# Sukurkime prisijungimo konfigūraciją

---

Sukurkime failą spring-security.xml ir jame patalpinkime tokį tekstą:

```
<beans:beans xmlns="http://www.springframework.org/schema/security"
xmlns:beans="http://www.springframework.org/schema/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-3.0.xsd
http://www.springframework.org/schema/security
http://www.springframework.org/schema/security/spring-security-4.2.xsd">

<http auto-config="true">
  <intercept-url pattern="/**" access="hasRole('ROLE_USER')" />
</http>

<authentication-manager>
  <authentication-provider>
    <user-service>
      <user name="admin" password="{noop}labas" authorities="ROLE_USER" />
    </user-service>
  </authentication-provider>
</authentication-manager>

</beans:beans>
```

# Web.xml failas

---

Tuomet web.xml faile turëtumëme pridëti tokias eilutes:

```
<filter>
  <filter-name>springSecurityFilterChain</filter-name>
  <filter-class>org.springframework.web.filter.DelegatingFilterProxy</filter-class>
</filter>
<filter-mapping>
  <filter-name>springSecurityFilterChain</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>
```

# Užduotis

---

Išbandykime standartinį prisijungimą, patikrinkime ar veikia slaptažodžio įrašymas

# Užduotis

---

Modifikuokime prisijungimo sistemą taip, kad mūsų slaptažodžio įvedimo laukai būtų mūsų stiliaus.

Mes juos galime aprašyti pakeitę prisijungimo konfigūraciją:

```
<http use-expressions="true">
  <intercept-url pattern="/Login" access="permitAll" />
  <intercept-url pattern="/**" access="hasRole('ROLE_USER')" />
  <form-login default-target-url="/home" login-page="/Login" authentication-failure-url="/Login?error=true" />
  <logout logout-success-url="/Login" />
</http>
```

# Security failas

---

Security faile galime nurodyti įvairias strategijas, kur ir kaip bus įleidžiami vartotojai, į kurias dalis vartotojai privalės prisijungti:

```
<intercept-url pattern="/" access="permitAll" />
```

```
<intercept-url pattern="/home" access="permitAll" />
```

```
<intercept-url pattern="/admin**" access="hasRole('ADMIN')" />
```

```
<intercept-url pattern="/dba**" access="hasRole('ADMIN') and hasRole('DBA')" />
```



# Užduotis

---

Sukurkime kontrolerį LoginController ir jame sukurkime route:

/login

# Užduotis

---

Sukurkime prisijungimo formos atvaizdavimą. Joje turėtų būti šie komponentai:

```
<form action="{pageContext.request.contextPath}/Login" method="POST">
  <input type="text" name="username">
  <input type="password" name="password" >
  <input type="submit" value="Login" />
  <input type="hidden" name="{_csrf.parameterName}" value="{_csrf.token}"/>
</form>
```

Formoje galėtumėme naudoti ir standartinį kintamąjį į login url: `{loginUrl}`

# Užduotis

---

Pasinaudodami Bootstrap modifikuokime formą taip, kad ji atrodytų tvarkingai.

# Sukurkime metodą atsijungimui

---

Sukurkime metodą skirtą atsijungimui, jo pagalba suprogramuokime atsijungimą mūsų puslapyje:

```
@RequestMapping(value="/logout", method = RequestMethod.GET)
public String logoutPage (HttpServletRequest request, HttpServletResponse response)
{
    Authentication auth = SecurityContextHolder.getContext().getAuthentication();
    if (auth != null){
        new SecurityContextLogoutHandler().logout(request, response, auth);
    }
    return "redirect:/login?logout";
}
```

# Užduotis

---

Modifikuokime prisijungimo formą taip, kad sėkmingai atsijungus užrašytų jog sėkmingai atsijungėme:

```
<c:if test="${param.logout != null}">
```

```
    You have been logged out successfully.
```

```
</c:if>
```