

1g

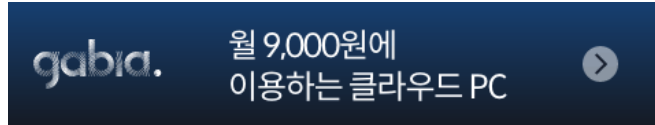
프링 시큐리티 기초 따라가기 (3) - HTTPS

이승현 (wowlsh93@gmail.com) 2015. 7. 15. 14:29

HAMA 블로그



[하마] 이승현...



현재에서는 이전에 만들었던 소스에 아래와 같은 기능을 추가할 것이다.

HTTPS 기능

HTTPS는 HTTP에 SSL 기능을 추가한 것인데, HTTP는 문자를 가지고 누가 엿보기가 쉽다. 따라서

!하는데 해당 문자를 암호화해주며, 암호화 하기 위한 키에 대해 안전성을 보장해주는 기술이 들어가있다.

참고 : <https://wiki.kldp.org/HOWTO/html/SSL-Certificates-HOWTO/x70.html>

security-context.xml 파일에 설정 추가

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<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-3.2.xsd
```

```
http://www.springframework.org/schema/security
```

```
http://www.springframework.org/schema/security/spring-security-3.2.xsd">
```

```
<http auto-config='true' use-expressions="true">
```

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

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

```
<intercept-url pattern="/*" access="hasRole('ROLE_USER')" requires-channel="https" />
```

```
<form-login login-page="/login"
```

```
    default-target-url="/monitoring"
```

```
    username-parameter="username"
```

```
    password-parameter="password"
```

```
    authentication-failure-url="/login?error"
```

```
    always-use-default-target='true'
```

```
/>
```

```
<logout invalidate-session="true"
```

```
    delete-
```

```
    cookies="JSESSIONID,SPRING_SECURITY_REMEMBER_ME_COOKIE"
```

```
    logout-success-url="/login?logout" />
```

```
<remember-me key="wmoskey" token-validity-seconds="2419200"/> <!-- 4 weeks -->
```

```
<!-- enable csrf protection -->
```

```
<csrf/>
```

```
</http>
```

```
<authentication-manager>
```

```
    <authentication-provider user-service-ref="memberService"/>
```

```
</authentication-manager>
```

```
<beans:bean id="memberService" class="com.company.wmos.auth.MemberService">
```

```
</beans:bean>
```

```
</beans:beans>
```

HAMA 블로그



[하마] 이승현...

res-channel 를 추가하였다. (한줄이면됨)

위에 한줄이면 되며, 톰캣에 HTTPS 설정을 하면되는데 아래 사이트를 참고하자

HAMA 블로그



<https://visu4l.tistory.com/419>

트 내용중에 참고로 국가코드는 KR 을 넣어주면 되며,
에서 명령에서 trustscerts 들은 모두 오타이다. trustacerts 로 바꿔주자.

```
tool -import -alias Root -trustscerts -file TrialRoot.pem -keystore testserver
```

```
tool -import -alias Intermediate -trustscerts -file TrialIntermediate.pem -keystore testserver
```

```
tool -import -alias testserver -trustscerts -file cert.pem -keystore testserver
```

TPS 설정을 위한 다른 방법 (번역글)

[/www.javacodegeeks.com/2012/12/securing-your-tomcat-app-with-ssl-and-spring-security.html](https://www.javacodegeeks.com/2012/12/securing-your-tomcat-app-with-ssl-and-spring-security.html)

과 Spring Security 를 가지고 톰캣 웹어플리케이션을 보호해보자.

나의 마지막 블로그를 읽었다면 **ten things that you can do with Spring Security** 를 알게되었을것이다.
스프링 시큐리티를 시작하기 전에 해야할것은 웹어플리케이션이 HTTPS 를 지원해야한다는것인데, 그것은
보통같은것을 평문으로 인터넷상에 돌아다니는것을 막아줄것이다.

TPS 설정을 시작해볼까?

Key Store 만들기



2로 할것은 인증서를 포함한 사설 키스토어 를 만들것것이다. 그것을 생성할 가장 간단한 방법은

keytool 유틸리티를 사용하는것이다. 자바SDK 설치했으면 /bin 디렉토리안에 있을것이다.

```
keytool -genkey -alias MyKeyAlias -keyalg RSA -keystore /Users/Roger/tmp/roger.keystore
```

명령어에서

- **alias** 키에 대한 유니크한 ID
- **keyalg** 'RSA', 'DSA', 'DES' 같은 키를 만들기 위한 알고리즘.
- **keystore** 키스토어가 저장될 위치

하면 다음과 같은 것들을 물을 것이다. 적절히 써넣자. (대한민국 국가코드는 **KR**)

HAMA 블로그



[하마] 이승현...

```
ger$ keytool -genkey -alias MyKeyAlias -keyalg RSA -keystore /Users/Roger/tmp/roger.keystore
```

ter keystore password:

-enter new password:

What is your first and last name?

Unknown]: localhost

What is the name of your organizational unit?

Unknown]: MyDepartmentName

What is the name of your organization?

Unknown]: MyCompanyName

What is the name of your City or Locality?

Unknown]: Stafford

What is the name of your State or Province?

Unknown]: NA

What is the two-letter country code for this unit?

Unknown]: UK

CN=localhost, OU=MyDepartmentName, O=MyCompanyName, L=Stafford, ST=UK, C=UK correct?

Yes]: Y

ter key password for

(RETURN if same as keystore password):

Tomcat 설정 변경



이 SSL connector 를 가져야하는데 /conf 디렉토리에 보통 위치한 server.xml 파일에 아래를 추가한다.

```
<connector port="8443" protocol="HTTP/1.1" SSLEnabled="true"
    maxThreads="150" scheme="https" secure="true"
    clientAuth="false" sslProtocol="TLS" />
```

I making it look something like this:

```
connector SSLEnabled="true" keystoreFile="/Users/Roger/tmp/roger.keystore" keystorePass="pass
```

너 password에 평문을 넣는것은 보안에 좋지 않다. 여러 방법이 있는데 이 포스트를 넘어서는것이다.

security-context.xml 설정

```
xml version="1.0" encoding="UTF-8"?>
<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-3.1.xsd">
```

```
<http auto-config='true' >
    <intercept-url pattern="/*" requires-channel="https" />
</http>
```

```
<authentication-manager>
</authentication-manager>
```

```
beans:beans>
```

res-channel="https" 를 추가하고 톰캣을 이용해서 시작하면 HTTPS 을 사용해서 접근할수있다.

://localhost:8443/my-app 요렇게 입력하면 작동할것이다.

://localhost:8080/my-app 요렇게 입력하면 자동으로 https 로 바뀔것이다.



1

구독하기

ing' 카테고리의 다른 글

링 시큐리티 기초 따라가기 (3) - HTTPS (0)	2015.0
링 시큐리티 기초 따라가기 (2) - Remember Me (0)	2015.0
링 시큐리티 기초 따라가기 (1) - 환경설정 및 기본 로그인 시스템 (1)	2015.0
Angular JS and Spring Security 1 ~4 편 (번역) (0)	2015.0
Spring Security 버전별 의존성 요구사항 (0)	2015.0
Spring Security Scan / annotation-config / annotation-driven 차이점 (2)	2015.0

스프링 보안 HTTPS

ing' Related Articles

스프링 시큐리티 기초 따라

스프링 시큐리티 기초 따라

Angular JS and Spring

Mybatis-spring 버전별

가기 (2) - Remember Me

2015.07.14

가기 (1) - 환경설정 및 기본 서비스 현황

로그인 시스템

2015.07.13

Security 1 ~4 편 (번역)

2015.07.11

준성 요구사항

2015.07.10

HAMA 블로그



omments

Name

Password

여러분의 소중한 댓글을 입력해주세요

Secret Send

1 ... 528 529 530 531 532 533 534 535 536 ... 669 N