**Тестирование запросов на защищённые и незащищённые урлы до, при и после авторизации.**

**Цель:**

*Узнать методологию работы SpringSecurity с защищёнными урлами и сессиями, вычислить настройки при которых можно будет хранить информацию в сессии.*

**Проблема:**

*Менеджмент сессии (включая сохранённую внутри сессии информацию) при работе через разные протоколы: HTTP/HTTPS, сохранение информации в сессии и подтверждение работы метода логаут.*

Статья: <http://www.baeldung.com/spring-channel-security-https>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ЦИТАТА:  «[Session fixation](https://en.wikipedia.org/wiki/Session_fixation) is a problem which can’t be avoided when switching between HTTP and HTTPS.  By default Spring creates a new session-id after a successful login. When a user loads the HTTPS login page the user’s session-id cookie will be marked as secure. After logging in, the context will switch to HTTP and [the cookie will be lost](http://docs.spring.io/spring-security/site/faq/faq.html#faq-tomcat-https-session) as HTTP is insecure.  To avoid this**setting**[***session-fixation-protection***](http://docs.spring.io/spring-security/site/docs/current/reference/html/ns-config.html#ns-session-fixation)**to none is required**.   |  |  | | --- | --- | | 1  2  3 | http.sessionManagement()    .sessionFixation()    .none(); |   Or via XML:   |  |  | | --- | --- | | 1 | <session-management session-fixation-protection="none"/> |   **Disabling session fixation protection might have security implications**, therefore you need to weigh the pros and cons if you’re concerned about session fixation based attacks.» |

**Оффициальная документация Spring Security:**

#### Session Fixation Attack Protection

[Session fixation](https://en.wikipedia.org/wiki/Session_fixation) attacks are a potential risk where it is possible for a malicious attacker to create a session by accessing a site, then persuade another user to log in with the same session (by sending them a link containing the session identifier as a parameter, for example). Spring Security protects against this automatically by creating a new session or otherwise changing the session ID when a user logs in. If you don’t require this protection, or it conflicts with some other requirement, you can control the behavior using the session-fixation-protection attribute on <session-management>, which has four options

* none - Don’t do anything. The original session will be retained.
* newSession - Create a new "clean" session, without copying the existing session data (Spring Security-related attributes will still be copied).
* migrateSession - Create a new session and copy all existing session attributes to the new session. This is the default in Servlet 3.0 or older containers.
* changeSessionId - Do not create a new session. Instead, use the session fixation protection provided by the Servlet container (HttpServletRequest#changeSessionId()). This option is only available in Servlet 3.1 (Java EE 7) and newer containers. Specifying it in older containers will result in an exception. This is the default in Servlet 3.1 and newer containers.

When session fixation protection occurs, it results in a SessionFixationProtectionEvent being published in the application context. If you use changeSessionId, this protection will also result in any javax.servlet.http.HttpSessionIdListener s being notified, so use caution if your code listens for both events. See the [Session Management](https://docs.spring.io/spring-security/site/docs/current/reference/htmlsingle/#session-mgmt) chapter for additional information.

**Маппинги URL:**

* Orders – незащищённый, ничего не требует
* Orders/id – незащищённый, требует юзера в сессии
* Login – защищённый, вставляет юзера в сессию
* giveSessionData – защищённый, требует юзера в сессии
* logout – незащищённый, должен удалять сессию.

**Последовательность запросов для тестов:**

1. **Orders**
2. **Orders/id –** *проверка незагруженности юзера в сессию*
3. **Login *–*** *логин, загрузка юзера в сессию*
4. **giveSessionData –** *проверка загруженности юзера в сессию*
5. **Orders –** *проверка того, что сессия не изменила ID, т.е. не была пересоздана*
6. **Orders/id –** *проверка того, что сессия не изменяется между обычными запросами*
7. **giveSessionData –** *проверка того, что https-сессия не умирает после обычных запросов*
8. **Orders**
9. **Logout –** *логаут, уничтожение сессии*
10. **Orders/id –** *проверка того, что сессия не содержит загруженного юзера*
11. **Orders/id –** *проверка того, что логика создания сессий не изменилась*
12. **giveSessionData –** *проверка того, что логаут произошёл*

**1 этап тестирования**

Данный этап тестировки: включена защита по сертификату на два метода: login, giveSessionData.

В конфигурацию добавлена строка: **http.sessionManagement().sessionFixation().none();**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | СОБЫТИЕ | УРЛ | ID СЕССИИ | ОТВЕТ ОТ СЕРВЕРА В ЛОГАХ |
| 1 | Запрос на незащищённый урл, сессия автосоздаётся | /orders | 18968071 | orders SESSION=org.apache.catalina.session.StandardSessionFacade@18968071 FROM DATE 1496833467493 |
| 2 | Запрос на незащищённый урл, юзера в сессии ещё нет, сессия пересоздаётся | /orders/301 | 5cabeb62 | order/id SESSION=org.apache.catalina.session.StandardSessionFacade@5cabeb62 FROM DATE 1496833570214  USERFROMSESSION=null |
| 3 | Запрос на защищённый урл, сессия фиксируется для защищённых запросов, юзер загружается в сессию | /auth/login  **(HTTPS)** | **bf6ebe7** | login SESSION=org.apache.catalina.session.StandardSessionFacade@bf6ebe7 FROM DATE 1496833787848 |
| 4 | Запрос на другой защищённый урл, айди сессии не изменился, **значит что сессия зафиксирована** | /auth/giveSessionData  **(HTTPS)** | **bf6ebe7** | giveSessionData SESSION=org.apache.catalina.session.StandardSessionFacade@bf6ebe7 FROM DATE 1496833787848 |
| 5 | Запрос на незащищённый урл, айди сессии опять поменялся - сессия не была запомнена после авторизации на https | /orders | 678222a6 | orders SESSION=org.apache.catalina.session.StandardSessionFacade@678222a6 FROM DATE 1496833988383 |
| 6 | Запрос на второй незащищённый урл, опять смена айди - **значит что сессия каждый раз пересоздаётся при доступе к незащищённым урлам** | /orders/301 | 3306aec2 | order/id SESSION=org.apache.catalina.session.StandardSessionFacade@3306aec2 FROM DATE 1496834079191  USERFROMSESSION=null |
| 7 | Запрос на защищённый урл, айди сессии - то что было после авторизации, **значит сессия сохраняется только для https** | /auth/giveSessionData  **(HTTPS)** | **bf6ebe7** | giveSessionData SESSION=org.apache.catalina.session.StandardSessionFacade@bf6ebe7 FROM DATE 1496833787848  **В этом запросе юзер вернулся в качестве JSON-ответа, полноценно заполненный данными.** |
| 8 | Запрос опять на незащищённый урл, **опять смена айди сессии** | /orders | 34f12546 | orders SESSION=org.apache.catalina.session.StandardSessionFacade@34f12546 FROM DATE 1496834246985 |
| 9 | Запрос на логаут, защищённая сессия должна быть распущена. | /auth/logout |  |  |
| 10 | Запрос на незащищённый урл, сессия создана заново | /orders/301 | 7ed9fe56 | order/id SESSION=org.apache.catalina.session.StandardSessionFacade@7ed9fe56 FROM DATE 1496834774850  USERFROMSESSION=null |
| 11 | Повтор запроса на незащищённый урл, **сессия каждый раз пересоздаётся при любом незащищённом запросе** | /orders/301 | 6ee36cab | order/id SESSION=org.apache.catalina.session.StandardSessionFacade@6ee36cab FROM DATE 1496834828098  USERFROMSESSION=null |
| 12 | Запрос на защищённый урл, вернулся неразлогиненный пользователь и старый айди сессии | /auth/giveSessionData  **(HTTPS)** | **bf6ebe7** | giveSessionData SESSION=org.apache.catalina.session.StandardSessionFacade@bf6ebe7 FROM DATE 1496833787848 |

***Выводы:***

***При любом виде запроса на незащищённый урл сессия пересоздаётся заново, хранение любого вида информации в объекте сессии невозможно.***

***Сессия для запросов по сертификату сохраняется как и все данные в ней, но логаут не работает по незащищённому урл.***

***Защиты от Session Fixation – атак не предоставляется.***

**2 этап тестирования**

Данный этап тестировки: включена защита по сертификату на два метода: login, giveSessionData.

В конфигурацию добавлена строка: **http.sessionManagement().sessionFixation().migrateSession();**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | СОБЫТИЕ | УРЛ | ID СЕССИИ | ОТВЕТ ОТ СЕРВЕРА В ЛОГАХ |
| 1 | Загружаем незащищённый урл | /orders | 6a5acb81 | orders SESSION=org.apache.catalina.session.StandardSessionFacade@6a5acb81 FROM DATE 1496839119162 |
| 2 | Сессия пересоздаётся | /orders/301 | 787706ba | order/id SESSION=org.apache.catalina.session.StandardSessionFacade@787706ba FROM DATE 1496839164936  USERFROMSESSION=null |
| 3 | Логинимся, получаем новую сессию | /auth/login  **(HTTPS)** | **3d5486dd** | login SESSION=org.apache.catalina.session.StandardSessionFacade@3d5486dd FROM DATE 1496839243240 |
| 4 | Проверяем защищённую сессию, юзер загружается из сессии | /auth/giveSessionData  **(HTTPS)** | **3d5486dd** | giveSessionData SESSION=org.apache.catalina.session.StandardSessionFacade@3d5486dd FROM DATE 1496839243240 |
| 5 | **Сессия пересоздаётся при доступе к незащищённому урл** | /orders | **1ab62b53** | orders SESSION=org.apache.catalina.session.StandardSessionFacade@1ab62b53 FROM DATE 1496839463153 |
| 6 | Сессия каждый раз пересоздаётся, юзера в ней нет | /orders/301 | 2ee84470 | order/id SESSION=org.apache.catalina.session.StandardSessionFacade@2ee84470 FROM DATE 1496839538521  USERFROMSESSION=null |
| 7 | Авторизированная сессия сохраняется | /auth/giveSessionData  **(HTTPS)** | **3d5486dd** | giveSessionData SESSION=org.apache.catalina.session.StandardSessionFacade@3d5486dd FROM DATE 1496839243240 |
| 8 | Прыгаем на незащищённый урл | /orders | 2778fe86 | orders SESSION=org.apache.catalina.session.StandardSessionFacade@2778fe86 FROM DATE 1496839744658 |
| 9 | Разлогиниваемся | /auth/logout |  |  |
| 10 | Пробуем | /orders/301 | 6a26db39 | order/id SESSION=org.apache.catalina.session.StandardSessionFacade@6a26db39 FROM DATE 1496839852076  USERFROMSESSION=null |
| 11 |  | /orders/301 | 1d300d9 | order/id SESSION=org.apache.catalina.session.StandardSessionFacade@1d300d9 FROM DATE 1496839930889  USERFROMSESSION=null |
| 12 | Разлогин не сработал | /auth/giveSessionData  **(HTTPS)** | **3d5486dd** | giveSessionData SESSION=org.apache.catalina.session.StandardSessionFacade@3d5486dd FROM DATE 1496839243240 |

***Выводы:***

***Методика работы Spring Security подозрительно похожа на первый этап тестирования. Нужны дополнительные тесты.***

**3 этап тестирования**

Данный этап тестировки: включена защита по сертификату на два метода: login, giveSessionData.

В конфигурацию добавлена строка: **http.sessionManagement().sessionFixation().newSession();**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | СОБЫТИЕ | УРЛ | ID СЕССИИ | ОТВЕТ ОТ СЕРВЕРА В ЛОГАХ |
| 1 | Пробуем | /orders | b560083 | orders SESSION=org.apache.catalina.session.StandardSessionFacade@b560083 FROM DATE 1496840520376 |
| 2 | Тестируем – сессия пересоздаётся | /orders/301 | 39447861 | order/id SESSION=org.apache.catalina.session.StandardSessionFacade@39447861 FROM DATE 1496840614740  USERFROMSESSION=null |
| 3 | Логинимся | /auth/login  **(HTTPS)** | **118b9333** | login SESSION=org.apache.catalina.session.StandardSessionFacade@118b9333 FROM DATE 1496839243240 |
| 4 | Проверяем – сессия сохраняется | /auth/giveSessionData  **(HTTPS)** | **118b9333** | giveSessionData SESSION=org.apache.catalina.session.StandardSessionFacade@118b9333 FROM DATE 1496839243240 |
| 5 | Проверяем – сессия не сохраняется при доступе к незащищённому url | /orders | 267712d4 | orders SESSION=org.apache.catalina.session.StandardSessionFacade@267712d4 FROM DATE 1496840762247 |
| 6 | Стучимся на незащищённый – сессия каждый раз меняется | /orders/301 | 9b3f7d4 | order/id SESSION=org.apache.catalina.session.StandardSessionFacade@9b3f7d4 FROM DATE 1496840812842  USERFROMSESSION=null |
| 7 | Проверяем жизненный цикл авторизированной сессии | /auth/giveSessionData  **(HTTPS)** | **118b9333** | giveSessionData SESSION=org.apache.catalina.session.StandardSessionFacade@118b9333 FROM DATE 1496839243240 |
| 8 |  | /orders | 448e5f47 | orders SESSION=org.apache.catalina.session.StandardSessionFacade@448e5f47 FROM DATE 1496841108367 |
| 9 | Разлогиниваемся | /auth/logout |  |  |
| 10 |  | /orders/301 | 2ddd3560 | order/id SESSION=org.apache.catalina.session.StandardSessionFacade@2ddd3560 FROM DATE 1496841167224  USERFROMSESSION=null |
| 11 |  | /orders/301 | 70cca491 | order/id SESSION=org.apache.catalina.session.StandardSessionFacade@70cca491 FROM DATE 1496841202521  USERFROMSESSION=null |
| 12 | Проверяем – логаут не сработал | /auth/giveSessionData  **(HTTPS)** | **118b9333** | giveSessionData SESSION=org.apache.catalina.session.StandardSessionFacade@118b9333 FROM DATE 1496839243240 |

***Вывод: результаты не меняются от смены методики работы SpringSecurity, логаут не работает***

**4 этап тестирования**

Данный этап тестировки: включена защита по сертификату на три метода: login, logout, giveSessionData.

Логаут метод промаплен как защищённый.

В конфигурацию добавлена строка и перемещена в шапку метода: **http.sessionManagement().sessionFixation().changeSessionId ();**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | СОБЫТИЕ | УРЛ | ID СЕССИИ | ОТВЕТ ОТ СЕРВЕРА В ЛОГАХ |
| 1 | Тыкаемся на урл | /orders | 5060fa | orders SESSION=org.apache.catalina.session.StandardSessionFacade@5060fa FROM DATE 1496842067456 |
| 2 | Пробуем, незащищённый урл, новая сессия создаётся | /orders/301 | 30c54f0a | order/id SESSION=org.apache.catalina.session.StandardSessionFacade@30c54f0a FROM DATE 1496842121395  USERFROMSESSION=null |
| 3 | Создаём защищённую сессию | /auth/login  **(HTTPS)** | **71da8591** | login SESSION=org.apache.catalina.session.StandardSessionFacade@71da8591 FROM DATE 1496839243240 |
| 4 | Проверяем – сессия сохраняется | /auth/giveSessionData  **(HTTPS)** | **71da8591** | giveSessionData SESSION=org.apache.catalina.session.StandardSessionFacade@71da8591 FROM DATE 1496839243240 |
| 5 | Проверяем – незащищённая сессия не сохраняется | /orders | d7a60f1 | orders SESSION=org.apache.catalina.session.StandardSessionFacade@d7a60f1 FROM DATE 1496842307966 |
| 6 |  | /orders/301 | 49a764ab | order/id SESSION=org.apache.catalina.session.StandardSessionFacade@49a764ab FROM DATE 1496842349210  USERFROMSESSION=null |
| 7 | Сессия сохраняется если она защищённая | /auth/giveSessionData  **(HTTPS)** | **71da8591** | giveSessionData SESSION=org.apache.catalina.session.StandardSessionFacade@71da8591 FROM DATE 1496839243240 |
| 8 |  | /orders | 1264458d | orders SESSION=org.apache.catalina.session.StandardSessionFacade@1264458d FROM DATE 1496842434980 |
| 9 | Удаляем сессию по защищённому каналу | /auth/logout  **(HTTPS)** |  |  |
| 10 |  | /orders/301 | 27d94c1a | order/id SESSION=org.apache.catalina.session.StandardSessionFacade@27d94c1a FROM DATE 1496842538408  USERFROMSESSION=null |
| 11 |  | /orders/301 | 27b01c92 | order/id SESSION=org.apache.catalina.session.StandardSessionFacade@27b01c92 FROM DATE 1496842581135  USERFROMSESSION=null |
| 12 | Проверяем – работа алгоритма не изменилась | /auth/giveSessionData  **(HTTPS)** | **71da8591** | giveSessionData SESSION=org.apache.catalina.session.StandardSessionFacade@71da8591 FROM DATE 1496839243240 |
| 13 | Логаутимся без /auth/ контроллера | **/logout**  **(HTTPS)** |  |  |
| 14 | Логаут произошёл успешно – значит урл пишется без контроллера | /auth/giveSessionData  **(HTTPS)** | **658258e0** | giveSessionData SESSION=org.apache.catalina.session.StandardSessionFacade@658258e0 FROM DATE 1496842692736 |

***Вывод: результаты не меняются от смены методики работы SpringSecurity, но логаут работает успешно.***

**5 этап тестирования**

Данный этап тестировки: включена защита по сертификату на три метода: login, logout, giveSessionData.

В конфигурацию добавлена строка и перемещена в шапку метода: **http.sessionManagement().sessionCreationPolicy(SessionCreationPolicy.NEVER);**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | СОБЫТИЕ | УРЛ | ID СЕССИИ | ОТВЕТ ОТ СЕРВЕРА В ЛОГАХ |
| 1 | Стучимся | /orders | 35a37f2a | orders SESSION=org.apache.catalina.session.StandardSessionFacade@35a37f2a FROM DATE 1496843660478 |
| 2 | Сессия пересоздаётся | /orders/301 | ddd9e80 | order/id SESSION=org.apache.catalina.session.StandardSessionFacade@ddd9e80 FROM DATE 1496843743770  USERFROMSESSION=null |
| 3 | Логинимся | /auth/login  **(HTTPS)** | **340dafee** | login SESSION=org.apache.catalina.session.StandardSessionFacade@340dafee FROM DATE 1496842692736 |
| 4 | Проверяем – сессия держится | /auth/giveSessionData  **(HTTPS)** | **340dafee** | giveSessionData SESSION=org.apache.catalina.session.StandardSessionFacade@340dafee FROM DATE 1496842692736 |
| 5 |  | /orders | 45203e1d | orders SESSION=org.apache.catalina.session.StandardSessionFacade@45203e1d FROM DATE 1496843901050 |
| 6 |  | /orders/301 | 4489e6b | order/id SESSION=org.apache.catalina.session.StandardSessionFacade@4489e6b FROM DATE 1496843923345  USERFROMSESSION=null |
| 7 | Проверяем – защищённая сессия сохраняется | /auth/giveSessionData  **(HTTPS)** | **340dafee** | giveSessionData SESSION=org.apache.catalina.session.StandardSessionFacade@340dafee FROM DATE 1496842692736 |
| 8 |  | /orders | 24a5f996 | orders SESSION=org.apache.catalina.session.StandardSessionFacade@24a5f996 FROM DATE 1496844144955 |
| 9 | Логаутимся | / logout  **(HTTPS)** |  |  |
| 10 |  | /orders/301 | 601bf9a4 | order/id SESSION=org.apache.catalina.session.StandardSessionFacade@601bf9a4 FROM DATE 1496844192044  USERFROMSESSION=null |
| 11 |  | /orders/301 | 576c045b | order/id SESSION=org.apache.catalina.session.StandardSessionFacade@576c045b FROM DATE 1496844256093  USERFROMSESSION=null |
| 12 | Сессия уничтожена | /auth/giveSessionData  **(HTTPS)** | **4dfc932e** | giveSessionData SESSION=org.apache.catalina.session.StandardSessionFacade@4dfc932e FROM DATE 1496844286692 |

***Вывод: результаты не меняются от смены методики работы SpringSecurity, но логаут работает успешно. Строка sessionCreationPolicy(SessionCreationPolicy.NEVER) ничего не изменила. Возможно требуется изменить и политику создания сессии и SessionFixation.***

**6 этап тестирования**

Данный этап тестировки: включена защита по сертификату на три метода: login, logout, giveSessionData.

В конфигурацию добавлены строки: **http.sessionManagement().sessionCreationPolicy(SessionCreationPolicy.ALWAYS);**

**http.sessionManagement().sessionFixation().migrateSession();**

Согласно логике спринг-секьюрити сессия должна каждый раз пересоздаваться и в неё копироваться информация с предыдущей сессии.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | СОБЫТИЕ | УРЛ | ID СЕССИИ | ОТВЕТ ОТ СЕРВЕРА В ЛОГАХ |
| 1 | Стучимся | /orders | 6787ed93 | orders SESSION=org.apache.catalina.session.StandardSessionFacade@6787ed93 FROM DATE 1496844996769 |
| 2 | Сессия пересоздаётся | /orders/301 | 39e95b0 | order/id SESSION=org.apache.catalina.session.StandardSessionFacade@39e95b0 FROM DATE 1496845030268  USERFROMSESSION=null |
| 3 | Логинимся | /auth/login  **(HTTPS)** | **5e8c595** | login SESSION=org.apache.catalina.session.StandardSessionFacade@5e8c595 FROM DATE 1496844286692 |
| 4 | Проверяем – сессия держится, не пересоздаётся, настройка не влияет на ход работы | /auth/giveSessionData  **(HTTPS)** | **5e8c595** | giveSessionData SESSION=org.apache.catalina.session.StandardSessionFacade@5e8c595 FROM DATE 1496844286692 |
| 5 |  | /orders | 748b4e17 | orders SESSION=org.apache.catalina.session.StandardSessionFacade@748b4e17 FROM DATE 1496845171520 |
| 6 |  | /orders/301 | 3fb017f4 | order/id SESSION=org.apache.catalina.session.StandardSessionFacade@3fb017f4 FROM DATE 1496845210266  USERFROMSESSION=null |
| 7 | Проверяем – защищённая сессия сохраняется | /auth/giveSessionData  **(HTTPS)** | **5e8c595** | giveSessionData SESSION=org.apache.catalina.session.StandardSessionFacade@5e8c595 FROM DATE 1496844286692 |
| 8 |  | /orders | 15a5a6af | orders SESSION=org.apache.catalina.session.StandardSessionFacade@15a5a6af FROM DATE 1496845272072 |
| 9 | Логаутимся | / logout  **(HTTPS)** |  |  |
| 10 |  | /orders/301 | 131121b4 | order/id SESSION=org.apache.catalina.session.StandardSessionFacade@131121b4 FROM DATE 1496845312799  USERFROMSESSION=null |
| 11 |  | /orders/301 | 66d7cdd0 | order/id SESSION=org.apache.catalina.session.StandardSessionFacade@66d7cdd0 FROM DATE 1496845403077  USERFROMSESSION=null |
| 12 | Сессия уничтожена | /auth/giveSessionData  **(HTTPS)** | **1999d2e1** | giveSessionData SESSION=org.apache.catalina.session.StandardSessionFacade@1999d2e1 FROM DATE 1496845304070 |

***Вывод: результаты не меняются от смены методики работы SpringSecurity. Подобная методика работы SpringSecurity указывает на то, что данная проблема есть ограничение BasicAuthentication, реализованная для того чтобы не допускать дырок в защите.***

***Обратимся к официальной документации:***

*«Basic and digest authentication are alternative authentication mechanisms which are popular in web applications.* ***Basic authentication is often used with stateless clients which pass their credentials on each request.*** *It’s quite common to use it in combination with form-based authentication where an application is used through both a browser-based user interface and as a web-service. However, basic authentication transmits the password as plain text so it should only really be used over an encrypted transport layer such as HTTPS.»*

*-*[*http://docs.spring.io/autorepo/docs/spring-security/current/reference/html/basic.html*](http://docs.spring.io/autorepo/docs/spring-security/current/reference/html/basic.html)

**«Digest Authentication will NOT work if you are using encoded passwords in your DAO****[[15]](http://docs.spring.io/autorepo/docs/spring-security/current/reference/html/basic.html" \l "ftn.d5e2867).»**

*-* [*http://docs.spring.io/autorepo/docs/spring-security/current/reference/html/basic.html*](http://docs.spring.io/autorepo/docs/spring-security/current/reference/html/basic.html)