

Spring Security

Практика

Spring Security

Аутентификация - это то, как мы проверяем личность того, кто пытается получить доступ к определенному ресурсу. Обычным способом аутентификации пользователей является требование ввести имя пользователя и пароль. После аутентификации мы узнаем личность пользователя и сможем выполнить авторизацию.

Spring Security

Авторизация - предоставление определённому лицу или группе лиц прав на выполнение определённых действий

Spring Security

Пароль может храниться разными способами.

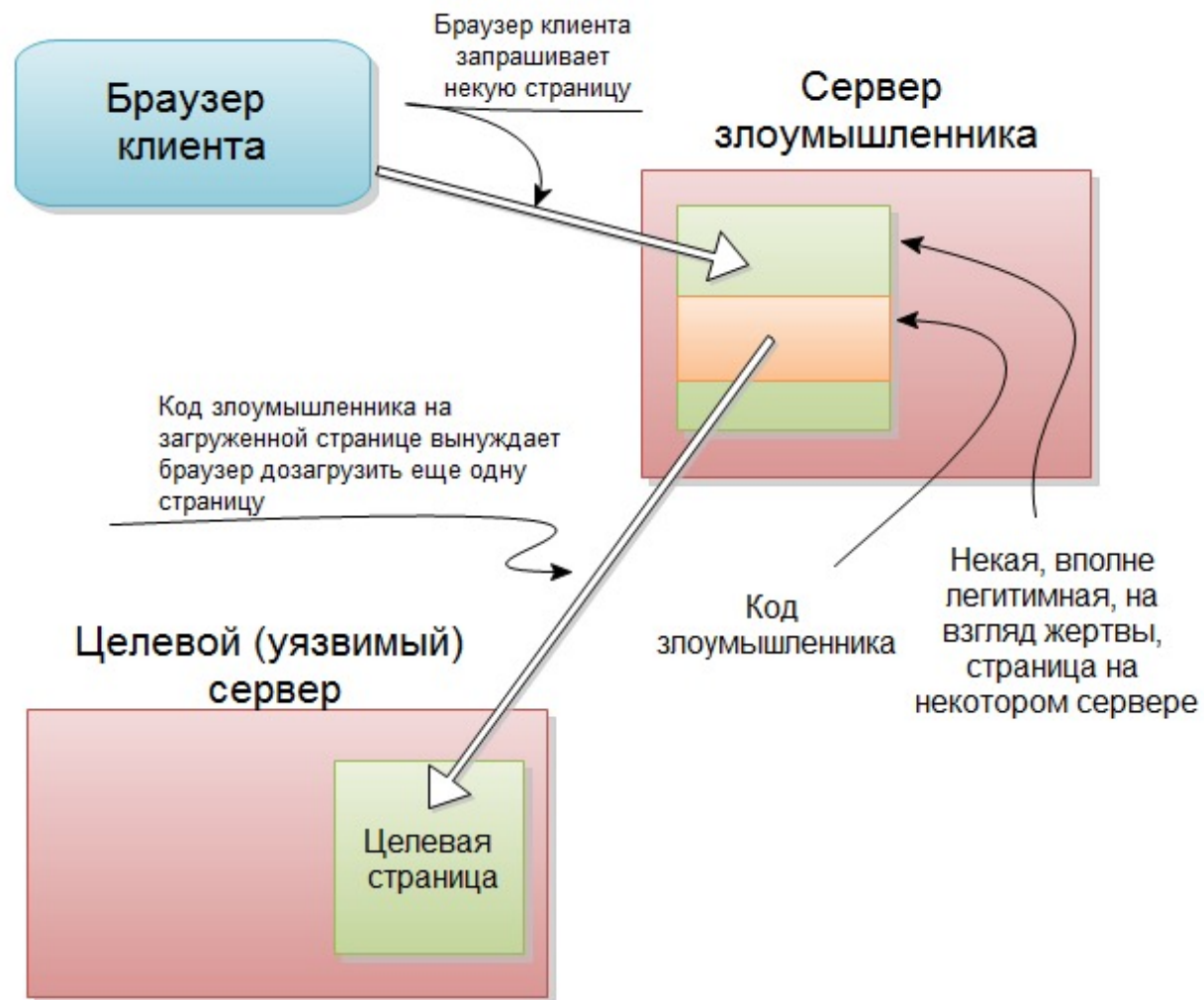
- Открытый текст
- Хеш
- Хеш с солью
- Адаптивные хеш-функции

Spring Security

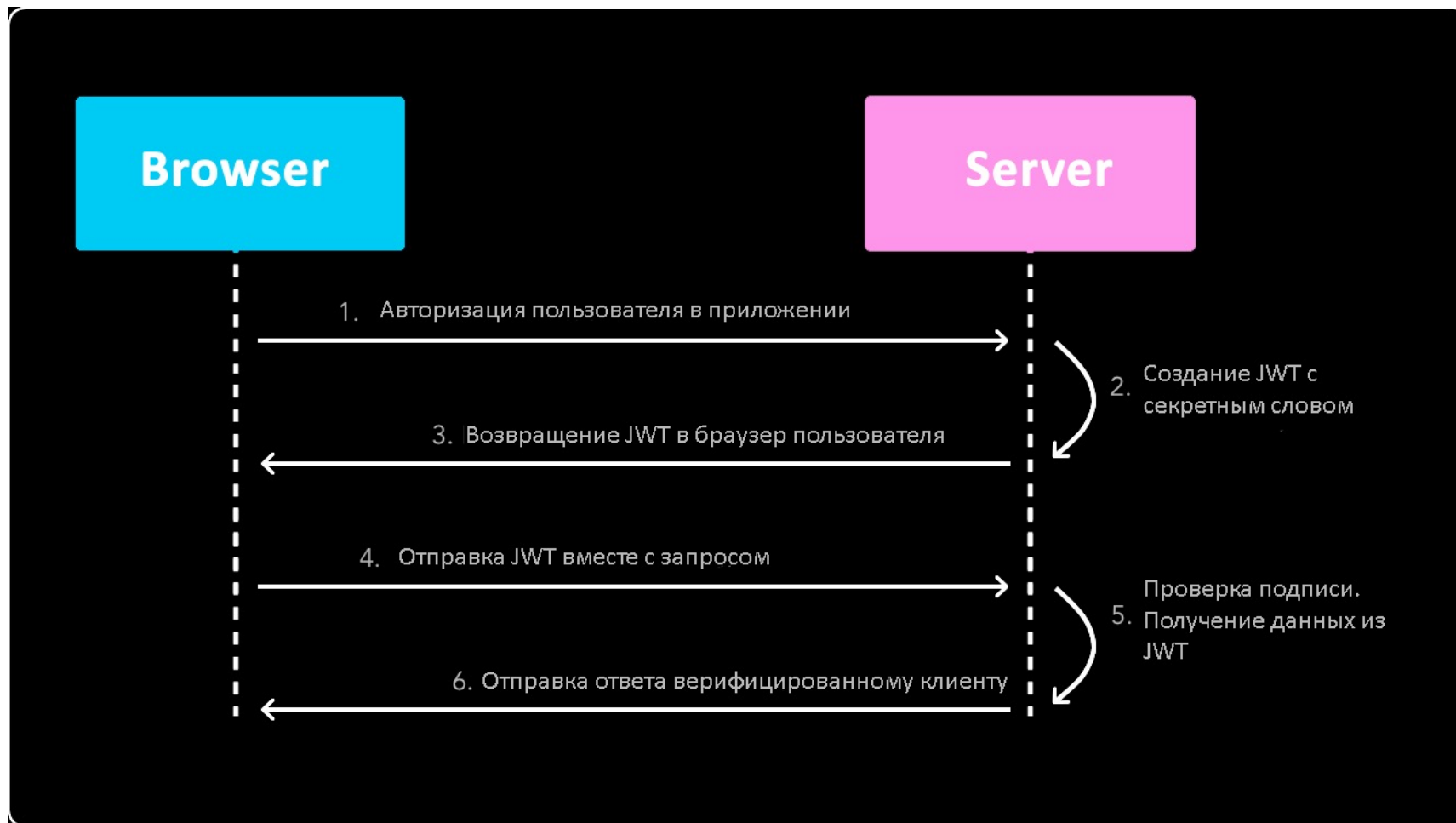
Как выглядит пароль, зашифрованный разными способами:

- {bcrypt}\$2a\$10\$dXJ3SW6G7P50lGmMkkmwe.20cQQubK3.HZWzG3YB1tIRy.fqvM/BG
- {noop}password
- {pbkdf2}5d923b44a6d129f3ddf3e3c8d29412723dcbde72445e8ef6bf3b508fbf17fa4ed4d6b99ca763d8dc
- {scrypt}\$e0801\$8bWJaSu2IKSn9Z9kM+TPXfOc/9bdYSrN1oD9qfVThWEwdRTnO7re7Ei+fUZRJ68k9lTyuTeUp4of4g24hHnazw==\$OAOec05+bXxvuu/1qZ6NUR+xQYvYv7BeL1QxwRpY5Pc=
- {sha256}97cde38028ad898ebc02e690819fa220e88c62e0699403e94fff291cfffaf8410849f27605abcbcb0

Защита от CSRF (Cross-site request forgery)



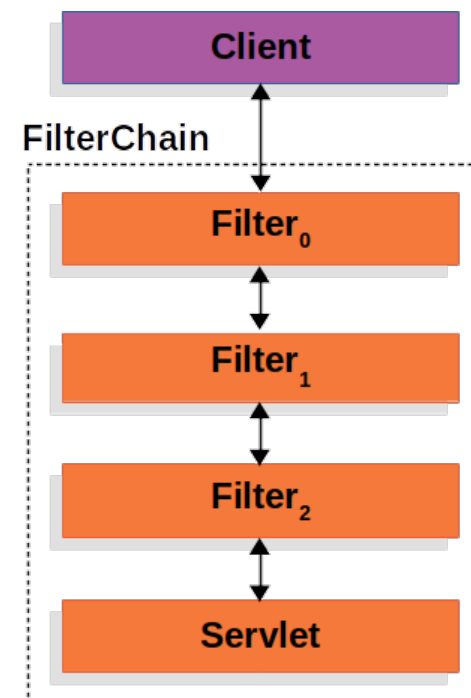
JWT-токен



Spring Security autoconfiguration

Spring Boot автоматически:

- Включает конфигурацию Spring Security по умолчанию, которая создает сервлет Filter с именем `springSecurityFilterChain`. Этот фильтр отвечает за всю безопасность (защита URL-адресов приложения, проверка имени пользователя и пароля, перенаправление на форму входа и т.д.) в вашем приложении.
- Создает бин `UserDetailsService` с именем пользователя `user` и случайно сгенерированным паролем, который записывается в консоль.
- Регистрирует фильтр `springSecurityFilterChain` в контейнере сервлетов
- Генерирует форму входа по умолчанию и пароль пользователю `admin`
- Предотвращает атак CSRF

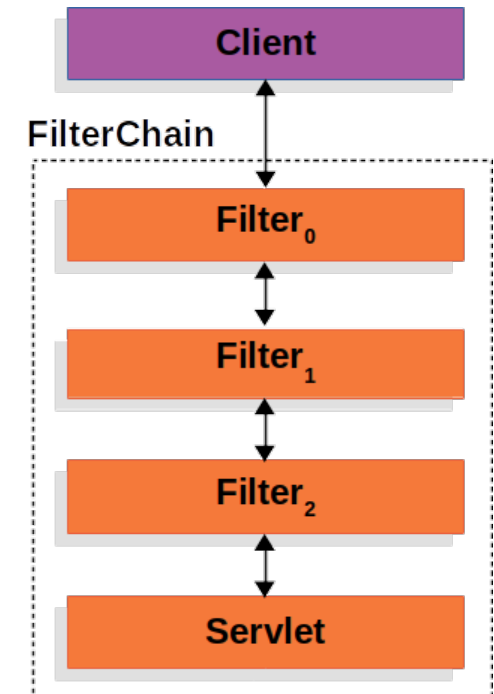


Spring Security autoconfiguration

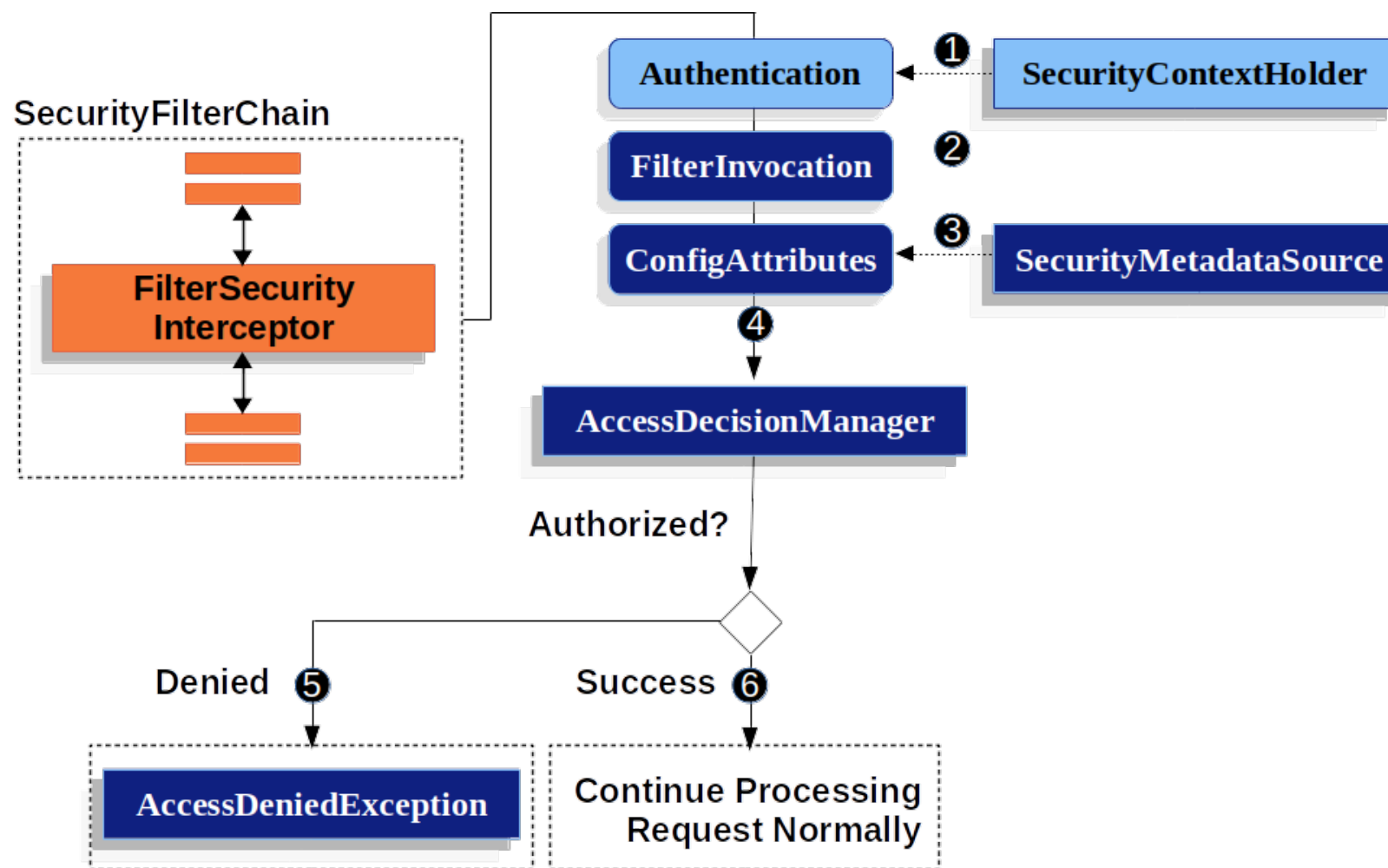
Всего 32 фильтра:

ChannelProcessingFilter
WebAsyncManagerIntegrationFilter
SecurityContextPersistenceFilter
HeaderWriterFilter
CorsFilter
CsrfFilter
LogoutFilter
OAuth2AuthorizationRequestRedirectFilter
Saml2WebSsoAuthenticationRequestFilter
X509AuthenticationFilter
AbstractPreAuthenticatedProcessingFilter
CasAuthenticationFilter
OAuth2LoginAuthenticationFilter
Saml2WebSsoAuthenticationFilter

UsernamePasswordAuthenticationFilter
OpenIDAuthenticationFilter
DefaultLoginPageGeneratingFilter
DefaultLogoutPageGeneratingFilter
ConcurrentSessionFilter
DigestAuthenticationFilter
BearerTokenAuthenticationFilter
BasicAuthenticationFilter
RequestCacheAwareFilter
SecurityContextHolderAwareRequestFilter
JaasApiIntegrationFilter
RememberMeAuthenticationFilter
AnonymousAuthenticationFilter
OAuth2AuthorizationCodeGrantFilter
SessionManagementFilter
ExceptionTranslationFilter
FilterSecurityInterceptor
SwitchUserFilter



Spring Security – доступ к ресурсам



Практика

Подключаем Security

Конфигурация SpringSecurity

1. Базовая аутентификация
2. Настройка пользователей через UserDetails
3. Кастомный password encoder
4. Роли пользователя
5. Permissions – как дополнение ролей
6. Авторизация через БД
7. JWT-token

Спасибо!