Аутентификация в SSO через внешнюю систему

Схема аутентификации

```
@startuml
autonumber
title Аутентификация пользователя по логину/паролю через SSO External JWT Login Module

' Actors
actor Пользователь as customer #3F9C35

participant "Личный кабинет\n(или другая закрытая страница)" as selfcare #7B7BC0

participant "SSO Server" as sso #F78F1E

participant "Authentication Service\n(система выполняющая аутентификацию)" as auth
#F78F1E

' Use Case
note over customer, auth : Успешный вход пользователя в личный кабинет

customer -> selfcare : запрашивает страницу личного кабинета
            selfcare -> sso : запрашивает статус пользователя
            selfcare <- sso : отвечает, что пользователь не аутентифицирован
customer <-- selfcare : возвращает редирект на SSO

customer -> sso : происходит редирект
customer <-- sso : записывает в cookie URL для обратного перехода и возвращает
редирект на сервис аутентификации

customer ->  auth : происходит редирект
customer <-  auth : возвращает страницу логина
customer --> auth :
customer <-- auth : происходит процесс аутентификации, завершается успешно
customer --> auth
             auth -> auth : генерирует токен JWT,\n записывает в cookie
customer <-- auth : возвращает редирект на SSO

customer -> sso : происходит редирект
            sso -> sso : проверяет токен JWT из cookie,\nдобавляет session cookie
customer <-- sso : возвращает редирект на страницу личного кабинета

customer -> selfcare : происходит редирект
            selfcare ->  sso : запрашивает статус пользователя по session cookie
            selfcare <- sso : отвечает, что пользователь аутентифицирован
customer <- selfcare : отображает содержимое личного кабинета
```

```
' Use Case
note over customer, auth : Неуспешный вход пользователя в личный кабинет

customer -> selfcare : запрашивает страницу личного кабинета
            selfcare -> sso : проверяет статус пользователя
            selfcare <- sso : отвечает, что пользователь не аутентифицирован
customer <-- selfcare : возвращает редирект на SSO

customer -> sso : происходит редирект
customer <-- sso : записывает в cookie URL для обратного перехода и возвращает
редирект на сервис аутентификации

customer ->  auth : происходит редирект
customer <-  auth : возвращает страницу логина
customer --> auth
customer <-- auth : происходит процесс аутентификации, завершается неуспешно
customer --> auth
             auth -> auth : генерирует токен JWT,\n записывает в cookie
customer <-- auth : возвращает редирект на SSO

customer -> sso : происходит редирект
customer <-- sso : проверяет токен из cookie, возвращает редирект на сервис
аутентификации с кодами ошибок

customer -> auth : происходит редирект
customer <- auth : возвращает страницу логина с ошибками


@enduml
```

Логин Модуль ExternalJwtLoginModule

Модуль делегирует процесс аутентификации некоторой внешней системе путем редиректа на ее адрес Внешняя система может использовать любую цепочку аутентификации, после чего результат отправить редиректом обратно в SSO

Интеграция

SSO → External Auth Service

Задать настройку com.rooxteam.federation.authentication.baseUrl входной точкой логина во внешней системе. по этому адресу будут приходить GET запросы для аутентификации.

External Auth Service → SSO

В случае успешной аутентификации нужно сохранить имя пользователя и справочную информацию в виде зашифрованного токена (JWT) и записать в cookie из настройки com.rooxteam.federation.authentication.cookie.name (по умолчанию iPlanetDirectoryPro), время

жизни токена должно быть несколько секунд. В случае неуспешной аутентификации нужно передать токен без информации о пользователе. В cookie из com.rooxteam.federation.authentication.backUrlCookie (по умолчанию sso_goto) должен быть записан обраный URL SSO, к этому URL необходимо добавить GET-параметр из com.rooxteam.federation.authentication.backStatusParam (по умолчанию auth_status) о статусе аутентификации (SUCCESS) и по полученному URL выполнить редирект.

Настройки

- com.rooxteam.federation.authentication.decryptor.key - Ключ расшифровки JWT токена
- com.rooxteam.federation.authentication.baseUrl - URL внешней системы аутентификации
- com.rooxteam.federation.authentication.backUrlCookie - Имя Cookie где находится URL для обратного редиректа в SSO
- com.rooxteam.federation.authentication.backStatusParam - Имя GET-параметра, в который внешняя система должна передать статус аутентификации
- com.rooxteam.federation.authentication.cookie.name - Имя Cookie в которой передается JWT
- com.rooxteam.federation.authentication.cookie.domain - Домен Cookie в которой передается JWT
- com.rooxteam.federation.authentication.cookie.path - Путь Cookie в которой передается JWT