IAM. Аутентификация и авторизация

Бизнес-требования

- Обеспечить аутентификацию пользователей исключительно через IAM по протоколу OIDC
- Обеспечить обработку результата аутентификации в IAM с получением данных пользователя и созданием или обновлением учетной записи пользователя в системе.
- Обеспечить сопоставление пользователей IAM с пользователями сервиса на основе уникального идентификатора, передаваемого провайдером.
- Обеспечить получение tenant_name и ролей пользователя из IAM и их использование в логике.
- Сохранить разграничение доступа к функциональности сервиса на основе ролей, полученных из IAM, и внутренних правил сервиса.
- Обеспечить сохранение минимально необходимого представления пользователя и тенанта в системе без хранения учетных данных и механизмов аутентификации IAM.
- Обеспечить управление пользовательской сессией на стороне сервиса после успешной аутентификации.
- Обеспечить автоматическое перенаправление пользователя в IAM при истечении сессии или при попытке доступа к защищённым ресурсам без валидной сессии.
- Обеспечить безопасную обработку пользовательских данных и токенов без их утечки в логи и ответы API.

Решение

Аутентификация

Системная логика (бэкенд)

Сервис реализует аутентификацию через IAM по протоколу OIDC с использованием authorization code flow.

Бэкенд выступает OIDC-клиентом и взаимодействует с IAM, используя client_id и client_secret.

При обращении к защищённым ресурсам без активной сессии бэкенд инициирует процесс аутентификации, возвращая редирект на IAM.

Редирект на IAM:

```
GET /auth/login


→ 302 Redirect:
https://<iam>/authorize?
client_id=<client_id>&
redirect_uri=<redirect_uri>&
response_type=code&
scope=openid profile email&
state=<random_state>
```

Параметр state используется для защиты от CSRF и должен проверяться при возврате пользователя.

После успешной аутентификации IAM перенаправляет пользователя на redirect_uri сервиса с authorization code.

Ответ от IAM:

```
GET /auth/callback?code=<code>&state=<state>
```

Бэкенд:

- валидирует параметр state;
- извлекает code.

Бэкенд выполняет обмен authorization code на токены.

Запрос к IAM:

```
POST /token
Content-Type: application/x-www-form-urlencoded


grant_type=authorization_code
code=<code>
redirect_uri=<redirect_uri>
client_id=<client_id>
```

```
client_secret=<client_secret>
```

Ответ IAM:

```
{
"access_token": "...",
"id_token": "...",
"expires_in": 3600,
"refresh_token": "..."
}
```

Бэкенд обрабатывает полученные токены:

- декодирует токен
- извлекает:
  - user_id (sub);
  - email / username;
  - tenant_name;
  - роли пользователя.

На основании полученных данных:

Работа с пользователем:

- если пользователь существует → используется;
- если нет → создаётся новый.

Работа с тенантом:

- поиск tenant по tenant_name;
- если отсутствует → создаётся новый tenant;
- определяется tenant_id;
- после этого бэкенд создаёт пользовательскую сессию.

Бэкенд:

- извлекает сессию;
- проверяет её валидность;
- при успехе допускает запрос.

При отсутствии или истечении сессии:

```
→ 302 Redirect → /auth/login
```

IAM не участвует в обработке каждого запроса и используется только на этапе аутентификации.

Токены IAM не передаются на фронтенд и не используются как основной механизм авторизации после создания сессии.

Пользовательский интерфейс

Фронтенд не реализует собственную форму логина и не взаимодействует напрямую с IAM API

Сценарии использования

Пользователь открывает приложение. При отсутствии сессии происходит редирект в IAM. После логина пользователь возвращается в приложение, где создаётся сессия.

Пользователь переходит по прямой ссылке к защищённому ресурсу. При отсутствии сессии выполняется редирект в IAM, после чего пользователь возвращается к исходному ресурсу.

Пользователь уже аутентифицирован в IAM. При переходе в приложение вход происходит автоматически без повторного ввода учетных данных.

Системная логика (фронтенд)

Фронтенд инициирует обращение к бэкенду для загрузки приложения и API.

Фронтенд не обрабатывает токены IAM и не хранит их.

Если сессия отсутствует:

- фронтенд получает редирект на /auth/login
- браузер автоматически перенаправляется в IAM.

После успешной аутентификации:

IAM → redirect → callbck

Далее пользователь возвращается в приложение уже с установленной сессией.

Если сессия истекла:

- фронтенд получает редирект;
- инициирует переход на /auth/login

Авторизация пользователей

Системная логика (бэкенд)

Авторизация в сервисе выполняется на основе ролей, полученных от IAM в процессе аутентификации и сохранённых в пользовательской сессии.

Получение ролей от IAM (этап аутентификации)

После обмена code → tokens бэкенд извлекает роли из токена IAM:

```
id_token / access_token → decode
```

Пример ответа:

```
{
"sub": "user-123",
"realm_access": {
"roles": ["admin", "user"]
},
"tenant_name": "company_a"
}
```

Бэкенд:

- извлекает список ролей;
- при необходимости выполняет маппинг во внутренние роли;
- сохраняет роли в сессии.

Обновление ролей

Роли пользователя обновляются только при повторной аутентификации: новый логин → новый токен → новые роли

Сервис не синхронизирует роли с IAM в фоне и не запрашивает их дополнительно.

Если в токене отсутствуют ожидаемые роли, то у пользователя будет закрыт доступ

Сценарии использования

Пользователь проходит аутентификацию через IAM. Бэкенд получает роли из токена и сохраняет их в сессии.

Пользователь выполняет запросы к API. Сервис использует роли из сессии без обращения к IAM.

Роли пользователя изменяются в IAM. После повторного входа пользователь получает обновлённые роли.

Revision #4
Created 3 April 2026 11:58:03 by Артём
Updated 7 April 2026 16:09:09 by Артём