IAM. Добавление пользователей

Данные от ГосТех по добавлению пользователей

Термины

- ОГВ – инициатор заказа.
- Заказчик – ФКУ Гостех.
- Оператор платформы – ЕЦП, управляет платформой и IAM.
- Администратор платформы – сотрудник Оператора, выполняет настройку.
- Вендор ПО – разработчик прикладного SaaS решения.
- Администратор тенанта – представитель ОГВ, которому делегируются права.

Сценарий

1. **Оператор** в **системе управления заказами** забирает:
   - tenant_id – уникальный идентификатор тенанта;
   - tenant_name – наименование организации ОГВ;
   - фиксирует список пользователей с ролями (администратор тенанта, пользователь).
   - Фиксирует название подключенной услуги (например service_name). Может быть более одной услуги.

2. **Оператор** формирует задачу для **Администратора платформы** (с указанием всех данных) и назначает её.
3. **Администратор платформы** (или, при разделении прав, **Администратор IAM**) в Keycloak:
   - tenant_id (из заказа)
   - tenant_name (название организации)
   - использует штатный механизм экспорта Keycloak (CLI или Partial Export, если поддерживается для пользователей);

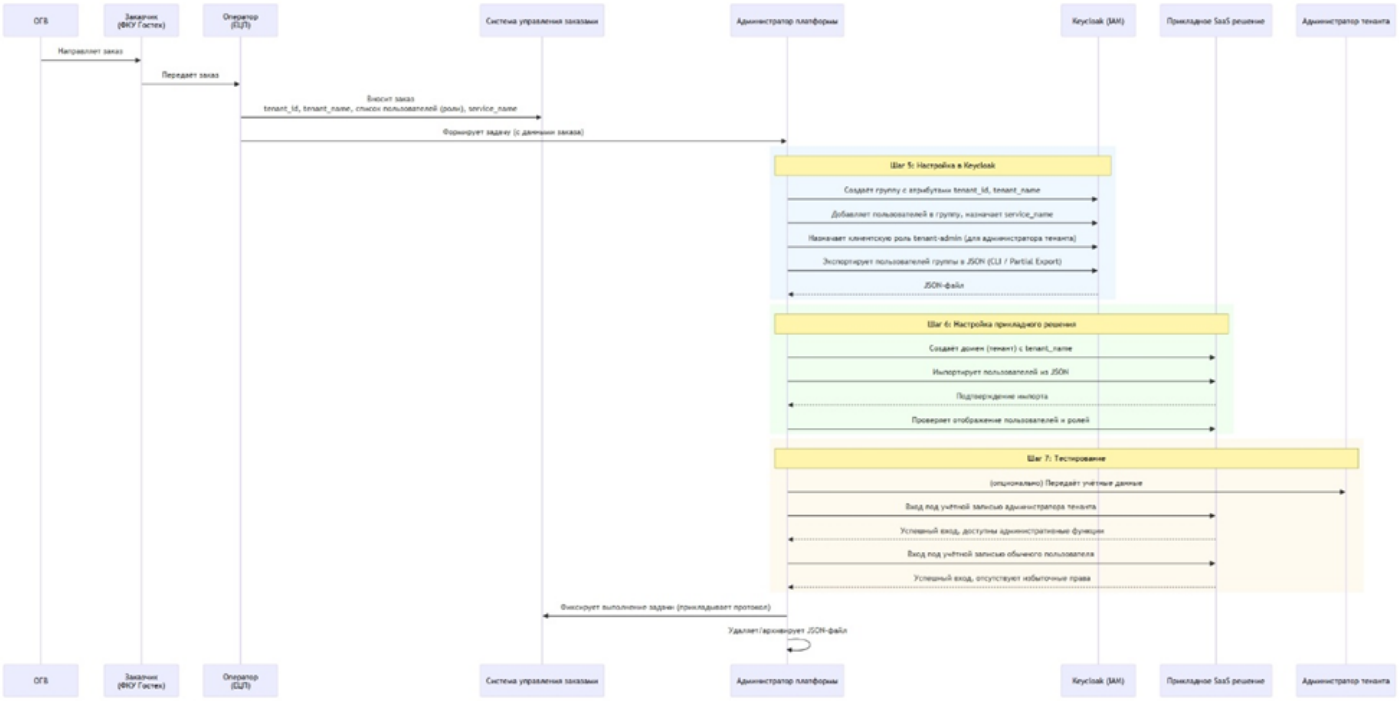
- убеждается, что в экспортированном файле присутствуют все необходимые поля (включая id, username, email, атрибуты, а также связь с группой).
  - в соответствующем **realm** создаёт **группу** с атрибутами:
  - добавляет в группу всех пользователей из заказа;
  - присваивает пользователям параметр service_name (по числу доступных к заказу услуг);
  - для пользователя, назначенного администратором тенанта, дополнительно назначает **клиентскую роль** (например, tenant-admin), которая будет передаваться в токен;
  - выполняет **экспорт пользователей группы** в **JSON**:

4. **Администратор платформы** (или, при разделении, **Администратор прикладного сервиса**) передаёт JSON-файл (через защищённый канал внутри контура) и приступает к настройке прикладного SaaS решения:
   - создаёт в прикладном решении **домен** (тенант) с именем, соответствующим tenant_name;
   - выполняет **импорт пользователей** из JSON-файла, используя возможности импорта, предоставленные вендором;
   - проверяет, что все пользователи отобразились в интерфейсе;
   - убеждается, что **администратор тенанта** получил роль, позволяющую управлять доменом;
   - всем остальным пользователям по умолчанию присваивается **базовая роль** (без административных привилегий).

5. **Администратор платформы** проводит **тестирование**:
   - входит в прикладное решение под учётной записью администратора тенанта;
   - проверяет, что доступны административные функции (управление пользователями, настройки домена);
   - входит под учётной записью обычного пользователя;
   - проверяет, что избыточные права отсутствуют.

6. **Администратор платформы** фиксирует выполнение задачи в системе управления заказами, прикладывает протокол тестирования.
7. JSON-файл с пользовательскими данными **удаляется** (или перемещается в архив с ограничением доступа) для предотвращения несанкционированного использования.

Схема сценария



Бизнес-требования

- Требуется подготовить шаблон JSON-файла для загрузки в БД Fastboard
- Требуется инструмент для загрузки JSON-файла с пользователями из IAM в FB

Решение

Стркутура JSON

Массив объектов (пользователей). Для каждого объекта передаются:

- tenant_name – название основного тенанта, к которому привязывается пользователь (обязательно)
- login – логин пользователя в сервисе (обязательно)
- password – пароль для входа пользователя (обязательно)
- name – имя пользователя для отображения в сервисе
- surname – фамилия пользователя для отображения в сервисе
- email – почта пользователя для взаимодействия в сервисе (обязательно)
- role – роль пользователя в системе, состоит из один из вариантов (обязательно):
  - admin (Администратор сервиса)
  - tenant_admin (Администратор потребителя)

- developer (Авторизованный пользователь с доступом к подключению данных интерактивных отчетов)
- analyst (Авторизованный пользователь с доступом к формированию интерактивных отчетов)
- viewer (Авторизованный пользователь с доступом к просмотру интерактивных отчетов)

```
[
  {
    "tenant_name": "string",
    "login": "string",
    "password": "string",
    "name": "string",
    "surname": "string",
    "email": "string",
    "role": "string"
  }
  ...
]
```

Создание пользователей в системе из JSON

Пользовательский интерфейс

Расположение: Панель администратора → Вкладка "Пользователи" → Список пользователей

Внизу списка пользователей должна быть кнопка "Импортировать пользователей"

Системная логика (frontend)

При нажатии на кнопку открывается filepicker для выбора JSON-файла с устройства пользователя.

После выбора файла отправляется запрос со списком пользователей для добавления в систему:

- POST-запрос
- Передаёт в себе выбранный файл

В случае успеха приходит ответ со списком пользователей и их атрибутов (как в запросе GET back/api/v2/admin/users).

После успешного ответа необходимо обновить список пользователей слева всеми данными из ответа.

В успешном ответе также может прийти список с пользователями, которых не удалось добавить в систему (остутствует часть обязательных полей). Этот список необходимо показывать в предупреждении (желтое всплывающее сообщение) с текстом: "Пользователи [массив логинов] не удалось зарегистрировать в системе: отсутствует одно или несколько обязательных полей". Если ни одного такого пользователя не пришло в ответе, то показывать сообщение не требуется.

С случае ошибки в интерфейсе отображается всплывающее сообщение с ошибкой с серверной части.

Системная логика (backend)

Новый POST-запрос на передачу с клиентской части списка новых пользователей в JSON-файле, передаёт в себе сам файл.

При получении файла с клиентской части:

- Проверяется его корректность и соответствие струткture. Также файл должен быть направлен авторизованным пользователем с ролью Администратор системы/сервиса. Если это условие не выполнено, то возвращается ошибка "Невалидный файл" или "Отказано в доступе"
- Если файл прошел проверку, то из каждого объекта в JSON-файле создается новый пользователь по следующим правилам:
  - Пользователю автоматиечски присваивается id
  - Пользователю обязательно устанавливаются login, password и email из файла
  - Пользователь обязательно закрепляется за тенантом с tenant_name. Пользователю выдается лицензия в рамках этого тенанта
  - Для пользователя также могут передаваться name и surname
  - Роль из файла устанавливается в качестве role: id, role: name устанавливается автоматически (роли соответствуют ролям в системе)
- Если для какого-то пользователя отстутсвует один или несколько обязательных элементов, он пропускается и не создается в системе, для ответа формируется дополнительная запись (+1 элемент массива, в котором передаются логины пользователей, которые не были созданы; если у пользователя не указан логин, то он полностью игнорируется)
- После создания всех пользователей отправляется ответ, аналогичный ответу на запрос GET back/api/v2/admin/users (вовращается список всех пользователей системы, включая новых созданных) + массив с логинами пользователей, которые не были созданы в результате запроса

Revision #4
Created 3 April 2026 10:00:24 by Артём
Updated 3 April 2026 11:56:51 by Артём