OAuth2 Server

Описание

Сервер авторизации на базе JWT и протокола OAuth2 с использованием Public/Private Keys

Генерация ключей

Формируем пару ключей, используя JDK

```
keytool -genkeypair -alias <alias>
                    -keyalg RSA
                    -keypass <keypass>
                    -keystore <keystore>.jks
                    -storepass <storepass>
```

.jks размещаем в resources/jks на сервере + прописываем настройки в application.yml

Экспортируем публичный ключ

```
keytool -export -alias <alias> -keystore <keystore>.jks -file publickey.pem
```

publickey.cert отдаем клиентам для размещения в resources/cert + прописываем настройки в их application.yml

Раздача публичного ключа

Kafka

Если отправка публичного ключа через кафку включена, тогда приложение читает ключ из хранилища и отправляет его в топик кафки. Затем происходит повторная отправка ключа с топик раз в сутки. Ключ шифруется перед отправкой Base64

Настройки

```
 jprime.security.authserver.publickey.kafka.producers.kafkaServers - _ip:port_
jprime.security.authserver.publickey.kafka.producers.kafkaTopic имя топика ex.:publicKey
jprime.security.authserver.publickey.kafka.producers.enabled если true, значит отправка публичного ключ
```

REST

Точка доступа: `GET "oauth/v1/publickey"` Возвращает ключ в виде массива байтов, дополнительные настройки отсутствуют

REST запросы

Аутентификация

`POST auth/token`

header

```
 "Authorization": Base64("<clientId>:<clientPassword>")
"Content-Type":"application/x-www-form-urlencoded"
```

body

```
 "username":"<user>"
"password":"<password>"
"grant_type":"password"
```

ответ вида

```
{
    "access_token": "<access_token>",
    "token_type": "bearer",
    "refresh_token": "<refresh_token>",
    "expires_in": 1799,
    "scope": "READ",
    "userId": 9999999912,
    "jti": "d97dd96d-4eec-431e-a31c-a7bab131ef61"
}
```

Обновление токена

`POST auth/token`

header

```
 "Authorization": Base64("<clientId>:<clientPassword>")
"Content-Type":"application/x-www-form-urlencoded"
```changepassword

body

"refresh_token":"" "grant_type":"refresh_token"

ответ вида

```json
{
    "access_token": "<access_token>",
    "token_type": "bearer",
    "refresh_token": "<refresh_token>",
    "expires_in": 1799,
    "scope": "READ",
    "userId": 9999999912,
    "jti": "d97dd96d-4eec-431e-a31c-a7bab131ef61"
}
```

Получение списка ролей по указанному токену (v1)

`GET oauth/v1/token_roles`

с указанием заголовка

```
Authorization = Bearer <token>
```

токен должен содержать роль `AUTH_ACCESS`

ответ вида

```
{
    "roles": [
        "AUTH_ACCESS"
    ]
}
```

Получение списка ролей по указанному токену (v2)

`GET oauth/v2/token_roles`

с указанием заголовка

```
Authorization = Bearer <token>
```

токен должен содержать роль `AUTH_ACCESS`

ответ вида

```
{
    "roles": [
        {
            "code": "AUTH_ACCESS",
            "name": "Общий доступ"
        },
        {
            "code": "ARTICLE_ADMIN",
            "name": "Администратор статей"
        }
    ]
}
```

Получение списка ролей

`GET oauth/v1/roles/`

с указанием заголовка

```
Authorization = Bearer <token>
```

токен должен содержать роль `AUTH_ADMIN`

ответ вида

```
{
    "roles": [
        {
            "code": "ADMIN",
            "name": "Администратор с полными правами"
        },
        {
            "code": "AUTH_ACCESS",
            "name": "Общий доступ"
        }
    ]
}
```

Получение описание роли

`GET oauth/v1/roles/<код роли>`

с указанием заголовка

```
Authorization = Bearer <token>
```

токен должен содержать роль `AUTH_ADMIN`

ответ вида

```
{
    "code": "ADMIN",
    "name": "Администратор с полными правами"
}
```

Удаление роли

`DELETE oauth/v1/roles/<код роли>`

с указанием заголовка

```
Authorization = Bearer <token>
```

токен должен содержать роль `AUTH_ADMIN`

Обновление роли

`PUT oauth/v1/roles`

с указанием заголовка

```
Authorization = Bearer <token>
```

токен должен содержать роль `AUTH_ADMIN`

запрос вида

```
{
    "code": "ADMIN",
    "name": "Администратор с полными правами [обновление]"
}
```

- ответ вида

```
{
    "code": "ADMIN",
    "name": "Администратор с полными правами [обновление]"
}
```

Создание роли

`POST oauth/v1/roles`

с указанием заголовка

```
Authorization = Bearer <token>
```

токен должен содержать роль `AUTH_ADMIN`

запрос вида

```
{
    "code": "TEST",
    "name": "Тестовая роль"
}
```

ответ вида

```
{
    "code": "TEST",
    "name": "Тестовая роль"
}
```

Получение списка групп

`GET oauth/v1/groups/`

с указанием заголовка

```
Authorization = Bearer <token>
```

токен должен содержать роль `AUTH_ADMIN`

ответ вида

```
{
    "groups": [
        {
            "code": "ORGplusPERS",
            "name": "Оператор подсистем населения и предприятий",
            "roles": [
                "PERS",
                "ORG"
            ]
        }
    ]
}
```

Получение описание группы

`GET oauth/v1/groups/<код группы>`

с указанием заголовка

```
Authorization = Bearer <token>
```

токен должен содержать роль `AUTH_ADMIN`

ответ вида

```
{
    "code": "ORGplusPERS",
    "name": "Оператор подсистем населения и предприятий",
    "roles": [
        "PERS",
        "ORG"
    ]
}
```

Удаление группы

`DELETE oauth/v1/groups/<код группы>`

с указанием заголовка

```
Authorization = Bearer <token>
```

токен должен содержать роль `AUTH_ADMIN`

Создание группы

`POST oauth/v1/groups`

с указанием заголовка

```
Authorization = Bearer <token>
```

токен должен содержать роль `AUTH_ADMIN`

запрос вида

```
{
    "code": "TEST",
    "name": "тест",
    "roles": [
        "PERS",
        "ORG"
    ]
}
```

ответ вида

```
{
    "code": "TEST",
    "name": "тест",
    "roles": [
        "PERS",
        "ORG"
    ]
}
```

Обновление группы

`PUT oauth/v1/groups`

с указанием заголовка

```
Authorization = Bearer <token>
```

токен должен содержать роль `AUTH_ADMIN`

запрос вида

```
{
    "code": "TEST",
    "name": "тест",
    "roles": [
        "PERS",
        "ORG"
    ]
}
```

ответ вида

```
{
    "code": "TEST",
    "name": "тест",
    "roles": [
        "PERS",
        "ORG"
    ]
}
```

Поиск пользователей

`POST oauth/v1/users/search`

с указанием заголовка

```
Authorization = Bearer <token>
```

токен должен содержать роль `AUTH_ADMIN`

запрос вида

```
{
   "limit": 50,
   "offset": 0,
   "filter": {
      "or": {
         "username":"TEST",
         "orgId":0,
         "email":null,
         "fio":"Сотрудник",
         "roles": ["AUTH_ADMIN","ADMIN"]
       }
    }
}
```

Параметр	Описание
username	Логин (поиск по содержит)
orgId	Идентификатор организации (поиск по равно)
email	Электронная почта (поиск по содержит)
fio	ФИО (поиск по содержит)
roles	Кодовые имена ролей
limit	Количество объектов в выборке
offset	Смещение выборки

ответ вида

```
{
    "users": [
        {
            "id": "da50068f-3c28-4377-a314-e0d4534511e94",
            "username": "TEST",
            "needPasswordChange": false,
            "orgId": 0,
            "depId": 0,
            "userId": 9999999912,
            "fio": "Сотрудник",
            "position": null,
            "workplace": null,
            "phone": null,
            "email": null,
            "changeDate": "2019-04-02",
            "creationDate": "2019-04-02",
            "disable": false,
            "disableDate": null,
            "groups": [
                "ORGplusPERS"
            ],
            "roles": [
                "AUTH_ADMIN",
                "ADMIN",
                "AUTH_ACCESS"
            ],
            "allRoles": [
                "PERS",
                "ORG",
                "AUTH_ADMIN",
                "ADMIN",
                "AUTH_ACCESS"
            ]
        }
    ]
}
```


Получение списка пользователей

`GET oauth/v1/users/`

с указанием заголовка

```
Authorization = Bearer <token>
```

токен должен содержать роль `AUTH_ADMIN`

ответ вида

```
{
    "users": [
        {
            "id": "da50068f-3c28-4377-a314-e0d4534511e94",
            "username": "TEST",
            "needPasswordChange": false,
            "orgId": 0,
            "depId": 0,
            "userId": 9999999912,
            "fio": "Сотрудник",
            "position": null,
            "workplace": null,
            "phone": null,
            "email": null,
            "changeDate": "2019-04-02",
            "creationDate": "2019-04-02",
            "disable": false,
            "disableDate": null,
            "groups": [
                "ORGplusPERS"
            ],
            "roles": [
                "AUTH_ADMIN",
                "ADMIN",
                "AUTH_ACCESS"
            ],
            "allRoles": [
                "PERS",
                "ORG",
                "AUTH_ADMIN",
                "ADMIN",
                "AUTH_ACCESS"
            ]
        }
    ]
}
```


Получение описание пользователя

`GET oauth/v1/users/<id пользователя>`

с указанием заголовка

```
Authorization = Bearer <token>
```

токен должен содержать роль `AUTH_ADMIN`

ответ вида

```
{
    "id": "6345340d6cc-d04c-4b7d-93a3-d7d7b77db4d9",
    "username": "MPUSER",
    "needPasswordChange": false,
    "orgId": 0,
    "depId": 0,
    "userId": 9999999911,
    "fio": "Разработчик (МП)",
    "position": null,
    "workplace": null,
    "phone": null,
    "email": null,
    "changeDate": "2019-04-02",
    "creationDate": "2019-04-02",
    "disable": false,
    "disableDate": null,
    "groups": [],
    "roles": [
        "AUTH_ADMIN",
        "ADMIN",
        "AUTH_ACCESS"
    ],
    "allRoles": [
        "AUTH_ADMIN",
        "ADMIN",
        "AUTH_ACCESS"
    ]
}
```


Удаление пользователя

`DELETE oauth/v1/users/<id пользователя>`

с указанием заголовка

```
Authorization = Bearer <token>
```

токен должен содержать роль `AUTH_ADMIN`

Создание пользователя

`POST oauth/v1/users`

с указанием заголовка

```
Authorization = Bearer <token>
```

токен должен содержать роль `AUTH_ADMIN`

запрос вида

```
{
    "username": "TESTUSER",
    "password": "TESTUSER",
    "needPasswordChange": false,
    "orgId": 0,
    "depId": 0,
    "userId": 0,
    "fio": "Разработчик",
    "position": null,
    "workplace": null,
    "phone": null,
    "email": null,
    "disable": false,
    "disableDate": null,
    "groups": [],
    "roles": [
        "AUTH_ACCESS"
    ]
}
```

ответ вида

```
{
    "id": "bb590f2d-3eec-4039-a584-7f814d586bcb",
    "username": "TESTUSER",
    "password": null,
    "needPasswordChange": false,
    "orgId": 0,
    "depId": 0,
    "userId": 0,
    "fio": "Разработчик",
    "position": null,
    "workplace": null,
    "phone": null,
    "email": null,
    "changeDate": "2019-04-09T14:07:56.280+0000",
    "creationDate": "2019-04-09T14:07:56.280+0000",
    "disable": false,
    "disableDate": null,
    "groups": [],
    "roles": [
        "AUTH_ACCESS"
    ],
    "allRoles": [
        "AUTH_ACCESS"
    ]
}
```

Обновление пользователя

`PUT oauth/v1/users`

с указанием заголовка

```
Authorization = Bearer <token>
```

токен должен содержать роль `AUTH_ADMIN`

запрос вида

```
{
    "id": "bb590f2d-3eec-4039-a584-7f814d586bcb",
    "username": "TESTUSER",
    "password": "TESTUSER2",
    "needPasswordChange": false,
    "orgId": 0,
    "depId": 0,
    "userId": 0,
    "fio": "Разработчик",
    "position": null,
    "workplace": null,
    "phone": null,
    "email": null,
    "changeDate": "2019-04-09T14:07:56.280+0000",
    "creationDate": "2019-04-09T14:07:56.280+0000",
    "disable": false,
    "disableDate": null,
    "groups": [],
    "roles": [
        "AUTH_ACCESS"
    ],
    "allRoles": [
        "AUTH_ACCESS"
    ]
}
```

ответ вида

```
{
    "id": "bb590f2d-3eec-4039-a584-7f814d586bcb",
    "username": "TESTUSER",
    "password": null,
    "needPasswordChange": false,
    "orgId": 0,
    "depId": 0,
    "userId": 0,
    "fio": "Разработчик",
    "position": null,
    "workplace": null,
    "phone": null,
    "email": null,
    "changeDate": "2019-04-09T14:09:09.566+0000",
    "creationDate": "2019-04-09T14:07:56.280+0000",
    "disable": false,
    "disableDate": null,
    "groups": [],
    "roles": [
        "AUTH_ACCESS"
    ],
    "allRoles": [
        "AUTH_ACCESS"
    ]
}
```


Обновление ролей пользователя

`PUT oauth/v1/userroles`

с указанием заголовка

```
Authorization = Bearer <token>
```

токен должен иметь одну (или несколько) из ролей:

- `AUTH_ADMIN` - полный доступ;
- `AUTH_USER_ADMIN` - доступ по своим ролям;
- `AUTH_ORG_ADMIN` - доступ по своим ролям и организации.

запрос вида


```
{
    "userId": 0,
    "roles": [
        "AUTH_ACCESS"
    ]
}
```


Параметр	Описание

Параметр	Описание
userId	Идентификатор пользователя
roles	Роли

ответ вида

Код	Описание
400	Неверный запрос
401	Обновление ролей невозможно
404	Адрес не найден
500	Ошибка сервера
200	Обновление ролей успешно

Обновление пароля пользователя

`PUT oauth/v1/changepassword`

с указанием заголовка

```
Authorization = Bearer <token>
```

токен должен иметь одну (или несколько) из ролей:

- `AUTH_ADMIN` - полный доступ;
- `AUTH_USER_ADMIN` - доступ по своим ролям;
- `AUTH_ORG_ADMIN` - доступ по своим ролям и организации.

запрос вида

```
{
    "userId": 0,
    "password": 123
}
```

Параметр	Описание
userId	Идентификатор пользователя
password	Пароль

ответ вида

Код	Описание
400	Неверный запрос

Код	Описание
401	Обновление ролей невозможно
404	Адрес не найден
500	Ошибка сервера
200	Обновление ролей успешно

Блокировка пользователя

`PUT oauth/v1/blockuser/{userId}`

с указанием заголовка

```
Authorization = Bearer <token>
```

токен должен иметь одну (или несколько) из ролей:

- `AUTH_ADMIN` - полный доступ;
- `AUTH_USER_ADMIN` - доступ по своим ролям;
- `AUTH_ORG_ADMIN` - доступ по своим ролям и организации.

запрос вида

```
{
}
```

ответ вида

Код	Описание
400	Неверный запрос
401	Обновление ролей невозможно
404	Адрес не найден
500	Ошибка сервера
200	Обновление ролей успешно

Разблокировка пользователя

`PUT oauth/v1/unblockuser/{userId}`

с указанием заголовка

```
Authorization = Bearer <token>
```

токен должен иметь одну (или несколько) из ролей:

- `AUTH_ADMIN` - полный доступ;
- `AUTH_USER_ADMIN` - доступ по своим ролям;
- `AUTH_ORG_ADMIN` - доступ по своим ролям и организации.

запрос вида

```
{
}
```

ответ вида

Код	Описание
400	Неверный запрос
401	Обновление ролей невозможно
404	Адрес не найден
500	Ошибка сервера
200	Обновление ролей успешно

Блокировка пользователей

`PUT oauth/v1/blockusers`

с указанием заголовка

```
Authorization = Bearer <token>
```

токен должен иметь одну (или несколько) из ролей:

- `AUTH_ADMIN` - полный доступ;
- `AUTH_USER_ADMIN` - доступ по своим ролям;
- `AUTH_ORG_ADMIN` - доступ по своим ролям и организации.

запрос вида

```
{
  "ids": [111, 112]
}
```

Параметр	Описание
ids	Идентификаторы пользователей

ответ вида

Код	Описание

Код	Описание
400	Неверный запрос
401	Обновление ролей невозможно
404	Адрес не найден
500	Ошибка сервера
200	Обновление ролей успешно

Разблокировка пользователей

`PUT oauth/v1/unblockusers`

с указанием заголовка

```
Authorization = Bearer <token>
```

токен должен иметь одну (или несколько) из ролей:

- `AUTH_ADMIN` - полный доступ;
- `AUTH_USER_ADMIN` - доступ по своим ролям;
- `AUTH_ORG_ADMIN` - доступ по своим ролям и организации.

запрос вида

```
{
  "ids": [111, 112]
}
```

ответ вида

Код	Описание
400	Неверный запрос
401	Обновление ролей невозможно
404	Адрес не найден
500	Ошибка сервера
200	Обновление ролей успешно

Обновление организации у пользователя

`PUT oauth/v1/orgid`

с указанием заголовка

```
Authorization = Bearer <token>
```

токен должен иметь одну (или несколько) из ролей:

- `AUTH_ADMIN` - полный доступ;
- `AUTH_USER_ADMIN` - доступ по своим ролям;
- `AUTH_ORG_ADMIN` - доступ по своим ролям и организации.

запрос вида

```
{
  "userId": 111,
  "orgId": 111
}
```

Атрибут	Описание
userId	Идентификатор пользователя
orgId	Идентификатор организации

ответ вида

Код	Описание
400	Неверный запрос
401	Обновление ролей невозможно
404	Адрес не найден
500	Ошибка сервера
200	Обновление ролей успешно

Данные текущего профиля

`GET oauth/v1/profiles`

с указанием заголовка

```
Authorization = Bearer <token>
```

токен должен содержать роль `AUTH_ACCESS`

ответ вида

```
{
    "orgId": "12",
    "username": "TESTUSER",
    "needPasswordChange": false,
    "fio": "Разработчик",
    "position": null,
    "workplace": null,
    "phone": null,
    "email": null
}
```

Обновление текущего профиля

`PUT oauth/v1/profiles`

с указанием заголовка

```
Authorization = Bearer <token>
```

токен должен содержать роль `AUTH_ACCESS`

запрос вида

```
{
    "username": "TESTUSER",

    "fio": "Разработчик",
    "position": null,
    "workplace": null,
    "phone": null,
    "email": null
}
```

ответ вида

```
{
    "username": "TESTUSER",

    "fio": "Разработчик",
    "position": null,
    "workplace": null,
    "phone": null,
    "email": null
}
```

Смена текущего пароля

`PUT oauth/v1/password`

с указанием заголовка

```
Authorization = Bearer <token>
```

токен должен содержать роль `AUTH_ACCESS`

запрос вида

```
{
    "curPassword":"<p1>",
    "newPassword":"<p2>"
}
```

- ответ

Код	Описание
400	Неверный запрос
401	Смена пароля невозможна
404	Адрес не найден
500	Ошибка сервера
200	Смена пароля успешна

Настройки

Код	Описание
jprime.authserver.jks.name	Имя контейнера ключей в jks папке
jprime.authserver.jks.storepass	Пароль к криптоконтейнеру
jprime.authserver.jks.alias	Алиас для пары ключей