# Keycloak.ЕСИА. Руководство по установке и настройке

## Оглавление



Программное обеспечение “Keycloak.ЕСИА” (далее — ПО и Keycloak.ЕСИА) является самостоятельным модулем прохождения процедуры аутентификации через Единую систему идентификации и аутентификации (далее — ЕСИА) для сервера Keycloak, который подключается посредством встроенного механизма расширения Keycloak для сторонних провайдеров аутентификации. Keycloak.ЕСИА позволяет надлежащим образом выполнить процедуру проверки подлинности пользователей сервера через Единую систему идентификации и аутентификации.

Установка модуля производится путём копирования дополнительных файлов в директории установленного сервера.
Настройка модуля выполняется в основном через административный интерфейс сервера Keycloak.

## Предварительные требования

Перед установкой и настройкой модуля через административный интерфейс необходимо убедиться, что выполнены следующие шаги:

- вашей организации и информационной системе разрешён доступ в ЕСИА, вы подключены как минимум к тестовому контуру ЕСИА
- вам известна мнемоника вашей системы в ЕСИА, имеется доступ в административный интерфейс ЕСИА

- для сервера выделено имя, разрешаемое через публичные сервера DNS в валидный («белый) IP адрес, на это имя получен TLS сертификат (например, с помощью сервиса Letsencrypt)
- для промышленной эксплуатации – необходимо дополнительно приобрести серверную лицензию на CryptoPRO JCP 2.0 версии 2.0.41940-A, соответствующую характеристикам вашего сервера. Для тестовой среды дистрибутив можно скачать с сайта CrryptoPRO после бесплатной регистрации (https://cryptopro.ru/products/csp/jcp/downloads).

## Установка сервера с модулем Keycloak.ЕСИА

Установка сервера возможна на любую операционную систему, для которой существует реализация Java Development Kit (JDK) версии 17. Для примера рассматривается установка в Linux с использованием дистрибутива ALT Server версии 10 (реестровая запись № 1541 в РОПО), установка в других дистрибутивах будет отличаться только в деталях.

### Подготовка к установке

#### *Создание пользователя*

Создайте пользователя, от которого будет работать сервер. В данном примере – это пользователь keycloak

#### *Установка JDK*

JDK версии 17. Рекомендуется использование Axiom JDK (реестровая запись № 5493 в РОПО) - https://axiomjdk.ru/pages/downloads/#/java-17-lts , также можно использовать любую другую сборку (Eclipse, Amazon, Bellsoft и т.п.)

#### *Установка и конфигурирование СУБД*

Реляционная база данных. Рекомендуется использование БД PostgresPro Standard 16 (реестровая запись № 104 в РОПО) - https://postgrespro.ru/products/download/postgrespro/latest , также можно использовать стандартную сборку Postgres 16 (https://www.postgresql.org/)
После установки СУБД создайте пользователя keycloak и базу данных keycloak

После установки JDK и БД в соответствии с инструкциями производителей, необходимо установить Keycloak.

#### *Установка и конфигурирование Keycloak*

Скачайте дистрибутив Keycloak необходимой версии (в примере - 26.0.4) (https://github.com/keycloak/keycloak/releases/download/26.0.4/keycloak-26.0.4.tar.gz)

Установите Keycloak в директорию /opt/keycloak-26.0.4 и сконфигурируйте в соответствии с инструкцией производителя (https://www.keycloak.org/guides#server).

В приложении 1 приведены примеры конфигурационных файлов для запуска сервера Keycloak с помощью стандартного для большинства дистрибутивов Linux механизма systemd

### Установка провайдера и сопутствующих файлов

Скопируйте в директорию providers установленного ранее сервера Keycloak следующие файлы:

1. Файл провайдера keycloak-esia-jcp.jar из дистрибутивного комплекта
2. Файлы поддержки JsonPath из дистрибутивного комплекта

- json-path-2.7.0.jar
- json-smart-2.4.7.jar

- accessors-smart-2.4.7.jar

3. Файлы из ранее полученного дистрибутива JCP 2.0 версии 2.0.41940-A:

- AdES-core.jar
- JCPRevCheck.jar
- JCPRequest.jar
- JCryptoP.jar
- CAdES.jar
- asn1rt.jar
- JCPRevTools.jar
- JCP.jar
- ASN1P.jar

В случае использования для управления сервисом механизма systemd необходимо также добавить в директорию providers файлы

- quarkus-systemd-notify-deployment-1.0.2.jar
- quarkus-systemd-notify-1.0.2.jar

### Создание директорий для работы CryptoPRO

Создайте на сервере следующие директории (подразумевается что сервис запускается от имени пользователя keycloak):
/var/opt/cprocsp/keys/keycloak - владелец keycloak, права 0700
/var/opt/cprocsp/tmp – владелец root, права 0777

### Запуск и проверка сервера

После выполнения всех шагов запустите настройку сервера
${KEYCLOAK_HOME}/bin/kc.sh build

И запустите его. Проверьте по логам что сервис стартовал успешно.

### Конфигурация защищённого соединения с сервером (TLS)

Хотя TLS можно сконфигурировать непосредственно в Keycloak, рекомендуется установка проксирующего веб-сервера и конфигурация TLS на нём. Помимо упрощения процедуры конфигурации TLS это даст возможность более гибко управлять видимостью отдельных компонентов сервера Keycloak для внешнего мира. В промышленной эксплуатации не рекомендуется открывать для внешнего доступа административный интерфейс Keycloak, оставив доступ из внешних сетей только для стандартных OIDC URL.
В приложении 2 приведён пример конфигурационного файла nginx для тестовой среды.

# Настройка модуля

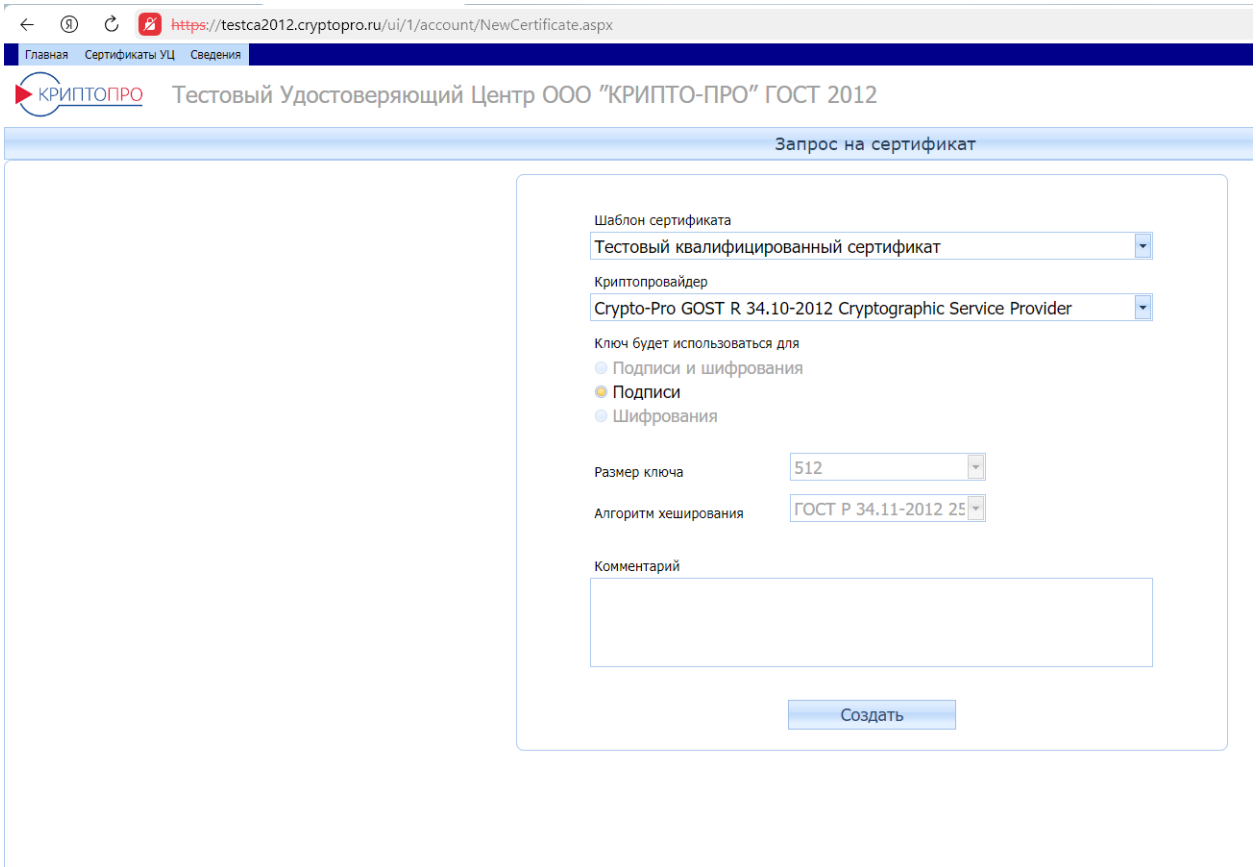
## Получение ключей в формате CryptoPro (ГОСТ)

В соответствии с протоколом обмена с ЕСИА запрос должен быть подписан электронной подписью в формате ГОСТ, для этого понадобится контейнер, содержащий закрытый ключ в формате CryptoPro и сертификат.  Для промышленной среды ЕСИА эту пару вы должны получить в аккредитованном УЦ.

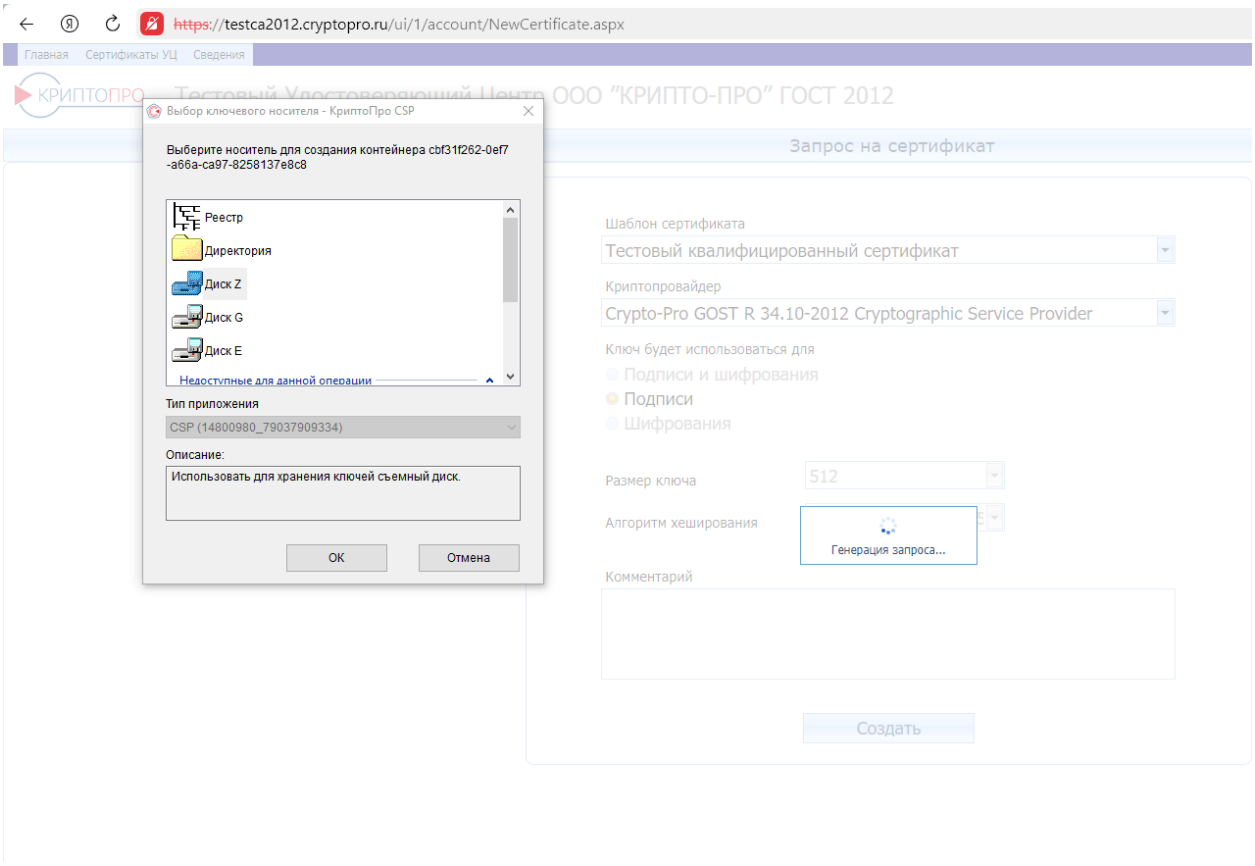
### *Генерация ключей для тестовой среды*

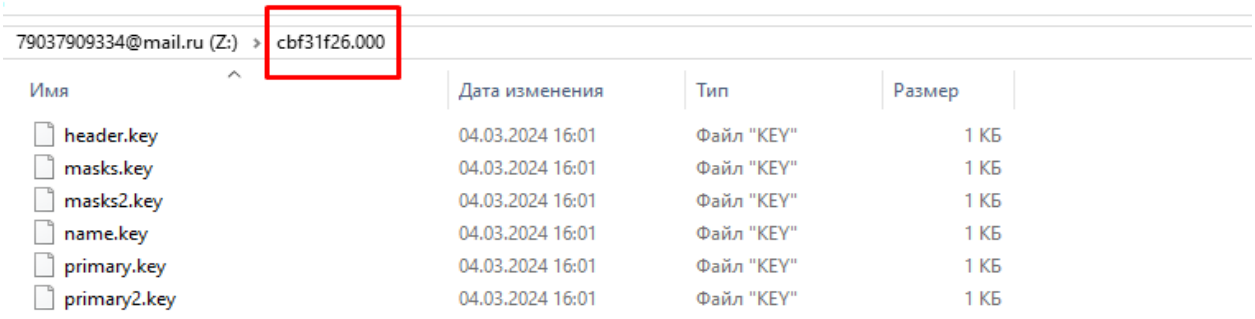
Для тестовой среды эту пару можно сгенерировать в тестовом УЦ компании «КриптоПро»: https://testca2012.cryptopro.ru/ui/Default.aspx (для доступа требуется браузер, поддерживающий ГОСТ TLS, например Яндекс.Браузер или Chromium-ГОСТ). На машине, с которой вы отправляете запрос, должно быть установлено ПО CryptoPro CSP версии 5 (https://www.cryptopro.ru/products/csp).

После бесплатной регистрации зайдите в личный кабинет и запросите выпуск сертификата, следуйте инструкциям УЦ. **При выборе типа запрашиваемого сертификата – выбирайте КЭП.**

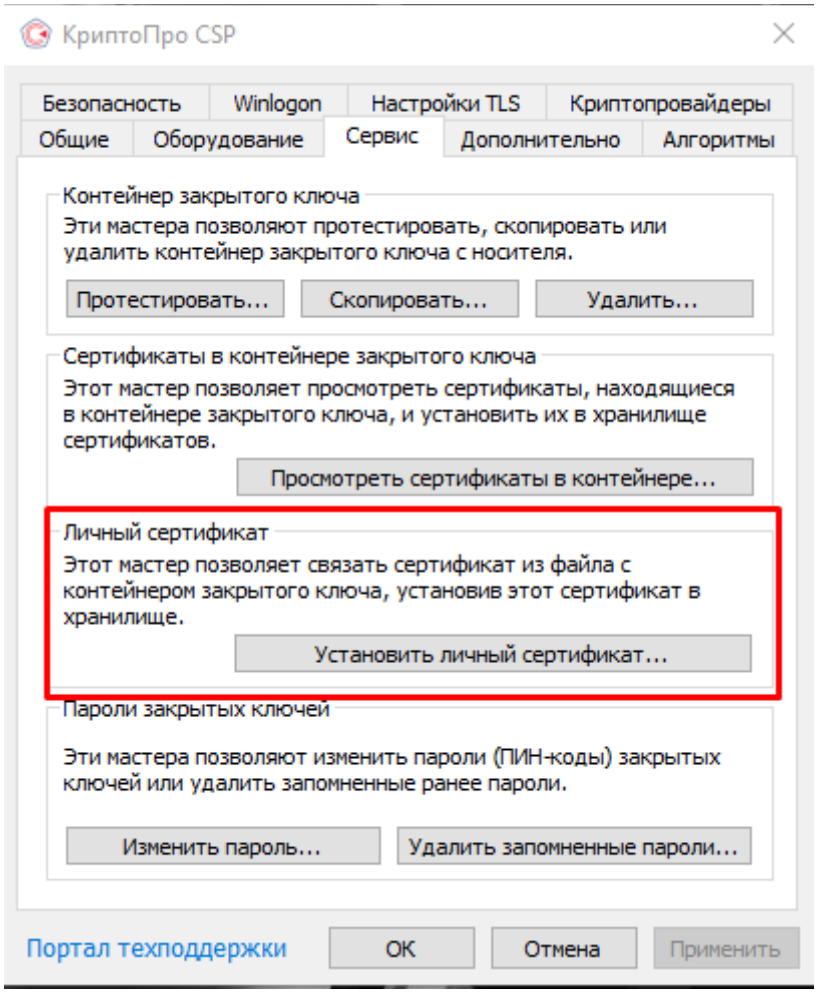


При сохранении закрытого ключа – сохраните его на диск. В результате у вас на диске должна появиться директория, содержащая закрытый ключ в формате CryptoPro (набор файлов с расширением key).

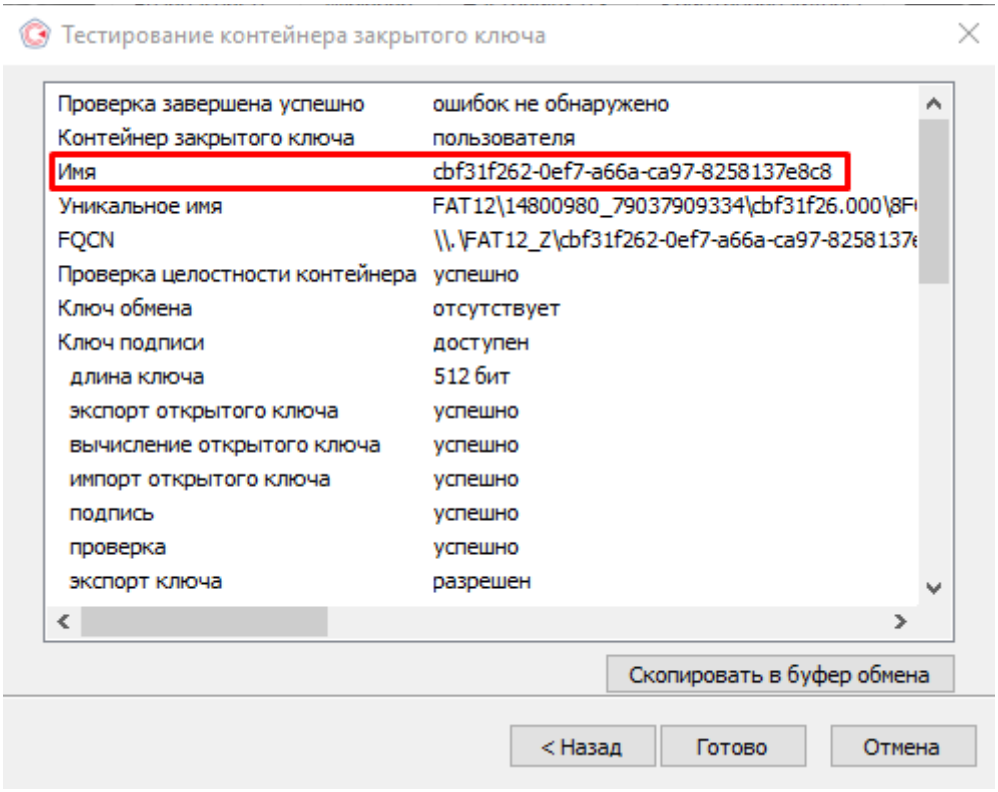


Скачайте сформированный по вашему запросу сертификат и с помощью CryptoPro CSP запишите его в этот контейнер.



Также с помощью CryptoPro CSP найдите и скопируйте идентификатор закрытого ключа – это понадобится при настройке модуля через административный интерфейс Keycloak.




## Проверка настроек в личном кабинете ЕСИА

В соответствии с руководством по работе с ЕСИА войдите в личный кабинет ЕСИА, найдите в списке свою организацию и в ней информационную систему, для которой вы настраиваете доступ, введите в список разрешённых URL адрес сервера, на котором вы установили ваш экземпляр Keycloak.

После этого добавьте полученный от УЦ сертификат в список сертификатов информационной системы.

## Конфигурация модуля

Копирование контейнера закрытого ключа (с сертификатом) на сервер
Скопируйте файлы контейнера закрытого ключа на сервер в директорию /var/cprocsp/keys/keycloak

Пример файлов в директории:

```
[root@esia-test ~]# ls -lR /var/opt/cprocsp/
/var/opt/cprocsp/:
итого 8
drwxr-xr-x 3 root root 4096 мар  4 16:17 keys
drwxrwxrwx 2 root root 4096 мар  4 16:35 tmp

/var/opt/cprocsp/keys:
итого 4
drwx------ 3 keycloak root 4096 мар  4 16:18 keycloak

/var/opt/cprocsp/keys/keycloak:
итого 4
drwx------ 2 keycloak esia 4096 мар  4 16:16 cbf31f26.000

/var/opt/cprocsp/keys/keycloak/cbf31f26.000:
итого 24
-rw-r--r-- 1 keycloak esia 2182 мар  4 16:16 header.key
-rw-r--r-- 1 keycloak esia   56 мар  4 16:16 masks2.key
-rw-r--r-- 1 keycloak esia   56 мар  5 11:54 masks.key
-rw-r--r-- 1 keycloak esia   40 мар  4 16:16 name.key
-rw-r--r-- 1 keycloak esia   36 мар  4 16:16 primary2.key
-rw-r--r-- 1 keycloak esia   36 мар  5 11:54 primary.key

/var/opt/cprocsp/tmp:
итого 0
-rwxrwxrwx 1 keycloak keycloak 0 мар  4 16:35 CertifiedRandom_class_RandomSeed
-rwxrwxrwx 1 keycloak keycloak 0 мар  4 16:35 mutexfordefrep
```

### *Создание реалма*

Создайте реалм, в котором будет сконфигурирован провайдер аутентификации через ЕСИА. Крайне не рекомендуется использовать для реальных задач создаваемый по умолчанию реалм master.  В примере созданный реалм называется esia, название выбрано просто для примера.

### *Конфигурация провайдера аутентификации через ЕСИА.*

Создайте провайдер аутентификации и сконфигурируйте следующие параметры:

1. Идентификатор клиента – мнемоника вашей ИС в ЕСИА.

2. Секрет клиента - мнемоника вашей ИС в ЕСИА (данное поле не используется в обмене).

3. Хост ЕСИА – для тестового контура это esia-portal1.test.gosuslugi.ru

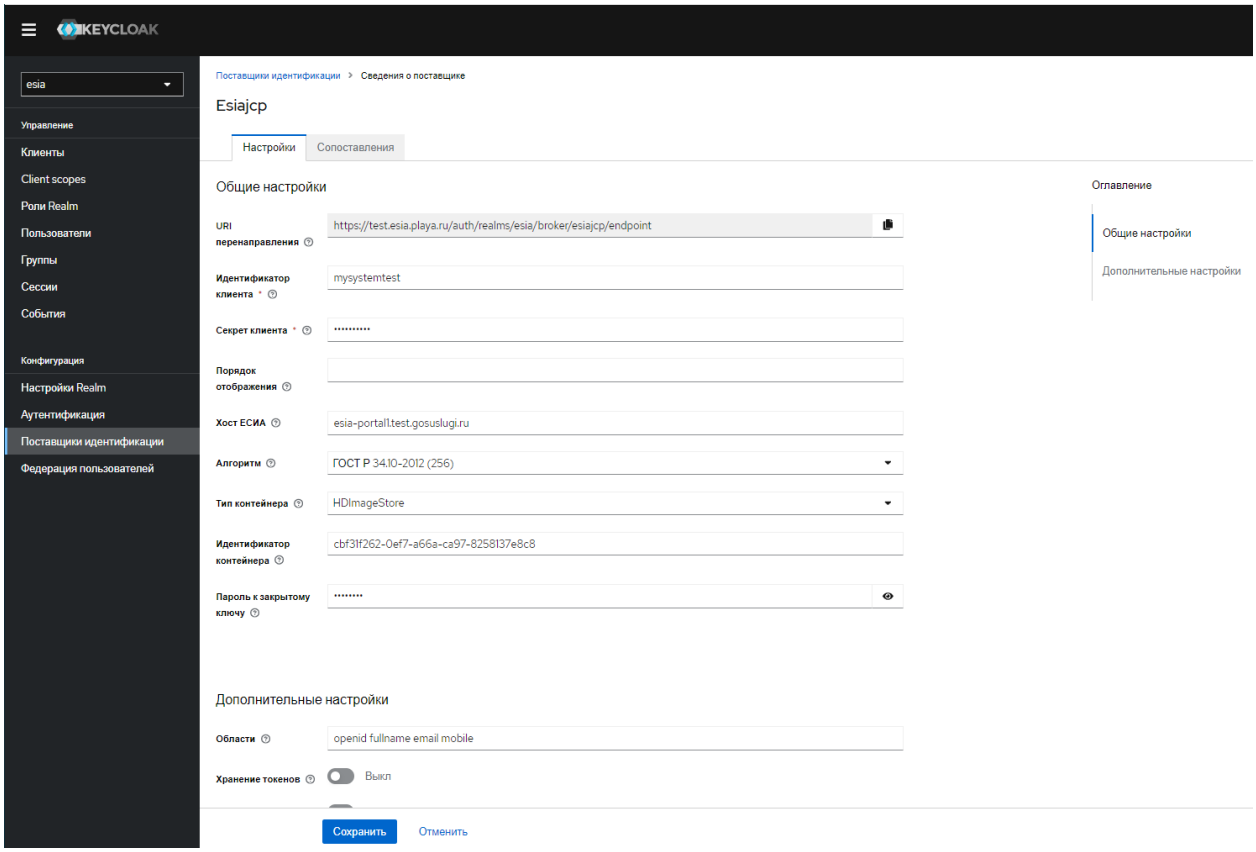
4. Алгоритм - выбираем алгоритм шифрования, по которому сгенерирован открытый и закрытый ключ.

5. Тип контейнера - выбираем тип контейнера, в котором хранится закрытый ключ. Если ключ хранится в файловой системе сервера – выберите HDImageStore

6. Идентификатор контейнера.

7. Пароль к закрытому ключу - если пароля нет, то оставляем поле пустым.

После сохранения зайдите в редактирование и в секции «Дополнительные параметры» заполните параметр «Области», указав там те области (scope), которые вы указывали в заявке на подключение к ЕСИА. Сохраните настройки




## Информация о пользователе, получаемая из ЕСИА

Информация, получаемая из ЕСИА, преобразуется в общий объект и может быть использована для создания атрибутов пользователя, прошедшего аутентификацию в провайдере.

### *Состав информации*

Состав информации зависит от разрешённых областей доступа (scope) и транслируется в соответствующие коллекции объекта:

1. Контактная информация (scope `contacts`, `email`, `mobile`) хранится в коллекции `ctts`
2. Транспортные средства (scope `vehicles`) хранится в коллекции `vhls`
3. Адреса (scope `addresses`) хранятся в коллекции `addrs`
4. Документы удостоверяющие личность (scope `id_doc`) в профиле хранятся в коллекции `docs`
5. Организации, сотрудником которых является данный пользователь (scope `orgs`) хранятся в коллекции `orgs`
6. Дети (scope `kid_usr_inf`, `kid_fullname`, `kid_birthdate`, `kid_gender`, `kid_snils`, `kid_inn`,`kid_birth_cert_doc`, `kid_medical_doc`) хранятся в коллекции `kids`

Структура информации, хранящейся в коллекциях, соответствует структуре информации, получаемой из ЕСИА (см. Методические рекомендации по использованию ЕСИА, Приложение Б.)

Пример профиля

Пример профиля пользователя с разрешёнными scope `contacts`, `email`, `mobile`, `vehicles`, `addresses`, `id_doc` приведён в приложении 3

### *Определение имени пользователя*

Имя пользователя, под которым он будет фигурировать в Keycloak, по умолчанию формируется следующим образом:

1. Если доступен мобильный телефон, то в качестве имени пользователя используется он (только цифры, остальные символы фильтруются)

2. Если доступен адрес электронной почты, то используется он

3. Если доступен СНИЛС, то используется он

Переопределить имя пользователя, под которым он будет фигурировать в Keycloak, можно через стандартный для Keycloak механизм Username Template Importer (ESIA-JCP)

### *Заполнение атрибутов пользователя*

По умолчанию данными из ЕСИА заполняются следующие атрибуты пользователя:

* email

* firstName

* lastName

Для создания дополнительных атрибутов пользователя из объекта используется Attribute Importer и технология JsonPath

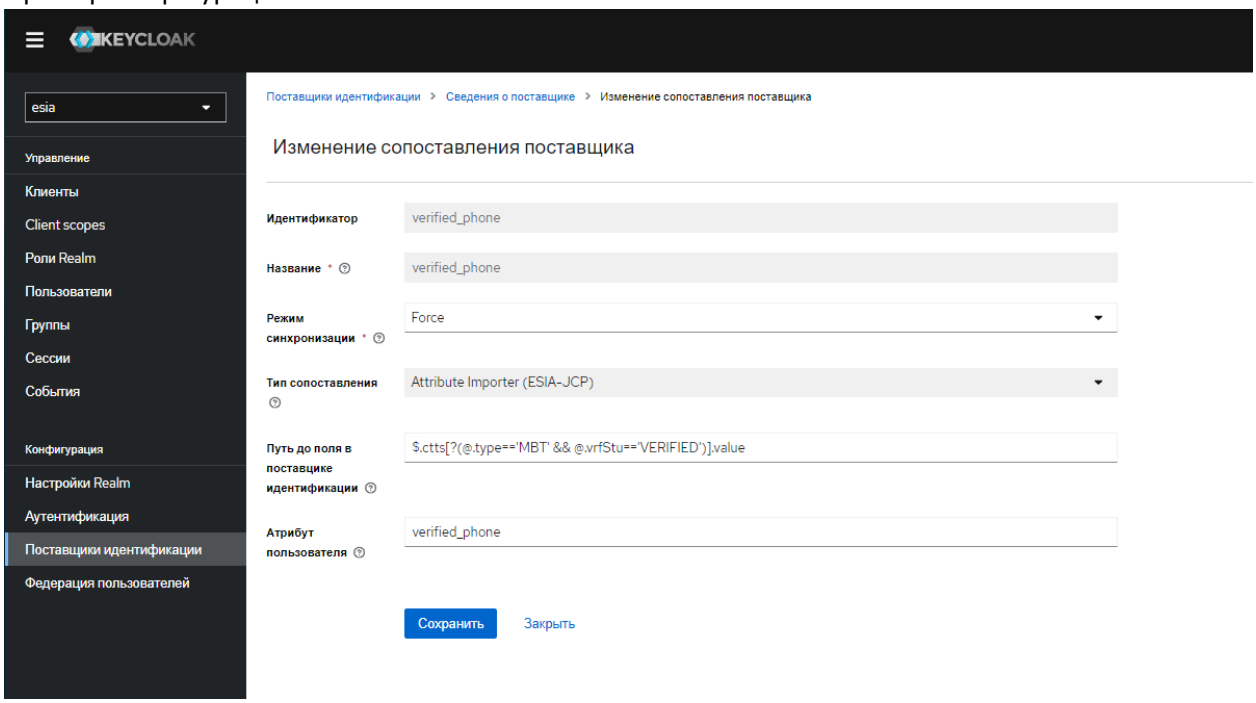
Примеры определения дополнительных атрибутов.

1. Получение СНИЛС пользователя: **$.snils (требуется разрешение на scope = snils)**

2. Получение верифицированного мобильного телефона пользователя: **$.ctts[?(@.type=='MBT' && @.vrfStu=='VERIFIED')].value (требуется разрешение на scope = phone)**

3. Получение паспортных данных: **$.concat($.docs[?(@.type=='RF_PASSPORT')].series, " ", $.docs[?(@.type=='RF_PASSPORT')].number, " Выдан ", $.docs[?(@.type=='RF_PASSPORT')].issuedBy) (требуется разрешение на scope = id_doc)**

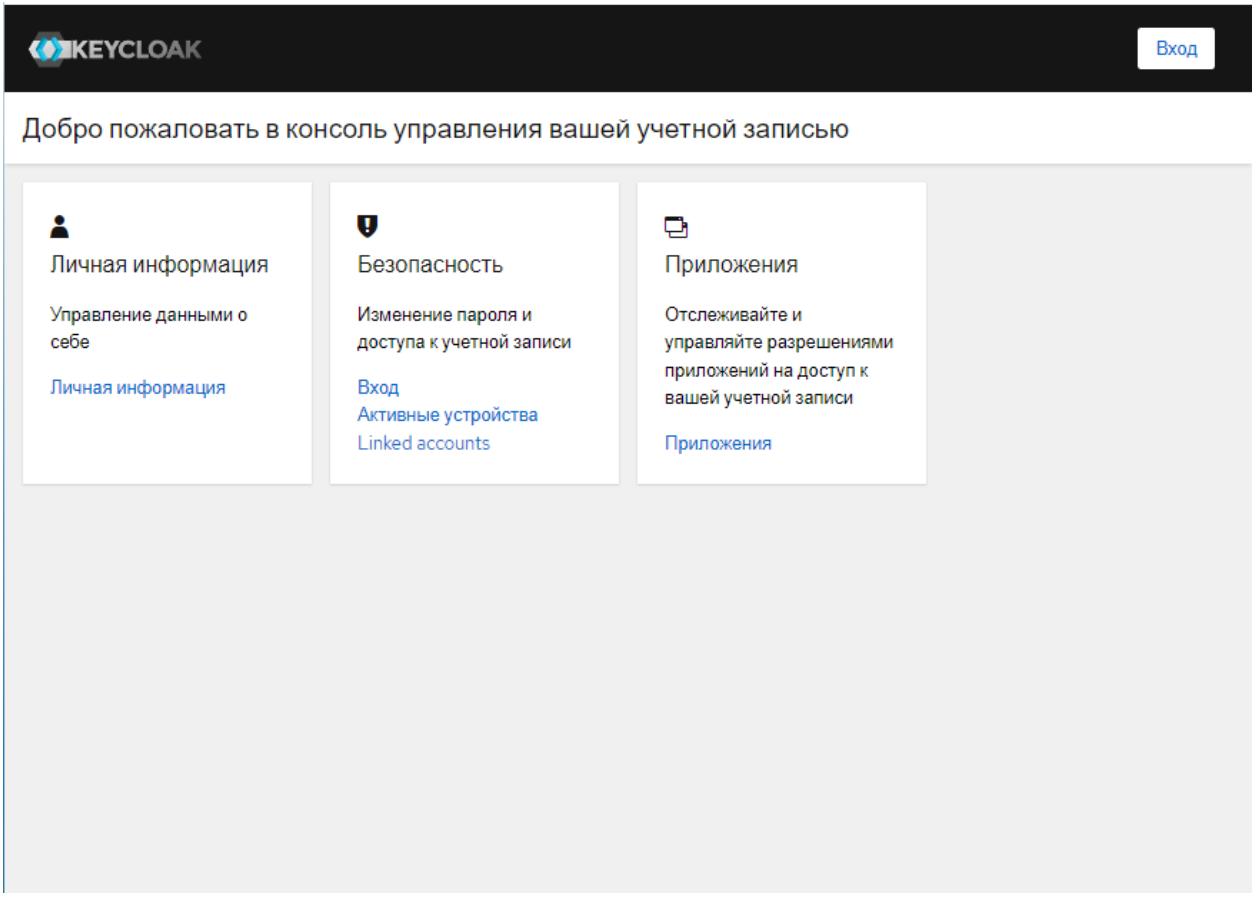
Пример конфигурации:



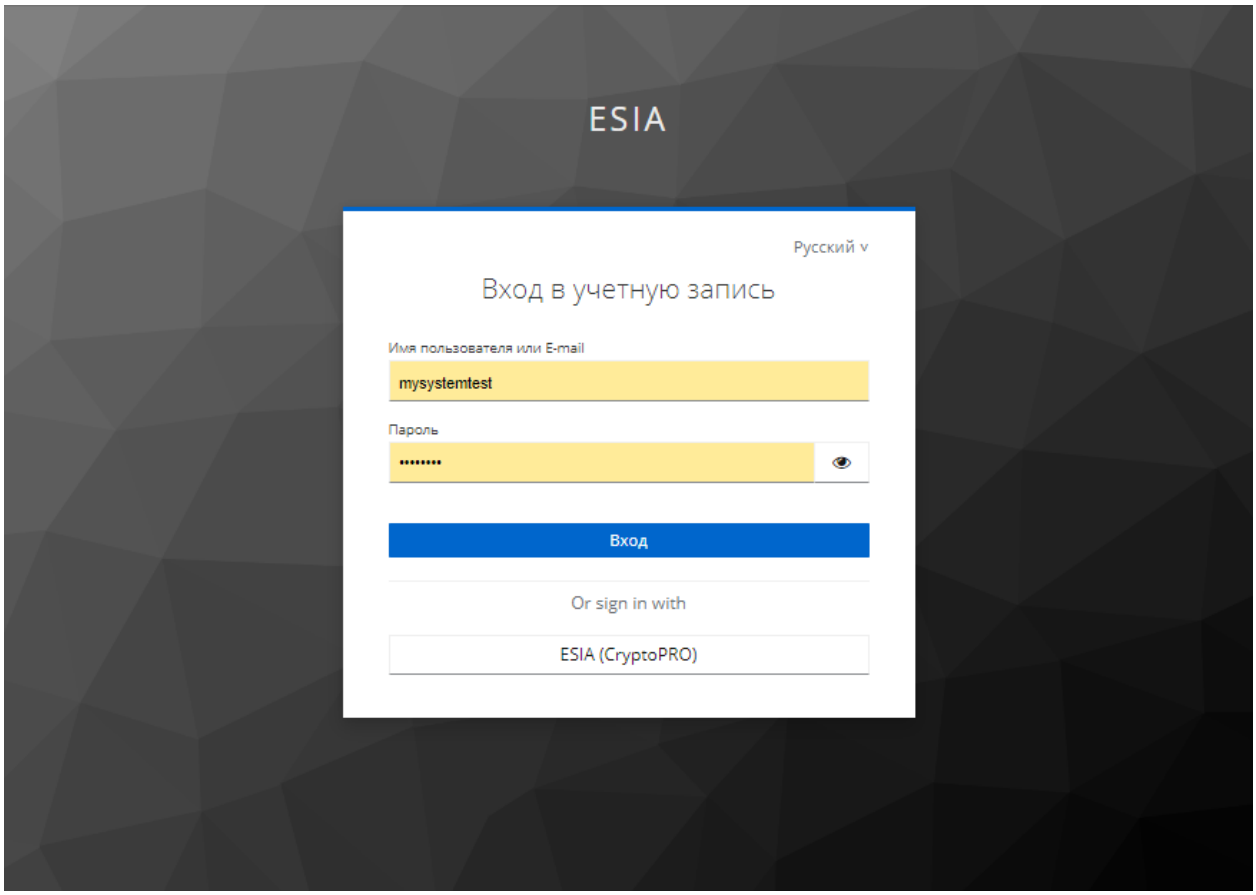
## Проверка работоспособности провайдера

Для проверки работоспособности провайдера можно воспользоваться встроенным в Keycloak приложением управления аккаунтом пользователя, которое доступно по адресу https://${keycloak_host}/${keycloak_path/}realms/${realm_name}/account/#/

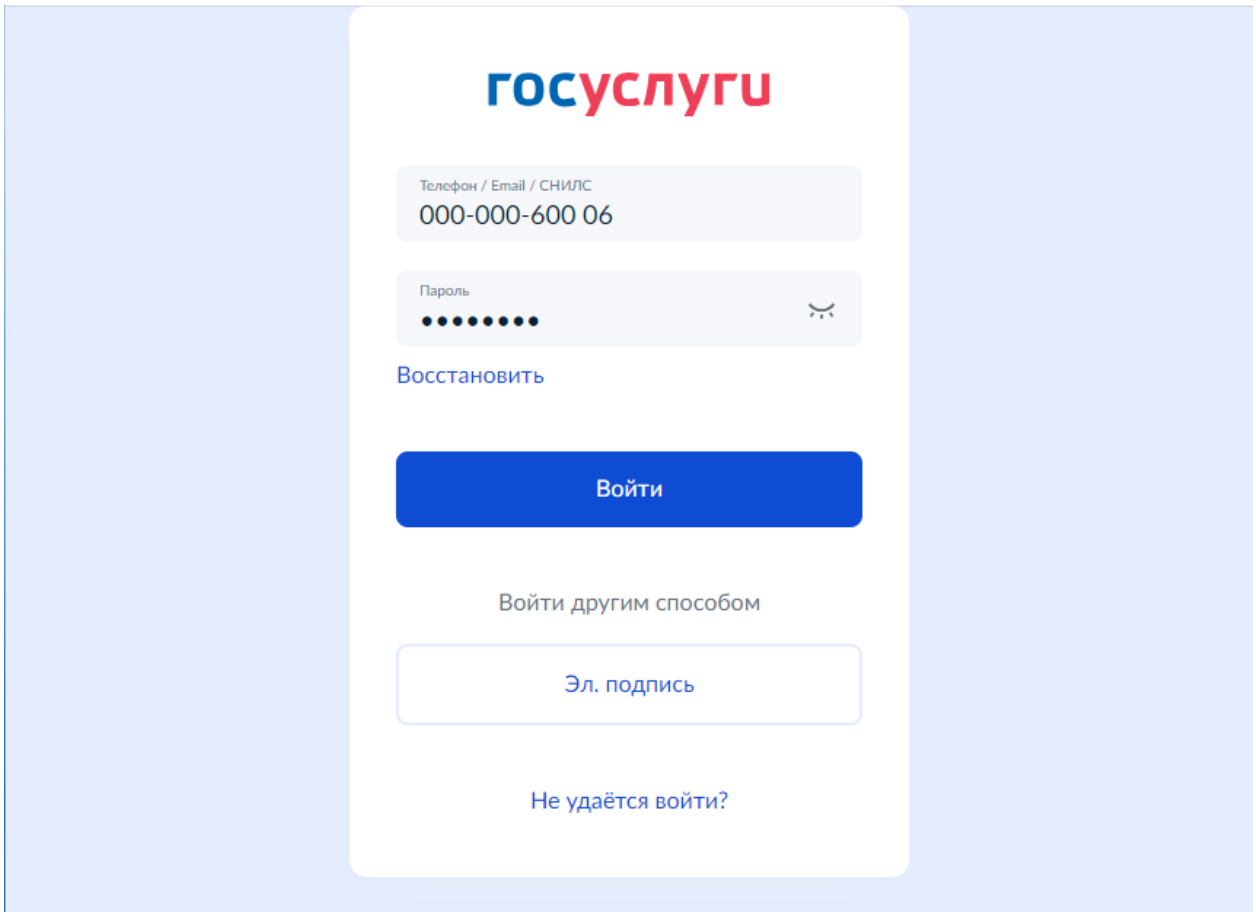
Для проверки – зайдите в это приложение с помощью ЕСИА.



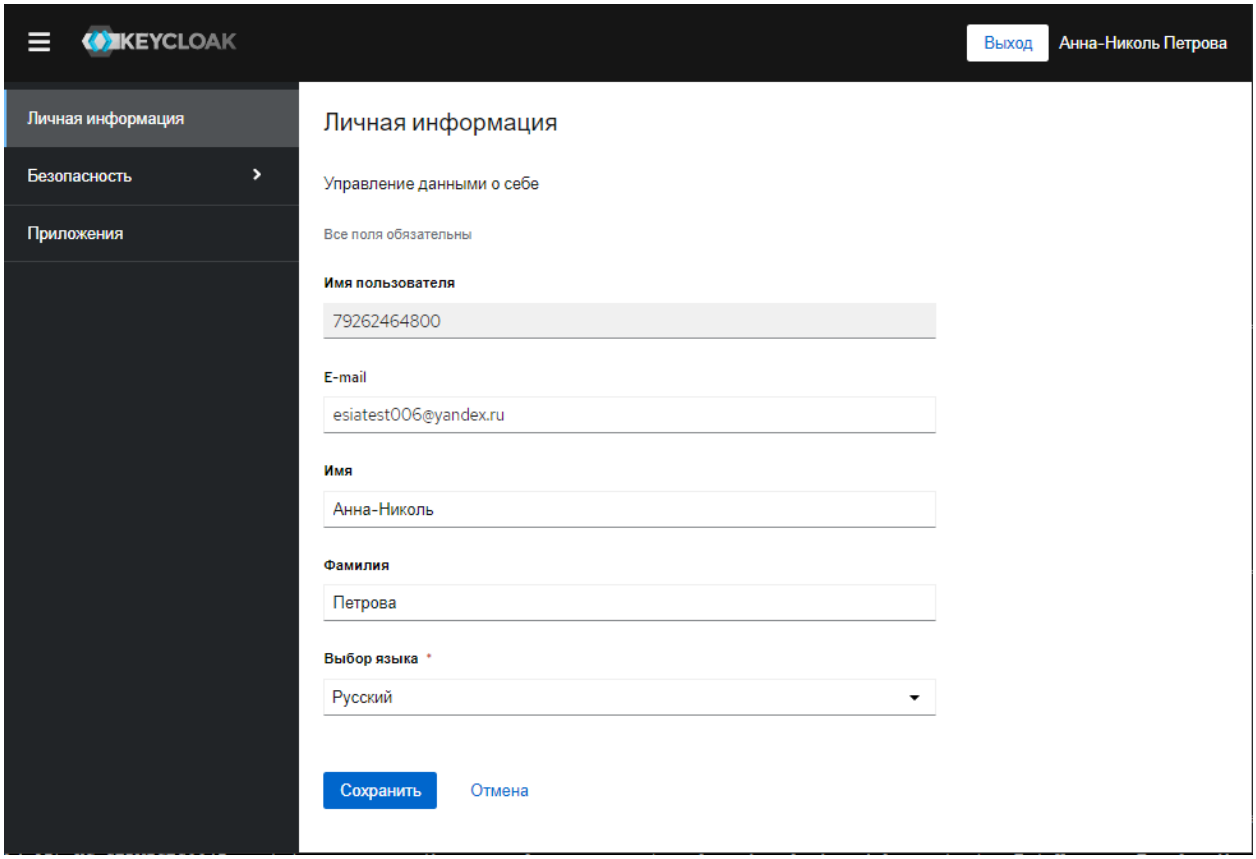
При входе нажимайте на кнопку логина через провайдера «ESIA (CryptoPRO)»



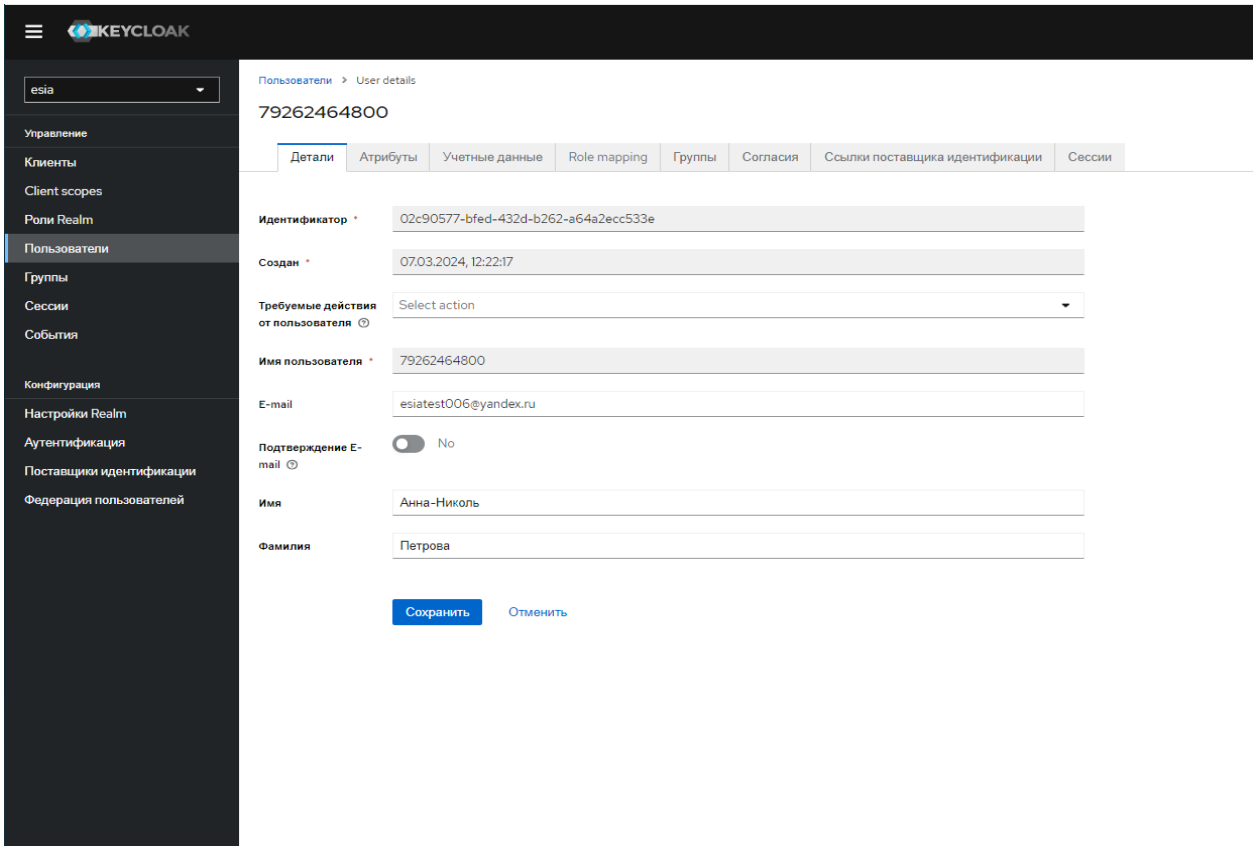

На странице логина ЕСИА введите данные пользователя, зарегистрированного на этом контуре

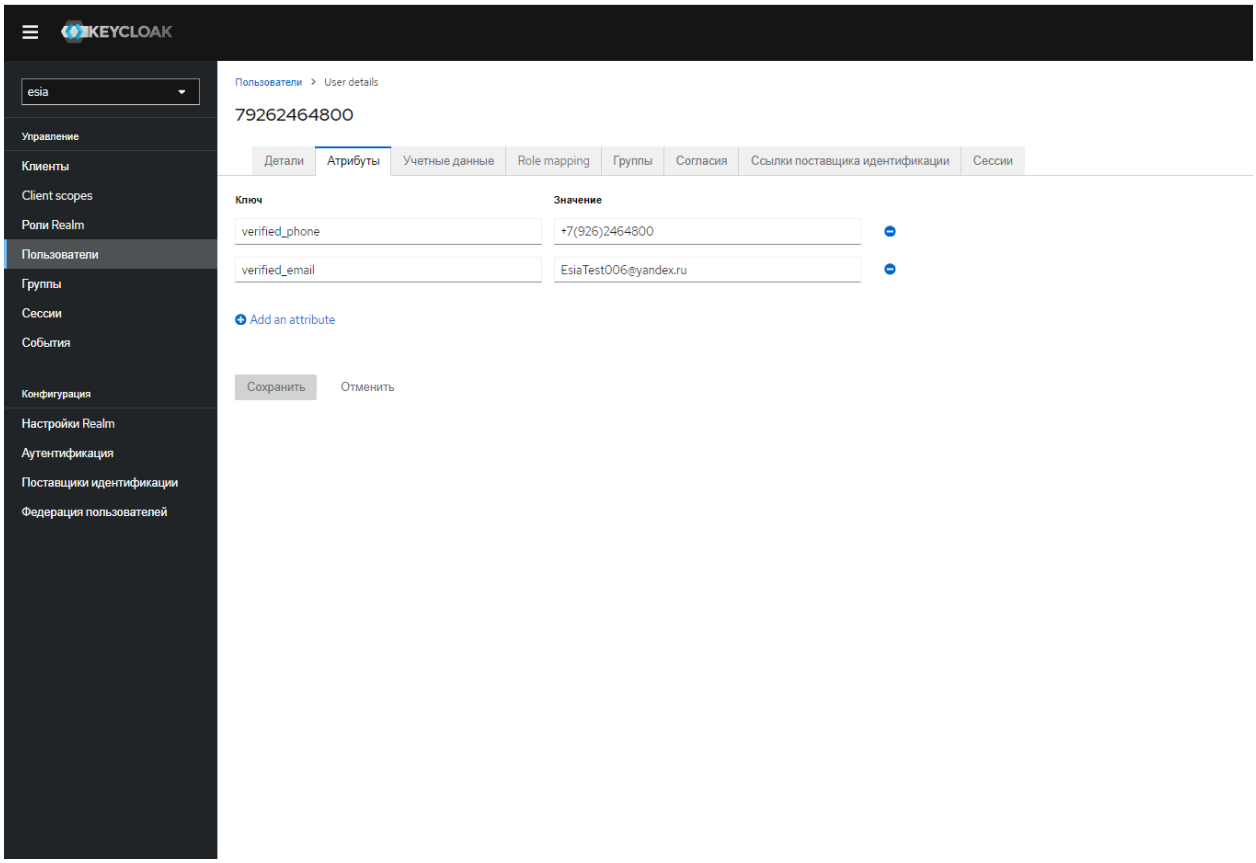


После логина в ЕСИА вы можете увидеть данные пользователя, соответствующую информации из ЕСИА




В административном интерфейсе Keycloak появится пользователь, также у него будут заполнены атрибуты, на которые настроены сопоставления (в данном примере – это verified_phone и verified_email)




KEYCLOAK
esia
Управление
Клиенты
Client scopes
Роли Realm
Пользователи
Группы
Сессии
События
Конфигурация
Настройки Realm
Аутентификация
Поставщики идентификации
Федерация пользователей
Пользователи > User details
79262464800
Детали
Атрибуты
Учетные данные
Role mapping
Группы
Согласия
Ссылки поставщика идентификации
Сессии
Ключ
Значение
verified_phone
+7(926)2464800
verified_email
EsiaTest006@yandex.ru
Add an attribute
Сохранить
Отменить

## Приложение 1. Конфигурационные файлы для запуска сервера Keycloak с помощью механизма system.

**/etc/systemd/system/keycloak.service**

```
[Unit]

Description=Keycloak Server

After=network.target

Wants=network.target

[Service]

Type=notify

AmbientCapabilities=CAP_SYS_ADMIN

User=keycloak

Group=keycloak

EnvironmentFile=/etc/keycloak/keycloak.conf

ExecStart=/opt/keycloak-26.0.4/bin/kc.sh start --http-port=8080 --log=console,file --log-file=/var/log/keycloak/keycloak.log

[Install]

WantedBy=multi-user.target
```

**/etc/keycloak/keycloak.conf**

```
JAVA_HOME=/opt/jdk-pro-17.0.10

KC_DB=postgres

KC_DB_PASSWORD=***

KC_DB_USERNAME=keycloak

KC_DB_URL=jdbc:postgresql://localhost/keycloak

KC_FEATURES=token-exchange

KC_PROXY=edge

KC_HOSTNAME_STRICT=false

KC_HOSTNAME=test.esia.playa.ru

KC_HTTP_RELATIVE_PATH=/auth

KC_LOG_LEVEL=ru.playa:DEBUG
```

## Приложение 2. Пример конфигурационного файл nginx для тестового контура.

```
server {
    listen 443 ssl http2;
    server_name test.esia.playa.ru;
    proxy_buffer_size 64k;
    proxy_buffers 4 64k;
    proxy_busy_buffers_size 64k;
    ssl_certificate     /etc/letsencrypt/live/test.esia.playa.ru/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/test.esia.playa.ru/privkey.pem;
    ssl_session_cache shared:SSL:1m;
    ssl_session_timeout  5m;
    keepalive_timeout   70;
    ssl_ciphers  HIGH:!aNULL:!MD5;
    ssl_prefer_server_ciphers   on;
    access_log /srv/www/test.esia.playa.ru/log/access.log;
    error_log /srv/www/test.esia.playa.ru/log/error.log;
    root /srv/www/test.esia.playa.ru/www;
    gzip on;
    gzip_proxied any;
    gzip_comp_level 6;
    gzip_buffers 16 8k;
    gzip_min_length 256;
    gzip_types
      text/plain
      text/css
      application/json
      application/javascript
      application/x-javascript
      text/xml
      application/xml
      application/xml+rss
      text/javascript
```

```
        application/vnd.ms-fontobject
        application/x-font-ttf
        font/opentype
        image/svg+xml
        image/x-icon;

    location /auth {
        proxy_pass http://localhost:8080;
        proxy_http_version 1.1;
        proxy_set_header Host $host;
        proxy_set_header X-Url-Scheme $scheme;
        proxy_set_header X-Host $http_host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
        client_max_body_size 400m;
        client_body_buffer_size 128k;
        proxy_connect_timeout 7200;
        proxy_send_timeout 7200;
        proxy_read_timeout 7200;
        expires -1;
    }

    location /.well-known {
        root /srv/www/test.esia.playa.ru/www;
    }
}

server {
    listen      80;
    server_name  test.esia.playa.ru;
    location /.well-known {
```

```
        root /srv/www/test.esia.playa.ru/www;
    }
    location / {
        return 301 https://test.esia.playa.ru$request_uri;
    }
}
```

## Приложение 3 Пример профиля пользователя.

Пример профиля пользователя с разрешёнными scope `contacts`, `email`, `mobile`, `vehicles`, `addresses`, `id_doc`


```
{
 "stateFacts": [
  "EntityRoot"
 ],
 "firstName": "Имя004",
 "lastName": "Фамилия004",
 "middleName": "Отчество004",
 "birthDate": "04.12.1986",
 "gender": "M",
 "trusted": true,
 "citizenship": "RUS",
 "snils": "000-000-600 04",
 "updatedOn": 1675139896,
 "vehicles": {
  "totalSize": 1
 },
 "rfgUOperatorCheck": false,
 "status": "REGISTERED",
 "verifying": false,
 "rIdDoc": 222686,
 "containsUpCfmCode": false,
 "eTag": "4D8A144DBDA5DD536CE39E95B86DF65CAA84ADB6",
 "addrs": [
  {
   "stateFacts": [
    "Identifiable"
   ],
   "id": 137916,
   "type": "PLV",
```

```
    "countryId": "RUS",
    "addressStr": "обл. Московская, г. Балашиха, мкр. 1 Мая",
    "fiasCode": "da5f6bba-281f-4900-8627-2543c85abf71",
    "zipCode": "143911",
    "region": "Московская",
    "city": "Балашиха",
    "settlement": "1 Мая",
    "house": "36",
    "flat": "140",
    "eTag": "210D568CDDBC5FE5EED5816ECFBD50B340885378"
  }
 ],
 "vhls": [
  {
    "stateFacts": [
      "Identifiable"
    ],
    "id": "84670",
    "name": "шруслер империал лебароша",
    "numberPlate": "A301BX76",
    "regCertificate": {
      "series": "1111",
      "number": "111111"
    },
    "vrfStu": "NOT_VERIFIED",
    "duplicate": false,
    "eTag": "D71F669F2296EB4DE25CE400EC91DC7F4927093A"
  }
 ],
 "ctts": [
  {
    "stateFacts": [
```

```
    "Identifiable"
  ],
  "id": 14292556,
  "type": "MBT",
  "vrfStu": "VERIFIED",
  "value": "+7(930)7537466",
  "otpCodeLength": 4,
  "eTag": "6C388EFECFB95EBD8F88539E4BB370C56607A550"
},
{
  "stateFacts": [
    "Identifiable"
  ],
  "id": 14292556,
  "type": "MBT",
  "vrfStu": "NOT_VERIFIED",
  "value": "+7(930)7537477",
  "otpCodeLength": 4,
  "eTag": "6C388EFECFB95EBD8F88539E4BB370C56607A550"
},
{
  "stateFacts": [
    "Identifiable"
  ],
  "id": 14434559,
  "type": "PHN",
  "vrfStu": "NOT_VERIFIED",
  "value": "+7(499)3213211",
  "otpCodeLength": 4,
  "eTag": "AA1806DDEED6FC8841B3DA99A99EA53A8E6EF735"
},
{
```

```
    "stateFacts": [
      "Identifiable"
    ],
    "id": 14492654,
    "type": "EML",
    "vrfStu": "VERIFIED",
    "value": "EsiaTest004@yandex.ru",
    "otpCodeLength": 4,
    "eTag": "EBD4F3D407CBF92F2367108838483F2B3542CCC6"
   }
  ],
  "docs": [
   {
    "stateFacts": [
     "EntityRoot"
    ],
    "id": 222686,
    "type": "RF_PASSPORT",
    "vrfStu": "VERIFIED",
    "series": "4545",
    "number": "445587",
    "issueDate": "04.01.2023",
    "issueId": "545444",
    "issuedBy": "мвд",
    "eTag": "C374DDAC25E11CCE5407FB1CB81CB0047A8C43D9"
   }
  ],
  "username": "79307537466",
  "email": "EsiaTest004@yandex.ru",
  "sbjID": "1000299656"
}
```