Приложения ТПУ

Здесь представлены основные моменты при работе с приложениями ТПУ, разработанными для цифровизации бизнес процессов.

- Введение
  - Знакомство с приложениями ТПУ
  - Типы приложений
  - Доверенные приложения ТПУ
  - Платформы приложения
  - Версии приложения
  - Разработчики приложения
  - Личности приложения ТПУ
  - Ключ доступа
- Создание приложения
  - Регистрация нового приложения
  - Добавление платформы приложения
  - Добавлении версии приложения
  - Расчет контрольной суммы
- Управление приложением
  - Настройки единой авторизации
  - Способы авторизации и получения доступа
  - Добавление разработчиков

Введение

Знакомство с приложениями ТПУ

Данная книга описывает основные принципы работы с приложениями в единой информационной системе ТПУ (ЕИС ТПУ).

Любой корпоративный пользователь может управлять своим приложением.

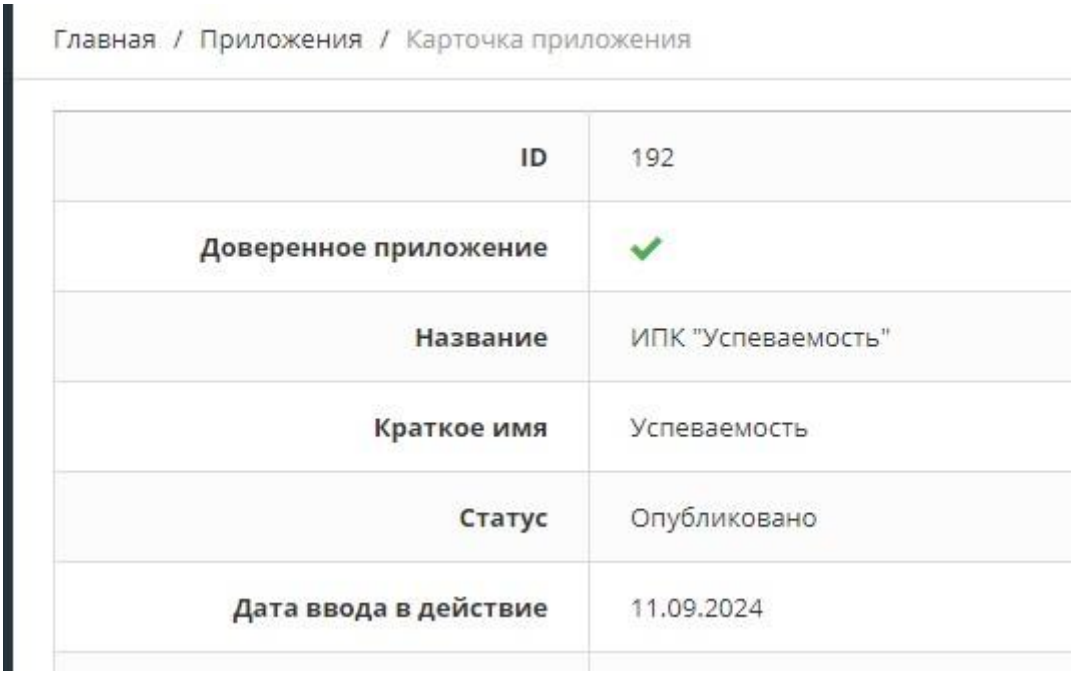
Типы приложений

Приложения ТПУ могут быть классифицированы на следующие категории:

1. Web-приложение;
2. Web-сайт;
3. Десктопное приложение;
4. Мобильное приложение.

Доверенные приложения ТПУ

Приложения, которые имеют значок (**Доверенное приложение**) имеют больший доступ к функционалу через API.



К дополнительному функционалу относятся следующие функции:

- Получение Идентификаторов
- Без ограничений доступа по IP
- Сквозная авторизация между приложениями

> Установить данный атрибут могут только сотрудники УЦ после проверки приложения

Платформы приложения

Каждое приложение может иметь несколько реализаций под разные платформах, но только одну реализацию на каждую платформу.

Типы платформ приложений:

1. Android
2. iOS
3. Web
4. Winphone
5. Desktop

Версии приложения

На каждой платформе приложения может быть несколько версий, но только одна может быть активна в рамках каждой платформы.

Версия приложения содержит важные данные для взаимодействия с API.

- Публичный ключ
- Приватный ключ

> Публичный ключ вы можете использовать для всех взаимодействий с API (https://api.tpu.ru)

> Приватный ключ используется только для расчета контрольной суммы параметров в запросах и должен быть скрыт от посторонних пользователей

Разработчики приложения

Разработчиками приложения являются пользователи, у которых есть корпоративный логин и пароль в ТПУ. Разработчики могут добавлять других разработчиков, а также менять параметры своего приложения и обращаться к закрытой части документации по API с функцией "Попробовать".

Личности приложения ТПУ

Данный раздел содержит ограниченный список личностей, которые так или иначе связаны с приложением, но не обязательно имеющие доступ к этому приложению.

Личности из этого списка могут присутствовать в выходных документах или быть ответственными за выполнение задач в приложении.

Ключ доступа

Ключ доступа - это публичный SSL ключ, который использует сервер авторизации для расшифровки сообщений, зашифрованные приватным ключом данного приложения.

> Публичный ключ добавляет разработчик приложения

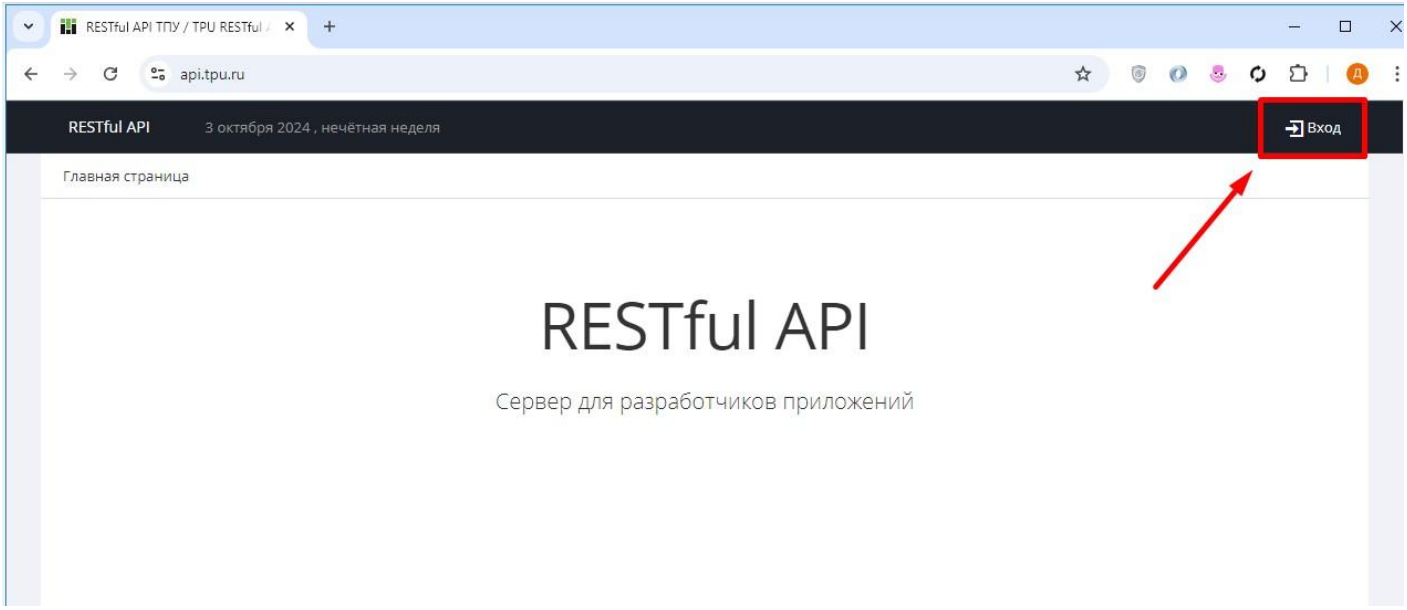
> Приватный ключ находится в самом приложении и не загружается в систему приложений ТПУ

Создание приложения

Регистрация нового приложения

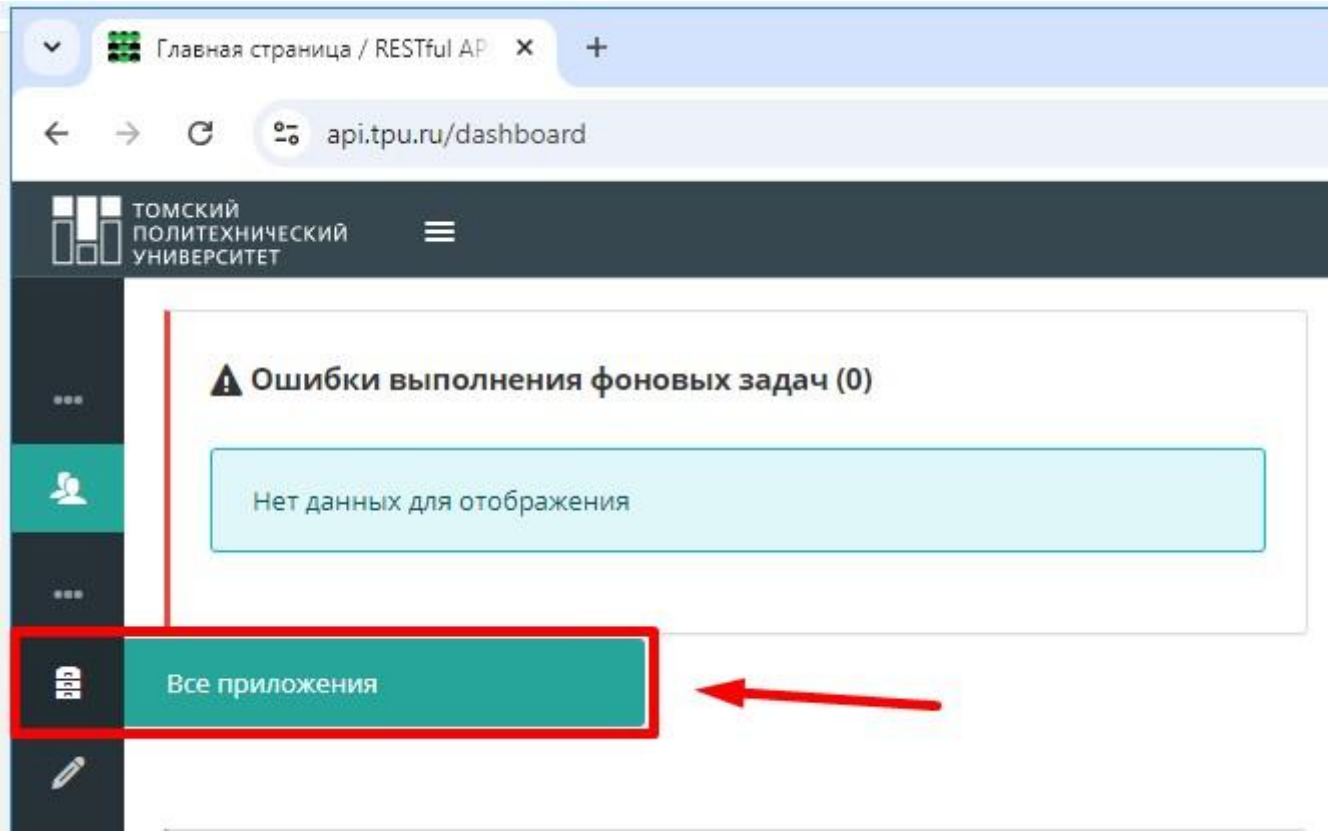
Для начала работы надо авторизоваться в приложение API TPU (https://api.tpu.ru)

1. Перейти на сайт API TPU (https://api.tpu.ru)
2. Нажать на кнопку вход (см. рисунок) следуя инструкциям авторизации ввести свой корпоративный логин и пароль.

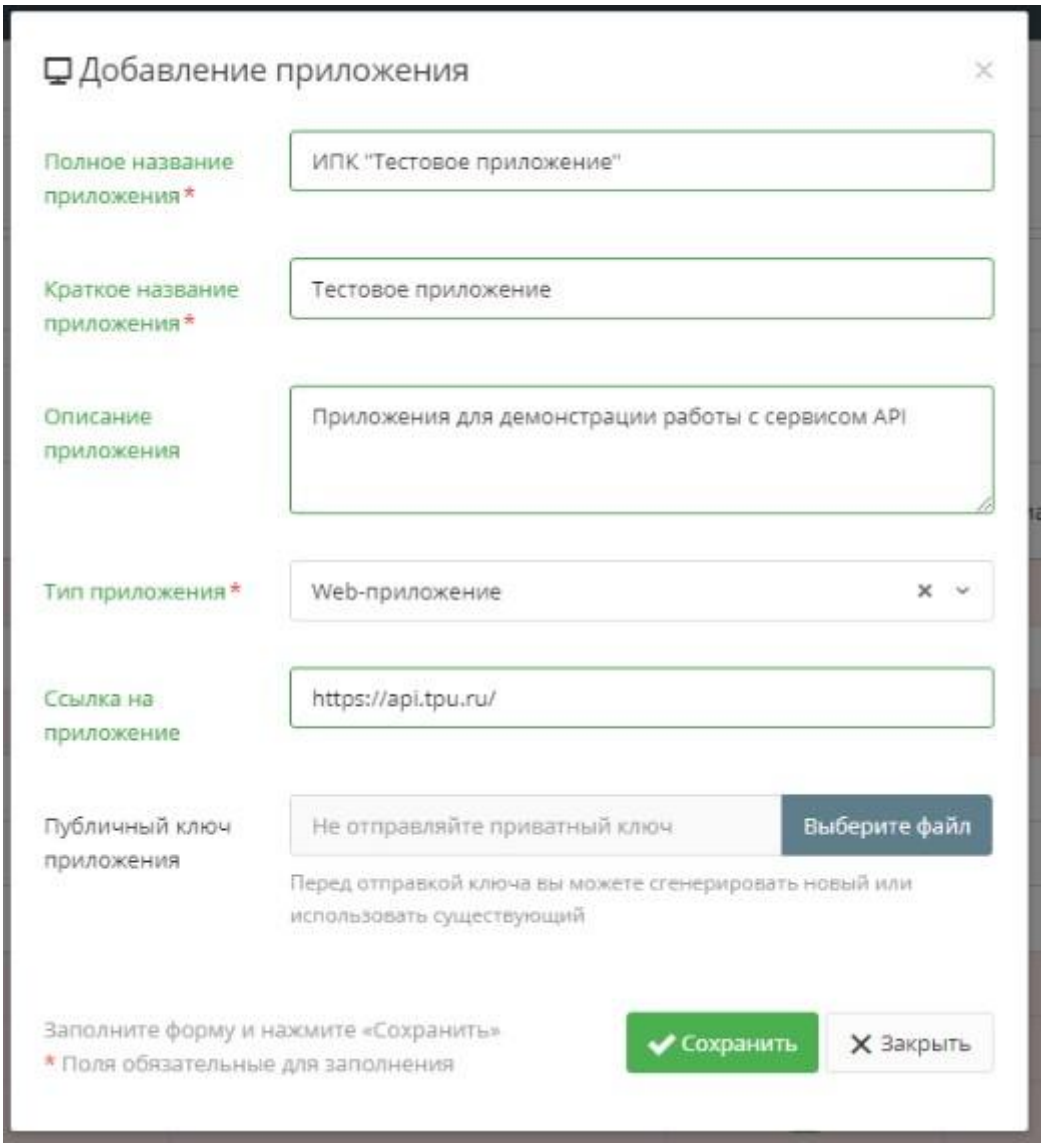


3. После авторизации в боковом меню выбрать раздел "Все приложения"

> В списке будут отображаться приложения, где вы являетесь разработчиком. Если кто-то вас добавит к своему приложению - оно отобразиться у вас в боковом меню.



4. В правом верхнем углу нажмите кнопку "Добавить" и заполните все поля, при необходимости можете сразу указать публичный SSL ключ вашего приложения

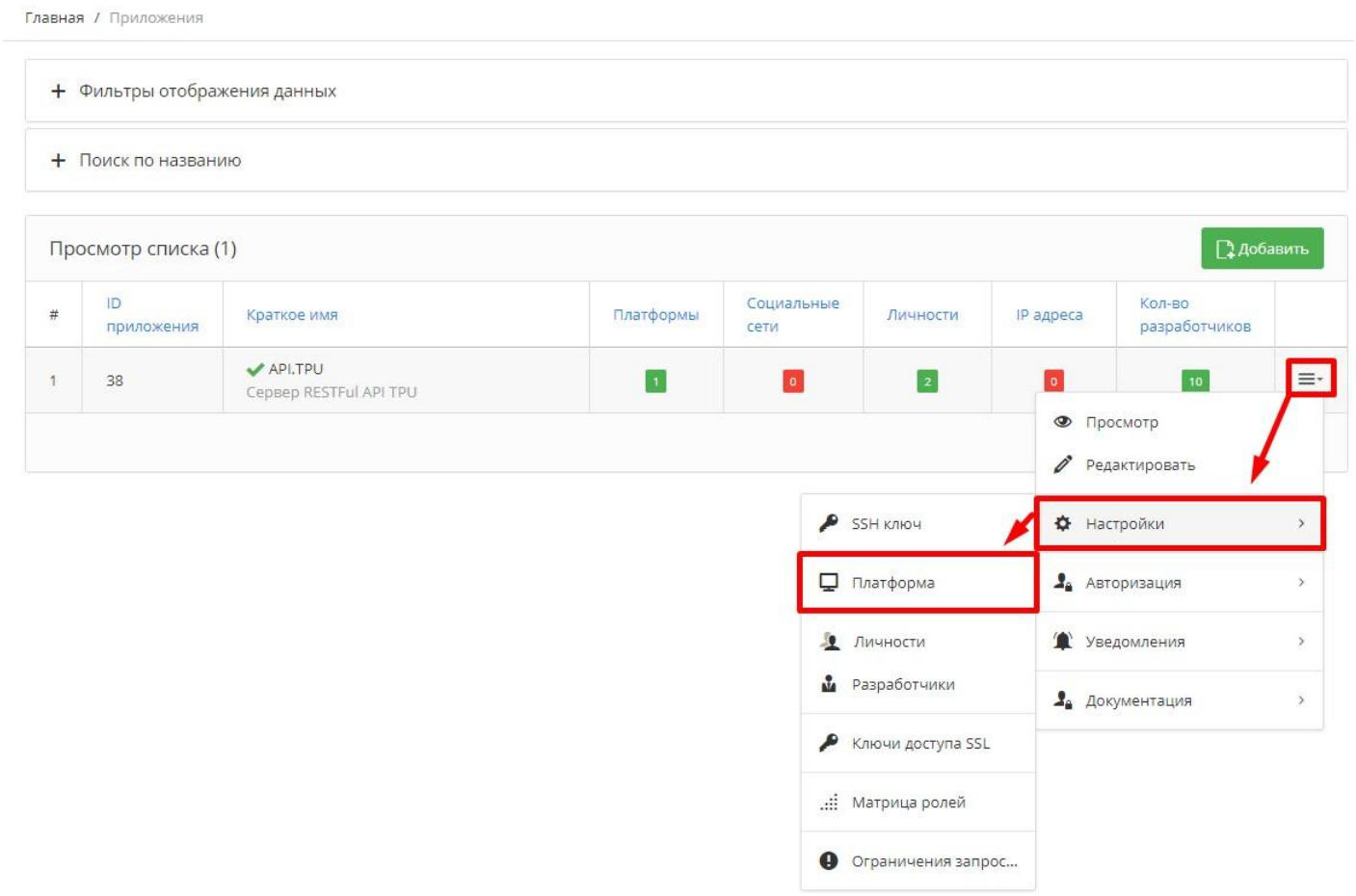


5. После добавления вашего приложения отправьте заявку на почту support@tpu.ru для подтверждения вашего приложения.

Добавление платформы приложения

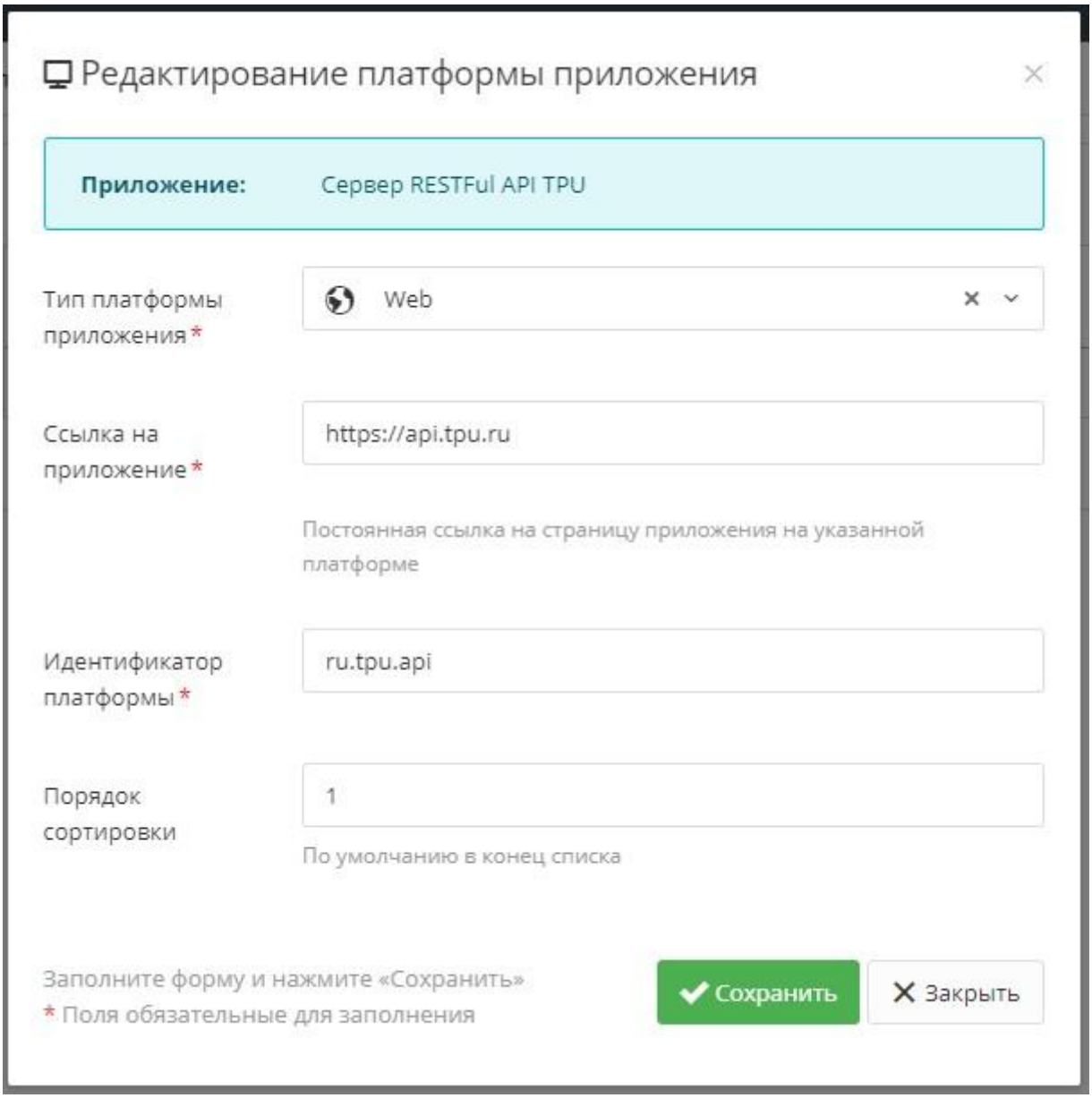
После добавления вашего приложения, требуется добавить платформы и получить API Key для работы с методами API.

Для этого в меню выберите следующие пункты "Настройки" -> "Платформы".



В появившимся окне необходимо выбрать тип платформы приложения и заполнить все поля. Нажать "Сохранить".

> Идентификатор платформы должен содержать только латинские буквы. Обычно это алиас системы или доменное имя сайта, записанное в обратном порядке

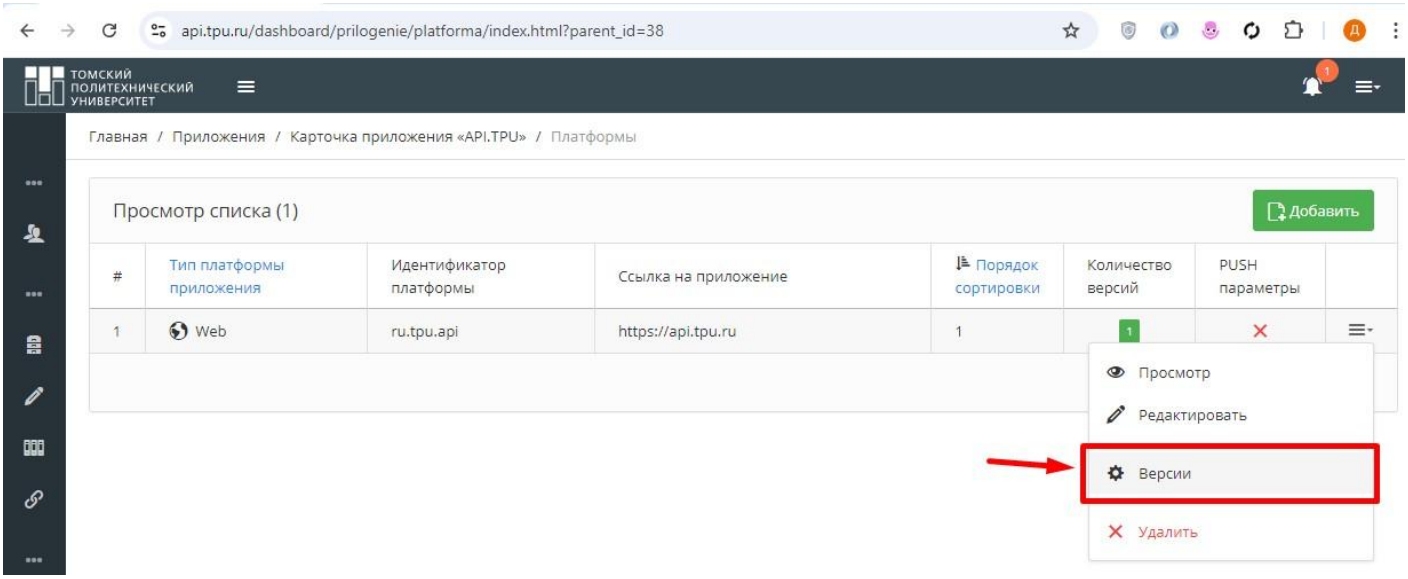

Редактирование платформы приложения
Приложение: Сервер RESTFul API TPU
Тип платформы приложения *
Web
Ссылка на приложение *
https://api.tpu.ru
Постоянная ссылка на страницу приложения на указанной платформе
Идентификатор платформы *
ru.tpu.api
Порядок сортировки
1
По умолчанию в конец списка
Заполните форму и нажмите «Сохранить»
* Поля обязательные для заполнения
Сохранить
Закрыть

Добавлении версии приложения

Следующий шаг - это добавление версии приложения, для того чтобы получить API Key и Private Key для работы с методами https://api.tpu.ru

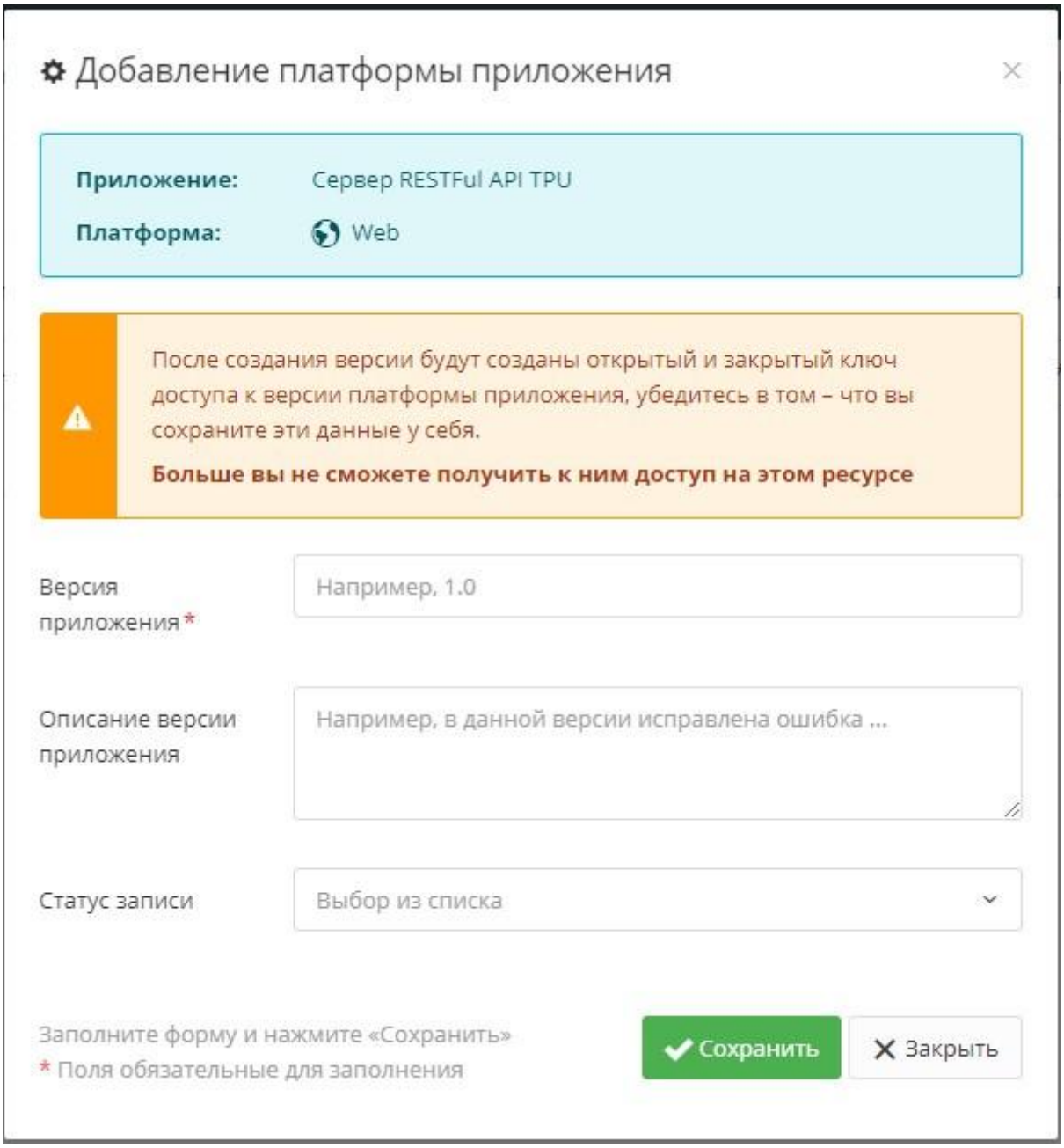
Выберите пункт "Версии" напротив интересующей вас платформы приложения.

> Каждая платформа приложения может иметь несколько разных версий

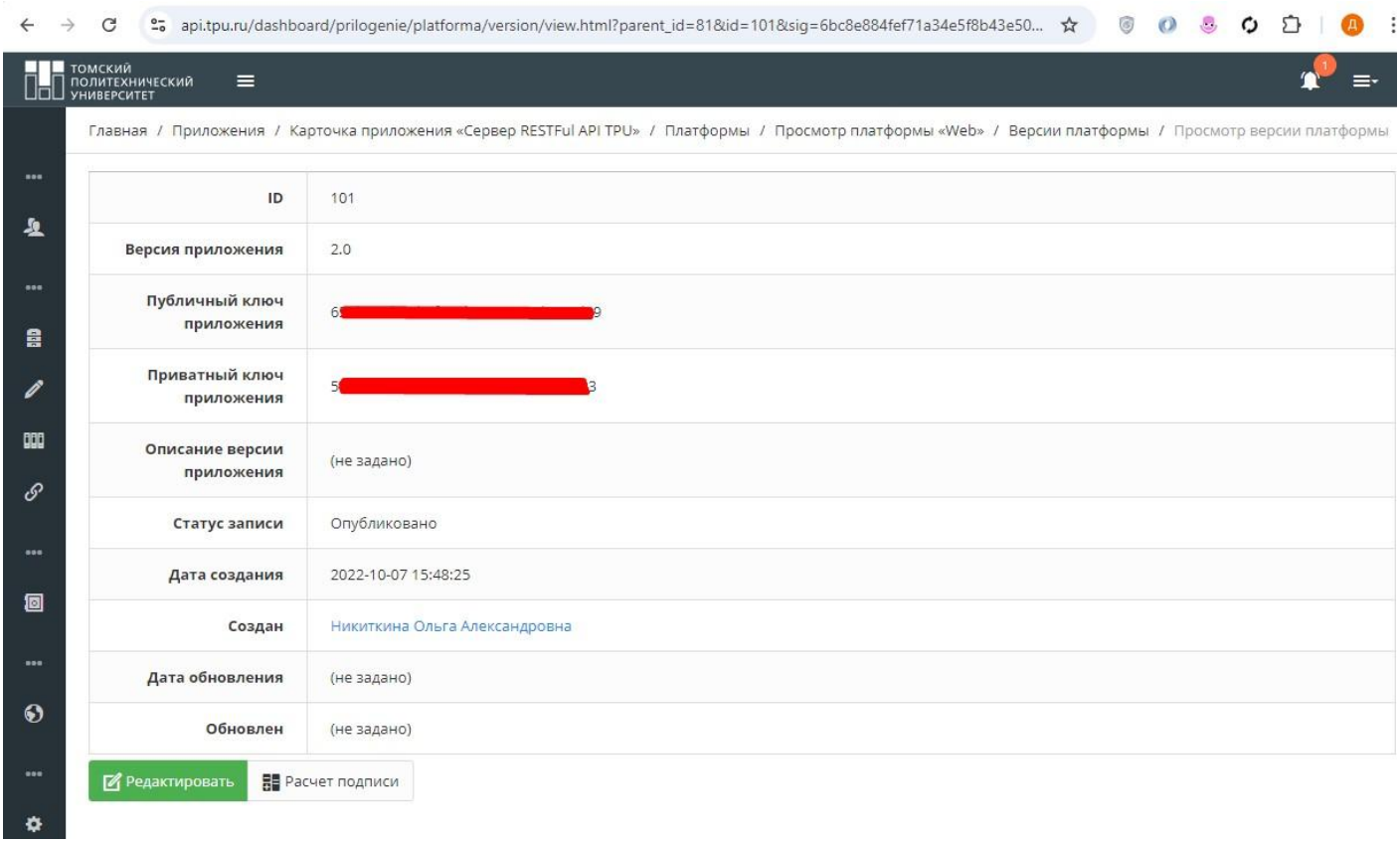


В модальном окне заполните номер версии приложения, описание и статус.

> Только *версия приложения* в статусе "Опубликовано" будет работать в API TPU

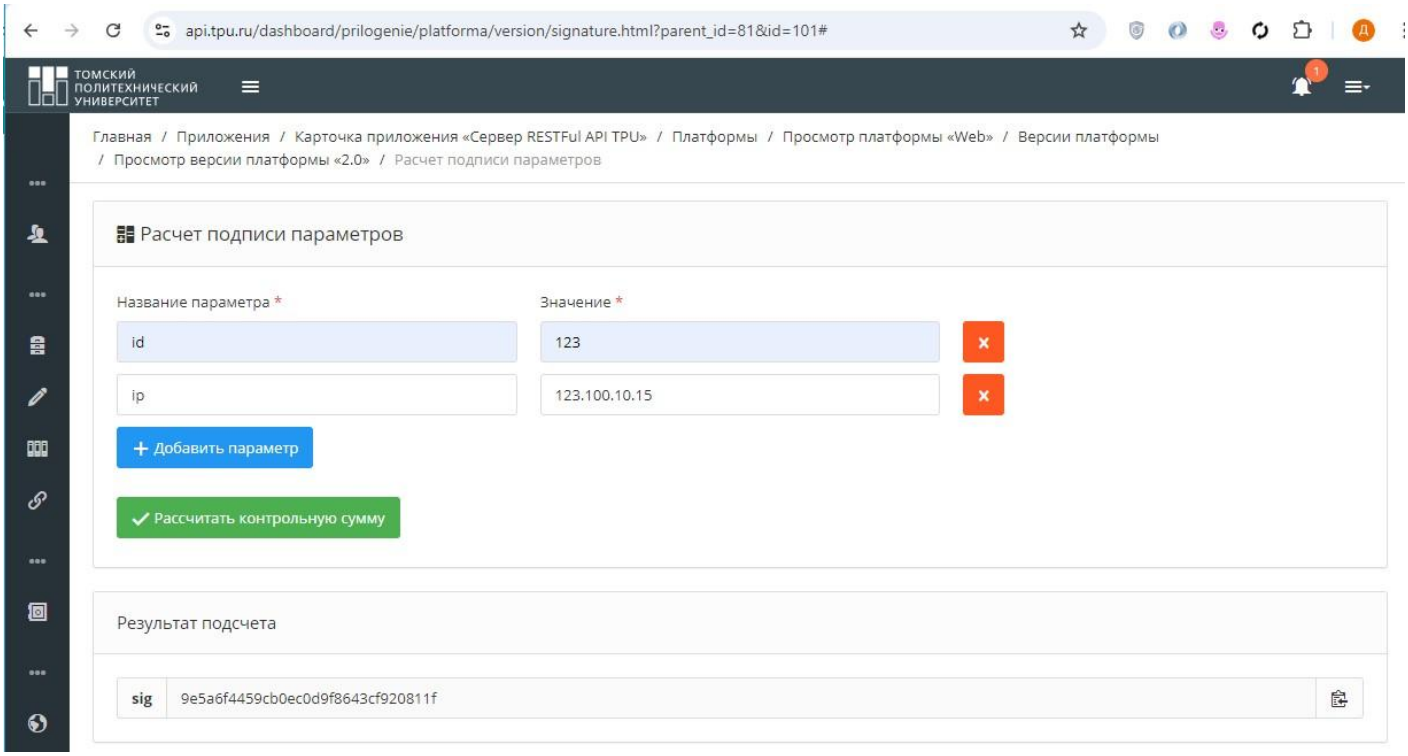


После добавления версии приложения вы получите два ключа *публичный* и *приватный.* Вы их будете использовать при работе с методами API TPU.


api.tpu.ru/dashboard/prilogenie/platforma/version/view.html?parent_id=81&id=101&sig=6bc8e884fef71a34e5f8b43e50...
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
Главная / Приложения / Карточка приложения «Сервер RESTFul API TPU» / Платформы / Просмотр платформы «Web» / Версии платформы / Просмотр версии платформы
ID
101
Версия приложения
2.0
Публичный ключ приложения
Приватный ключ приложения
Описание версии приложения
(не задано)
Статус записи
Опубликовано
Дата создания
2022-10-07 15:48:25
Создан
Никиткина Ольга Александровна
Дата обновления
(не задано)
Обновлен
(не задано)
Редактировать
Расчет подписи

Расчет контрольной суммы

Для некоторых методов требуется указать дополнительный параметр безопасности sig. В интерфейсе вы сможете проверить алгоритм расчета этого параметра на сервере. Разработчику приложения нужно реализовать свой собственный метод для получения данного параметра.



> Список параметров, которые могут быть переданы через HTTP-Get запрос, игнорируемые при расчёте контрольной суммы. т.к. эти параметры системные и не могут влиять на работу.
>
> 1. **sig**
> 2. **apikey**
> 3. **suppress_response_code**
> 4. **access_token**

Пример на языке PHP

```
public function sig(array $arguments, $secret)
{
```

```
    ksort($arguments);
    $string = '';

    foreach ($arguments as $key => $value) {
        $string .= "$key=$value";
    }

    return md5($string . $secret);
}
```

Пример на языке Java

```
import java.security.MessageDigest;
import java.util.List;
import java.util.Map;
import java.util.stream.Collectors;
import java.security.NoSuchAlgorithmException;

public String sig(Map<String, String> arguments, String secret) throws NoSuchAlgorithmException {
    StringBuilder builder = new StringBuilder();

    List<String> keys = arguments.keySet().stream().sorted().collect(Collectors.toList());

    for(String key : keys) {
        builder.append(key).append("=").append(arguments.get(key));
    }

    builder.append(secret);
    byte[] digest = MessageDigest.getInstance("MD5").digest(builder.toString().getBytes());

    StringBuilder resultBuilder = new StringBuilder();

    for(byte b : digest) {
        resultBuilder.append(String.format("%02x", b));
    }

    return resultBuilder.toString();
}
```

Пример на языке Scala

```
import java.security.MessageDigest
import scala.collection.immutable.ListMap

def sig(arguments: Map[String, String], secret: String): String = {
    MessageDigest.getInstance("MD5").digest {
        s"${ListMap(arguments.toSeq.sortBy({ case (key, value) => key }):_*).map({ case (key, value) =>
s"$key=$value" }).mkString}$secret".getBytes
    }.map(byte => String.format("%02x", byte)).mkString
}
```

Пример на языке Python

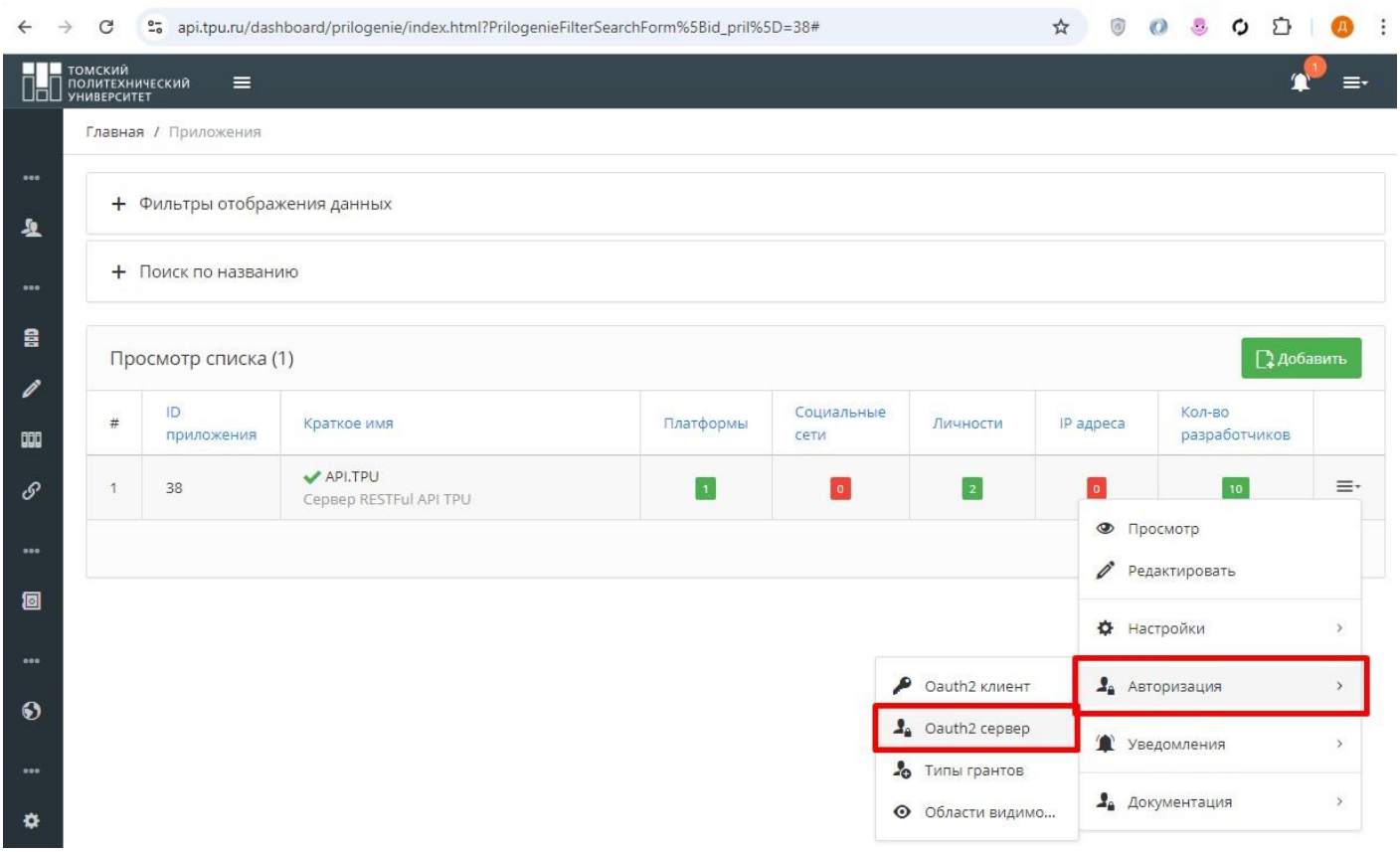
```
import collections
import hashlib

def sig(arguments, secret):
    ordered = collections.OrderedDict(sorted(arguments.items()))
    concated = "".join(list(map(lambda k,v: k + "=" + v, ordered.keys(), ordered.values()))) + secret
    return hashlib.md5(concated.encode()).hexdigest()
```

Управление приложением

Настройки единой авторизации

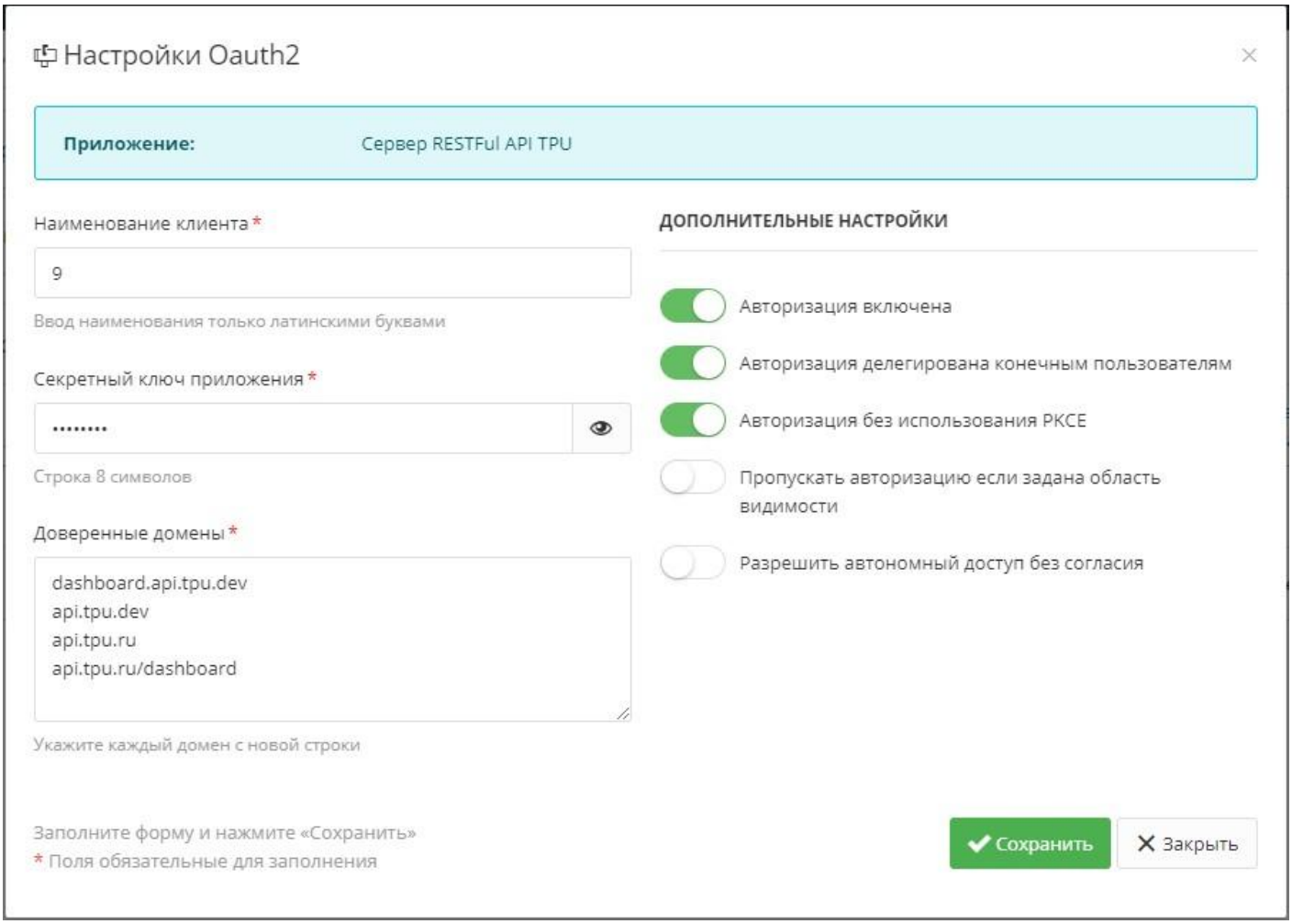
Для подключение единой авторизации в приложении требуется настроить параметры *Oauth2 сервер*



Здесь отображаются

- *Идентификатор клиента **(client_id)***
- *Секретный ключ приложения **(client_secret)***

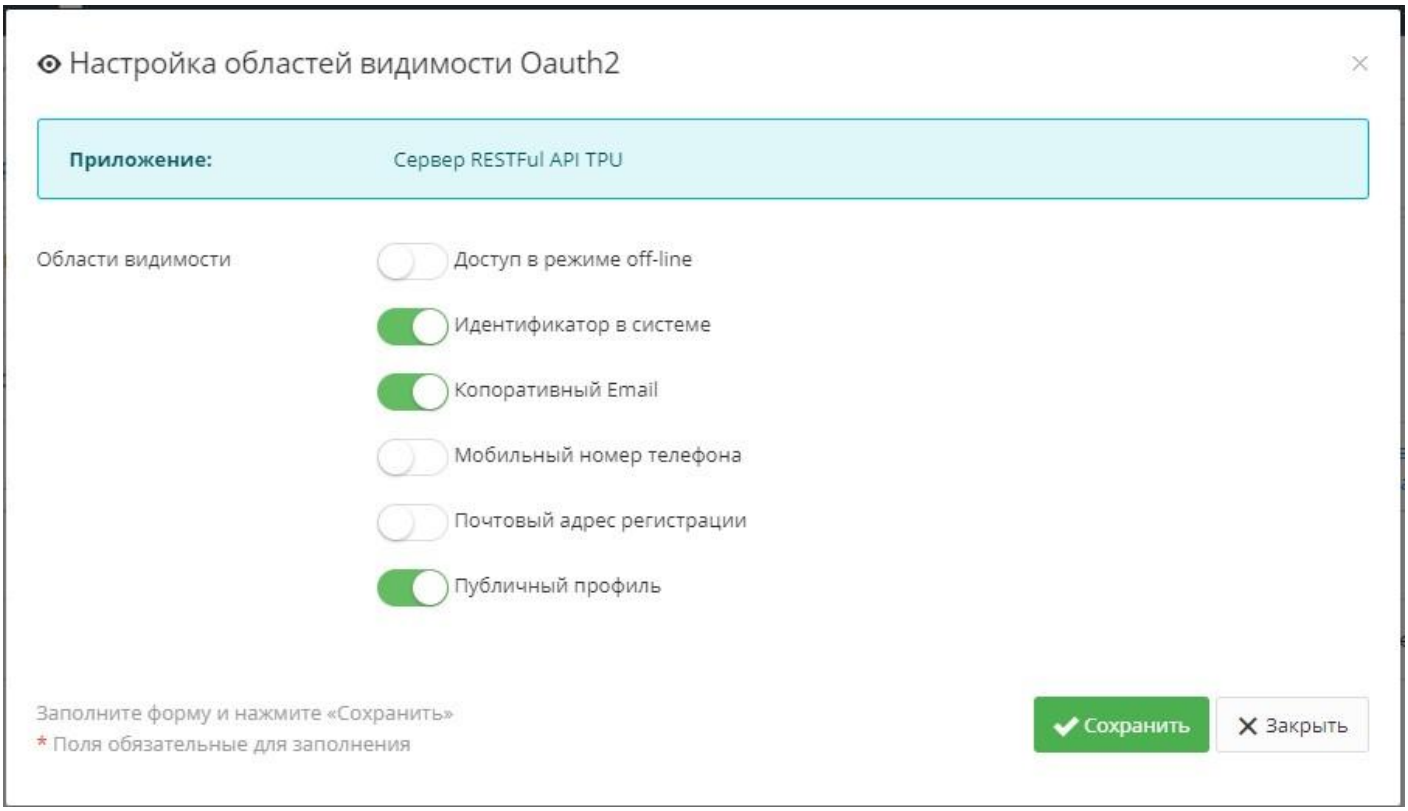
> *Для параметра **client_secret** должно быть обеспечено хранение в защищенном хранилище*



Доверенные домены - в этом списка на каждой новой строке указывается перечень доменных имён, с которых разрешено подключение вашего приложения.

Дополнительные параметры	**Описание**
Авторизация включена	Основной переключатель - доступна или нет авторизация
Авторизация делегирована конечным пользователям	Пользователи могу авторизоваться в системе
Авторизация без использования PKCE	Для повышения безопасности можно использовать *Proof Key for Code Exchange*
Пропускать авторизацию если задана область видимости	
Разрешить автономный доступ без согласия	Разрешено для только доверенных приложений ТПУ

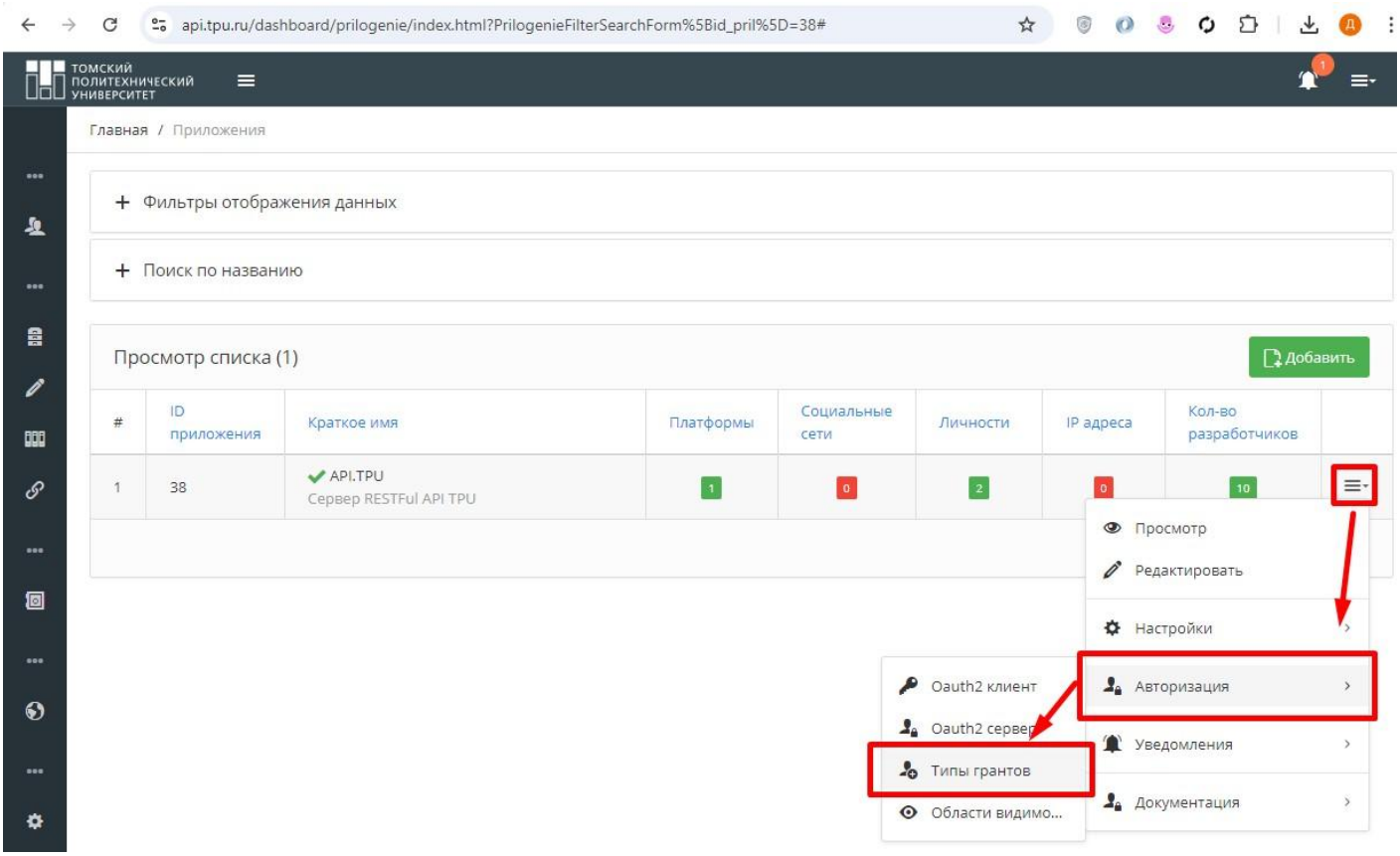
Настройки области видимости - это список параметров которые сервис единой авторизации сможет предоставить приложению, если пользователь согласиться их передать.


Настройка областей видимости Oauth2
Приложение:
Сервер RESTFul API TPU
Области видимости
Доступ в режиме off-line
Идентификатор в системе
Копоративный Email
Мобильный номер телефона
Почтовый адрес регистрации
Публичный профиль
Заполните форму и нажмите «Сохранить»
* Поля обязательные для заполнения
Сохранить
Закрыть

Управление приложением

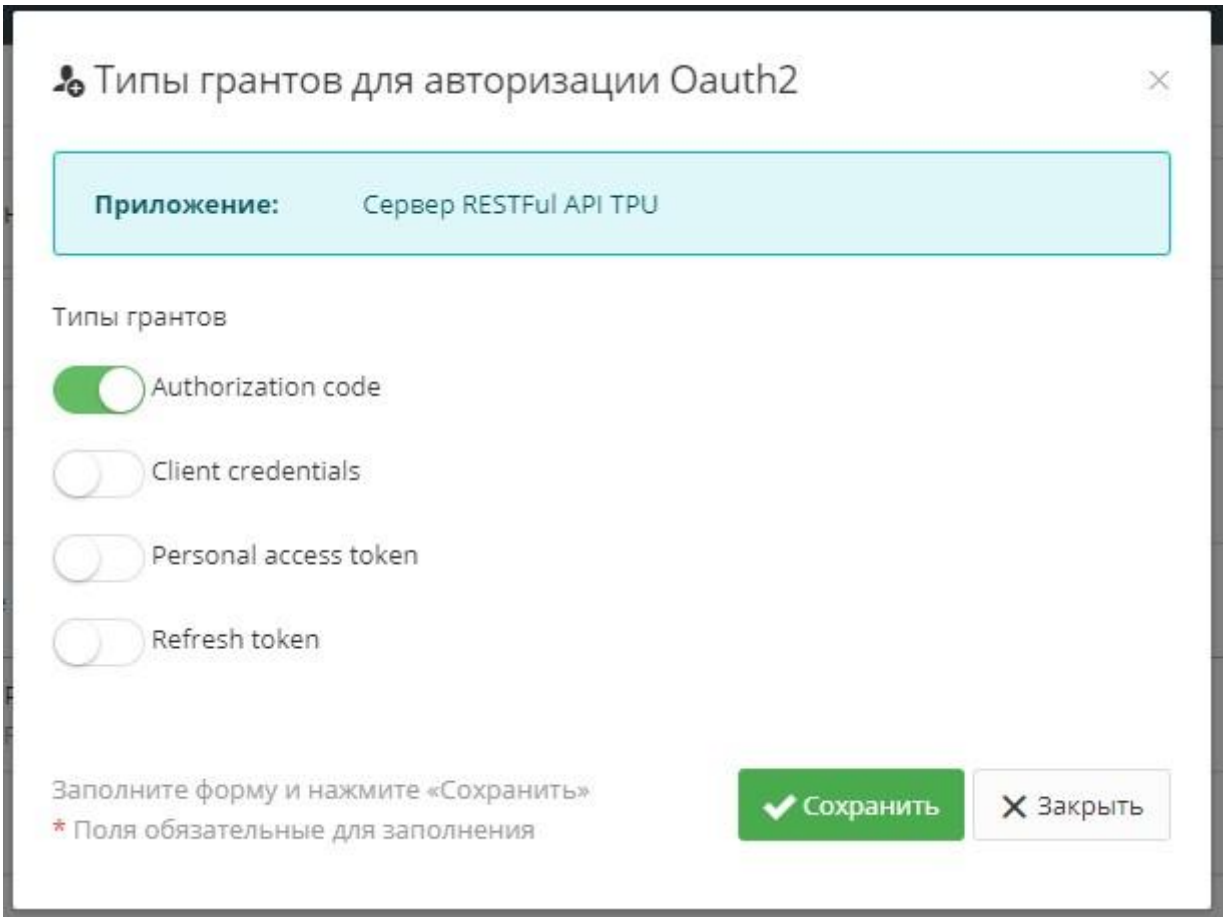
Способы авторизации и получения доступа

Настройка способов авторизации осуществляется в меню "Типы грантов"



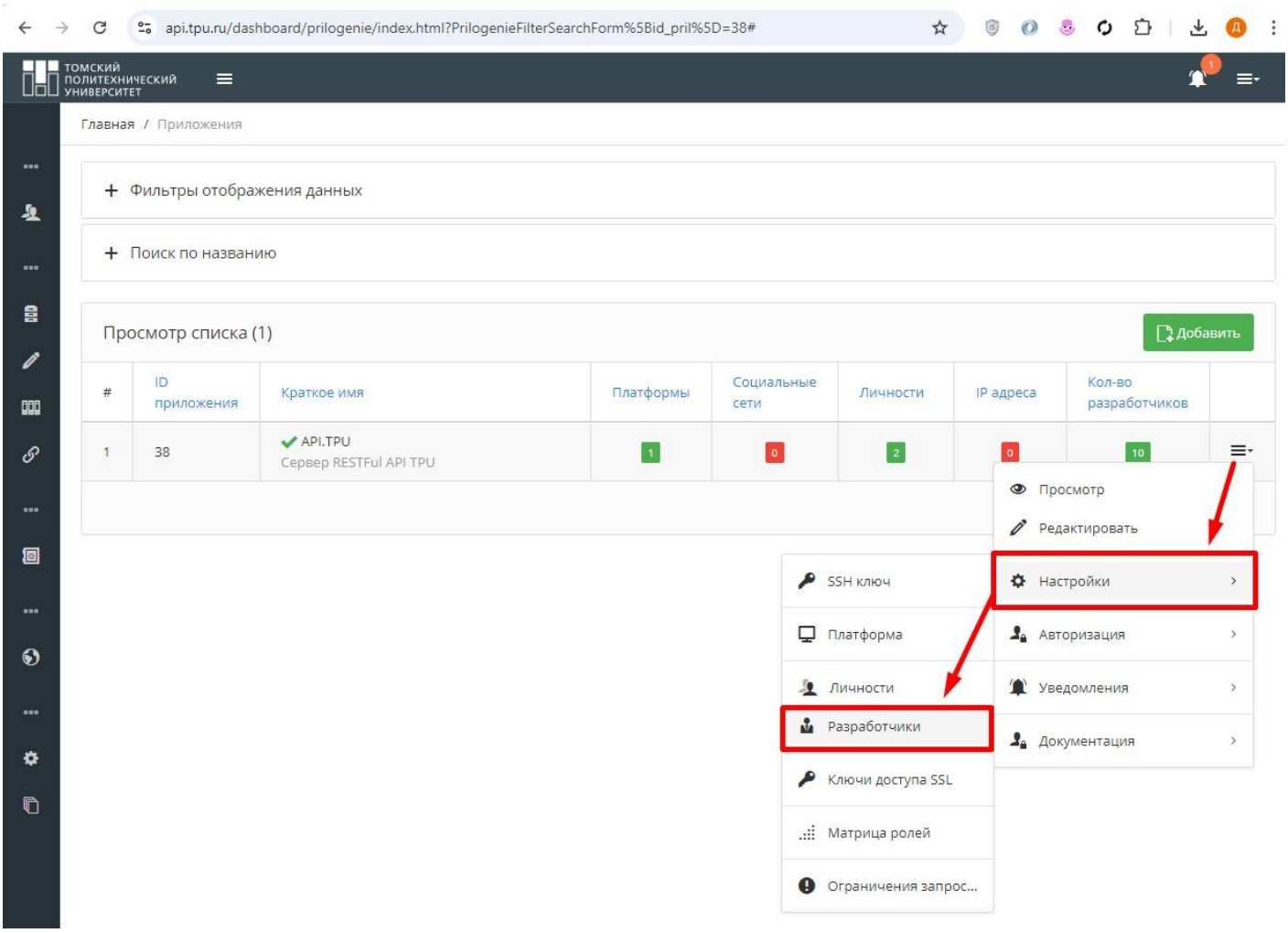
В этом разделе указывается способы получения авторотационного ключа пользователя, требуется указать дополнительные способы если ваше приложение их использует.

> Способ авторизации ***Personal access token*** доступен только для доверенных приложений ТПУ

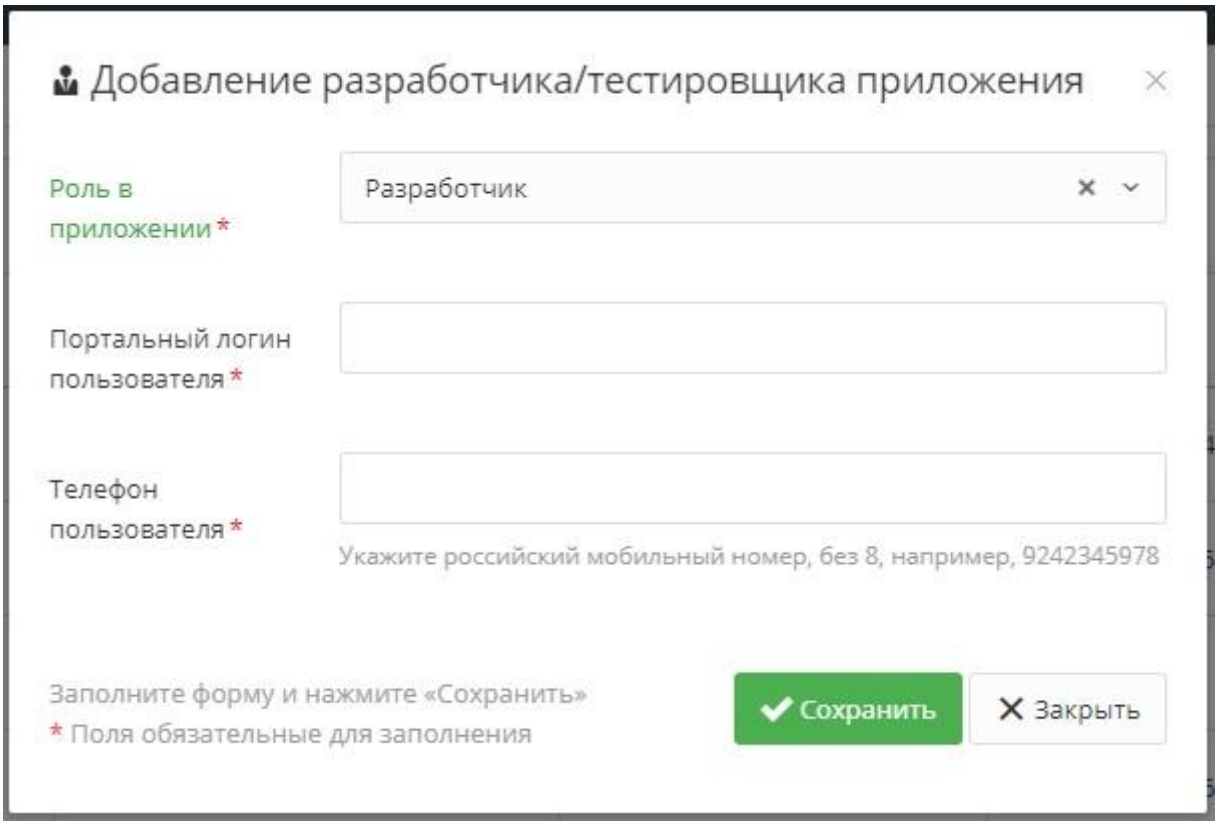

Типы грантов для авторизации Oauth2
Приложение:
Сервер RESTFul API TPU
Типы грантов
Authorization code
Client credentials
Personal access token
Refresh token
Заполните форму и нажмите «Сохранить»
* Поля обязательные для заполнения
Сохранить
Закрыть

Добавление разработчиков

Для добавления разработчика приложения необходимо в меню выбрать "Настройки" -> "Разработчики"



В списке разработчиков - нажать кнопку "Добавить" и в появившимся окне выбрать роль пользователя в приложении, указать корпоративный логин и мобильный номер телефона.


Добавление разработчика/тестировщика приложения
Роль в приложении *
Разработчик
Портальный логин пользователя *
Телефон пользователя *
Укажите российский мобильный номер, без 8, например, 9242345978
Заполните форму и нажмите «Сохранить»
* Поля обязательные для заполнения
Сохранить
Закрыть