

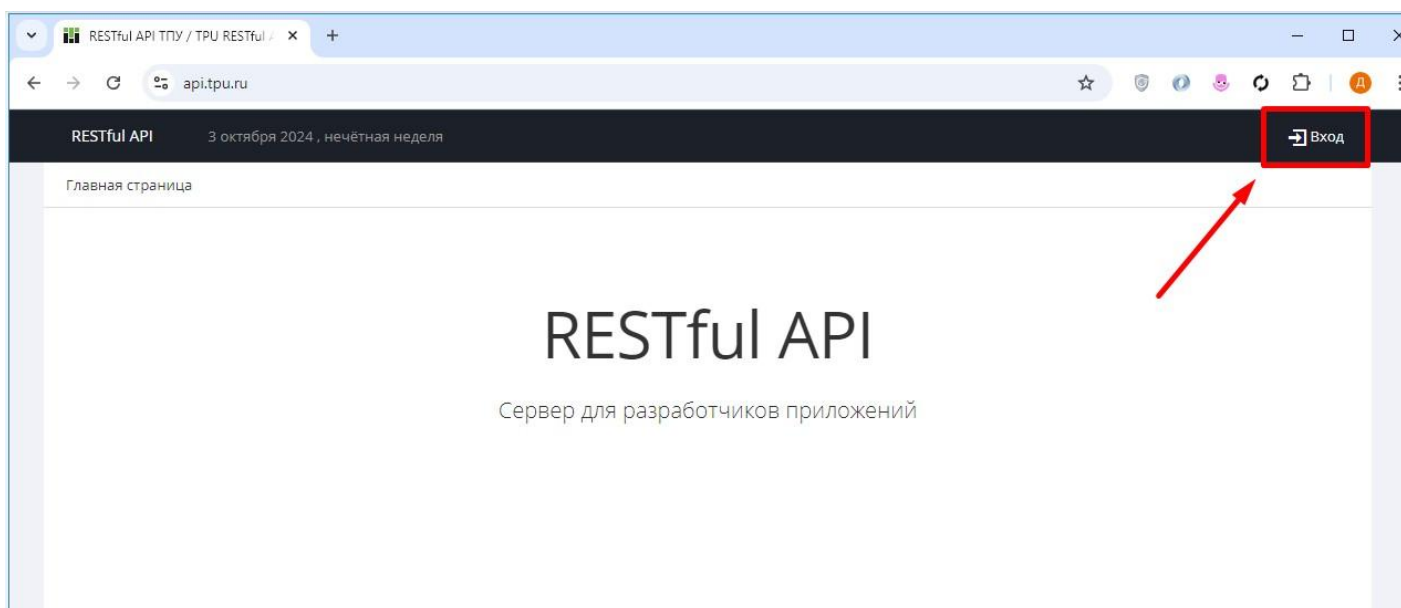
# Создание приложения

- [Регистрация нового приложения](#)
- [Добавление платформы приложения](#)
- [Добавлении версии приложения](#)
- [Расчет контрольной суммы](#)

# Регистрация нового приложения

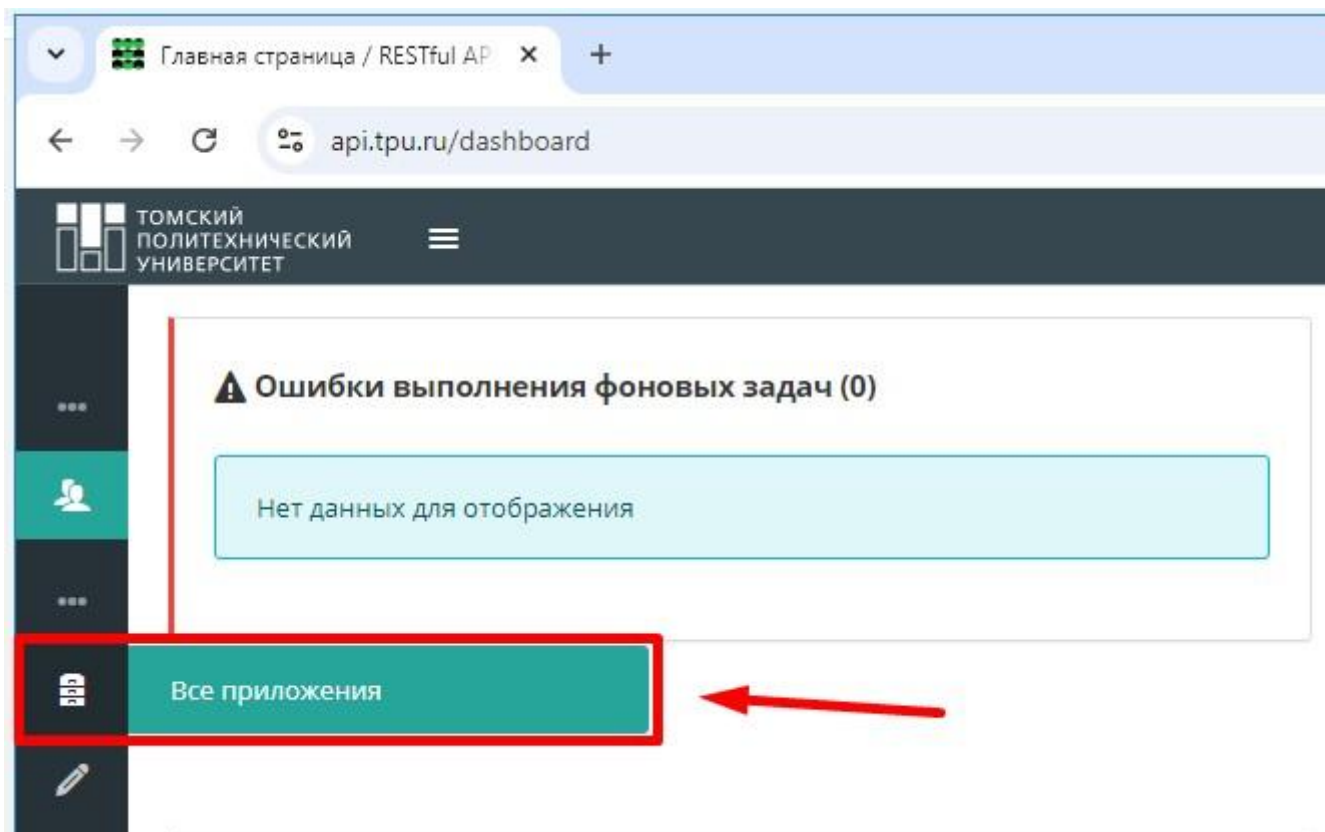
Для начала работы надо авторизоваться в приложение API TPU (<https://api.tpu.ru>)

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 отобразится 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

Добавление приложения

Полное название приложения \*

ИПК "Тестовое приложение"

Краткое название приложения \*

Тестовое приложение

Описание приложения

Приложения для демонстрации работы с сервисом API

Тип приложения \*

Web-приложение

Ссылка на приложение

https://api.tpu.ru/

Публичный ключ приложения

Не отправляйте приватный ключ

Выберите файл

Перед отправкой ключа вы можете сгенерировать новый или использовать существующий

Заполните форму и нажмите «Сохранить»

\* Поля обязательные для заполнения

✓ Сохранить

✕ Закрыть

5. После добавления вашего приложения отправьте заявку на почту [support@tpu.ru](mailto: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".

Главная / Приложения

+ Фильтры отображения данных

+ Поиск по названию

Просмотр списка (1) Добавить

#	ID приложения	Краткое имя	Платформы	Социальные сети	Личности	IP адреса	Кол-во разработчиков	
1	38	✓ API.TPU Сервер RESTful API TPU	1	0	2	0	10	<div><div>Просмотр</div><div>Редактировать</div><div>Настройки</div><div>Авторизация</div><div>Уведомления</div><div>Документация</div></div>

SSH ключ

Платформа

Личности

Разработчики

Ключи доступа SSL

Матрица ролей

Ограничения запрос...

Настройки

Авторизация

Уведомления

Документация

В появившемся 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

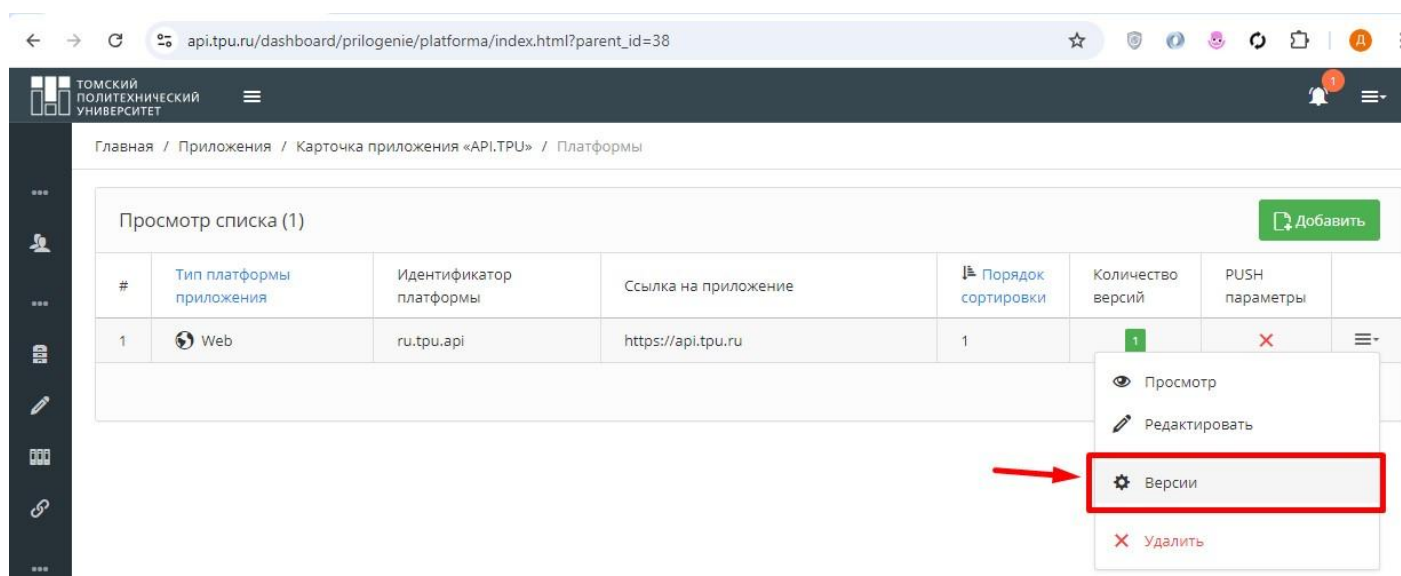
✕ Закрыть

# 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

⚙


Добавление платформы приложения


✕

Приложение:

Сервер RESTFul API TPU

Платформа:

 Web



После создания версии будут созданы открытый и закрытый ключ доступа к версии платформы приложения, убедитесь в том – что вы сохраните эти данные у себя.

**Больше вы не сможете получить к ним доступ на этом ресурсе**

Версия приложения \*

Описание версии приложения

Статус записи

Заполните форму и нажмите «Сохранить»

\* Поля обязательные для заполнения

✓ Сохранить

✕ Закрыть

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



ID	101
Версия приложения	2.0
Публичный ключ приложения	6 [REDACTED] 9
Приватный ключ приложения	5 [REDACTED] 3
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.

The screenshot shows a web application interface for calculating a signature. The browser address bar displays `api.tpu.ru/dashboard/prilogenie/platforma/version/signature.html?parent_id=81&id=101#`. The page title is "Расчет подписи параметров" (Signature Calculation). The interface includes a sidebar with navigation icons and a main content area. The main content area has a form with two columns: "Название параметра \*" (Parameter Name) and "Значение \*" (Value). The form contains two rows of input fields: one for "id" with value "123" and one for "ip" with value "123.100.10.15". There are red "X" buttons next to each input field. Below the input fields is a blue button labeled "+ Добавить параметр" (+ Add parameter) and a green button labeled "✓ Рассчитать контрольную сумму" (✓ Calculate checksum). Below the form is a section titled "Результат подсчета" (Calculation Result) which displays the calculated signature: "sig 9e5a6f4459cb0ec0d9f8643cf920811f".

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()
```