

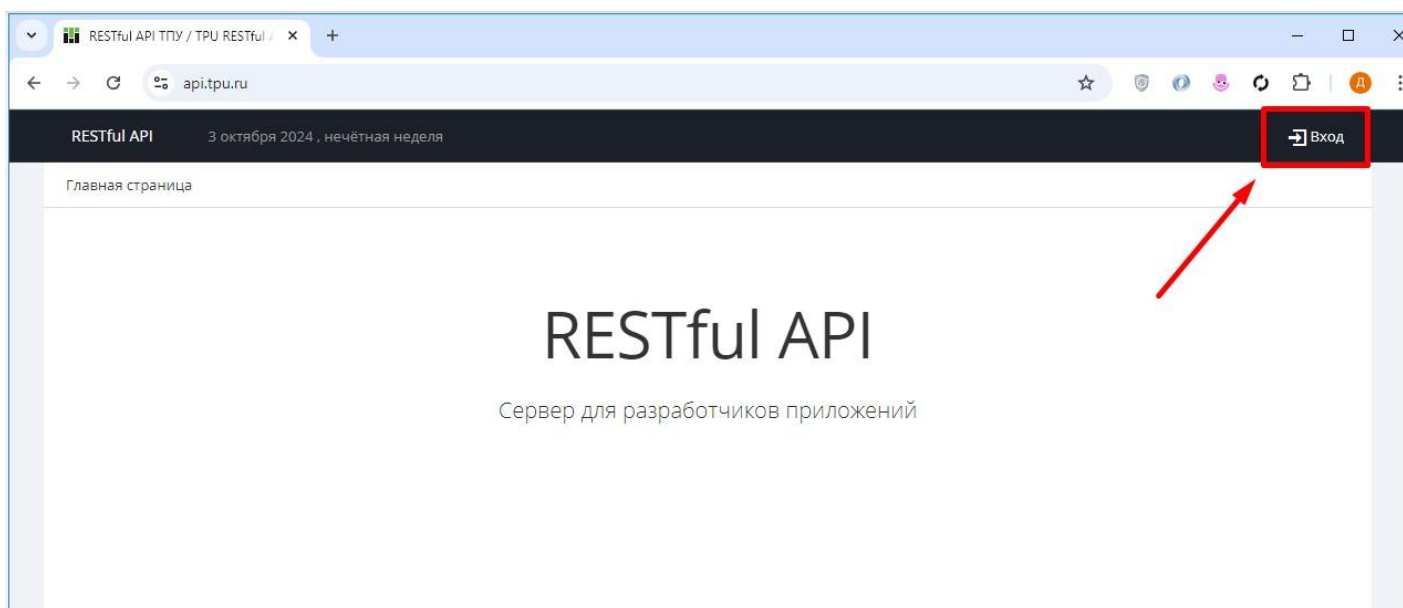
# Создание приложения

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

# Регистрация нового приложения

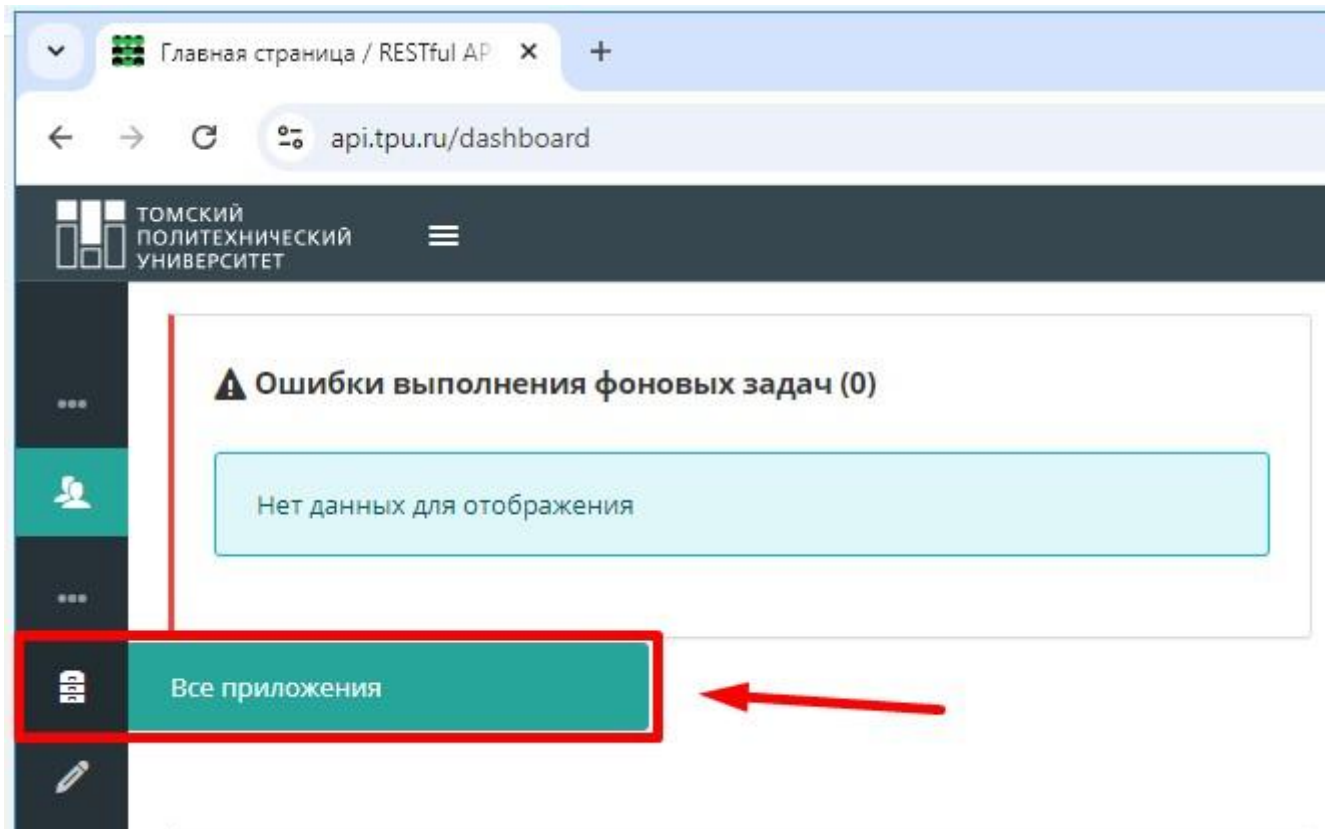
Для начала работы надо авторизоваться в приложение API TPU (<https://api.tpu.ru>)

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 \*

Краткое название приложения \*

Описание приложения

Тип приложения \* ✕ ▾

Ссылка на приложение

Публичный ключ приложения

Перед отправкой ключа вы можете сгенерировать новый или использовать существующий

Заполните форму и нажмите «Сохранить»  
\* Поля обязательные для заполнения

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	☰

- Просмотр
- Редактировать
- Настройки
- Платформа
- Личности
- Разработчики
- Ключи доступа SSL
- Матрица ролей
- Ограничения запрос...
- 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

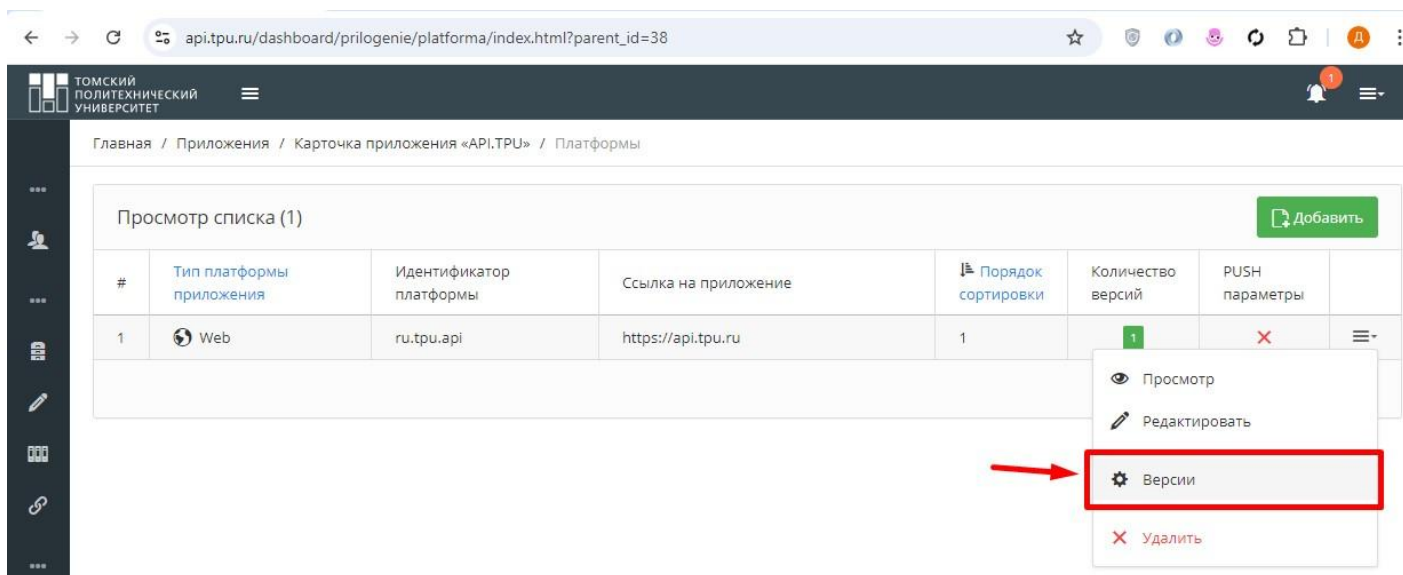
Закрыть

# Добавления 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



The screenshot shows a web interface for the Tomsk Polytechnic University API. The breadcrumb trail is: Главная / Приложения / Карточка приложения «API.TPU» / Платформы. The main content area displays a table titled "Просмотр списка (1)" with a "Добавить" button. The table has the following columns: #, Тип платформы приложения, Идентификатор платформы, Ссылка на приложение, Порядок сортировки, Количество версий, PUSH параметры, and a menu icon. The first row contains: 1, Web, ru.tpu.api, https://api.tpu.ru, 1, 1, X, and a menu icon. A context menu is open over the menu icon, with options: "Просмотр", "Редактировать", "Версии" (highlighted with a red box and a red arrow), and "Удалить".


#	Тип платформы приложения	Идентификатор платформы	Ссылка на приложение	Порядок сортировки	Количество версий	PUSH параметры	
1	Web	ru.tpu.api	https://api.tpu.ru	1	1	X	⋮

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

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	<a href="#">Никиткина Ольга Александровна</a>
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.

api.tpu.ru/dashboard/prilogenie/platforma/version/signature.html?parent\_id=81&id=101#

ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Главная / Приложения / Карточка приложения «Сервер RESTFul API TPU» / Платформы / Просмотр платформы «Web» / Версии платформы / Просмотр версии платформы «2.0» / Расчет подписи параметров

### Расчет подписи параметров

Название параметра *	Значение *
id	123
ip	123.100.10.15

+ Добавить параметр

✓ Рассчитать контрольную сумму

### Результат подсчета

sig	9e5a6f4459cb0ec0d9f8643cf920811f
-----	----------------------------------

“ 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()
```