API. Инструкция по интеграции

Версия 1.2
от 12 ноября 2020

Содержание

Используемые термины и сокращения

Аутентификация - процесс установления, действительно ли пользователь является тем, за кого себя выдает.

OTP - (англ. One-Time Password) одноразовый пароль - пароль, действительный только для одного сеанса аутентификации.

Токен - физическое или виртуальное устройство для генерации одноразового пароля.

Ресурс - объект, который должен быть защищен двухфакторной аутентификацией.

Общие сведения

Команда Protectimus предоставляет набор инструментов, которые помогут легко интегрироваться с любым ресурсом или проектом. SDK для популярных языков программирования, Java, Ruby, Pyton, .Net, PHP[1], значительно экономят время и силы на интеграцию с нашим решением, мы рекомендуем использовать именно их. Если же у Вас возникли особые потребности, Вы можете напрямую использовать API, описание которого представлено в этом документе.

Наш API построен в соответствии с принципами REST. Данные передаются в формате XML или JSON. Значение параметров одинаково для обоих форматов. По умолчанию ответ передается в формате XML.

Предварительные действия

При использовании **сервиса** Protectimus, API необходимо активировать. Для активации API Вам требуется активировать выбранный тарифный план в нашей системе (https://service.protectimus.com/pricing). После этого с Вашего счета один раз в день будет списываться плата за использование сервиса. Вы можете в любой момент приостановить использование системы и снятие средств, деактивировав тарифный план, но учтите, что и API в таком случае также будет отключено.

Авторизация

API Protectimus доступен только авторизованным пользователям. В нашем решении используется Basic-аутентификация. В качестве имени пользователя используется логин администратора, который выполняет запрос, а в качестве пароля - токен аутентификации.
Токен аутентификации является хэшом от строки, состоящей из следующих элементов: <ApiKey>:<YYYYMMDD>:<HH>, где:

- *ApiKey* - ключ API, он различный для каждого администратора, доступен и может быть изменен на странице управления профилем https://service.protectimus.com/profile
- *YYYYMMDD* - текущая дата в указанном формате
- *HH* - время UTC в формате HH (только часы в 24-х часовом формате, без минут и секунд)

Пример: В профиле адмиминистратора указан ApiKey - MySecureApiKey, дата - 30 января 2014 года, время по UTC 17:42.

Строка, от которой следует брать хэш: MySecureApiKey:20140130:17

Hash SHA256 для данного текста:

62704fb3a9dcf7b5b3cf7bda6ac9d0b0aa37c6fce8d0fae6b466c91ba68894f5

[1] На момент написания данного документа существуют клиенты для Java, Python, PHP

Выполнение запросов

Запросы к Protectimus API передаются по протоколу HTTPS.

Формат запроса:

```
<HTTP-метод>
https://api.protectimus.com/api/v<версия_API>/<раздел_API>/<метод_API>.<формат_ответа>
```

Указанные параметры имеют следующее значение:

- <HTTP-метод> - метод, который характерен для текущего запроса
- <версия_API> - версия API, которую Вы хотите использовать. На сегодня доступна только первая версия, соответственно, эта часть запроса будет выглядеть как: “v1”
- <раздел_API> - раздел, к которому относится вызываемый Вами метод. Доступны следующие разделы: auth-service, resource-service, token-service, user-service. Описание методов API разбито на разделы, к которым они относятся.
- <метод_API> - вызываемый Вами метод
- <формат_ответа> - формат, в котором Вы хотите получить ответ: XML или JSON. По умолчанию установлен формат XML.

В случае возникновения ошибки обработка запроса прекращается и возвращается сообщение об ошибке. Список ошибок и их описания приведены в разделе Сообщение об ошибках.
Большинство действий доступных через АПИ, доступны в сервисе через наш графический интерфейс. Возможно, ознакомление с ним поможет Вам лучше понять принцип работы нашей системы.

Описание методов API

Получение общих сведений

Общий адрес раздела:

```
https://api.protectimus.com/api/v1/auth-service/
```

GET balance

Получение сведений о текущем балансе средств на счету клиента. Эта информация доступна только главному администратору системы.

URL:

```
https://api.protectimus.com/api/v1/auth-service/balance
```

Входные данные
Отсутствуют.

Выходные данные
Возвращает текущий баланс средств на счету.

Структура выходных данных для формата XML:

```
<responseHolder>
   <response>
      <balance>{баланс_клиента}</balance>
   </response>
   <status>OK</status>
</responseHolder>
```

Структура выходных данных для формата JSON:

```
{
    "responseHolder":{
        "response": {
            "balance":"{баланс_клиента}"
        },
        "status":"OK"
    }
}
```

Описание параметров:

Параметр	Тип	Значение
balance	число	Текущий баланс средств на счету в долларах США.

Процесс аутентификации

Этот раздел API предназначен для аутентификации пользователей и токенов на Ваших ресурсах.

Общий адрес раздела:

```
https://api.protectimus.com/api/v1/auth-service/
```

В зависимости от выбранной Вами модели пользователь может быть аутентифицирован по статическому паролю, по одноразовому паролю или же по статическому и одноразовому паролю одновременно. Для того, чтобы пользователь или токен мог пройти аутентификацию - он должен быть назначен на запрашиваемый ресурс (а если выполняется аутентификация пользователя и токена на ресурсе одновременно, то пользователь должен быть назначен на ресурс вместе с токеном).

Обратите внимание
1. Если пользователь или токен не пройдет аутентификацию, то для него будет увеличен счетчик неуспешных попыток аутентификации. При превышении порогового значения количества неверных попыток для указанного ресурса пользователь будет заблокирован. Разблокировать пользователя можно через веб-интерфейс либо с помощью API (метод редактирования пользователя).

При успешной аутентификации счетчик неуспешных попыток обнулится, если он не превысил допустимый предел для ресурса и пользователь еще не был заблокирован.

2. Токены ROTECTIMUS SMS, PROTECTIMUS MAIL, PROTECTIMUS ULTRA, PROTECTIMUS BOT требуют вызова метода POST prepare, чтобы подготовить процесс аутентификации. Подробнее об этом можно прочитать в описании самого метода.

POST prepare

Для некоторых видов токенов, таких как Protectimus SMS, Protectimus BOT, Protectimus MAIL, Protectimus ULTRA, OATH_OCRA требуется выполнение определенных действий перед их аутентификацией. Этот метод должен быть вызван для SMS, MAIL, BOT токенов, чтобы отправить пользователю одноразовый пароль, а для Protectimus ULTRA и OATH_OCRA чтобы получить запрос (challenge), который пользователь должен будет ввести в токен для генерации пароля по алгоритму Challenge-Response. Для других видов токенов вызов этого метода не требуется.

Для указания токена, который должен быть подготовлен к аутентификации Вы можете передать один из следующих параметров: tokenId, userId, userLogin. Используйте параметр, который более удобен для Вас, но помните: токен должен быть назначен на текущий ресурс вместе с пользователем для правильного определения токена по параметру userId или userLogin.

URL:

```
https://api.protectimus.com/api/v1/auth-service/prepare
```

Входные данные

Параметр	Обязательный параметр	Описание
resourceId	да, если не указан параметр resourceName	Идентификатор ресурса, на котором токен должен быть подготовлен к аутентификации
resourceName	да, если не указан параметр resourceId	Имя ресурса, на котором токен должен быть подготовлен к аутентификации
tokenId	да, если не указан параметр userId или userLogin	Идентификатор токена, который должен быть подготовлен к аутентификации
userId	да, если не указан параметр tokenId или userLogin	Идентификатор пользователя, которому назначен токен. Для использования этого параметра токен должен быть назначен на ресурс вместе с этим пользователем
userLogin	да, если не указан параметр userId или tokenId	Логин пользователя. Для использования этого параметра пользователю должен быть назначен токен и “токен с пользователем” должен быть назначен на ресурс
templateIdOrName	нет	Идентификатор или имя шаблона который будет использоваться для доставки OTP

authType	нет, по умолчанию OTP	Тип аутентификации, нужен для выбора способа аутентификации токена PROTECTIMUS BOT. Допустимые значения : OTP, INTERACTIVE
message	да, если значение параметра authType - INTERACTIVE	Сообщение, которое будет отображаться при INTERACTIVE аутентификации

Выходные данные
Возвращает строку-вопрос (challenge) для токена Protectimus ULTRA, или сообщение об успешном выполнении операции для Protectimus MAIL, BOT и SMS токенов

Структура выходных данных для формата XML:

```
<responseHolder>
  <response>
    <challenge>{challenge_для_CR_токена}</challenge>
    <tokenName>{имя_токена}</tokenName>
    <tokenType>{тип_токена}</tokenType>
    <authId>{идентификатор_аутентификации}</authId>
  </response>
  <status>OK</status>
</responseHolder>
```

Структура выходных данных для формата JSON:

```
{
  "responseHolder" : {
    "response" : {
      "challenge" : {challenge_для_CR_токена},
      "tokenName": "{имя_токена}",
      "tokenType": "{тип_токена}",
      "authId": "{идентификатор_аутентификации}"
    },
    "status" : "OK"
  }
}
```

Описание параметров:

Параметр	Тип	Описание
challenge	число	Число-вопрос, который пользователь должен ввести в токен, на основании которого токен сгенерирует ответ
tokenName	строка	Имя токена который был подготовлен для аутентификации

tokenType	строка	Тип токена который был подготовлен для аутентификации
authId	строка	Идентификатор аутентификации необходим для дальнейшего сопоставления пользователя с результатом аутентификации. Используется для INTERACTIVE аутентификации PROTECTIMUS BOT

Обратите внимание

- Указанный для подготовки токен может иметь тип, который не требует этого действия. Этот метод необходим только для PROTECTIMUS SMS, BOT, BOT, MAIL и ULTRA токенов. Если же он будет вызван для другого типа токена, то будет возвращен код ошибки 6001, что означает неверно указанный параметр.
- Параметр challenge в ответе характерен только для токена PROTECTIMUS ULTRA, который работает по алгоритму OCRA. Для SMS, BOT, BOT и MAIL токенов позитивным ответом будет стандартное сообщение об успешном выполнении операции.
- Указанный токен может быть не назначен на указанный ресурс, в таком случае Вы получите код ошибки 5002 с описанием проблемы.
- Параметр authId в выходных данных характерен только для токена PROTECTIMUS BOT при интерактивной аутентификации.

POST prepare-user

Этот метод предназначен для быстрой подготовки пользователя к аутентификации на указанном ресурсе, а именно:

1. Создаётся пользователь (в случае его отсутствия).
2. Создаётся SMS или MAIL токен.
3. Токен с пользователем назначаются на указанный ресурс.
4. Пользователю отправляется OTP.

Используется только для PROTECTIMUS SMS, PROTECTIMUS MAIL токенов!

URL:

```
https://api.protectimus.com/api/v1/auth-service/prepare-user
```

Входные данные

Параметр	Обязательный параметр	Описание
resourceId	да, если не указан параметр resourceName	Идентификатор ресурса, на котором токен должен быть подготовлен к аутентификации
resourceName	да, если не указан параметр resourceId	Имя ресурса, на котором токен должен быть подготовлен к аутентификации
userLogin	да	Логин пользователя.
emailOrPhoneNumber	да	Email или номер телефона пользователя на который будет доставлен OTP
templateIdOrName	нет	Идентификатор или имя шаблона который будет использоваться для доставки OTP

Выходные данные
Возвращает сообщение об успешном выполнении операции для Protectimus MAIL, SMS токенов

Структура выходных данных для формата XML:

```
<responseHolder>
  <response>
    <tokenType>{тип_токена}</tokenType>
  </response>
  <status>OK</status>
</responseHolder>
```

Структура выходных данных для формата JSON:

```
{
  "responseHolder" : {
    "response" : {
      "tokenType": "{тип_токена}"
    },
    "status" : "OK"
  }
}
```

Описание параметров:

Параметр	Тип	Описание
tokenName	строка	Имя токена который был подготовлен для аутентификации
tokenType	строка	Тип токена который был подготовлен для аутентификации

POST authenticate/token

Выполняет аутентификацию указанного токена на указанном ресурсе.

URL:

```
https://api.protectimus.com/api/v1/auth-service/authenticate/token
```

Входные данные

Параметр	Обязательный параметр	Описание
resourceId	да, если не указан resourceName	Идентификатор ресурса, на котором токен должен быть аутентифицирован
resourceName	да, если не указан resourceId	Имя ресурса, на котором токен должен быть аутентифицирован
tokenId	да	Идентификатор токена, который должен быть аутентифицирован
otp	да	Одноразовый пароль, введенный пользователем
ip	частично	IP-адрес пользователя. Должен быть указан, чтобы выполнялась проверка по гео-фильтру.

Выходные данные

Возвращает true если аутентификация прошла успешно или false если токен не прошел проверку.

Структура выходных данных для формата XML:

```
<responseHolder>
  <response>
    <result>{результат_аутентификации}</result>
  </response>
  <status>OK</status>
</responseHolder>
```

Структура выходных данных для формата JSON:

```
{
  "responseHolder" : {
    "response" : {
      "result" : {результат_аутентификации}
    },
    "status" : "OK"
  }
}
```

Описание параметров

Параметр	Тип	Описание
result	логический	Результат аутентификации токена на указанном ресурсе.

Обратите внимание

- Проверяемый токен должен быть назначен на указанный ресурс самостоятельно либо с пользователем.

POST authenticate/user-password

Выполняет аутентификацию пользователя по статическому паролю на указанном ресурсе. Чтобы выполнялась аутентификация с использованием этого метода пользователь должен быть назначен на указанный ресурс и у пользователя должен быть установлен пароль, по которому он будет аутентифицироваться.

URL:

```
https://api.protectimus.com/api/v1/auth-service/authenticate/user-password
```

Входные данные

Параметр	Обязательный параметр	Описание
resourceId	да, если не указан resourceName	Идентификатор ресурса, на котором пользователь должен быть аутентифицирован
resourceName	да, если не указан resourceId	Имя ресурса, на котором пользователь должен быть аутентифицирован
userId	да, если не указан userLogin	Идентификатор пользователя, который должен быть аутентифицирован
userLogin	да, если не указан userId	Логин пользователя, который должен быть аутентифицирован
pwd	да	пароль, который ввел пользователь
ip	частично	IP-адрес пользователя. Должен быть указан, чтобы выполнялась проверка по гео-фильтру.

Выходные данные

Возвращает true если аутентификация прошла успешно или false если пользователь не прошел проверку.

Структура выходных данных для формата XML:

```
<responseHolder>
  <response>
    <result>{результат_аутентификации}</result>
  </response>
  <status>OK</status>
</responseHolder>
```

Структура выходных данных для формата JSON:

```
{
  "responseHolder" : {
    "response" : {
      "result" : {результат_аутентификации}
    },
    "status" : "OK"
  }
}
```

Описание параметров:

Параметр	Тип	Описание
result	логический	Результат аутентификации пользователя по паролю на указанном ресурсе.

Обратите внимание

- Проверяемый пользователь должен быть назначен на указанный ресурс
- У проверяемого пользователя должен присутствовать пароль в системе

Если описанное выше не будет выполнено - Вы получите код ошибки 5002 с описанием возникшей проблемы.

POST authenticate/user-token

Выполняет аутентификацию пользователя по одноразовому паролю на указанном ресурсе. Для этого у пользователя должен существовать токен, вместе с которым он должен быть назначен на указанный ресурс.

URL:

```
https://api.protectimus.com/api/v1/auth-service/authenticate/user-token
```

Входные данные

Параметр	Обязательный параметр	Описание
resourceId	да, если не указан resourceName	Идентификатор ресурса, на котором пользователь должен быть аутентифицирован
resourceName	да, если не указан resourceId	Имя ресурса, на котором пользователь должен быть аутентифицирован
userId	да, если не указан userLogin	Идентификатор пользователя, который должен быть аутентифицирован
userLogin	да, если не указан userId	Логин пользователя, который должен быть аутентифицирован
otp	да	Одноразовый пароль, который ввел пользователь
ip	частично	IP-адрес пользователя. Должен быть указан, чтобы выполнялась проверка по гео-фильтру.

Выходные данные

Возвращает true если аутентификация прошла успешно или false если пользователь не прошел проверку.

Структура выходных данных для формата XML:

```
<responseHolder>
  <response>
    <result>{результат_аутентификации}</result>
  </response>
  <status>OK</status>
</responseHolder>
```

Структура выходных данных для формата JSON:

```
{
  "responseHolder" : {
    "response" : {
      "result" : {результат_аутентификации}
    },
    "status" : "OK"
  }
}
```

Описание параметров

Параметр	Тип	Описание
result	логический	Результат аутентификации пользователя по токену на указанном ресурсе.

Обратите внимание

- У пользователя должен присутствовать токен
- Пользователь должен быть назначен на ресурс вместе с токеном.

При несоблюдении описанных выше случаев Вы получите код ошибки 5002 с описанием возникшей проблемы.

POST authenticate/user-password-token

Выполняет аутентификацию пользователя по статическому и одноразовому паролю на указанному ресурсе. Для этого пользователь должен быть назначен на ресурс вместе с токеном, а также у него должен быть задан статический пароль. Если токен пользователя будет отключен, то проверка OTP выполняться не будет, в таком случае будет проверен только статический пароль и проходит ли пользователь фильтры, если они существуют.

URL:

```
https://api.protectimus.com/api/v1/auth-service/authenticate/user-password-token
```

Входные данные

Параметр	Обязательный параметр	Описание
resourceId	да, если не указан resourceName	Идентификатор ресурса, на котором пользователь должен быть аутентифицирован
resourceName	да, если не указан resourceId	Имя ресурса, на котором пользователь должен быть аутентифицирован
userId	да, если не указан userLogin	Идентификатор пользователя, который должен быть аутентифицирован
userLogin	да, если не указан userId	Логин пользователя, который должен быть аутентифицирован
otp	да	Одноразовый пароль, который ввел пользователь
pwd	да	Пароль, который предоставил пользователь
ip	частично	IP-адрес пользователя. Должен быть указан, чтобы выполнялась проверка по гео-фильтру.

Выходные данные
Возвращает true если аутентификация прошла успешно или false если пользователь не прошел проверку.

Структура выходных данных для формата XML:

```
<responseHolder>
  <response>
    <result>{результат_аутентификации}</result>
  </response>
  <status>OK</status>
</responseHolder>
```

Структура выходных данных для формата JSON:

```
{
  "responseHolder" : {
    "response" : {
      "result" : {результат_аутентификации}
    },
    "status" : "OK"
  }
}
```

Описание параметров

Параметр	Тип	Описание
result	логический	Результат аутентификации пользователя по токену и паролю на указанном ресурсе.

Обратите внимание

- У пользователя должен быть назначен токен и присутствовать статический пароль.
- Пользователь должен быть назначен на ресурс вместе с токеном.

При несоблюдении описанных выше случаев Вы получите код ошибки 5002 с описанием возникшей проблемы.

Управление ресурсами (проектами)

Ресурс является средством группировки пользователей и предоставляет гибкие возможности делегирования полномочий. Под ресурсом следует понимать веб-проект, портал, приложение или отдел Ваших сотрудников. Главный администратор может добавить других администраторов в систему и назначить их на определенные ресурсы. Обычным администраторам доступны только действия в пределах ресурса, на которые они назначены, но они могут видеть всех пользователей и все токены, которые существуют в системе, независимо от того, назначены они на ресурсы, которыми управляет администратор, или нет.

Общий адрес раздела:

```
https://api.protectimus.com/api/v1/resource-service/resources
```

GET resources

Позволяет получить список Ваших ресурсов (до 10 элементов начиная с заданного сдвига). По умолчанию сдвиг равен нулю.

URL:

```
https://api.protectimus.com/api/v1/resource-service/resources
```

Входные данные

Параметр	Обязательный параметр	Описание
start	нет	Сдвиг, с которого должен начаться список ресурсов. По умолчанию равен 0.

Выходные данные
Список ресурсов авторизованного клиента

Структура выходных данных для формата XML:

```
<responseHolder>
  <response>
    <resources>
      <resource>
        <creatorId>{идентификатор_создателя_ресурса}</creatorId>
        <creatorUsername>{логин_создателя_ресурса}</creatorUsername>
        <failedAttemptsBeforeLock>
            {количество_неуспешных_попыток_аутентификации_до_блокировки}
        </failedAttemptsBeforeLock>
        <id>{идентификатор_ресурса}</id>
        <name>{имя_ресурса}</name>
      </resource>
       . . .
    </resources>
  </response>
  <status>OK</status>
</responseHolder>
```

Структура выходных данных для формата JSON:

```
{

   "responseHolder":
     {
        "response":
         {"resources":
            [
```

```
            {
                "creatorId":{идентификатор_создателя_ресурса},
                "creatorUsername":"{логин_создателя_ресурса}",

"failedAttemptsBeforeLock":{количество_неуспешных_попыток_аутентификации_до_блокировки},
                "id":{идентификатор_ресурса},
                "name":"{имя_ресурса}"
             },
              . . .
          ]
       },
   "status":"OK"
  }
}
```

Описание параметров

Параметр	Тип	Описание
creatorId	число	Идентификатор администратора, который создал ресурс
creatorUsername	строка	Логин администратора, который создал ресурс
failedAttemptsBeforeLock	число	Количество неуспешных попыток аутентификации, при превышении которого пользователь будет заблокирован.
id	число	Идентификатор ресурса
name	строка	Имя ресурса

Обратите внимание

- Вы ошибочно можете указать размер сдвига больше, чем количество ресурсов у клиента. В таком случае будет возвращен пустой список

GET resources/quantity

Позволяет получить информацию о количестве ресурсов администратора, от имени которого выполняется запрос.

URL:

```
https://api.protectimus.com/api/v1/resource-service/resources/quantity
```

Входные данные
Отсутствуют.

Выходные данные
Возвращает количество ресурсов (проектов) авторизованного клиента.

Структура выходных данных для формата XML:

```
<responseHolder>
   <response>
      <quantity>{количество_ресурсов}</quantity>
   </response>
   <status>OK</status>
</responseHolder>
```

Структура выходных данных для формата JSON:

```
{
   "responseHolder":   {
      "response": {
         "quantity": количество_ресурсов
       },
       "status":"OK"
   }
}
```

Описание параметров

Параметр	Тип	Описание
quantity	число	Количество ресурсов авторизованного клиента

POST resources

Создание нового ресурса (проекта).

URL:

```
https://api.protectimus.com/api/v1/resource-service/resources
```

Входные данные

Параметр	Обязательный параметр	Описание
resourceName	да	Имя создаваемого ресурса
failedAttemptsBeforeLock	нет	Количество не успешных попыток аутентификации, при превышении которого пользователь будет заблокирован. Значение этого параметра должно быть в диапазоне от 3 до 10. Если этот параметр не указан, по умолчанию он будет равным 5 попыткам.

Выходные данные
Возвращает идентификатор созданного ресурса.

Структура выходных данных для формата XML:

```
<responseHolder>
  <response>
    <id>{идентификатор_ресурса}</id>
  </response>
  <status>OK</status>
</responseHolder>
```

Структура выходных данных для формата JSON:

```
{
 "responseHolder" : {
  "response" : {
    "id" : {идентификатор_ресурса}
  },
  "status" : "OK"
 }
}
```

Описание параметров:

Параметр	Тип	Описание
id	число	Идентификатор созданного ресурса

Обратите внимание

- Количество доступных для создания ресурсов (проектов) ограничивается выбранным Вами тарифным планом. Если Вам необходимо создать больше ресурсов - выберите желаемое количество ресурсов настроив собственный тарифный план. Действия по выбору тарифного плана доступны только главному администратору.

GET resources/{id}

Позволяет получить информацию о конкретном ресурсе клиента.

URL:

```
https://api.protectimus.com/api/v1/resource-service/resources/{id}
```

Входные данные

Параметр	Обязательный параметр	Описание
id	да	Идентификатор ресурса, о котором необходимо получить информацию.

Выходные данные
Возвращает информацию об указанном ресурсе клиента.

Структура выходных данных для формата XML:

```
<responseHolder>
  <response>
    <resource>
        <creatorId>{идентификатор_создателя_ресурса}</creatorId>
        <creatorUsername>{логин_создателя_ресурса}</creatorUsername>
        <failedAttemptsBeforeLock>
            {количество_неуспешных_попыток_аутентификации_до_блокировки}
        </failedAttemptsBeforeLock>
        <id>{идентификатор_ресурса}</id>
        <name>{имя_ресурса}</name>
      </resource>
  </response>
  <status>OK</status>
</responseHolder>
```

Структура выходных данных для формата JSON:

```
{
    "responseHolder":
     {
        "response":
         {"resource":
            [
               {
                   "creatorId":{идентификатор_создателя_ресурса},
                   "creatorUsername":"{логин_создателя_ресурса}",
```

```
"failedAttemptsBeforeLock":{количество_неуспешных_попыток_аутентификации_до_блокировки},
                    "id":{идентификатор_ресурса},
                    "name":"{имя_ресурса}"
                }
            ]
        },
    "status":"OK"
  }
}
```

Описание параметров

Параметр	Тип	Описание
creatorId	число	Идентификатор администратора, который создал ресурс
creatorUsername	строка	Логин администратора, который создал ресурс
failedAttemptsBeforeLock	число	Количество неуспешных попыток аутентификации, при превышении которого пользователь будет заблокирован.
id	число	Идентификатор ресурса
name	строка	Имя ресурса

Обратите внимание

- Вы можете указать неверный идентификатор, или идентификатор ресурса, который Вам не принадлежит. В таком случае Вы получите сообщение об ошибке.

PUT resources/{id}

Позволяет редактировать информацию о Вашем ресурсе (проекте).

URL:

```
https://api.protectimus.com/api/v1/resource-service/resources/{id}
```

Входные данные

Параметр	Обязательный параметр	Описание
id	да, если не указано старое имя ресурса	Идентификатор ресурса, информацию о котором Вы желаете изменить
resourceName	да, если не указан id, или если необходимо изменить имя ресурса	Старое либо новое имя ресурса. Если не указан идентификатор ресурса, то поиск ресурса будет вестись по указанному имени. Чтобы изменить имя ресурса укажите его идентификатор (id) и передайте новое имя в этом параметре.
failedAttemptsBeforeLock	нет	Количество неуспешных попыток аутентификации пользователя, после которого он будет заблокирован. Разрешенный диапазон значений: от 3-х до 10-ти. Если этот параметр не будет указан - он останется прежним для этого ресурса.

Выходные данные
Возвращает информацию об отредактированном ресурсе.

Структура выходных данных для формата XML:

```
<responseHolder>
  <response>
    <resource>
      <creatorId>{идентификатор_создателя}</creatorId>
      <creatorUsername>{имя_создателя}</creatorUsername>
      <failedAttemptsBeforeLock>
         {новый_лимит_неуспешных_аутентификаций}
      </failedAttemptsBeforeLock>
      <id>{идентификатор_ресурса}</id>
      <name>{новое_имя_ресурса}</name>
    </resource>
  </response>
  <status>OK</status>
</responseHolder>
```

Структура выходных данных для формата JSON:

```
{
  "responseHolder" : {
    "response" : {
      "resource" : {
        "creatorId" : {идентификатор_создателя},
        "creatorUsername" : "{имя_создателя}",
        "failedAttemptsBeforeLock" : {новый_лимит_неуспешных_аутентификаций},
        "id" : {идентификатор_ресурса},
        "name" : "{новое_имя_ресурса}"
      }
    },
    "status" : "OK"
  }
}
```

Описание параметров

Параметр	Тип	Описание
creatorId	число	Идентификатор администратора, который создал ресурс
creatorUsername	строка	Логин администратора, который создал ресурс
failedAttemptsBeforeLock	число	Количество неуспешных попыток аутентификации, при превышении которого пользователь будет заблокирован.
id	число	Идентификатор ресурса
name	строка	Имя ресурса

Обратите внимание

- Вы можете указать неверный идентификатор ресурса, что приведет к нежелательному изменению информации о другом Вашем ресурсе.

DELETE resources/{id}

Удаление Вашего ресурса.

Удалить ресурс может только администратор, который его создал, либо главный администратор.

URL:

```
https://api.protectimus.com/api/v1/resource-service/resources/{id}
```

Входные данные

Параметр	Обязательный параметр	Описание
id	да	Идентификатор удаляемого ресурса

Выходные данные
Возвращает информацию об удаленном ресурсе.

Структура выходных данных для формата XML:

```
<responseHolder>
  <response>
    <resource>
      <creatorId>{идентификатор_создателя}</creatorId>
      <creatorUsername>{имя_создателя}</creatorUsername>

<failedAttemptsBeforeLock>{лимит_неуспешных_аутентификаций}</failedAttemptsBeforeLoc
k>
      <id>{идентификатор_ресурса}</id>
      <name>{имя_ресурса}</name>
    </resource>
  </response>
  <status>OK</status>
</responseHolder>
```

Структура выходных данных для формата JSON:

```
{
  "responseHolder" : {
    "response" : {
      "resource" : {
        "creatorId" : {идентификатор_создателя},
        "creatorUsername" : "{имя_создателя}",
        "failedAttemptsBeforeLock" : {лимит_неуспешных_аутентификаций},
        "id" : {идентификатор_ресурса},
        "name" : "{имя_ресурса}"
```

```
    }
   },
   "status" : "OK"
  }
}
```

Описание параметров

Параметр	Тип	Описание
creatorId	число	Идентификатор администратора, который создал ресурс
creatorUsername	строка	Логин администратора, который создал ресурс
failedAttemptsBeforeLock	число	Количество неуспешных попыток аутентификации, при превышении которого пользователь будет заблокирован.
id	число	Идентификатор ресурса
name	строка	Имя ресурса

Обратите внимание

- У Вас может не быть прав на удаление ресурса, если Вы не являетесь его создателем или главным пользователем системы.

GET resources/{id}/webhook

Используйте этот метод для получения статуса webhook по заданному ID ресурса.

URL:

```
https://api.protectimus.com/api/v1/resource-service/resources/{id}/webhook
```

Входные данные

Параметр	Обязательный параметр	Описание
id	да	Идентификатор ресурса

Выходные данные
Информацию о статусе webhook указанном в ресурсе клиента.

Структура выходных данных для формата XML:

```
<responseHolder>
   <response>
      <url>{url}</url>
      <isWebhookCertified>false</isWebhookCertified>
   </response>
   <status>OK</status>
</responseHolder>
```

Структура выходных данных для формата JSON:

```
{
   "responseHolder": {
      "response": {
         "url": {url},
         "isWebhookCertified": false
      },
      "status": "OK"
   }
}
```

Описание параметров

Параметр	Тип	Описание
url	строка	Webhook URL установленный для ресурса. Может быть пустым если webhook не установлен
isWebhookCertified	логический	True, если для проверки webhook сертификата был предоставлен специальный сертификат

Обратите внимание

- Вы можете указать неверный идентификатор, или идентификатор ресурса, который Вам не принадлежит. В таком случае Вы получите сообщение об ошибке.

PUT resources/{id}/webhook

Позволяет добавлять или изменять webhook по заданному id ресурса.

При определенных событиях связанных с ресурсами вам будет отправлено уведомление, содержащее детальную информации в JSON формате (пример ниже). Для отправки уведомлений используется POST запрос на указанный webhook. В случае неудачного запроса попытка повторится несколько раз.

В настоящее время webhook используется для получения результата интерактивной (INTERACTIVE) аутентификации. Интерактивная аутентификация поддерживается токеном PROTECTIMUS_BOT. Альтернативный вариант получения уведомлений является использования метода GET resources/{id}/updates работающего с помощью механизм long polling.

URL:

```
https://api.protectimus.com/api/v1/resource-service/resources/{id}/webhook
```

Входные данные

Параметр	Обязательный параметр	Описание
id	да	Идентификатор ресурса
url	да	HTTPS url для отправки уведомлений по заданному ресурсу.
certificate	нет	Сертификат открытого ключа удостоверяет принадлежность открытого ключа указанному webhook. Предоставленный сертификат должен быть в кодировке PEM (ASCII BASE64), файл pem должен содержать только открытый ключ начинаться с "-----BEGIN CERTIFICATE----- " и заканчиваться "-----END CERTIFICATE----- "

Выходные данные
Сообщение об успешном выполнении операции

Структура уведомлений которые будут приходить на заданный webhook:

```
{
   "update": {
      "authentication": {
         "id": {идентификатор_аутентификации},
         "hash": {хэш_для_проверки},
         "date": {дата_аутентификации},
         "accepted": {результат}
      }
   }
}
```

Описание параметров

Параметр	Тип	Описание
id	строка	Идентификатор аутентификации необходим для сопоставления результа аутентификации с пользователем.
hash	строка	Хэш необходим для проверки целостности входящих данных. Для генерации используется алгоритм SHA256. Хэш берется от строки состоящей из следующих элементов : <id>:<date>:<accepted>:<ApiKey>
date	число	Дата аутентификации (Unix-time)
accepted	логический	Результат аутентификации

Обратите внимание

- ApiKey - ключ API, он различный для каждого администратора, доступен и может быть изменен на странице управления профилем https://service.protectimus.com/profile
- Вы можете указать неверный идентификатор, или идентификатор ресурса, который Вам не принадлежит. В таком случае Вы получите сообщение об ошибке.

DELETE resources/{id}/webhook

Удаление webhook по заданному id ресурса.

URL:

```
https://api.protectimus.com/api/v1/resource-service/resources/{id}/webhook
```

Входные данные

Параметр	Обязательный параметр	Описание
id	да	Идентификатор ресурса

Выходные данные
Сообщение об успешном выполнении операции

Обратите внимание

- Вы можете указать неверный идентификатор, или идентификатор ресурса, который Вам не принадлежит. В таком случае Вы получите сообщение об ошибке.

GET resources/{id}/updates

Используйте этот метод для получения уведомлений о событиях связанных с указанным ресурсом.

На данный момент реализована только доставка событий о результатах интерактивной (INTERACTIVE) аутентификации. Интерактивную аутентификацию поддерживают PROTECTIMUS_BOT токен.

Метод использует long polling механизм что позволяет обрабатывать уведомления без указания webhook с помощью метода PUT resources/{id}/webhook.

URL:

```
https://api.protectimus.com/api/v1/resource-service/resources/{id}/updates
```

Входные данные

Параметр	Обязательный параметр	Описание
id	да	Идентификатор ресурса, о котором необходимо получить информацию.

Выходные данные
Возвращает новые события связанные с указанным ресурсом.

Структура выходных данных для формата XML:

```
<responseHolder>
   <response>
      <updates>
         <update>
            <authentication>
               <id>{идентификатор_аутентификации}</id>
               <hash>{хэш_для_проверки}</hash>
               <date>{дата_аутентификации}</date>
               <accepted>{результат}</accepted>
            </authentication>
         </update>
         ...
      </updates>
   </response>
   <status>OK</status>
</responseHolder>
```

Структура выходных данных для формата JSON:

```
{
```

```
  "responseHolder": {
     "response": {
        "updates": [
           {
              "authentication": {
                 "id": {идентификатор_аутентификации},
                 "hash": {хэш_для_проверки},
                 "date": {дата_аутентификации},
                 "accepted": {результат}
              }
           },
           ...
        ]
     },
     "status": "OK"
  }
}
```

Описание параметров

Параметр	Тип	Описание
id	строка	Идентификатор аутентификации необходим для сопоставления результа аутентификации с пользователем
hash	строка	Хэш необходимый для проверки подлинности уведомления. Для генерации используется алгоритм SHA256. Хэш берется от строки состоящей из следующих элементов : <id>:<date>:<accepted>:<ApiKey>
date	число	Дата аутентификации (Unix-time)
accepted	логический	Результат аутентификации

Обратите внимание

- ApiKey - ключ API, он различный для каждого администратора, доступен и может быть изменен на странице управления профилем https://service.protectimus.com/profile
- Вы можете указать неверный идентификатор, или идентификатор ресурса, который Вам не принадлежит. В таком случае Вы получите сообщение об ошибке.

POST assign/user

Выполняет назначение пользователя с указанным идентификатором на указанный ресурс.

Используйте этот метод, если Вы хотите проводить аутентификацию этого пользователя на ресурсе используя только статические пароли. Если же Вы хотите проводить проверку статического и динамического пароля (OTP), либо только OTP - Вам следует назначить пользователя на ресурс вместе с токеном (метод assign/token-with-user) либо назначить токен (метод assign/token).

URL:

```
https://api.protectimus.com/api/v1/resource-service/assign/user
```

Входные данные

Параметр	Обязательный параметр	Описание
resourceId	да, если не указан параметр resourceName	Идентификатор ресурса, на который необходимо назначить пользователя
resourceName	да, если не указан параметр resourceId	Имя ресурса, на который необходимо назначить пользователя
userId	да, если не указан параметр userLogin	Идентификатор пользователя, которого необходимо назначить на ресурс
userLogin	да, если не указан параметр userId	Логин пользователя, которого необходимо назначить на ресурс

Выходные данные

Отсутствуют (в ответе лишь сообщение об успешном выполнении операции либо стандартные сообщения о возникших ошибках)

Обратите внимание

- В этом методе, как и некоторых других, где это позволено, для указания объектов, с которыми следует работать Вы можете использовать один из удобных для Вас параметров. Например, здесь для указания ресурса Вы можете передать его идентификатор в нашей системе resourceId либо имя ресурса resourceName, а для указания пользователя - ID пользователя userId либо его логин - userLogin.

POST assign/token

Назначение токена на ресурс.

После успешного выполнения этого метода Вы сможете проводить проверку OTP с этого токена на ресурсе без привязки к пользователю. Используйте этот метод, если Вы не хотите хранить информацию о пользователях в нашей системе. Но в таком случае, Вам будет доступна только проверка OTP (и пин-кода, если он назначен).

URL:

```
https://api.protectimus.com/api/v1/resource-service/assign/token
```

Входные данные

Параметр	Обязательный параметр	Описание
resourceId	да, если не указан параметр resourceName	Идентификатор ресурса, на который должен быть назначен токен
resourceName	да, если не указан параметр resourceId	Имя ресурса, на который необходимо назначить токен
tokenId	да	Идентификатор токена, который должен быть назначен на ресурс

Выходные данные

Сообщение об успешном выполнении операции

Обратите внимание

- Даже если этот токен назначен пользователю и пользователь назначен на этот же ресурс, Вы не сможете провести аутентификацию пользователя по OTP на этом ресурсе, если Вы явно не назначили пользователя на ресурс ВМЕСТЕ с токеном. Для этого используйте метод assign/user-token или выполните операцию назначения пользователя вместе с токеном через веб-интерфейс.

POST assign/user-token

Назначение пользователя и токена на ресурс.

После успешного выполнения этого метода становится доступным аутентифицировать пользователя на ресурсе по одноразовым паролям либо по комбинации одноразового и статического пароля.

URL:

```
https://api.protectimus.com/api/v1/resource-service/assign/user-token
```

Входные данные

Параметр	Обязательный параметр	Описание
resourceId	да, если не указан параметр resourceName	Идентификатор ресурса, на который должен быть назначен пользователь с токеном
resourceName	да, если не указан параметр resourceId	Имя ресурса, на который необходимо назначить пользователя с токеном
userId	да, если не указан параметр userLogin	Идентификатор пользователя, который должен быть добавлен на ресурс вместе с токеном
userLogin	да, если не указан параметр userId	Логин пользователя, который должен быть добавлен на ресурс вместе с токеном
tokenId	да	Идентификатор токена, который должен быть назначен на ресурс вместе с пользователем

Выходные данные

Сообщение об успешном выполнении операции

Обратите внимание

- Пользователь может иметь несколько токенов, но токен может быть только у одного пользователя.

POST assign/token-with-user

Назначение токена на ресурс вместе с пользователем, которому принадлежит данный токен.

Выполняет ту же операцию, что и метод assign/user-token, но не требует указания идентификатора пользователя, т.к. используется пользователь, которому назначен токен.

URL:

```
https://api.protectimus.com/api/v1/resource-service/assign/token-with-user
```

Входные данные

Параметр	Обязательный параметр	Описание
resourceId	да, если не указан параметр resourceName	Идентификатор ресурса, на который должен быть назначен токен с пользователем
resourceName	да, если не указан параметр resourceId	Имя ресурса, на который необходимо назначить токен с пользователем
tokenId	да	Идентификатор токена

Выходные данные
Сообщение об успешном выполнении операции

Обратите внимание

- Токен должен быть назначен пользователю для успешного выполнения этого метода.

POST unassign/user

Снимает назначение пользователя с ресурса.

После выполнения этого метода указанный пользователь не сможет быть аутентифицирован на данном ресурсе.

URL:

```
https://api.protectimus.com/api/v1/resource-service/unassign/user
```

Входные данные

Параметр	Обязательный параметр	Описание
resourceId	да, если не указан параметр resourceName	Идентификатор ресурса
resourceName	да, если не указан параметр resourceId	Имя ресурса
userId	да, если не указан параметр userLogin	Идентификатор пользователя
userLogin	да, если не указан параметр userId	Логин пользователя

Выходные данные
Сообщение об успешном выполнении операции

Обратите внимание

- Пользователь должен быть назначен на указанный ресурс.

POST unassign/token

Снимает назначение токена с ресурса. После выполнения этого метода указанный токен не сможет быть аутентифицирован на данном ресурсе.

URL:

```
https://api.protectimus.com/api/v1/resource-service/unassign/token
```

Входные данные

Параметр	Обязательный параметр	Описание
resourceId	да, если не указан параметр resourceName	Идентификатор ресурса
resourceName	да, если не указан параметр resourceId	Имя ресурса
tokenId	да	Идентификатор токена

Выходные данные
Сообщение об успешном выполнении операции

Обратите внимание

- Токен должен быть назначен на ресурс

POST unassign/token-with-user

Снимает назначение токена с ресурса вместе с пользователем, который является владельцем данного токена.

URL:

```
https://api.protectimus.com/api/v1/resource-service/unassign/token-with-user
```

Входные данные

Параметр	Обязательный параметр	Описание
resourceId	да, если не указан параметр resourceName	Идентификатор ресурса
resourceName	да, если не указан параметр resourceId	Имя ресурса
tokenId	да	Идентификатор пользователя

Выходные данные

Сообщение об успешном выполнении операции

Обратите внимание

- Токен и пользователь должны быть назначены на ресурс.
- Токен должен принадлежать данному пользователю.

POST unassign/user-token

Снимает назначение пользователя и токена с ресурса.

URL:

```
https://api.protectimus.com/api/v1/resource-service/unassign/user-token
```

Входные данные

Параметр	Обязательный параметр	Описание
resourceId	да, если не указан параметр resourceName	Идентификатор ресурса
resourceName	да, если не указан параметр resourceId	Имя ресурса
userId	да, если не указан параметр userLogin	Идентификатор пользователя
userLogin	да, если не указан параметр userId	Логин пользователя
tokenId	да	Идентификатор токена

Выходные данные

Сообщение об успешном выполнении операции.

Обратите внимание

- Для корректной работы метода необходимо, чтобы пользователь был назначен на ресурс вместе с этим токеном.

Управление токенами

Мы условно разделяем токены на физические и программные. Это разделение связано с особенностями введения и использования секретного ключа в конкретном виде токенов. К программным токенам относятся: Protectimus SMART, Google Authenticator, Protectimus SMS, Protectimus BOT и MAIL токены, к физическим - все остальные.

Токены, которые есть в системе могут быть использованы не только Вашими пользователями, но и Вами или Вашими администраторами для защиты доступа к Protectimus. Поэтому, токеном, который назначены администратору, может управлять только администратор, которому он назначен. Для остальных токенов подход такой же, как и для других объектов в системе: редактировать могут все, удалить может только создатель либо главный администратор.

Если пользователь потерял токен, но Вы хотите предоставить ему срочный доступ - Вам достаточно лишь выключить токен, тогда токен не будет участвовать в процессе аутентификации пользователя.
Если же токен утерян администратором - ему необходимо воспользоваться механизмом резервного доступа для подтверждения оставшихся аутентификаторов. После этого будет создана заявка на отключение токена. Удовлетворить ее может главный администратор либо служба технической поддержки.

В нашей системе могут использоваться любые токены, которые работают по стандартным алгоритмам OATH. Это может быть сделано в режиме добавления универсального токена. Конечно же, в таком случае от Вас потребуется больше знаний о Вашем токене. Поэтому мы реализовали поддержку нескольких популярных токенов других производителей, что значительно облегчает Вашу задачу. Мы постоянно работаем над усовершенствованием сервиса и расширением номенклатуры поддерживаемых токенов.

Общий адрес раздела:

```
https://api.protectimus.com/api/v1/token-service
```

GET secret-key/google-authenticator

Позволяет получить секретный ключ для Google Authenticator, который будет использован для генерации OTP. Этот ключ единоразово передается в устройство и на сервер, после чего его не должен знать никто, кроме токена и сервера, который выполняет проверку OTP с этого токена.

URL::

```
https://api.protectimus.com/api/v1/token-service/secret-key/google-authenticator
```

Входные данные
Отсутствуют.

Выходные данные
Возвращает секретный ключ для Google Authenticator.

Структура выходных данных для формата XML:

```
<responseHolder>
   <response>
      <key>{секретный_ключ}</key>
   </response>
   <status>OK</status>
</responseHolder>
```

Структура выходных данных для формата JSON:

```
{
   "responseHolder": {
      "response": {
         "key":"{секретный_ключ}"
       },
       "status":"OK"
    }
}
```

Описание параметров

Параметр	Тип	Описание
key	строка	Секретный ключ, который нужен для создания токена Google Authenticator

GET secret-key/protectimus-smart

Позволяет получить секретный ключ, который необходим для создания токена Protectimus SMART. Этот ключ используется для генерации OTP и должен быть передан устройству и серверу единоразово на этапе создания токена, после чего его не должен знать никто кроме этих двух сторон. Также, этот ключ содержит контрольную сумму, так что пользователь не сможет создать токен с неверным ключом на своем устройстве.

URL:

```
https://api.protectimus.com/api/v1/token-service/secret-key/protectimus-smart
```

Входные данные
Отсутствуют.

Выходные данные
Возвращает секретный ключ, который может быть использован для токена Protectimus SMART.

Структура выходных данных для формата XML:

```
<responseHolder>
   <response>
      <key>{секретный_ключ}</key>
   </response>
   <status>OK</status>
</responseHolder>
```

Структура выходных данных для формата JSON:

```
{
    "responseHolder": {
        "response": {
            "key":"{секретный_ключ}"
         },
         "status":"OK"
     }
}
```

Описание параметров

Параметр	Тип	Описание
key	строка	Секретный ключ, который нужен для создания токена Protectimus SMART

GET tokens

Позволяет получить список Ваших токенов.

URL:

https://api.protectimus.com/api/v1/token-service/tokens

Входные данные

Параметр	Обязательный параметр	Описание
start	нет, по умолчанию 0	Сдвиг, начиная с которого следует возвратить определенное количество (параметр limit) следующих токенов.
limit	нет, по умолчанию 10	Ограничение по количеству возвращаемых результатов
tokenName	нет	Имя токена
tokenType	нет	Тип токена. Доступные значения: PROTECTIMUS, PROTECTIMUS_SLIM, PROTECTIMUS_ULTRA, PROTECTIMUS_SMART, GOOGLE_AUTHENTICATOR, SAFENET_ETOKEN_PASS, YUBICO_OATH_MODE, UNIFY_OATH_TOKEN, SMS, MAIL, PROTECTIMUS_BOT
serialNumber	нет	Серийный номер токена. Он указан на обратной стороне устройства. Либо же является адресом электронной почты для Mail-токена и номером телефона для SMS-токена
enabled	нет	Включен или выключен токен
block	нет	Статус блокировки токена. Допустимые значения: NONE_BLOCKED, BLOCKED_BY_ADMIN, TOO_MANY_OTP_FAILED_ATTEMPTS_BLOCKED, TOO_MANY_OTP_FAILED_SYNCHRONIZATION_ATTEMPTS_BLOCKED
username	нет	Имя пользователя
resourceIds	нет	Идентификаторы ресурсов на которые назначен пользователь. Id должны быть перечислены через запятую. Пример: “2,3,5”
useBlankNames	нет	При значении true будут отображаться только токены без имени

Выходные данные

Список токенов с информацией о каждом из них.

Структура выходных данных для формата XML:

```
<responseHolder>
  <response>
    <tokens>
       <token>
         <apiSupport>{поддержка_через_API}</apiSupport>
         <creatorId>{идентификатор_создателя}</creatorId>
         <creatorUsername>{логин_создателя}</creatorUsername>
         <enabled>{включен_или_выключен}</enabled>
         <id>{идентификатор_токена}</id>
         <serialNumber>{серийный_номер_токена}</serialNumber>
         <type>{тип_токена}</type>
         <counter>{счетчик}</counter>
      </token>
      . . .
    </tokens>
  </response>
  <status>OK</status>
</responseHolder>
```

Структура выходных данных для формата JSON:

```
{
   "responseHolder": {
      "response":{
         "tokens":
          [
             {
                "apiSupport":{поддержка_через_API},
                "creatorId":{идентификатор_создателя},
                "creatorUsername":"{логин_создателя}",
                "enabled":{включен_или_выключен},
                "id":{идентификатор_токена},
                "serialNumber":"{серийный_номер_токена}",
                "type":"{тип_токена}"
             },
              . . .
          ]},
      "status":"OK"
   }
}
```

Описание параметров

Параметр	Тип	Описание
apiSupport	логический	Показывает, поддерживает ли токен аутентификацию через API. Если этот параметр false - значит токен не может быть аутентифицирован через API.
creatorId	число	Идентификатор администратора, который создал токен.
creatorUsername	строка	Имя (логин) администратора, который создал токен
enabled	логический	Показывает, включен или выключен токен. Если токен выключен, то при любом переданном OTP результат проверки ВСЕГДА будет положительным. Проще говоря, при выключенном токене он не участвует в процессе аутентификации, OTP не проверяется. Это полезно, если пользователь не может воспользоваться своим токеном по каким-либо причинам. Чтобы открыть ему доступ Вам необходимо просто отключить его токен.
id	число	Идентификатор токена
serialNumber	строка	Серийный номер токена. Он указан на обратной стороне устройства. Либо же является адресом электронной почты для Mail-токена и номером телефона для SMS-токена
type	строка (перечисление)	Тип токена. В системе имеются следующие типы токенов: PROTECTIMUS - физические токены Protectimus ONE PROTECTIMUS_SLIM- физические токены SLIM PROTECTIMUS_ULTRA - токен, который работает по алгоритму Challenge-Response PROTECTIMUS_SMART - токен, который устанавливается на мобильные устройства Android или iOs. GOOGLE_AUTHENTICATOR - токен от Google для мобильных устройств SAFENET_ETOKEN_PASS - токен от SafeNet YUBICO_OATH_MODE - токен от Yubico UNIFY_OATH_TOKEN - универсальный токен, который работает по стандартам OATH SMS - отправка одноразовых паролей по SMS

		MAIL - отправка одноразовых паролей по почте PROTECTIMUS_BOT - отправка одноразовых паролей с помощью ботов

Обратите внимание

- При задании сдвига больше, чем количество токенов, Вы получите пустой список.

GET tokens/quantity

Возвращает количество Ваших токенов.

URL:

```
https://api.protectimus.com/api/v1/token-service/tokens/quantity
```

Входные данные
Отсутствуют

Выходные данные
Возвращает количество токенов клиента

Структура выходных данных для формата XML:

```
<responseHolder>
   <response>
      <quantity>{количество_токенов}</quantity>
   </response>
   <status>OK</status>
</responseHolder>
```

Структура выходных данных для формата JSON:

```
{
    "responseHolder":{
        "response":{
            "quantity":количество_токенов
        },
        "status":"OK"
    }
}
```

Описание параметров

Параметр	Тип	Описание
quantity	число	Количество токенов клиента

POST tokens/unify

Создание универсального токена.

Данный метод позволяет добавить токен при отсутствии подходящих готовых вариантов.

URL:

https://api.protectimus.com/api/v1/token-service/tokens/unify

Входные данные

Параметр	Обязательный параметр	Описание
userId *	нет	Идентификатор пользователя, которому следует назначить создаваемый токен.
userLogin *	нет	Логин пользователя, которому следует назначить создаваемый токен.
unifyType	да	Тип токена. Доступные значения : OATH_HOTP (Event-Based), OATH_TOTP (Time-Based), OATH_OCRA (Challenge-Response)
unifyKeyAlgo	да	Алгоритм криптографического хеширования использующийся для генерации OTP. Доступные значения : SHA1, SHA256, SHA512
unifyKeyFormat		Формат секретного ключа. Доступные значения : HEX, BASE32, BASE64
serial	да	Серийный номер токена. Служит для идентификации токена во внешнем мире. Этим параметром может быть любая уникальная строка.
name	нет	Имя токена.
secret	да	Секретный ключ токена. Должен быть представлен строкой Base32. Сгенерировать секретный ключ можно с помощью методов: secret-key/google-authenticator и secret-key/protectimus-smart
otp	да	Одноразовый пароль из этого токена.
otpLength	да, для токена Protectimus SMART	Длина получаемого одноразового пароля. Допустимая длина 6 или 8 символов.

pin	нет	Пин-код, который пользователь должен будет вводить в поле ввода вместе с одноразовым паролем. Пароль и пин-код должны вводится одной строкой, без пробелов и других символов между ними. Положение пин-кода определяется параметром pinOtpFormat. Длина пин-кода должна быть равна 4 символам.
pinOtpFormat	да, если задан параметр pin	Формат пин-кода. Означает положение пин-кода в поле ввода: до одноразового пароля или после него. Допустимые значения: PIN_BEFORE_OTP и PIN_AFTER_OTP.
counter	нет	Счетчик токена.
challenge	да, если unifyType = OATH_OCRA	Число-вопрос, который пользователь должен ввести в токен, на основании которого токен сгенерирует ответ

* Параметр userId и userLogin служат для указания пользователя, которому следует назначить этот токен. Поиск пользователя ведется только по одному из этих параметров. Сначала выполняется поиск по параметру userId и если пользователь не будет найден - выполняется поиск по userLogin. Следовательно, если Вы указали Id одного пользователя, а логин другого - токен будет назначен пользователю, чей Id был указан.
Эти параметры не являются обязательными, если они не будут указаны - токен не будет назначен никакому пользователю. Назначить токен пользователю можно в любой момент при помощи соответствующих методов этого раздела.

Выходные данные
Идентификатор (ID) созданного токена.

Структура выходных данных для формата XML:

```
<responseHolder>
  <response>
    <id>{идентификатор_токена}</id>
  </response>
  <status>OK</status>
</responseHolder>
```

Структура выходных данных для формата JSON:

```
{
  "responseHolder" : {
    "response" : {
      "id" : {идентификатор_токена}
    },
    "status" : "OK"
```

```
  }
}
```

Описание параметров

Параметр	Тип	Описание
id	число	Идентификатор созданного токена

Обратите внимание

- Ваш тарифный план может не позволять добавить больше сущностей.

POST tokens/software

Создание программного токена.

Программными токенами считаются: Protectimus SMART, Google Authenticator, а также BOT, SMS и MAIL токены.

URL:

```
https://api.protectimus.com/api/v1/token-service/tokens/software
```

Входные данные

Параметр	Обязательный параметр	Описание
userId *	нет	Идентификатор пользователя, которому следует назначить создаваемый токен.
userLogin *	нет	Логин пользователя, которому следует назначить создаваемый токен.
type	да	Указывает на тип токена. Для этого метода доступны только программные типы токенов: PROTECTIMUS_SMART, GOOGLE_AUTHENTICATOR, SMS, MAIL, PROTECTIMUS_BOT
serial	да	Серийный номер токена. Служит для идентификации токена во внешнем мире. Для SMS-токена этим параметром является номер телефона, а для MAIL-токена - адрес электронной почты. Для остальных видов токенов этим параметром может быть любая уникальная строка.
name	нет	Имя токена.
secret	да	Секретный ключ токена. Должен быть представлен строкой Base32. Для Google Authenticator токенов длина ключа должна быть не менее 16 символов. Для Protectimus SMART длина ключа должна составлять 18 символов. Получить секретный ключ для этих токенов можно с помощью методов: secret-key/google-authenticator и secret-key/protectimus-smart Внимание! Для SMS и Mail токенов параметр secret должен быть эквивалентным значению параметра otp.
otp	да	Одноразовый пароль из этого токена. При создании SMS и Mail токена этот параметр должен совпадать с параметром secret. На этом этапе пользователю НЕ будет выслан одноразовый

		пароль и Вы не сможете проверить, правильный ли номер указал пользователь. Для того, чтобы сделать это, Вам следует создать SMS или Mail токен с помощью этого метода, указав в качестве параметра secret и otp одну и ту же случайную строку, после чего вызвать метод prepare чтобы отправить пароль, а затем проверить указанный пароль одним из методов аутентификации раздела auth-service. Остальные типы токенов позволяют получить одноразовый пароль сразу. Его и следует указывать в качестве этого параметра.
otpLength	да, для токена Protectimus SMART	Длина получаемого одноразового пароля. Допустимая длина 6 или 8 символов.
keyType	да, для токена Protectimus SMART	Режим, в котором генерируется одноразовый пароль. Допустимые значения: TOTP, HOTP. TOTP - генерация OTP на основании времени. HOTP - генерация OTP на основании счетчика.
pin	нет	Пин-код, который пользователь должен будет вводить в поле ввода вместе с одноразовым паролем. Пароль и пин-код должны вводится одной строкой, без пробелов и других символов между ними. Положение пин-кода определяется параметром pinOtpFormat. Длина пин-кода должна быть равна 4 символам.
pinOtpFormat	да, если задан параметр pin	Формат пин-кода. Означает положение пин-кода в поле ввода: до одноразового пароля или после него. Допустимые значения PIN_BEFORE_OTP и PIN_AFTER_OTP.
botPlatform	да, для токена PROTECTIMUS_BOT	Название мессенджера. Доступные значения: TELEGRAM, VIBER, FACEBOOK
botChatId	да, для токена PROTECTIMUS_BOT	Id чата в мессенджере (можно получить при обращении к соответствующему боту)

* Параметр userId и userLogin служат для указания пользователя, которому следует назначить этот токен. Поиск пользователя ведется только по одному из этих параметров. Сначала выполняется поиск по параметру userId и если пользователь не будет найден - выполняется поиск по userLogin. Следовательно, если Вы указали Id одного пользователя, а логин другого - токен будет назначен пользователю, чей Id был указан.
Эти параметры не являются обязательными, если они не будут указаны - токен не будет назначен никакому пользователю. Назначить токен пользователю можно в любой момент при помощи соответствующих методов этого раздела.

Выходные данные

Идентификатор (ID) созданного токена.

Структура выходных данных для формата XML:

```
<responseHolder>
  <response>
    <id>{идентификатор_токена}</id>
  </response>
  <status>OK</status>
</responseHolder>
```

Структура выходных данных для формата JSON:

```
{
 "responseHolder" : {
  "response" : {
    "id" : {идентификатор_токена}
  },
  "status" : "OK"
 }
}
```

Описание параметров

Параметр	Тип	Описание
id	число	Идентификатор созданного токена

Обратите внимание

- Ваш тарифный план может не позволять добавить больше сущностей.

POST tokens/hardware

Создание физического токена.

К физическим относится следующие виды токенов: PROTECTIMUS, PROTECTIMUS_SLIM, SAFENET_ETOKEN_PASS, PROTECTIMUS_ULTRA, YUBICO_OATH_MODE.

URL:

```
https://api.protectimus.com/api/v1/token-service/tokens/hardware
```

Входные данные

Параметр	Обязательный параметр	Описание
userId *	нет	Идентификатор пользователя, которому следует назначить создаваемый токен.
userLogin *	нет	Логин пользователя, которому следует назначить создаваемый токен.
type	да	Указывает на тип токена. Для этого метода доступны только аппаратные е типы токенов: PROTECTIMUS - токены Protectimus ONE PROTECTIMUS_SLIM - токены SLIM от Protectimus PROTECTIMUS_ULTRA - токены Protectimus, работающие по алгоритму Challenge-Response YUBICO_OATH_MODE - токены Yubico, работающие по стандартам OATH SAFENET_ETOKEN_PASS - токены SafeNet, работающие по алгоритму HOTP.
serial	да	Серийный номер токена. Обычно он указан на обратной стороне устройства.
secret	да, если токен не был заказан у Protectimus	Секретный ключ токена. Этот ключ вшит в токен и используется для генерации одноразового пароля, его не должен знать никто кроме токена и сервера, который будет проводить проверку OTP. Если токены были заказаны у Protectimus мы сами проведем обслуживание ключей в режиме строжайшей секретности и Вам не нужно будет указывать этот параметр.
name	нет	Имя токена
otp	да	Одноразовый пароль с токена. Он нужен для того, чтобы подтвердить, что токен действительно имеется в наличии. Для токенов SAFENET_ETOKEN_PASS необходимо указать два последовательных OTP, разделенных запятой,

		например: “147852,963258”. Это необходимо для определения сдвига счетчика.
existed	да	Указывает, создаете ли Вы существующий токен (заказан у Protectimus) или нет. Допустимые значения: “true” или “false”.
pin	нет	Пин-код, который пользователь должен будет вводить в поле ввода вместе с одноразовым паролем. Пароль и пин-код должны вводится одной строкой, без пробелов и других символов между ними. Положение пин-кода определяется параметром pinOtpFormat. Длина пин-кода должна быть равна 4 символам.
pinOtpFormat	да, если задан параметр pin	Формат пин-кода. Означает положение пин-кода в поле ввода: до одноразового пароля или после него. Допустимые значения PIN_BEFORE_OTP и PIN_AFTER_OTP.

* Параметр userId и userLogin служат для указания пользователя, которому следует назначить этот токен. Поиск пользователя ведется только по одному из этих параметров. Сначала выполняется поиск по параметру userId и если пользователь не будет найден - выполняется поиск по userLogin. Следовательно, если Вы указали Id одного пользователя, а логин другого - токен будет назначен пользователю, чей Id был указан.
Эти параметры не являются обязательными, если они не будут указаны - токен не будет назначен никакому пользователю. Назначить токен пользователю можно в любой момент при помощи соответствующих методов этого раздела.

Выходные данные
Возвращает идентификатор созданного токена.

Структура выходных данных для формата XML:

```
<responseHolder>
  <response>
    <id>{идентификатор_токена}</id>
  </response>
  <status>OK</status>
</responseHolder>
```

Структура выходных данных для формата JSON:

```
{
  "responseHolder" : {
    "response" : {
      "id" : {идентификатор_токена}
    },
    "status" : "OK"
  }
}
```

Описание параметров:

Параметр	Тип	Описание
id	число	Идентификатор созданного токена.

Обратите внимание

- Вы можете указать секретный ключ токена не в том формате, что не позволит сгенерировать правильный OTP на сервере и аутентифицировать токен.
- Параметр otp для токенов которые работают по алгоритму HOTP должен содержать два последовательных значения, разделенных запятой, чтобы Protectimus мог определить положение счетчика.

GET tokens/{id}

Позволяет получить информацию о Вашем токене.

URL:

```
https://api.protectimus.com/api/v1/token-service/tokens/{id}
```

Входные данные

Параметр	Обязательный параметр	Описание
id	да	Идентификатор токена.

Выходные данные
Возвращает информацию о токене

Структура выходных данных для формата XML:

```
<responseHolder>
   <response>
      <token>
         <apiSupport>{поддержка_через_API}</apiSupport>
         <creatorId>{идентификатор_создателя}</creatorId>
         <creatorUsername>{логин_создателя}</creatorUsername>
         <enabled>{проверка_OTP_включена_или_нет}</enabled>
         <id>{идентификатор_токена}</id>
         <name>{имя_токена}</name>
         <serialNumber>{серийный_номер}</serialNumber>
         <type>{тип_токена}</type>
      </token>
   </response>
   <status>OK</status>
</responseHolder>
```

Структура выходных данных для формата JSON:

```
{
    "responseHolder":{
        "response":{
            "token":{
                "apiSupport":{поддержка_через_API}, "creatorId":{идентификатор_создателя},
                "creatorUsername":"{логин_создателя}",
"enabled":{проверка_OTP_включена_или_нет},

"id":{идентификатор_токена},"name":{имя_токена},"serialNumber":"{серийный_номер}",
                "type":"{тип_токена}"
            }
```

```
        },
        "status":"OK"
    }
}
```

Описание параметров:

Параметр	Тип	Описание
apiSupport	логический	Показывает, поддерживает ли токен аутентификацию через API. Если этот параметр false - значит токен не может быть аутентифицирован через API.
creatorId	число	Идентификатор администратора, который создал токен.
creatorUsername	строка	Имя (логин) администратора, который создал токен
enabled	логический	Показывает, включен или выключен токен. Если токен выключен, то при любом переданном OTP результат проверки ВСЕГДА будет положительным. Проще говоря, при выключенном токене он не участвует в процессе аутентификации, OTP не проверяется. Это полезно, если пользователь не может воспользоваться своим токеном по каким-либо причинам. Чтобы открыть ему доступ Вам необходимо просто отключить его токен.
id	число	Идентификатор токена
name	строка	Имя токена.
serialNumber	строка	Серийный номер токена. Он указан на обратной стороне устройства. Либо же является адресом электронной почты для Mail-токена и номером телефона для SMS-токена
type	строка (перечисление)	Тип токена. Описание типов Вы можете найти в методах создания программных и физических токенов.

PUT tokens/{id}

Позволяет редактировать информацию о токене.

Помните: нельзя менять информацию о токене, который принадлежит другому администратору.

URL:

```
https://api.protectimus.com/api/v1/token-service/tokens/{id}
```

Входные данные

Параметр	Обязательный параметр	Описание
name	нет	Новое имя токена
enabled *	нет	Включен или выключен токен.
apiSupport *	нет	Поддержка аутентификации через API.

* Эти параметры подробно были описаны выше, в других методах API, например в методе GET tokens/{id}

Выходные данные

Возвращает информацию об отредактированном токене. Структура выходных данных и значение параметров такое же, как и у метода GET tokens/{id}

DELETE tokens/{id}

Удаление токена.

Помните: нельзя удалить токен, который принадлежит другому администратору.

URL:

```
https://api.protectimus.com/api/v1/token-service/tokens/{id}
```

Входные данные

Параметр	Обязательный параметр	Описание
id	да	Идентификатор удаляемого токена.

Выходные данные
Информация об удаленном токене. Структура выходных данных и значение параметров такое же, как и у метода GET tokens/{id}

POST tokens/{id}/unassign

Снимает назначение токена с пользователя.

URL:

```
https://api.protectimus.com/api/v1/token-service/tokens/{id}/unassign
```

Входные данные

Параметр	Обязательный параметр	Описание
id	да	Идентификатор токена, который должен быть снят с назначения.

Выходные данные
Сообщение об успешном выполнении операции либо стандартные сообщения об ошибках.

POST tokens/sign-transaction

Данный метода предназначен для генерации цифровой подписи.

Эта возможность позволяет обеспечивать контроль целостности подписанных данных и защиту от изменений или подделки данных.

URL:

```
https://api.protectimus.com/api/v1/token-service/tokens/sign-transaction
```

Входные данные

Параметр	Обязательный параметр	Описание
tokenId	да	Идентификатор токена, с помощью которого будет подписана информация
transactionData	да	Данные для подписи
hash	да	Хэш от transactionData необходим для проверки целостности входящих данных. Хэш должен быть сформирован при помощи HmacSHA256 алгоритма с <ApiKey> в качестве ключа
templateIdOrName	нет	Идентификатор или имя шаблона который будет использоваться для доставки подписанной информации и OTP

Выходные данные
Информация необходимая для проверки подписанных данных.

Структура выходных данных для формата XML:

```
<responseHolder>
   <response>
      <id>{идентификатор_токена}</id>
      <challenge>{challenge}</challenge>
      <tokenName>{имя_токена}</tokenName>
      <tokenType>{тип_токена}</tokenType>
      <transactionData>{подписанные_данные}</transactionData>
   </response>
   <status>OK</status>
</responseHolder>
```

Структура выходных данных для формата JSON:

```
{
   "responseHolder": {
      "response": {
         "id": {идентификатор_токена},
         "challenge": {challenge},
         "tokenName": {имя_токена},
         "tokenType": {тип_токена},
         "transactionData": {подписанные_данные}
      },
      "status": "OK"
   }
}
```

Описание параметров

Параметр	Тип	Описание
id	число	Идентификатор токена использованного для подписи данных
challenge	число	Число-вопрос, сформированное на основании подписываемых данных
tokenName	строка	Имя токена использованного для подписи
tokenType	строка	Тип токена использованного для подписи
transactionData	строка	Подписанные данные в зашифрованном виде

Обратите внимание

- ApiKey - ключ API, он различный для каждого администратора, доступен и может быть изменен на странице управления профилем https://service.protectimus.com/profile
- Для возможности проверки информации с помощью приложения Protectimus Smart необходимо сгенерировать QR код следующего формата : transaction://challenge={challenge}&transactionData={подписанные_данные}

POST tokens/verify-signed-transaction

Данный метода предназначен для проверки подписанных данных.

URL:

```
https://api.protectimus.com/api/v1/token-service/tokens/verify-signed-transaction
```

Входные данные

Параметр	Обязательный параметр	Описание
tokenId	да	Идентификатор токена использованного для подписи данных
transactionData	да	Данные для подписи
hash	да	Хэш от transactionData необходим для проверки целостности входящих данных. Хэш должен быть сформирован при помощи HmacSHA256 алгоритма с <ApiKey> в качестве ключа
otp	да	OTP необходимое для проверки подлинности подписанных данных

Выходные данные
Результат проверки подписанных данных.

Структура выходных данных для формата XML:

```
<responseHolder>
   <response>
      <result>false</result>
   </response>
   <status>OK</status>
</responseHolder>
```

Структура выходных данных для формата JSON:

```
{
   "responseHolder": {
      "response": {
         "result": true
      },
      "status": "OK"
   }
}
```

Описание параметров

Параметр	Тип	Описание
id	число	Идентификатор токена использованного для подписи
challenge	число	Число-вопрос, сформированное на основании подписываемых данных
tokenName	строка	Имя токена использованного для подписи
tokenType	строка	Тип токена использованного для подписи
transactionData	строка	Подписанная информация

Обратите внимание

- ApiKey - ключ API, он различный для каждого администратора, доступен и может быть изменен на странице управления профилем https://service.protectimus.com/profile

POST tokens/send-message-from-bot

Отправка сообщений с помощью ботов.

URL:

```
https://api.protectimus.com/api/v1/token-service/tokens/send-message-from-bot
```

Входные данные

Параметр	Обязательный параметр	Описание
tokenId	да, если не указан параметр tokenName	Идентификатор токена
tokenName	да, если не указан параметр tokenId	Имя токена
message	да	Сообщение которое будет отправлено с помощью бота

Выходные данные
Сообщение об успешном выполнении операции либо стандартные сообщения об ошибках.

Управление пользователями

Этот раздел предназначен для работы с пользователями. Вы можете хранить некоторый набор информации о Ваших пользователях в нашей системе, но основным параметром является логин пользователя.
В системе имеется механизм самообслуживания пользователей. Он настраивается для каждого ресурса отдельно, это можно сделать на вкладке “Самообслуживание” на странице просмотра детальной информации о ресурсе.

GET users

Позволяет получить список пользователей (10 экземпляров, начиная с заданного сдвига).

URL:

```
https://api.protectimus.com/api/v1/user-service/users
```

Входные данные

Параметр	Обязательный параметр	Описание
start	нет, по умолчанию 0	Сдвиг, начиная с которого следует возвратить определенное количество (параметр limit) следующих пользователей.
limit	нет, по умолчанию 10	Ограничение по количеству возвращаемых результатов
block	нет	Статус блокировки пользователя. Допустимые значения: NONE_BLOCKED, BLOCKED_BY_ADMIN, TOO_MANY_LOGIN_FAILED_ATTEMPTS_BLOCKED, TOO_MANY_OTP_FAILED_ATTEMPTS_BLOCKED, TOO_MANY_EMAIL_FAILED_ATTEMPTS_BLOCKED, TOO_MANY_PIN_FAILED_ATTEMPTS_BLOCKED
resourceIds	нет	Идентификаторы ресурсов на которые назначен пользователь. Id должны быть перечислены через запятую. Пример: “2,3,5”
login	нет	Логин пользователя
email	нет	Адрес электронной почты пользователя
firstName	нет	Имя пользователя
secondName	нет	Фамилия пользователя

Выходные данные
Список пользователей.

Структура выходных данных для формата XML:

```
<responseHolder>
  <response>
    <users>
      <user>
        <apiSupport>{поддержка_через_API}</apiSupport>
        <creatorId>{ID_создателя}</creatorId>
        <creatorUsername>{логин_создателя}</creatorUsername>
```

```
            <email>{адрес_электронной_почты}</email>
            <firstName>{имя_пользователя}</firstName>
            <hasTokens>{наличие_назначенных_токенов}</hasTokens>
            <id>{идентификатор_пользователя}</id>
            <login>{логин_пользователя}</login>
            <phoneNumber>{телефонный_номер_пользователя}</phoneNumber>
            <secondName>{фамилия_пользователя}</secondName>
        </user>
        . . .
      </users>
    </response>
    <status>OK</status>
</responseHolder>
```

Структура выходных данных для формата JSON:

```
{
   "responseHolder":{
      "response":{
         "users":[{
            "apiSupport":{поддержка_через_API},
            "creatorId":{ID_создателя},
            "creatorUsername":"{логин_создателя}",
            "email":"{адрес_электронной_почты}",
            "hasTokens":{наличие_назначенных_токенов},
            "id":{идентификатор_пользователя},
            "login":"{логин_пользователя}",
            "phoneNumber":"{телефонный_номер_пользователя}",
            "secondName":"{фамилия_пользователя}"
         },
            . . .
         ]
      },
      "status":"OK"
   }
}
```

Описание параметров

Параметр	Тип	Описание
apiSupport	логический	Поддержка аутентификации через API. Если этот параметр false, то пользователь
creatorId	число	Идентификатор создателя
creatorUsername	строка	Имя (логин) создателя
email	строка	адрес электронной почты

hasTokens	логический	Показывает, есть ли у пользователя назначенные токены
id	число	Идентификатор пользоателя
login	строка	логин пользователя
phoneNumber	строка	телефонный номер пользователя
secondName	строка	Фамилия пользователя

GET users/quantity

Позволяет получить информацию о количестве пользователей клиента.

URL:

```
https://api.protectimus.com/api/v1/user-service/users/quantity
```

Входные данные
Отсутствуют.

Выходные данные
Возвращает количество пользователей клиента.

Структура выходных данных для формата XML:

```
<responseHolder>
   <response>
      <quantity>{количество_пользователей}</quantity>
   </response>
   <status>OK</status>
</responseHolder>
```

Структура выходных данных для формата JSON:

```
{
   "responseHolder":{
      "response":{
         "quantity":{количество_пользователей}
      },
      "status":"OK"
   }
}
```

Описание параметров

Параметр	Тип	Описание
quantity	число	Количество пользователей клиента

POST users

Добавление пользователя.

URL:

```
https://api.protectimus.com/api/v1/user-service/users
```

Входные данные

Параметр	Обязательный параметр	Описание
login	да	Логин пользователя. Может содержать только латинские буквы, цифры, знаки: @ _ . - . Разрешенная длина: от 5 до 30 символов. Поле должно быть уникальным в пределах аккаунта Вашей компании. По этому параметру будет вестись поиск пользователя при аутентификации. В качестве логина Вы также можете указать e-mail или телефон пользователя, оставив заполнение соответствующих параметров на свое усмотрение.
alias	нет	Если вы используете Protectimus для защиты нескольких приложений, например Windows и OWA, и при этом у ваших пользователей заданы разные типы логинов в каждом из этих приложений (например, “Administrator” в Windows и “administrator@doomain.com” в OWA), создайте в сервисе Protectimus одного пользователя с логином “Administrator” и укажите алиас “administrator@doomain.com”. Все логины и алиасы должны быть уникальными.
email	нет	Адрес электронной почты пользователя.
phoneNumber	нет	Телефонный номер пользователя. Необходимо указывать в международном формате.
password	нет	Пароль пользователя. Вы можете передавать пароль в открытом виде. Но если у Вас есть только хеш, Вы можете воспользоваться двумя способами: 1. Использовать метод /users/password из раздела user-service, чтобы задать хеш и то, каким образом Protectimus может его получить. В этом случае пользователь сможет

		использовать свой пароль для доступа ко всем сервисам Protectimus.. 2. Передать в протектимус хеш с помощью этого или подобного метода и каждый раз перед аутентификацией самостоятельно формировать хеш от введенного пользователем пароля, после чего передавать результат хеширования для аутентификации. В этом случае Вы сможете работать только с методами API, в которые Вы должны передать пароль самостоятельно, а пользователь не сможет воспользоваться сервисами, где он должен будет сам указать свой пароль для входа.
firstName	нет	Имя пользователя. Длина поля от 1 до 50 символов.
secondName	нет	Фамилия пользователя. Длина поля от 1 до 50 символов.
apiSupport	нет	Поддержка аутентификации через API. Допустимые значения: true или false. По умолчанию параметр устанавливается в true, т.е. пользователь может аутентифицироавться через API.

Выходные данные
Идентификатор созданного пользователя

Структура выходных данных для формата XML:

```
<responseHolder>
  <response>
    <id>{идентификатор_пользователя}</id>
  </response>
  <status>OK</status>
</responseHolder>
```

Структура выходных данных для формата JSON:

```
{
  "responseHolder" : {
    "response" : {
      "id" : {идентификатор_пользователя}
    },
    "status" : "OK"
  }
}
```

Описание параметров

Параметр	Тип	Описание
id	число	Идентификатор созданного пользователя

GET users/{id}

Позволяет получить информацию о пользователе.

URL:

```
https://api.protectimus.com/api/v1/user-service/users/{id}
```

Входные данные

Параметр	Обязательный параметр	Описание
id	да	Идентификатор пользователя, о котором нужно получить информацию.

Выходные данные
Возвращает информацию о пользователе.

Структура выходных данных для формата XML:

```
<responseHolder>
   <response>
      <user>
            <apiSupport>{поддержка_через_API}</apiSupport>
            <creatorId>{ID_создателя}</creatorId>
            <creatorUsername>{логин_создателя}</creatorUsername>
           <email>{адрес_электронной_почты}</email>
            <firstName>{имя_пользователя}</firstName>
            <hasTokens>{наличие_назначенных_токенов}</hasTokens>
            <id>{идентификатор_пользователя}</id>
            <login>{логин_пользователя}</login>
            <alias>{алиас_пользователя}</alias>
            <phoneNumber>{телефонный_номер_пользователя}</phoneNumber>
            <secondName>{фамилия_пользователя}</secondName>
       </user>
   </response>
   <status>OK</status>
</responseHolder>
```

Структура выходных данных для формата JSON:

```
{
   "responseHolder":{
      "response":{
         "user":{
            "apiSupport":{поддержка_через_API},"creatorId":{ID_создателя},
            "creatorUsername":"{логин_создателя}","email":"{адрес_электронной_почты}",
            "hasTokens":{наличие_назначенных_токенов},"id":{идентификатор_пользователя},
```

```
            "login":"{логин_пользователя}", "alias":"{алиас_пользователя}",
            "phoneNumber":"{телефонный_номер_пользователя}",
           "secondName":"{фамилия_пользователя}"
        }
      },
      "status":"OK"
   }
}
```


Описание параметров

Параметр	Тип	Описание
apiSupport	логический	Поддержка аутентификации через API. Если этот параметр false, то пользователь не может аутентифицироваться через API.
creatorId	число	Идентификатор создателя
creatorUsername	строка	Имя (логин) создателя
email	строка	Адрес электронной почты
hasTokens	логический	Показывает, есть ли у пользователя назначенные токены
id	число	Идентификатор пользователя
login	строка	Логин пользователя
alias	строка	Алиас пользователя
phoneNumber	строка	Телефонный номер пользователя
secondName	строка	Фамилия пользователя

PUT users/{id}

Позволяет редактировать информацию о пользователе.

URL:

```
https://api.protectimus.com/api/v1/user-service/users/{id}
```

Входные данные

Параметр	Обязательный параметр	Описание
login	да	Логин пользователя. Может содержать только латинские буквы, цифры, знаки: @ _ . - . Разрешенная длина: от 5 до 30 символов. Поле должно быть уникальным в пределах аккаунта Вашей компании. По этому параметру будет вестись поиск пользователя при аутентификации. В качестве логина Вы также можете указать e-mail или телефон пользователя, оставив заполнение соответствующих параметров на свое усмотрение.
alias	нет	Если вы используете Protectimus для защиты нескольких приложений, например Windows и OWA, и при этом у ваших пользователей заданы разные типы логинов в каждом из этих приложений (например, “Administrator” в Windows и “administrator@doomain.com” в OWA), создайте в сервисе Protectimus одного пользователя с логином “Administrator” и укажите алиас “administrator@doomain.com”. Все логины и алиасы должны быть уникальными.
email	нет	Адрес электронной почты пользователя.
phoneNumber	нет	Телефонный номер пользователя. Необходимо указывать в международном формате.
password	нет	Пароль пользователя. Вы можете передавать пароль в открытом виде. Но если у Вас есть только хеш, Вы можете воспользоваться двумя способами: 1. Использовать метод /users/password из раздела user-service, чтобы задать хеш и то, каким образом Protectimus может его получить. В этом случае пользователь сможет использовать свой пароль для доступа ко всем сервисам Protectimus.. 2. Передать в протектимус хеш с помощью этого или подобного метода и каждый раз перед аутентификацией самостоятельно формировать хеш от введенного

		пользователем пароля, после чего передавать результат хеширования для аутентификации. В этом случае Вы сможете работать только с методами API, в которые Вы должны передать пароль самостоятельно, а пользователь не сможет воспользоваться сервисами, где он должен будет сам указать свой пароль для входа.
firstName	нет	Имя пользователя. Длина поля от 1 до 50 символов.
secondName	нет	Фамилия пользователя. Длина поля от 1 до 50 символов.
apiSupport	нет	Поддержка аутентификации через API. Допустимые значения: true или false. По умолчанию параметр устанавливается в true, т.е. пользователь может аутентифицироваться через API.

* Значение этих параметров такое же, как и у метода POST users

Выходные данные
Возвращает отредактированную информацию о пользователе. Структура выходных данных и описание параметров такое же, как и у методов POST users, GET users/{id}

POST users/password

Данный метод позволяет задавать или редактировать пароль пользователя в безопасной форме.

Скорее всего, пароли пользователей в Вашей базе данных хранятся в захешированном виде и Вы не знаете их. Если Вы хотите использовать эти же пароли для аутентификации пользователей через Protectimus - Вам следует использовать этот метод. Вы передаете хеш пароля пользователя и правила, по которым он формируется, чтобы Protectimus смог выполнить те же преобразования и получить тот же хеш, необходимый для аутентификации.

URL:

```
https://api.protectimus.com/api/v1/user-service/users/password
```

Входные данные

Параметр	Обязательный параметр	Описание
id	да, если не указан параметр login	Идентификатор пользователя, для которого должен быть установлен или изменен пароль.
login	да, если не указан параметр id	Логин пользователя, для которого должен быть установлен или изменен пароль.
rawPassword	да	Хеш пароля пользователя в HEX-формате.
rawSalt	нет	Соль, использованная при хешировании пароля.
encodingType	да	Метод, который использовался для получения хеша. Допустимы следующие значения: PLAIN - пароль не был захеширован и представлен в чистом виде; MD5 - был использован алгоритм MD5 SHA - был использован алгоритм SHA-1 SHA256 - был использован алгоритм SHA-256
encodingFormat	да	Формат строки (пароль и соль), которая была захеширована.

		При аутентификации пользователя Protectimus будет заменять в этой строке слово PASS на пароль, который введет пользователь, а слово PLAIN_SALT на соль, переданную Вами в этом методе. Остальные же символы будут сохранены. Полученная строка будет преобразована с помощью алгоритма encodingType и будет сравниваться с rawPassword для аутентификации пользователя.

Выходные данные
Возвращает информацию о пользователе. Структура выходных данных и описание параметров такое же, как и у методов POST users, GET users/{id}

DELETE users/{id}

Удаление пользователя.

Как и другие объекты системы, пользователь может быть удален только администратором, который его создал, либо главным администратором.

URL:

```
https://api.protectimus.com/api/v1/user-service/users/{id}
```

Входные данные

Параметр	Обязательный параметр	Описание
id	да	Идентификатор удаляемого пользователя.

Выходные данные

Возвращает информацию об удаленном пользователе. Структура ответа и значение параметров такое же, как и у метода получения информации о пользователе GET users/{id}.

GET /users/{id}/tokens

Позволяет получить список токенов пользователя (10 элементов начиная с заданного сдвига).

URL:

```
https://api.protectimus.com/api/v1/user-service/users/{id}/tokens
```

Входные данные

Параметр	Обязательный параметр	Описание
id	да	Идентификатор пользователя

Выходные данные
Возвращает список с информацией о токенах, которые назначены указанному клиенту.

Структура выходных данных для формата XML:

```
<responseHolder>
  <response>
    <tokens>
      <token>
        <apiSupport>{поддержка_через_API}</apiSupport>
        <creatorId>{идентификатор_создателя}</creatorId>
        <creatorUsername>{логин_создателя}</creatorUsername>
        <enabled>{включен_или_выключен}</enabled>
        <id>{идентификатор_токена}</id>
        <serialNumber>{серийный_номер_токена}</serialNumber>
        <type>{тип_токена}</type>
      </token>
      ...
    </tokens>
  </response>
  <status>OK</status>
</responseHolder>
```

Структура выходных данных для формата JSON:

```
{
   "responseHolder": {
      "response":{
         "tokens":
         [
            {
               "apiSupport":{поддержка_через_API},"creatorId":{идентификатор_создателя},
               "creatorUsername":"{логин_создателя}","enabled":{включен_или_выключен},
               "id":{идентификатор_токена},"serialNumber":"{серийный_номер_токена}",
```

```
                "type":"{тип_токена}"
            },
             . . .
        ]},
    "status":"OK"
  }
}
```

Описание параметров

Параметр	Тип	Описание
apiSupport	логический	Показывает, поддерживает ли токен аутентификацию через API. Если этот параметр false - значит токен не может быть аутентифицирован через API.
creatorId	число	Идентификатор администратора, который создал токен.
creatorUsername	строка	Имя (логин) администратора, который создал токен
enabled	логический	Показывает, включен или выключен токен. Если токен выключен, то при любом переданном OTP результат проверки ВСЕГДА будет положительным. Проще говоря, при выключенном токене он не участвует в процессе аутентификации, OTP не проверяется. Это полезно, если пользователь не может воспользоваться своим токеном по каким-либо причинам. Чтобы открыть ему доступ Вам необходимо просто отключить его токен.
id	число	Идентификатор токена
serialNumber	строка	Серийный номер токена. Он указан на обратной стороне устройства. Либо же является адресом электронной почты для Mail-токена и номером телефона для SMS-токена
type	строка (перечисление)	Тип токена. Подробнее типы токенов описаны в методах соответствующего раздела, например, в методах tokens/hardware и tokens/software

GET users/{id}/tokens/quantity

Позволяет узнать количество токенов, которые назначены пользователю.

URL:

```
https://api.protectimus.com/api/v1/user-service/users/{id}/tokens/quantity
```

Входные данные

Параметр	Обязательный параметр	Описание
id	да	Идентификатор пользователя, количество токенов которого необходимо узнать.

Выходные данные

Возвращает количество токенов, назначенных пользователю.

Структура выходных данных для формата XML:

```
<responseHolder>
  <response>
    <quantity>1</quantity>
  </response>
  <status>OK</status>
</responseHolder>
```

Структура выходных данных для формата JSON:

```
{
 "responseHolder" : {
  "response" : {
    "quantity" : 1
  },
  "status" : "OK"
 }
}
```

Описание параметров

Параметр	Тип	Описание
quantity	число	Количество токенов, которые назначены пользователю с указанным идентификатором.

POST users/{userId}/tokens/{tokenId}/assign

Выполняет назначение токена пользователю.

Пользователю может быть назначено сколько угодно токенов, но токен может быть назначен только одному пользователю.

URL:

```
https://api.protectimus.com/api/v1/user-service/users/{userId}/tokens/{tokenId}/assign
```

Входные данные

Параметр	Обязательный параметр	Описание
userId	да	Идентификатор пользователя, которому следует назначить токен
tokenId	да	Идентификатор токена, который следует назначить указанному пользователю.

Выходные данные
Сообщение об успешном выполнении операции либо стандартные сообщения об ошибках.

POST users/{userId}/tokens/{tokenId}/unassign

Снимает назначение токена с пользователя.

URL:

https://api.protectimus.com/api/v1/user-service/users/{userId}/tokens/{tokenId}/unassign

Входные данные

Параметр	Обязательный параметр	Описание
userId	да	Идентификатор пользователя, с которого нужно снять назначение токена.
tokenId	да	Идентификатор токена, который должен быть снят с назначения пользователю.

Выходные данные
Сообщение об успешном выполнении операции, либо стандартные сообщения об ошибках.

Коды ошибок и сообщений

Код ошибки	Описание
1001	Сущность уже существует. Возникает в двух случаях: 1. Когда Вы пытаетесь добавить объект с уникальным полем, которое уже существует в системе. Например, Вы пытаетесь добавить пользователя с логином, который уже зарегистрирован в системе. 2. Когда вы пытаетесь выполнить действие, которое уже было выполнено. Например, Вы пытаетесь назначить токен пользователю, но этот токен уже назначен этому или другому пользователю.
2001	Неправильная длина параметра. Возникает, когда один или несколько переданных параметров имеют неверную длину.
3001	Ошибка базы данных.
4001	Незарегистрированное имя.
5001	Возникает, когда не указан параметр, который является обязательным для вызываемого метода.
5002	Возникает, когда: 1. Сущность не может быть найдена в базе данных. Например, запрошена информация о пользователе с идентификатором, которого нет в базе данных. 2. В базе данных не существует связи для выполнения требуемого действия. Например, Вы хотите провести аутентификацию пользователя на ресурсе, но он не назначен на этот ресурс. Или Вы хотите снять назначение токена с ресурса, но токен не был назначен на этот ресурс ранее. Или Вы хотите назначить токен на ресурс вместе с пользователем, но токен не назначен никакому пользователю.
6001	Возникает в случае передачи неверного параметра. Например, ожидается получить число, но переданный параметр содержит посторонние символы.
6002	Неверный формат URL
7001	Ограничение доступа. Возникает, когда Вы не имеете права работать с запрашиваемым объектом. Причиной этому может быть отсутствие прав на доступ к объекту, блокировка Вашего аккаунта по разным причинам и т.д.
8001	Внутренняя ошибка сервера.
9001	Неизвестная ошибка.

Структура сообщений об ошибках в формате XML:

```
<responseHolder>
  <error>
    <code>{код_ошибки}</code>
    <message>{общее_пояснение_к_ошибке}</message>
    <developersMessage>{сообщение_с_техническими_деталями}</developersMessage>
  </error>
  <status>FAILURE</status>
</responseHolder>
```

Структура сообщений об ошибках в формате JSON:

```
{
  "responseHolder" : {
    "error" : {
      "code" : {код_ошибки},
      "message" : "{общее_пояснение_к_ошибке}"
      “developersMessage”: “{сообщение_с_техническими_деталями}”
    },
    "status" : "FAILURE"
  }
}
```

Сообщения об успешном выполнении операции

Сообщения об успешном выполнении операции в формате XML:

```
<responseHolder>
   <status>OK</status>
</responseHolder>
```

Сообщения об успешном выполнении операции в формате JSON:

```
{
  "responseHolder" : {
    "status" : "OK"
 }
```