API. Integration Instructions

Version 1.01
Date: 23 February 2014

Contents

Terms and Abbreviations

Authentication is a process of verifying a user's identity, i.e. verifying whether or not a user is the person that this user claims to be.

OTP (One-Time Password) is a password that is valid for only one authentication session.

Token is a physical or virtual device for generating one-time passwords.

Resource is an object that needs to be protected via two-factor authentication.

General Information

The Protectimus team provides a set of tools that will help you to easily perform integration with any resource or project. The software developer kits (SDK) for such popular programming languages as Java, Ruby, Pyton, .Net, and PHP[1] will save you time and effort during integration with our solution, and we recommend that you should use these SDK's. If you have any specific requirements, you can directly use our API described in this document.

Our API's design is based on the REST principles. Data is transmitted in the XML format or the JSON format. Parameter values are identical in these formats. By default, responses are transmitted in the XML format.

Preliminary Steps

When you use the Protectimus **service**, the API needs to be activated. To activate the API, you need to activate the service plan selected in our system (https://service.protectimus.com/pricing). After that, your account will be charged once a day as payment for your use of the service. You can suspend the use of the system and payments by deactivating your service plan, but you should note that in this case the API will also be deactivated.

Authorization

The Protectimus API is only accessible to authorized users. Our solution uses Basic authentication. The login (username) of the administrator that submits a request is used as the username, and an authentication token is used as the password.
An authentication token is the hash of a string that consists of the following elements: <ApiKey>:<YYYYMMDD>:<HH>, where:

- *ApiKey* is an API key, which is unique for each administrator; it is provided and may be changed on the profile management page https://service.protectimus.com/profile
- *YYYYMMDD* is the current date in the specified format
- *HH* is the UTC time in the HH format (only hours in the 24-hour format, without minutes or seconds)

Example: The administrator's profile contains the following information:
ApiKey – MySecureApiKey; Date - 30 January 2014; Time - 17:42 (UTC).

String for hash: MySecureApiKey:20140130:17

Hash SHA256 for this text:

62704fb3a9dcf7b5b3cf7bda6ac9d0b0aa37c6fce8d0fae6b466c91ba68894f5

1 Currently (as of the date of this document), there are customers that require Java, Pyton, and PHP

Request Submission

The protocol for transmitting all requests to the Protectimus API is HTTPS.

Request Format:

```
<HTTP-method>
https://api.protectimus.com/api/v<API_version>/<API_section>/<API_method>.<format>
```

The parameters specified above have the following values:

- <HTTP-method> is the method typical for the current request.
- <API_version> is the API version that you want to use. Currently, only the first version is available; therefore, this part of the request will look like this: “v1”.
- <API_section> is the section to which the method you are calling belongs. The following sections are available: auth-service, resource-service, token-service, and user-service. The API methods’ descriptions are divided into the sections to which these methods belong.
- <API_method> is the method you are calling.
- <format> is the format in which you want to receive a response: XML or JSON. By default, XML is the selected format.

If an error occurs, the processing of a request is terminated, and an error message is returned.
A list of errors and descriptions of errors are given in the Error Message section.
Most of the actions available through the API are available in the service through our graphical user interface (GUI). Familiarizing yourself with it will help you to understand the principles of our system’s operation better.

API Methods’ Descriptions

Obtaining General Information

This Section’s URL:

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

GET Balance

Obtain information on the current balance of funds in a customer’s account. This information is accessible only to the superuser.

URL:

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

Input Data
None.

Output Data
The current balance of funds in an account is returned.

Output Data Structure for XML Format:

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

Output Data Structure for JSON Format:

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

Parameter Description

Parameter	Type	Description
balance	Numeric	Current balance of funds in an account in USD

Authentication Process

This section of the API serves the purpose of authenticating users and tokens on your resources.

This Section's URL:

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

Depending on the chosen model, a user may be authenticated with a static password, a one-time password, or both a static password and a one-time password. For a user or a token to be authenticated, this user or token has to be assigned to the requested resource (if both a user and a token are authenticated on a resource simultaneously, this user has to be assigned to this resource with this token).

Please note:
If a user is not authenticated successfully, the number of failed authentication attempts will be increased for this user. When the threshold number of failed attempts for the specified resource is exceeded, this user will be locked. A user can be unlocked through the web interface or the API (the edit user method).

If a user is authenticated successfully, the number of failed authentication attempts will be set at zero, if the threshold number of failed attempts for the specified resource is not exceeded, and if this user has not yet been locked.

The PROTECTIMUS SMS, PROTECTIMUS MAIL, and PROTECTIMUS ULTRA tokens require that the *POST prepare* method should be called to prepare the authentication process. You can find more information about it in this method's description.

POST prepare

For some types of tokens, such as Protectimus SMS, Protectimus MAIL, and Protectimus ULTRA, certain actions are required before a token can be authenticated. This method must be called for the SMS and MAIL tokens for a one-time password to be sent to a user, and for the Protectimus ULTRA tokens – to receive a challenge that a user will have to enter in a token to generate a password under the Challenge-Response algorithm. For other types of tokens, this method does not need to be called.

To specify token, which must be prepared for authentication you can use one of the following parameters: tokenId, userId or userLogin

. But remember: to correctly determine token by the userId or userLogin, the token must be assigned on the resource with the user.

URL:

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

Input Data

Parameter	Mandatory Parameter	Description
resourceId	Yes, if the resourceName parameter is not specified	The identifier (ID) of the resource on which the token needs to be prepared for authentication
resourceName	Yes, if the resourceId parameter is not specified	The name of the resource on which the token needs to be prepared for authentication
tokenId	Yes, if the userId or userLogin parameter is not specified	The identifier (ID) of the token that needs to be prepared for authentication
userId	Yes, if the tokenId or userLogin parameter is not specified	The identifier (ID) of the user whose token needs to be prepared for authentication. The token must be assigned with the user on the resource to use this parameter.
userLogin	Yes, if the userId or tokenId parameter is not specified	The login of the user whose token needs to be prepared for authentication. The token must be assigned with the user on the resource to use this parameter.

Output Data

A question string (challenge) for a Protectimus ULTRA token, or the successful transaction completion message for Protectimus MAIL and SMS tokens is returned.

Output Data Structure for XML Format:

```
<responseHolder>
  <response>
    <challenge>{challenge_for_CR_token}</challenge>
  </response>
```

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

Output Data Structure for JSON Format:

```
{
 "responseHolder" : {
   "response" : {
     "challenge" : {challenge_for_CR_token}
   },
   "status" : "OK"
 }
}
```

Parameter Description

Parameter	Type	Description
challenge	Numeric	A number that a user has to enter in a token based on which this token will generate a response.

Please note:

- The specified token to be prepared may be of a type that does not require this action. This method is only required for the PROTECTIMUS SMS, MAIL, and ULTRA tokens. If it is called for a different type of token, the 6001 error code is returned, which means that a parameter is specified incorrectly.
- The challenge parameter in the answer appears only for the Protectimus ULTRA token.
- The specified token may not be assigned to the specified resource, in which case the 5002 error code with a problem's description is returned.

POST authenticate/token

This method performs the function of authenticating the specified token on the specified resource.

URL:

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

Input Data

Parameter	Mandatory Parameter	Description
resourceId	Yes, if the resourceName parameter is not specified	The identifier (ID) of the resource on which a token needs to be authenticated
resourceName	Yes, if the resourceId parameter is not specified	The name of the resource on which a token needs to be authenticated
tokenId	Yes	The identifier (ID) of the token that needs to be authenticated
otp	Yes	A one-time password entered by a user
ip	Partially	A user's IP address. It must be specified so as to perform verification with the geographic filter.

Output Data

The result "true" is returned if authentication is successful; the result "false" is returned if a token's authentication fails.

Output Data Structure for XML Format:

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

Output Data Structure for JSON Format:

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

Parameter Description

Parameter	Type	Description
result	Logical	The result of a token’s authentication on the specified resource

Please note:

- The token undergoing authentication may not be assigned to the specified resource.

POST authenticate/user-password

This method performs authentication of a user with a static password on the specified resource. To authenticate a user using this method, this user has to be assigned to the specified resource, and this user has to have an assigned password with which this user will be authenticated.

URL:

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

Input Data

Parameter	Mandatory Parameter	Description
resourceId	Yes, if the resourceName parameter is not specified	The identifier (ID) of the resource on which a user needs to be authenticated
resourceName	Yes, if the resourceId parameter is not specified	The name of the resource on which a user needs to be authenticated
userId	Yes, if the userLogin parameter is not specified	The identifier (ID) of the user that needs to be authenticated
userLogin	Yes, if the userId parameter is not specified	The login (username) of the user that needs to be authenticated
pwd	Yes	A password entered by a user
ip	Partially	A user's IP address. It must be specified so as to perform verification with the geographic filter.

Output Data

The result "true" is returned if authentication is successful; the result "false" is returned if a user's authentication fails.

Output Data Structure for XML Format:

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

Output Data Structure for JSON Format:

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

Parameter Description

Parameter	Type	Description
result	Logical	The result of a user’s authentication with a password on the specified resource

Please note:

- The user undergoing authentication may not be assigned to the specified resource.
- The user undergoing authentication may not have a password in the system.

In the cases described above, you will receive the 5002 error code with the description of the problem that occurred.

POST authenticate/user-token

This method performs authentication of a user with a one-time password on the specified resource. This user has to have a token, and both this user and this token have to be assigned to the specified resource.

URL:

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

Input Data

Parameters	Mandatory Parameter	Description
resourceId	Yes, if the resourceName parameter is not specified	The identifier (ID) of the resource on which a user needs to be authenticated
resourceName	Yes, if the resourceId parameter is not specified	The name of the resource on which a user needs to be authenticated
userId	Yes, if the userLogin parameter is not specified	The identifier (ID) of the user that needs to be authenticated
userLogin	Yes, if the userId parameter is not specified	The login (username) of the user that needs to be authenticated
otp	Yes	A one-time password entered by a user
ip	Partially	A user's IP address. It must be specified so as to perform verification with the geographic filter.

Output Data

The result "true" is returned if authentication is successful; the result "false" is returned if a user's authentication fails.

Output Data Structure for XML Format:

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

Output Data Structure for JSON Format:

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

Parameter Description

Parameter	Type	Description
result	Logical	The result of a user's authentication with a token on the specified resource

Please note:

- The user undergoing authentication may not have a token.
- The user undergoing authentication may not be assigned to the specified resource or be assigned to this resource without a token.

In the cases described above, you will receive the 5002 error code with the description of the problem that occurred.

POST authenticate/user-password-token

This method performs authentication of a user with a static and one-time password on the specified resource. This user and this token have to be assigned to the specified resource, and there has to be a static password assigned to this user. If this user's token is deactivated, the OTP authentication will not be performed; in this case, only this user's static password will be authenticated, and whether or not this user meets the filter parameters, if any.

URL:

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

Input Data

Parameters	Mandatory Parameter	Description
resourceId	Yes, if the resourceName parameter is not specified	The identifier (ID) of the resource on which a user needs to be authenticated
resourceName	Yes, if the resourceId parameter is not specified	The name of the resource on which a user needs to be authenticated
userId	Yes, if the userLogin parameter is not specified	The identifier (ID) of the user that needs to be authenticated
userLogin	Yes, if the userId parameter is not specified	The login (username) of the user that needs to be authenticated
otp	Yes	A one-time password entered by a user
ip	Partially	A user's IP address. It must be specified so as to perform verification with the geographic filter.
pwd	Yes	The password entered by user.

Output Data

The result "true" is returned if authentication is successful; the result "false" is returned if a user's authentication fails.

Output Data Structure for XML Format:

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

Output Data Structure for JSON Format:

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

Parameter Description

Parameter	Type	Description
result	Logical	The result of a user's authentication with a token and a password on the specified resource

Please note:

- The user undergoing authentication may not have an assigned token, or may have no static password.
- The user undergoing authentication may not be assigned with a token to the specified resource.

In the cases described above, you will receive the 5002 error code with the description of the problem that occurred.

Managing Resources (Projects)

A resource serves as a means to group users and provides flexible possibilities for delegating authorities and responsibilities. A resource may be a web project, a portal, an application, or a department of your employees. The chief system administrator may add other administrators to the system and assign them to specific resources. Such regular administrators may only perform actions within a resource to which they are assigned, but they may see all users and all tokens existing in the system, regardless of whether or not they are assigned to the resources under this administrator's management.

This Section's URL:

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

GET resources

This method allows you to obtain a list of your resources (up to 10 elements starting from the specified offset). By default, the offset is set at zero.

URL:

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

Input Data

Parameter	Mandatory Parameter	Description
start	No	The offset from which a list of resources should start. By default, the offset is set at 0.

Output Data

A list of an authorized user's resources is returned.

Output Data Structure for XML Format:

```
<responseHolder>
  <response>
    <resources>
      <resource>
        <creatorId>{resource_creator_identifier}</creatorId>
        <creatorUsername>{resource_creator_username}</creatorUsername>
        <failedAttemptsBeforeLock>
            {number_of_failed_authentication_attempts_before_locking}
        </failedAttemptsBeforeLock>
        <id>{resource_identifier}</id>
        <name>{resource_name}</name>
      </resource>
      . . .
    </resources>
  </response>
  <status>OK</status>
</responseHolder>
```

Output Data Structure for JSON Format:

```
{

   "responseHolder":
     {
       "response":
        {"resources":
          [
            {
               "creatorId":{resource_creator_identifier},
               "creatorUsername":"{resource_creator_username}",
```

```
                "failedAttemptsBeforeLock":
                   {number_of_failed_authentication_attempts_before_locking},
                "id":{resource_identifier},
                "name":"{resource_name}"
             },
             . . .
          ]
       },
   "status":"OK"
  }
}
```

Parameter Description

Parameter	Type	Description
creatorId	Numeric	The identifier (ID) of the administrator that created the resource
creatorUsername	String	The login (username) of the administrator that created the resource
failedAttemptsBeforeLock	Numeric	The number of failed authentication attempts, which, if exceeded, results in a user's being blocked.
id	Numeric	The identifier (ID) of the resource
name	String	The name of the resource

Please Note:

- You may erroneously specify an offset that exceeds the number of a customer's resources. In this case, an empty search result is returned.

GET resources/quantity

This method allows you to obtain information on the number of resources assigned to an administrator that submits a request.

URL:

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

Input Data
None.

Output Data
The number of an authorized user's resources (projects) is returned.

Output Data Structure for XML Format:

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

Output Data Structure for JSON Format:

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

Parameter Description

Parameter	Type	Description
quantity	Numeric	The number of an authorized user's resources

POST resources

This method allows you to create a new resource (project)

URL:

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

Input data

Parameter	Mandatory Parameter	Description
resourceName	Yes	The name of the resource created
failedAttemptsBeforeLock	No	The number of failed authentication attempts, which, if exceeded, results in a user's being blocked. The value of this parameter should be specified between 3 and 10. If this parameter is not specified, by default, it will be set at 5 attempts.

Output Data

The identifier (ID) of the resource created is returned.

Output Data Structure for XML Format:

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

Output Data Structure for JSON Format:

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

Parameter Description

Parameter	Type	Description
id	Numeric	The identifier (ID) of the resource created

Please note:
The number of resources (projects) that you may create depends on the service plan you select. If you need to create more resources, please select the desired or required number of resources by customizing your service plan.

GET resources/{id}

This method allows you to obtain information on a customer's specific resource.

URL:

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

Input Data

Parameter	Mandatory Parameter	Description
id	Yes	The identifier (ID) of the resource on which information needs to be obtained

Output Data

Information on the specified customer's resource is returned.

Output Date Structure for XML Format:

```
<responseHolder>
  <response>
    <resource>
         <creatorId>{resource_creator_identifier}</creatorId>
         <creatorUsername>{resource_creator_username}</creatorUsername>
         <failedAttemptsBeforeLock>
             {number_of_failed_authentication_attempts_before_locking}
         </failedAttemptsBeforeLock>
         <id>{resource_identifier}</id>
         <name>{resource_name}</name>
       </resource>
  </response>
  <status>OK</status>
</responseHolder>
```

Output Date Structure for JSON Format:

```
{
   "responseHolder":
    {
       "response":
        {"resource":
          [
             {
                "creatorId":{resource_creator_identifier},
                "creatorUsername":"{resource_creator_username}",
                "failedAttemptsBeforeLock":
{number_of_failed_authentication_attempts_before_locking},
                "id":{resource_identifier},
                "name":"{resource_name}"
             }
          ]
       },
```

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

Parameter Description

Parameter	Type	Description
creatorId	Numeric	The identifier (ID) of the administrator that created the resource
creatorUsername	String	The login (username) of the administrator that created the resource
failedAttemptsBeforeLock	Numeric	The number of unsuccessful authentication attempts, which if exceeded results in a user's being blocked.
id	Numeric	The identifier (ID) of the resource
name	String	The name of the resource

Please note:

- You may erroneously an incorrect identifier (ID) or the identifier (ID) of a resource that you are not assigned to. In this case, you will receive an error message.

PUT resources/{id}

This method allows you to edit information on your resource (project).

URL:

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

Input Data

Parameter	Mandatory Parameter	Description
id	Yes, if old name of resource is not specified	The identifier (ID) of the resource whose information you want to change
resourceName	Yes, if id is not specified, or if resource name needs to be changed	The resource's old or new name. If the resource's identifier is not specified, search will be performed with the specified name. To change the name of a resource, specify its identifier (ID) and submit the new name in this parameter.
failedAttemptsBeforeLock	No	The number of a user's failed authentication attempts, which, if exceeded, results in a user's being blocked. The permissible range of values: from 3 to 10. If this parameter is not specified, it will remain unchanged for this resource.

Output Data

Information on the resource edited is returned.

Output Date Structure for XML Format:

```
<responseHolder>
  <response>
    <resource>
      <creatorId>{creator_identifier}</creatorId>
      <creatorUsername>{creator_username}</creatorUsername>
      <failedAttemptsBeforeLock>
         {new_failed_authentication_attempts_limit}
      </failedAttemptsBeforeLock>
      <id>{resource_identifier}</id>
      <name>{new_resource_name}</name>
    </resource>
  </response>
  <status>OK</status>
</responseHolder>
```

Output Date Structure for JSON Format:

```
{
 "responseHolder" : {
   "response" : {
```

```
    "resource" : {
      "creatorId" : {creator_identifier},
      "creatorUsername" : "{creator_username}",
      "failedAttemptsBeforeLock" : {new_failed_authentication_attempts_limit},
      "id" : {resource_identifier},
      "name" : "{new_resource_name}"
    }
   },
   "status" : "OK"
 }
}
```

Parameter Description

Parameter	Type	Description
creatorId	Numeric	The identifier (ID) of the administrator that created the resource
creatorUsername	String	The login (username) of the administrator that created the resource
failedAttemptsBeforeLock	Numeric	The number of failed authentication attempts, which, if exceeded, results in a user's being blocked.
id	Numeric	The identifier (ID) of the resource
name	String	The name of the resource

Please note:

- You may erroneously specify an incorrect identifier (ID) of a resource, which will result in making changes in the information on another one of your resources.

DELETE resources/{id}

Deleting Your Resource

A resource may be deleted only by the administrator that created it or by the chief system administrator.

URL:

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

Input data

Parameter	Mandatory Parameter	Description
id	Yes	The identifier (ID) of the resource deleted

Output Data

Information on the resource deleted is returned.

Output Data Structure for XML Format:

```
<responseHolder>
  <response>
    <resource>
      <creatorId>{creator_identifier}</creatorId>
      <creatorUsername>{creator_username}</creatorUsername>
       <failedAttemptsBeforeLock>
         {failed_authentication_attempts_limit}
      </failedAttemptsBeforeLock>
      <id>{resource_identifier}</id>
      <name>{resource_name}</name>
    </resource>
  </response>
  <status>OK</status>
</responseHolder>
```

Output Data Structure for JSON Format:

```
{
 "responseHolder" : {
  "response" : {
   "resource" : {
    "creatorId" : {creator_identifier},
    "creatorUsername" : "{creator_username}",
    "failedAttemptsBeforeLock" : {failed_authentication_attempts_limit},
    "id" : {resource_identifier},
    "name" : "{resource_name}"
   }
  },
  "status" : "OK"
 }
}
```

Parameter Description

Parameter	Type	Description
creatorId	Numeric	The identifier (ID) of the administrator that created the resource
creatorUsername	String	The login (username) of the administrator that created the resource
failedAttemptsBeforeLock	Numeric	The number of unsuccessful authentication attempts, which if exceeded results in a user's being blocked.
id	Numeric	The identifier (ID) of the resource
name	String	The name of the resource

Please note:

- You may not be authorized to delete a resource, if you are not the creator of this resource or the chief system administrator.

POST assign/user

This method assigns a user with the specified identifier (ID) to the specified resource.

Use this method if you want to authenticate this user on a resource with only static passwords. If you want to verify both a static password and a dynamic (OTP) password, or only an OTP password, you need to assign a user together with a token to a resource (the assign/token-with-user method) or assign a token (the assign/token method).

URL:

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

Input Data

Parameter	Mandatory Parameter	Description
resourceId	Yes, if resourceName parameter is not specified	The identifier (ID) of the resource to which a user needs to be assigned
resourceName	Yes, if resourceId parameter is not specified	The name of the resource to which a user needs to be assigned
userId	Yes, if userLogin parameter is not specified	The identifier (ID) of the user that needs to be assigned to a resource
userLogin	Yes, if userId parameter is not specified	The login of the user that needs to be assigned to a resource

Output Data

None. (The response contains either a successful operation completion message or standard error messages.)

POST assign/token

This method assigns a token to a resource.

After the successful execution of this method, you will be able to authenticate an OTP from this token on a resource, without tying authentication to a specific user. Use this method if you do not want to store information on users in our system. But, in this case, you will only be able to authenticate an OTP (and a PIN, if assigned).

URL:

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

Input Data

Parameter	Mandatory Parameter	Description
resourceId	Yes, if resourceName parameter is not specified	The identifier (ID) of the resource to which a token needs to be assigned
resourceName	Yes, if resourceId parameter is not specified	The name of the resource to which a token needs to be assigned
tokenId	Yes	The identifier (ID) of the token that needs to be assigned to a resource

Output Data

A successful operation completion message is returned.

Please note:

Even if a token is assigned to a user, and this user is assigned to the same resource as this token, you will not be able to authenticate this user with an OTP on this resource unless you assigned this user to this resource WITH this token. To do that, use the assign/user-token method or perform the procedure for assigning a user with a token through the web interface.

POST assign/user-token

This method assigns a user and a token to a resource.

After the successful execution of this method, you will be able to authenticate a user on a resource with a one-time password or with a combination of a one-time password and a static password.

URL:

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

Input Data

Parameter	Mandatory Parameter	Description
resourceId	Yes, if resourceName parameter is not specified	The identifier (ID) of the resource to which a user with a token needs to be assigned
resourceName	Yes, if resourceId parameter is not specified	The name of the resource to which a user with token needs to be assigned
userId	Yes, if userLogin parameter is not specified	The identifier (ID) of the user that needs to be assigned to a resource with a token
userLogin	Yes, if userId parameter is not specified	The login of the user that needs to be assigned to a resource with a token
tokenId	Yes	The identifier (ID) of the token that needs to be assigned to a resource with a user

Output Data

A successful operation completion message is returned.

Please note:
A user may have several tokens; but a token can only be assigned to one user.

POST assign/token-with-user

This method assigns a token to a resource with the user to which this token is assigned.

It performs the same operation as the assign/user-token method, but it does not required that the user's identifier (ID) be specified since it is the user to which this token is assigned.

URL:

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

Input Data

Parameter	Mandatory Parameter	Description
resourceId	Yes, if resourceName parameter is not specified	The identifier (ID) of a resource
resourceName	Yes, if resourceId parameter is not specified	The name of the resource to which a token with user needs to be assigned
tokenId	Yes	The identifier (ID) of a token

Output Data
A successful operation completion message is returned.

Please note:

- For the successful execution of this method, a token must be assigned to a user.

POST unassign/user

This method unassigns a user from a resource.

After the successful execution of this method, this user will not be able to be authenticated on this resource.

URL:

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

Input Data

Parameter	Mandatory Parameter	Description
resourceId	Yesm if resourceName parameter is not specified	The identifier (ID) of a resource
resourceName	Yes, if resourceId parameter is not specified	The name of the resource
userId	Yes, if userLogin parameter is not specified	The identifier (ID) of a user
userLogin	Yes, if userId parameter is not specified	The login of the user

Output Data
A successful operation completion message is returned.

Please note:

- A user may not be assigned to the specified resource.

POST unassign/token

This method unassigns a token from a resource. After the successful execution of this method, this token will not be able to be authenticated on this resource.

URL:

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

Input Data

Parameter	Mandatory Parameter	Description
resourceId	Yes, if resourceName parameter is not specified	The identifier (ID) of a resource
resourceName	Yes, if resourceId parameter is not specified	The name of the resource
tokenId	Yes	The identifier (ID) of a token

Output Data

A successful operation completion message is returned.

Please note:

- A token may not be assigned to the specified resource.

POST unassign/token-with-user

This method unassigns a token from a resource with the user to which this token is assigned.

URL:

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

Input Data

Parameter	Mandatory Parameter	Description
resourceId	Yes, if resourceName parameter is not specified	The identifier (ID) of a resource
resourceName	Yes, if resourceId parameter is not specified	The name of the resource
tokenId	Yes	The identifier (ID) of a token

Output Data

A successful operation completion message is returned.

Please note:

- A token and/or a user may not be assigned to a resource (in conjunction with or separately from each other)
- A token may not be assigned to this user.

POST unassign/user-token

This method unassignes a user and a token from a resource.

URL:

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

Input Data

Parameter	Mandatory Parameter	Description
resourceId	Yes, if resourceName parameter is not specified	The identifier (ID) of a resource
resourceName	Yes, if resourceId parameter is not specified	The name of the resource
userId	Yes, if userLogin parameter is not specified	The identifier (ID) of a user
UserLogin	Yes, if userId parameter is not specified	The login of the user
tokenId	Yes	The identifier (ID) of a token

Output Data
A successful operation completion message is returned.

Please note:

- For this method to be executed correctly, a user must be assigned to a resource with a token.

Managing Tokens

We divide all tokens into hardware (physical) tokens and software (virtual) tokens. This classification is based on the peculiarities of entering and using the secret key for a specific type of tokens. Software tokens include the following: Protectimus SMART, Google Authenticator, Protectimus SMS, and MAIL tokens; all the remaining types of our tokens are hardware tokens.

The tokens provided in our system may be used not only by your users, but also by you or your administrators for protecting access to Protectimus. Therefore, a token assigned to an administrator may only be managed by the administrator to which it is assigned. For other tokens, the same approach applies as for all other objects in the system: any user can edit them, but only the creator or the chief system administrator can delete them.

If a user loses its token, but you want to provide urgent access to this user, you will only need to deactivate this token, in which case this token will not be involved in the user authentication process.
If an administrator loses a token, this administrator will need to use the backup access mechanism to verify this administrator’s other authenticators. After that, a request for deactivating a token will be created. Only the chief system administrator or the technical support service will be able to satisfy this request.

Any tokens that work based on the standard OATH algorithms may be used in our system. It may be done by adding a universal token. Of course, in this case, you will need to have sufficient knowledge about your token. We support several types of popular tokens from other manufacturers, which significantly simplifies your task. We continually improve our service and expand the range of tokens that we support.

This Section’s URL:

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

GET secret-key/google-authenticator

This method allows you to obtain the secret key for Google Authenticator, which will be used to generate an OTP. This one-time key is transmitted to the device and the server, after which operation this key should be known only to the token and the server which authenticates the OTP from this token.

URL::

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

Input Data
None.

Output Data
The secret key for Google Authenticator is returned.

Output Data structure for XML Format:

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

Output Data structure for JSON Format:

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

Parameter Description

Parameter	Type	Description
key	String	The secret key required for Google Authenticator to create a token

GET secret-key/protectimus-smart

This method allows you to obtain the secret key required to create a Protectimus SMART token. This key is used to generate an OTP and must be transmitted as a one-time key to the device and the server at the token creation stage, after which operation it should only be known to these two parties. Also, this token contains the checksum so that a user cannot create a token with an invalid key on this user's device.

URL:

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

Input Data
None.

Output Data
The secret key that can be used to create a Protectimus SMART token is returned.

Output Data Structure for XML Format:

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

Output Data Structure for JSON Format:

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

Parameter Description

Parameter	Type	Description
key	String	The secret key required to create a Protectimus SMART token

GET tokens

This method allows you to obtain a list of your tokens (10 elements starting from the specified offset).

URL:

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

Input Data

Parameter	Mandatory Parameter	Description
start	No	The offset starting from which a list of 10 tokens will be obtained. By default, the offset is set at 0.

Output Data

A list of tokens with information on each token is returned.

Output Data Structure for XML Format:

```
<responseHolder>
  <response>
    <tokens>
       <token>
         <apiSupport>{support_through_API}</apiSupport>
         <creatorId>{creator_identifier}</creatorId>
         <creatorUsername>{creator_username}</creatorUsername>
         <enabled>{enabled_or_disabled}</enabled>
         <id>{token_identifier}</id>
         <serialNumber>{token_serial_number}</serialNumber>
         <type>{token_type}</type>
       </token>
       . . .
    </tokens>
  </response>
  <status>OK</status>
</responseHolder>
```

Output Data Structure for JSON Format:

```
{
   "responseHolder": {
      "response":{
         "tokens":
         [
            {
               "apiSupport":{support_through_API},"creatorId":{creator_identifier},
               "creatorUsername":"{creator_username}","enabled":{enabled_or_disabled},
               "id":{token_identifier},"serialNumber":"{token_serial_number}",
               "type":"{token_type}"
```

```
            },
             . . .
          ]},
       "status":"OK"
   }
}
```

Parameter Description

Parameter	Type	Description
apiSupport	Logical	It shows whether or not a token supports authentication through the API. If this parameter's result is "false", a token cannot be authenticated through the API.
creatorId	Numeric	The identifier (ID) of the administrator that created a token
creatorUsername	String	The username (login) of the administrator that created a token
enabled	Logical	It shows whether a token is enabled or disabled. If a token is disabled, any OTP transmitted will ALWAYS return a positive response. In other words, when a token is disabled, it is not involved in the authentication process, and the OTP is not verified (authenticated). It is useful when a user cannot use its token for any reason. To give this user access, you will simply need to disable this user's token.
id	Numeric	The identifier (ID) of a token
serialNumber	String	The serial number of a token. It is given on the back side of the device. Or, it is the email address for a Mail-token and the phone number for an SMS-token.
type	String (Enumeration)	The type of a token. The system offers the following types of tokens: PROTECTIMUS – hardware tokens Protectimus ONE PROTECTIMUS_SLIM - hardware tokens SLIM PROTECTIMUS_ULTRA – tokens that work based on Challenge-Response algorithm PROTECTIMUS_SMART – tokens that are installed on Android or iOs mobile devices. GOOGLE_AUTHENTICATOR – a token from Google for mobile devices SAFENET_ETOKEN_PASS – a token from SafeNet YUBICO_OATH_MODE – a token from Yubico UNIFY_OATH_TOKEN – a universal token that works based on the OATH standards SMS – delivery of one-time passwords via SMS MAIL – delivery of one-time passwords via email

Please note:

- When the specified offset exceeds the number of tokens, an empty search result is returned.

GET tokens/quantity

This method allows you to obtain information on the number of your tokens.

URL:

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

Input Data
None.

Output Data
A customer’s number of tokens is returned.

Output Data Structure for XML Format:

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

Output Data Structure for JSON Format:

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

Parameter Description

Parameter	Type	Descriptions
quantity	Numeric	A customer’s number of tokens

POST tokens/software

This method allows you to create a software token.

The following types are software tokens: Protectimus SMART, Google Authenticator, as well as SMS and Mail tokens.

URL:

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

Input Data

Parameter	Mandatory Parameter	Description
userId *	No	The identifier (ID) of the user to which the token created should be assigned
userLogin *	No	The login (username) of the user to which the token created should be assigned
type	Yes	It specifies the type of a token. This method applies only to software tokens: PROTECTIMUS_SMART GOOGLE_AUTHENTICATOR, SMS, MAIL
serial	Yes	The serial number of a token. It serves the purpose of identifying a token in the outside world. For an SMS-token, it is a phone number; for a MAIL-token it is an email address. For all other types of tokens, this parameter may be any unique string.
name	No	The name of a token.
secret	Yes	The secret key of a token. It should be in the format of a Base32 string. For Google Authenticator tokens, the length of a string should be at least 16 characters. For Protectimus SMART tokens, the length of the key should be 18 characters. The secret key for these types of tokens can be obtained using the following methods: *secret-key/google-authenticator и secret-key/protectimus-smart* **Important!** For SMS and Mail tokens, the secret parameter should be equivalent to the value of the otp parameter.
otp	Yes	A one-time password from this token. When creating an SMS or a Mail token, this parameter should be the same as the *secret* parameter. At this stage, a user will NOT receive a one-time password, and you will not be able to verify whether this user specified a valid number. To do that, you need to create an SMS or a Mail token using this method specifying the same random string as the *secret* and *otp* parameter; after that, you need to call the *prepare* method to send the password and authenticate this password using one of the authentication methods in the *auth-service* section. Other types of tokens allow you to receive a one-time password immediately. This one-time password should be specified as this parameter.
otpLength	Yes, for a Protectimus	The length of the one-time password received. The

	SMART token	permissible length is 6 or 8 characters.
keyType	Yes, for a Protectimus SMART token	The mode in which a one-time password is generated. Permissible types: TOTP, HOTP. TOTP – time-based OTP generation HOTP – counter-based OTP generation
pin	No	A PIN-code that a user has to enter in the input field, together with a one-time password. This password and the PIN should be entered as one string without spaces or any other characters between them. The position of the PIN is determined by the pinOtpFormat parameter. The length of a PIN-code is 4 characters.
pinOtpFormat	Yes, if a pin is specified	The format of a PIN-code. It determines the position of a PIN code in the input field: before a one-time password or after it. Permissible format types: PIN_BEFORE_OTP and PIN_AFTER_OTP.

* The userId and userLogin parameters identify the user to which a token needs to be assigned. User search is performed with the use of only one of these two parameters. First, the userId parameter is used to perform user search, and if a user is not found – the userLogin parameter is used. Consequently, if you specify the identifier (ID) of one user, and the login (username) of another user, a token will be assigned to the user whose ID you specify.
These parameters are not mandatory; if they are not specified, a token will not be assigned to any user. A token can be assigned to a user at any time using the relevant methods from this section.

Output Data
The Identifier (ID) of the token created is returned.

Output Data Structure for XML Format:

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

Output Data Structure for JSON Format:

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

Parameter Description

Parameter	Type	Description
id	Numeric	The identifier (ID) of the token created

Please note:

- Your service plan may not allow you to add more entities.

POST tokens/hardware

This method allows you to create a hardware token.

The following types are hardware tokens: PROTECTIMUS, PROTECTIMUS_SLIM, SAFENET_ETOKEN_PASS, PROTECTIMUS_ULTRA, and YUBICO_OATH_MODE.

URL:

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

Input Data

Parameter	Mandatory Parameter	Description
userId *	No	The identifier (ID) of the user to which the token created should be assigned
userLogin *	No	The login (username) of the user to which the token created should be assigned
type	Yes	It specifies the type of a token. This method applies only to software tokens: PROTECTIMUS - tokens Protectimus ONE PROTECTIMUS_SLIM - tokens SLIM от Protectimus PROTECTIMUS_ULTRA - tokens Protectimus that work based on the Challenge-Response algorithm YUBICO_OATH_MODE - tokens Yubico that work based on the OATH standards SAFENET_ETOKEN_PASS - tokens SafeNet that work based on the HOTP algorithm
serial	Yes	The serial number of a token. It is usually given on the back side of a device.
secret	Yes, if a token is not ordered from Protectimus	The secret key of a token. This key is embedded into a token and used to generate a one-time password; it should not be known to any party except the token and the server from which an OTP will be authenticated. If tokens are ordered from Protectimus, we will service the keys ourselves in the strictest confidentiality, and you will not need to specify this parameter.
name	No	The name of a token
otp	Yes	The one-time password from a token. It is required to confirm that a token exists. For SAFENET_ETOKEN_PASS tokens, you need to specify two OTP’s separated by a comma; for example: “147852,963258”. This is necessary to determine the counter’s offset.
existed	Yes	It shows whether or not you create an existing token (ordered from Protectimus). Permissible values: “true” or “false”.
pin	No	A PIN-code that a user has to enter in the input field, together with a one-time password. This password and the PIN should be entered as one string without spaces or any other characters between them. The position of a PIN-code is determined by the pinOtpFormat parameter.

		The length of a PIN-code is 4 characters.
pinOtpFormat	Yes, if the pin parameter is specified	The format of a PIN-code. It determines the position of a PIN code in the input field: before a one-time password or after it. Permissible format types: PIN_BEFORE_OTP and PIN_AFTER_OTP.

* The userId and userLogin parameters identify the user to which a token needs to be assigned. User search is performed with the use of only one of these two parameters. First, the userId parameter is used to perform user search, and if a user is not found – the userLogin parameter is used. Consequently, if you specify the identifier (ID) of one user, and the login (username) of another user, a token will be assigned to the user whose ID you specify.
These parameters are not mandatory; if they are not specified, a token will not be assigned to any user. A token can be assigned to a user at any time using the relevant methods from this section.

Output Data
The Identifier (ID) of the token created is returned.

Output Data Structure for XML Format:

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

Output Data Structure for JSON Format:

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

Parameter Description

Parameter	Type	Description
id	Numeric	The identifier (ID) of the token created

Please note:

- You may specify a token’s secret key in an incorrect format, which will result in the inability to generate the correct OTP on the server and authenticate this token.

- The otp parameter for tokens that work based on the HOTP algorithm should include two consequtive values separated by a comma so that Protectimus can determine the position of the counter.

GET tokens/{id}

This method allows you to obtain information on your token.

URL:

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

Input Data

Parameter	Mandatory Parameter	Description
id	Yes	The identifier (ID) of a token

Output Data

Information on a token is returned.

Output Data Structure for XML Format:

```
<responseHolder>
  <response>
    <token>
      <apiSupport>{support_through_API}</apiSupport>
      <creatorId>{creator_identifier}</creatorId>
      <creatorUsername>{creator_username}</creatorUsername>
      <enabled>{token_enabled_or_not}</enabled>
      <id>{token_identifier}</id>
      <name>{token_name}</name>
      <serialNumber>{serial_number}</serialNumber>
      <type>{token_type}</type>
    </token>
  </response>
  <status>OK</status>
</responseHolder>
```

Output Data Structure for JSON Format:

```
{
   "responseHolder":{
      "response":{
         "token":{
            "apiSupport":{support_through_API}, "creatorId":{creator_identifier},
            "creatorUsername":"{creator_username}", "enabled":{token_enabled_or_not},
            "id":{token_identifier},"name":{token_name},"serialNumber":"{serial_number}",
            "type":"{token_type}"
         }
      },
      "status":"OK"
   }
}
```

Parameter Description

Parameter	Type	Description
apiSupport	Logical	It shows whether or not a token supports authentication through the API. If this parameter's result is "false", a token cannot be authenticated through the API.
creatorId	Numeric	The identifier (ID) of the administrator that created a token
creatorUsername	String	The username (login) of the administrator that created a token
enabled	Logical	It shows whether a token is enabled or disabled. If a token is disabled, any OTP transmitted will ALWAYS return a positive response. In other words, when a token is disabled, it is not involved in the authentication process, and the OTP is not verified (authenticated). It is useful when a user cannot use its token for any reason. To give this user access, you will simply need to disable this user's token.
id	Numeric	The identifier (ID) of a token
name	String	The name of a token
serialNumber	String	The serial number of a token. It is given on the back side of the device. Or, it is the email address for a Mail-token and the phone number for an SMS-token.
type	String (Enumeration)	The type of a token. You will find the description of token types in the methods for creating software and hardware tokens.

PUT tokens/{id}

This method allows you to edit information on a token.

Please note: you cannot edit any information on a token that is assigned to another administrator.

URL:

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

Input data

Parameter	Mandatory Parameter	Description
name	No	The new name of a token
enabled *	No	Enabled or disabled token
apiSupport *	No	Support of authentication through the API

* These parameters are described in detail above in the sections on other API methods, for example, in the section on the *GET tokens/{id}* method.

Output Data
Information on the token edited is returned. The output data structure and parameter value is the same as those for the *GET tokens/{id}* method.

DELETE tokens/{id}

This method allows you to delete a token.

Please note: you cannot delete a token that is assigned to another administrator.

URL:

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

Input data

Parameter	Mandatory Parameter	Description
id	Yes	Identifier (ID) of the token deleted

Output Data

Information on the token deleted is returned. The output data structure and parameter value is the same as those for the *GET tokens/{id}* method.

POST tokens/{id}/unassign

This method allows you to unassign a token from a user.

URL:

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

Input data

Parameter	Mandatory Parameter	Description
id	Yes	Identifier (ID) of the token to be unassigned

Output Data
A successful operation completion message or standard error messages are returned.

Managing Users

This section is devoted to working with and managing users. You may store a certain set of data on your users in our system, but the key parameter is a user’s login (username).
The system offers a user self-service mechanism. It is set up for every resource separately; it can be done in the Self-Service tab on the page containing a resource’s detailed information.

GET users

This method allows you to obtain a list of users (10 items starting from the specified offset).

URL:

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

Input Data

Parameter	Mandatory Parameter	Description
start	No	The offset starting from which the next 10 items should be returned.

Output data
A list of users is returned.

Output Data Structure for XML Format:

```
<responseHolder>
  <response>
    <users>
      <user>
        <apiSupport>{support_through_API}</apiSupport>
        <creatorId>{creator_identifier}</creatorId>
        <creatorUsername>{creator_username}</creatorUsername>
        <email>{email_address}</email>
        <firstName>{user_name}</firstName>
        <secondName>{user_last_name}</secondName>
        <hasTokens>{has_assigned_tokens}</hasTokens>
        <id>{user_identifier}</id>
        <login>{login}</login>
        <phoneNumber>{user_phone_number}</phoneNumber>
      </user>
      . . .
    </users>
  </response>
  <status>OK</status>
</responseHolder>
```

Output Data Structure for JSON Format:

```
{
   "responseHolder":{
      "response":{
         "users":[{
            "apiSupport":{support_through_API},"creatorId":{creator_identifier},
            "creatorUsername":"{creator_username}","email":"{email_address}",
            "hasTokens":{has_assigned_tokens},"id":{user_identifier},
            "login":"{login}", "phoneNumber":"{user_phone_number}",
```

```
          "secondName":"{user_last_name}"
        },
          . . .
        ]
      },
      "status":"OK"
    }
}
```

Parameter Description

Parameter	Type	Description
apiSupport	Logical	Support of authentication through the API. If this parameter's result is "false", a user cannot be authenticated through the API.
creatorId	Numeric	The identifier (ID) of the creator
creatorUsername	String	The username (login) of the creator
email	String	Email address
hasTokens	Logical	It shows whether or not a user has assigned tokens.
id	Numeric	The identifier (ID) of a user
login	String	The username (login) of a user
phoneNumber	String	The phone number of a user
secondName	String	The last name of a user

GET users/quantity

This method allows you to obtain information on a customer’s number of users.

URL:

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

Input Data
None.

Output Data
A customer’s number of users is returned.

Output Data Structure for XML Format:

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

Output Data Structure for JSON Format:

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

Parameter Description

Parameter	Type	Description
quantity	Numeric	The number of a customer’s users

POST users

This method allows you to add users.

URL:

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

Input Data

Parameter	Mandatory Parameter	Description
login	Yes	The username (login) of a user. It may only contain Latin characters, digits, and symbols: @ _ . - . The permissible length is from 5 to 30 characters. This field should be unique within your company's account. This parameter will be used to perform search when authenticating a user. You can also use a user's email address or phone number as this user's username (login), and decide whether or not you want to fill in the relevant parameters.
email	No	The email address of a user
phoneNumber	No	The phone number of a user. It should be entered in the international format.
password	No	User Password. You can transmit a password as plain text. But, if you only have hashed text, you can use two methods: 1. Use the /users/password method from the user-service section to enter the hash and specify how Protectimus can receive it. In this case, a user can use their password to access all of the Protectimus services. 2. Transmit the hash to Protectimus using this method or a similar method, and prior to every authentication independently hash the password entered by a user and submit the resulting hash for authentication. In this case, you will only be able to work with the API methods to which you should send your password independently, and a user will not be able to use the services for which they will be required to enter their password to login.
firstName	No	The first name of a user. The length of this field is from 1 to 50 characters.
secondName	No	The last name of a user. The length of this field is from 1 to 50 characters.
apiSupport	No	Support of authentication through the API. Permissible values: "true" or "false". By default, this parameter's result is set as "true", i.e. a user can be authenticated through the API.

Output Data
The identifier (ID) of the user created is returned.

Output Data Structure for XML Format:

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

Output Data Structure for JSON Format:

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

Parameter Description

Parameter	Type	Description
id	Numeric	The identifier (ID) of the user created

GET users/{id}

This method allows you to obtain information on a user.

URL:

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

Input Data

Parameter	Mandatory Parameter	Description
id	Yes	The identifier (ID) of the user whose information needs to be obtained

Output Data

The information on the requested user is returned.

Output Data Structure for XML Format:

```
<responseHolder>
  <response>
    <user>
          <apiSupport>{support_through_API}</apiSupport>
          <creatorId>{creator_identifier}</creatorId>
          <creatorUsername>{creator_username}</creatorUsername>
          <email>{email_address}</email>
          <firstName>{user_first_name}</firstName>
          <secondName>{user_last_name}</secondName>
          <hasTokens>{has_assigned_tokens}</hasTokens>
          <id>{user_identifier}</id>
          <login>{login}</login>
          <phoneNumber>{user_phone_number}</phoneNumber>
      </user>
  </response>
  <status>OK</status>
</responseHolder>
```

Output Data Structure for JSON Format:

```
{
   "responseHolder":{
      "response":{
         "user":{
            "apiSupport":{support_through_API},"creatorId":{creator_identifier},
            "creatorUsername":"{creator_username}","email":"{email_address}",
            "hasTokens":{has_assigned_tokens},"id":{user_identifier},
            "login":"{login}", "phoneNumber":"{user_phone_number}",
           "secondName":"{user_last_name}"
         }
```

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


Parameter Description

Parameter	Type	Description
apiSupport	Logical	Support of authentication through the API. If this parameter's result is "false", a user cannot be authenticated through the API.
creatorId	Numeric	The identifier (ID) of the creator
creatorUsername	String	The username (login) of the creator
email	String	Email address
hasTokens	Logical	It shows whether or not a user has assigned tokens.
id	Numeric	The identifier (ID) of a user
login	String	The username (login) of a user
phoneNumber	String	The phone number of a user
secondName	String	The last name of a user

PUT users/{id}

This method allows you to edit information on a user.

URL:

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

Input data

Parameter	Mandatory Parameter	Description
login	Yes	The username (login) of a user. It may only contain Latin characters, digits, and symbols: @ _ . - . The permissible length is from 5 to 30 characters. This field should be unique within your company's account. This parameter will be used to perform search when authenticating a user. You can also use a user's email address or phone number as this user's username (login), and decide whether or not you want to fill in the relevant parameters.
email	No	The email address of a user
phoneNumber	No	The phone number of a user. It should be entered in the international format.
password	No	User Password. You can transmit a password as plain text. But, if you only have hashed text, you can use two methods: 1. Use the /users/password method from the user-service section to enter the hash and specify how Protectimus can receive it. In this case, a user can use their password to access all of the Protectimus services. 2. Transmit the hash to Protectimus using this method or a similar method, and prior to every authentication independently hash the password entered by a user and submit the resulting hash for authentication. In this case, you will only be able to work with the API methods to which you should send your password independently, and a user will not be able to use the services for which they will be required to enter their password to login.
firstName	No	The first name of a user. The length of this field is from 1 to 50 characters.
secondName	No	The last name of a user. The length of this field is from 1 to 50 characters.
apiSupport	No	Support of authentication through the API. Permissible values: "true" or "false". By default, this parameter's result is set as "true", i.e. a user can be authenticated through the API.

* The values of these parameters are the same as those for the *POST users* method.

Output Data
Edited information on a user is returned. The output data structure and parameter descriptions are the same as those for the *POST users* and *GET users/{id}* methods.

POST users/password

This method allows you to assign or edit a user's password in a secure format.
Most likely, user passwords are stored in your database as hashed data, and you do not know them. If you want to use these passwords for authenticating users through Protectimus, you need to use this method. You transmit a user's hashed password and the rules for password hashing so that Protectimus could perform the same conversion to get the same hash required for authentication.

URL:

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

Input data

Parameter	Mandatory Parameter	Description
Id	Yes, if login parameter is not specified	User's identifier for which a password needs to be set or changed.
login	Yes, if id parameter is not specified	User's login for which a password needs to be set or changed.
rawPassword	Yes	User's password hash in the HEX format.
rawSalt	No	Salt used for password hashing.
encodingType	Yes	Method used for password hashing. The following values are acceptable: PLAIN – password was not hashed and was provided as plain text; MD5 - MD5 algorithm was used SHA - SHA-1 algorithm was used SHA256 - SHA-256 algorithm was used
encodingFormat	Yes	Format of the hashed string (password and salt). During user authentication, Protectimus will replace the word "PASS" in this string with the password entered by the user, and the PLAIN_SALT word will be replaced with the salt transmitted by you via this method. The remaining characters will be kept intact. The resulting string will be converted with the encodingType algorithm and compared with the rawPassword for user authentication.

Output Data
Information on a user is returned. The output data structure and parameter descriptions are the same as those for the *POST users* and *GET users/{id}* methods.

DELETE users/{id}

This method allows you to delete a user.

Like other items in the system, a user may only be deleted by the administrator that created this user or by the chief system administrator.

URL:

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

Input Data

Parameter	Mandatory Parameter	Description
id	Yes	The identifier (ID) of the user deleted

Output Data

Information on the user deleted is returned. The response structure and parameter values are the same as those for the GET users/{id} method used for obtaining information on a user.

GET /users/{id}/tokens

This method allows you to obtain a list of a user's tokens (10 elements starting from the specified offset).

URL:

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

Input Data

Parameter	Mandatory Parameter	Description
id	Yes	The identifier (ID) of a user

Output Data
A list containing information on tokens assigned to the specified customer is returned.

Output Data Structure for XML Format:

```
<responseHolder>
  <response>
    <tokens>
       <token>
         <apiSupport>{support_through_API}</apiSupport>
         <creatorId>{creator_identifier}</creatorId>
         <creatorUsername>{creator_username}</creatorUsername>
         <enabled>{enabled_or_disabled}</enabled>
         <id>{token_identifier}</id>
         <serialNumber>{token_serial_number}</serialNumber>
         <type>{token_type}</type>
       </token>
       . . .
    </tokens>
  </response>
  <status>OK</status>
</responseHolder>
```

Output Data Structure for JSON Format:

```
{
   "responseHolder": {
      "response":{
         "tokens":
         [
            {
               "apiSupport":{support_through_API},"creatorId":{creator_identifier},
               "creatorUsername":"{creator_username}","enabled":{enabled_or_disabled},
               "id":{token_identifier},"serialNumber":"{token_serial_number}",
               "type":"{token_type}"
```

```
            },
             . . .
          ]},
      "status":"OK"
   }
}
```

Parameter Description

Parameter	Type	Description
apiSupport	Logical	It shows whether or not a token supports authentication through the API. If this parameter's result is "false", a token cannot be authenticated through the API.
creatorId	Numeric	The identifier (ID) of the administrator that created a token
creatorUsername	String	The username (login) of the administrator that created a token
enabled	Logical	It shows whether a token is enabled or disabled. If a token is disabled, any OTP transmitted will ALWAYS return a positive response. In other words, when a token is disabled, it is not involved in the authentication process, and the OTP is not verified (authenticated). It is useful when a user cannot use its token for any reason. To give this user access, you will simply need to disable this user's token.
id	Numeric	The identifier (ID) of a token
serialNumber	String	The serial number of a token. It is given on the back side of the device. Or, it is the email address for a Mail-token and the phone number for an SMS-token.
type	String (Enumeration)	The type of a token. You will find detailed descriptions of token types in the methods contained in the relevant section, for example, the *tokens/hardware* and *tokens/software* methods.

GET users/{id}/tokens/quantity

This method allows you to obtain information on the number of tokens assigned to a user.

URL:

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

Input Data

Parameter	Mandatory Parameter	Description
id	Yes	The identifier (ID) of the user whose number of tokens needs to be obtained

Output Data

The number of tokens assigned to a user is returned.

Output Data Structure for XML Format:

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

Output Data Structure for JSON Format:

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

Parameter Description

Parameter	Type	Description
quantity	Numeric	The number of tokens assigned to the user with the specified identifier (ID)

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

This method assigns a token to a user.

An unlimited number of tokens may be assigned to a user, but a token may only be assigned to one user.

URL:

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

Input Data

Parameter	Mandatory Parameter	Description
userId	Yes	The identifier (ID) of the user to which a token needs to be assigned
tokenId	Yes	The identifier (ID) of the token which needs to be assigned to the specified user

Output Data
A successful operation completion message or standard error messages are returned.

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

This method unassigns a token from a user.

URL:

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

Input Data

Parameter	Mandatory Parameter	Description
userId	Yes	The identifier (ID) of the user from which a token needs to be unassigned
tokenId	Yes	The identifier (ID) of the token which needs to be unassigned from a user

Output Data

A successful operation completion message or standard error messages are returned.

Error Codes and Error Messages

Error Code	Description
1001	This entity already exists. This error occurs in two cases: 1. When you try to add an object with a unique field that already exists in the system. For example, when you try to add a user with a login that is already registered in the system. 2. When you try to perform an action that has already been performed. For example, when you try to assign a token to a user, but this token is already assigned to this or another user.
2001	Incorrect Parameter Length This error occurs when one or more of the parameters transmitted have incorrect length.
3001	Database Error
4001	Unregistered Name
5001	This error occurs when a parameter that is mandatory for the method called is not specified.
5002	This error occurs when: 1. The entity can not be found in the database. For example, when you request information on a user with an identifier that does not exist in the database. 2. There is no connection to perform the required action in the database. For example, when you want to authenticate a user on a resource, but this user is not assigned to this resource. Or, when you want to unassign a token from a resource, but this token was not assigned to this resource before. Or, when you want to assign a token to a resource with a user, but this token is not assigned to any user.
6001	This error occurs when an invalid parameter is transmitted. For example, a numeric parameter is expected, but the parameter transmitted contains extraneous characters.
6002	Incorrect URL Format
7001	Access Restriction This error occurs when you do not have the right to work with the requested object. The reason for this may be lack of access rights pertaining to this object, locking of your account for various reasons, etc.
8001	Internal Server Error
9001	Unknown Error

Error Message Structure

XML:

```
<responseHolder>
  <error>
    <code>{error_code}</code>
    <message>{general_error_explanation_message}</message>
    <developersMessage>{message_with_technical_details}</developersMessage>
  </error>
  <status>FAILURE</status>
</responseHolder>
```

JSON:

```
{
 "responseHolder" : {
  "error" : {
    "code" : {error_code},
    "message" : "{general_error_explanation_message}"
    “developersMessage”: “{message_with_technical_details}”
  },
  "status" : "FAILURE"
 }
}
```

Successful Operation Completion Messages

Successful operation completion messages look as follows:

XML:

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

JSON:

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