



# OPENOTP API WSDL

The specifications and information in this document are subject to change without notice. Companies, names, and data used in examples herein are fictitious unless otherwise noted. This document may not be copied or distributed by any means, in whole or in part, for any reason, without the express written permission of RCDevs.

Copyright (c) 2010-2017 RCDevs SA. All rights reserved.

<http://www.rcdevs.com>

WebADM and OpenOTP are trademarks of RCDevs. All further trademarks are the property of their respective owners.

Limited Warranty

No guarantee is given for the correctness of the information contained in this document. Please send any comments or corrections to [info@rcdevs.com](mailto:info@rcdevs.com).

### OpenOTP API Description

The OpenOTP authentication service is implemented over the SOAP/XML and RADIUS APIs. The SOAP/XML API is provided with a SOAP WSDL service description listed below. The OpenOTP API is very simple and provides 4 methods:

#### 1. openotpNormalLogin and openotpSimpleLogin

These methods are used to send an authentication request.

The request contains the following attributes:

- > username: User login name (mandatory).
- > domain: User login domain (optional if OpenOTP as a default domain setting set).
- > ldapPassword (openotpNormalLogin): User LDAP password (mandatory if OpenOTP login mode setting is LDAPOTP or LDAP).
- > otpPassword (openotpNormalLogin): One-time password (optional and usable only with Token OTPs).
- > anyPassword (openotpSimpleLogin): LDAP password or one-time password.
- > client: Client identifier (NAS) to be used in service logs (defaults to the client IP address).
- > source: IP address of the end user system (optional).
- > settings: List of OpenOTP settings which will override the user/group/application server-side settings (ex. *LoginMode=LDAPOTP, OTPTYPE=SMS*).

The response contains the following attributes:

- > code:
  - > 0 means an authentication failure.
  - > 1 means an authentication success.
  - > 2 means an authentication challenge.
- > message: The server reply message to be displayed to the user. With code 2, the message contains the challenge message.
- > session: With challenge, this is the session ID to be passed in the openotpChallenge request.
- > timeout: With challenge, this is the remaining session time to send the challenge response.
- > data: This attribute contains the ReplyData set in the LDAP user or group settings.
- > u2fChallenge: The list of U2F challenges in the form `<JSON Challenge 1>,<JSON Challenge 2>`.

With Radius, the data can be used by rule-based policies on a RADIUS VPN client for example.

In that case, OpenOTP RadiusBridge will return this data in a Filter-Id RADIUS attribute. In OpenOTP versions equal or greater

than 1.0.9, the `openotpChallenge` SOAP method includes the username and domain fields like in the `openotpLogin` method. This simplifies authentication programming in web applications as the developers do not have to ensure that the credentials passed via hidden fields in the challenge login form have been altered or not. Before, if a challenge-response was returned after an `openotpLogin` call, the website had to store the username and domain because it cannot trust this information when passed via hidden fields in the challenge HTML form. They can be altered on the client side before being posted again.

Now the `openotpChallenge` method requires the same username and domain as those given in the `openotpLogin` method. OpenOTP will also succeed only if the username and domain are identical in the `openotpLogin` and `openotpChallenge`. The website can also start a PHP session and use the information gathered by the hidden fields securely to get the user identity gathered in the first login form.

## 2. `openotpChallenge`

This method is used when the `openotpLogin` returned a challenge (code 2). This is the second request to be sent containing the user one-time password.

The request contains the following attributes:

- > `username`: User login name (mandatory).
- > `domain`: User login domain (optional if OpenOTP as a default domain setting set).
- > `session`: The session ID returned in the `openotpLogin` response.
- > `otpPassword`: The user one-time password (i.e. challenge response).
- > `u2fResponse`: The U2F response in JSON format (U2F challenge response).

The response contains the following attributes:

- > `code`:
- > 1 means authentication success.
- > 0 means authentication failure.
- > `message`: The server reply message to be displayed to the user.
- > `data`: See `openotpLogin` response above.

## 3. `openotpStatus`

This method is used to query a server status.

The request does not contain any attribute.

The response contains the following attributes:

- > `status`:
- > 1 if the server is willing to accept requests.
- > 0 if the server cannot accept new requests.

> message: The server status details.

## Note

The otpPassword attribute is usable in an openotpLogin request only with OATH HOTP onetime password. In this mode, the user can generate and enter the OTP in the first request (which is not possible with SMSOTP or MAILOTP).

*This manual was prepared with great care. However, RCDevs S.A. and the author cannot assume any legal or other liability for possible errors and their consequences. No responsibility is taken for the details contained in this manual. Subject to alternation without notice. RCDevs S.A. does not enter into any responsibility in this respect. The hardware and software described in this manual is provided on the basis of a license agreement. This manual is protected by copyright law. RCDevs S.A. reserves all rights, especially for translation into foreign languages. No part of this manual may be reproduced in any way (photocopies, microfilm or other methods) or transformed into machine-readable language without the prior written permission of RCDevs S.A. The latter especially applies for data processing systems. RCDevs S.A. also reserves all communication rights (lectures, radio and television). The hardware and software names mentioned in this manual are most often the registered trademarks of the respective manufacturers and as such are subject to the statutory regulations. Product and brand names are the property of RCDevs S.A. © 2021 RCDevs SA, All Rights Reserved*