

- [API Methods](#)
 - [Authenticate By Username and Password](#)
 - [Request](#)
 - [Example - Raw HTTP](#)
 - [Example - Curl](#)
 - [Response](#)
 - [Get User Info by User ID](#)
 - [Request](#)
 - [Example - Raw HTTP](#)
 - [Example - Curl](#)
 - [Response](#)
 - [Get user Info by Authentication Token](#)
 - [Request](#)
 - [Example - Raw HTTP](#)
 - [Example - Curl](#)
 - [Response](#)

Appendix A - Web Service API Specifications

To authenticate/authorize a user where user information is kept on the publisher using Web Service Integration, the publisher must implement a Web API based on the following requirements.

- A HTTP web API end-point using *HTTPS*
- Optional, but strongly advised to implement [HTTP Basic Authentication](#)
 - See [Web Service API HTTP Basic Authentication Example](#)
 - If implemented, the Basic Authentication credentials must be configured on Hub: Smart Edition > Stage Settings > Integration > Web Service

API Methods

- API methods must use **HTTP POST** and accept parameters the **HTTP Request Body** (x-www-form-urlencoded)
- Credentials in the request body should be in plain-text. They are secured over HTTPS

Authenticate By Username and Password

Request

Body Parameter (x-www-form-urlencoded)	Value
call	authenticate
username	<username>
password	<password>

Example - Raw HTTP

```
POST https://api-domain.com/api HTTP/1.1
Accept: application/xml
Authorization: Basic YXBpdXNlcm5hbWU6YXBpcGFzc3dvcmQ=
Content-Type: application/x-www-form-urlencoded
Host: api-domain.com
Content-Length: 59
Expect: 100-continue
```

```
call=authenticate&username=test@test.com&password=123456789
```

Example - Curl

```
curl --location --request POST 'https://api-domain.com/api' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--header 'Authorization: Basic YXBpdXNlcm5hbWU6YXBpcGFzc3dvcmQ=' \
--data-urlencode 'call=authenticate' \
--data-urlencode 'username=test@test.com' \
--data-urlencode 'password=123456789'
```

Response

Status	HTTP Response Status Code	Response Body
Success	200	A ticket based on one of the following: <ul style="list-style-type: none">• Appendix B1 - External Authentication Response Ticket Structure (Subscription)• Appendix B2 - External Authentication Response Ticket Structure (Bundles)

Fail	200	<p>If the authentication fails. Return an XML ticket with the publisher defined error code and message</p> <pre><error> <code>03</code> <message>Invalid credentials</message> </error></pre>
------	-----	---

Get User Info by User ID

Request

Body Parameter (x-www-form-urlencoded)	Value	Description
call	get_user_by_userid	
userid	<user ID>	user ID from the authentication ticket

Example - Raw HTTP

```
POST https://api-domain.com/api HTTP/1.1
Accept: application/xml
Authorization: Basic YXBpdXNlcm5hbWU6YXBpcGFzc3dvcmQ=
Content-Type: application/x-www-form-urlencoded
Host: api-domain.com
Content-Length: 67
Expect: 100-continue

call=get_user_by_userid&userid=FAE75C6E-622F-461F-BB4E-DDDFB7B5C982
```

Example - Curl

```
curl --location --request POST 'https://api-domain.com/api' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--header 'Authorization: Basic YXBpdXNlcm5hbWU6YXBpcGFzc3dvcmQ=' \
--data-urlencode 'call=get_user_by_userid' \
--data-urlencode 'userid==FAE75C6E-622F-461F-BB4E-DDDFB7B5C982'
```

Response

Status	HTTP Response Status Code	Response Body
--------	---------------------------	---------------

Success	200	<p>A ticket based on one of the following:</p> <ul style="list-style-type: none"> • Appendix B1 - External Authentication Response Ticket Structure (Subscription) • Appendix B2 - External Authentication Response Ticket Structure (Bundles)
Fail	200	<p>If the user cannot be retrieved by the specified user ID, return an XML ticket with the publisher defined error code and message</p> <pre><error> <code>04</code> <message>User not found</message> </error></pre>

Get user Info by Authentication Token

Request

Body Parameter (x-www-form-urlencoded)	Value	Description
call	get_user_by_token	
token	<authentication token>	Authentication token received from the publisher on redirection from login page hosted on the publisher's side

Example - Raw HTTP

```
POST https://api-domain.com/api HTTP/1.1
Accept: application/xml
Authorization: Basic YXBpdXNlcm5hbWU6YXBpcGFzc3dvcmQ=
Content-Type: application/x-www-form-urlencoded
Host: api-domain.com
Content-Length: 65
Expect: 100-continue
```

```
call=get_user_by_token&token=9A4EB6FB-FCF8-47AE-9D20-5DD5C21E3178
```

Example - Curl

```
curl --location --request POST 'https://api-domain.com/api' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--header 'Authorization: Basic YXBpdXNlcm5hbWU6YXBpcGFzc3dvcmQ=' \
--data-urlencode 'call=get_user_by_token' \
--data-urlencode 'token=9A4EB6FB-FCF8-47AE-9D20-5DD5C21E3178'
```

Response

Status	HTTP Response Status Code	Response Body
Success	200	<p>A ticket based on one of the following:</p> <ul style="list-style-type: none">• Appendix B1 - External Authentication Response Ticket Structure (Subscription)• Appendix B2 - External Authentication Response Ticket Structure (Bundles)
Fail	200	<p>If the user cannot be retrieved by the specified authentication token, return an XML ticket with the publisher defined error code and message</p> <pre><error> <code>05</code> <message>Invalid token</message> </error></pre>