



| Dackground | 1 |
|---|----|
| Creating TBA Keys | 2 |
| Create New Integration | 2 |
| Creating Access Token | 2 |
| Postman Setup | 3 |
| Import Collection | 3 |
| Webservice Example | 3 |
| SRS TBA Webservice (Search Customer by Phone Number) | 3 |
| Manual Setup of Postman Request | 3 |
| Relevant Screen Shots | 7 |
| Restlet Example | 9 |
| SRS TBA Restlet (Add User via Master Control Restlet) | 9 |
| Manual Setup of Postman Request | 9 |
| Relevant Screen Shots | 12 |
| Environment Variables | 15 |
| PHP Setup | 17 |
| TBA Restlet via PHP | 17 |
| TBA Webservice via PHP | 17 |
| Deprecating Username Password for Webservice | 19 |

Background

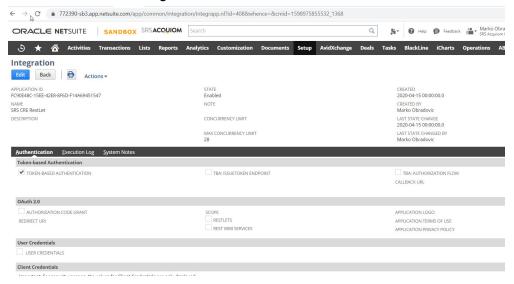
This document holds information how to setup oAuth 1.0 Token Based Authentication.



Creating TBA Keys

Create New Integration

- Create New Integration
- Or use "<u>SRS CRE RestLet</u>". Consumer Tokens for this integration are located in App Setting "General Settings" → "Restlet TBA SRS Consumer Token"
- Screen shot of new integration:



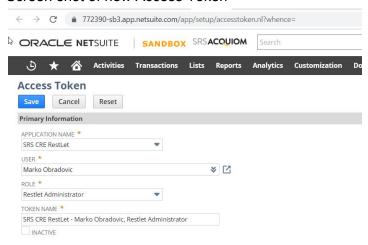
Creating Access Token

- Create new Access Token
- Or, add "Restlet Administrator" role 1072 to User and Create New Access Token
- Or use existing access tokens located in App Setting "General Settings" → "oAuth TBA
 Access Token" that was setup with role Restlet Administrator and integration SRS CRE





- Screen shot of new Access Token



Postman Setup

Import Collection

Download <u>SRS TBA.postman_collection_v2.1 (recommended).json</u> and import into Postman. Collection holds two examples.

Webservice Example

SRS TBA Webservice (Search Customer by Phone Number)

Manual Setup of Postman Request

Name: SRS TBA Webservice (Search Customer by Phone Number)

Method: POST

URL: https://{{Account_in_URL}}.suitetalk.api.netsuite.com/services/NetSuitePort_{{Version}}

Headers

SOAPAction: search
Content-Type: text/xml

Body





```
<soap:Envelope xmlns:xsd="http://www.w3.org/2001/XMLSchema"</pre>
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <soap:Header>
        <tokenPassport>
            <account>{ {Account} } </account>
            <consumerKey>{{ConsumerToken}}</consumerKey>
            <token>{ {AccessToken} } </token>
            <nonce>{ {Nonce} } </nonce>
            <timestamp>{{Timestamp}}</timestamp>
            <signature algorithm="HMAC-SHA256">{{Signature}}</signature>
        </tokenPassport>
    </soap:Header>
    <soap:Body>
        <search>
            <searchRecord xsi:type="q1:CustomerSearch"</pre>
xmlns:q1="urn:relationships_{{Version}}.lists.webservices.netsuite.com">
                <q1:basic>
                    <phone operator="is">
                        <searchValue>+442073462482
                    </phone>
                </q1:basic>
            </searchRecord>
        </search>
    </soap:Body>
</soap:Envelope>
```

Pre-request Script





```
pm.sendRequest('https://cdnjs.cloudflare.com/ajax/libs/crypto-js/4.0.0/crypto-js.min
.js', (err, res) => {
 (new Function(res.text()))();
});
let account = pm.variables.replaceIn('{{Account}}}');
let consumerKey = pm.variables.replaceIn('{{ConsumerToken}}');
let consumerSecret = pm.variables.replaceIn('{{ConsumerSecret}}');
let tokenId = pm.variables.replaceIn('{{AccessToken}}');
let tokenSecret = pm.variables.replaceIn('{{AccessSecret}}');
console.log("account " + account);
console.log("consumerKey " + consumerKey);
console.log("consumerSecret " + consumerSecret);
console.log("tokenId " + tokenId);
console.log("tokenSecret " + tokenSecret);
let timestamp = new Date().getTime().toString().substring(0, 10);
let nonce = CryptoJS.lib.WordArray.random(10).toString();
let baseString = account + '&' + consumerKey + '&' + tokenId + '&' + nonce + '&' +
timestamp;
let key = consumerSecret + '&' + tokenSecret;
let signature = CryptoJS.HmacSHA256(baseString, key).toString(CryptoJS.enc.Base64);
pm.environment.set("Nonce", nonce);
pm.environment.set("Timestamp", timestamp);
pm.environment.set("Signature", signature);
console.log("Nonce " + nonce);
console.log("Timestamp " + timestamp);
console.log("Signature " + signature);
```

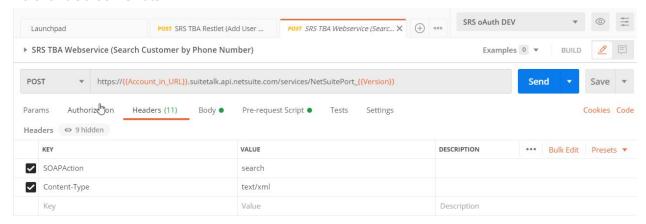


PROLECTO RESOURCES



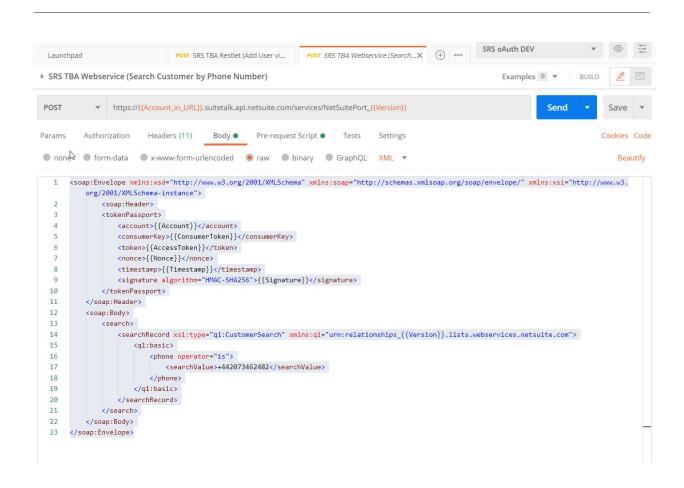


Relevant Screen Shots



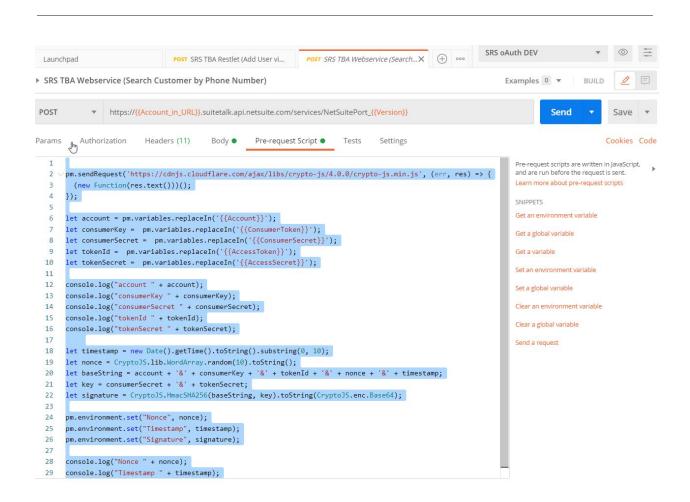












Restlet Example

SRS TBA Restlet (Add User via Master Control Restlet)

Manual Setup of Postman Request

Name: SRS TBA Restlet (Add User via Master Control Restlet)

Method: POST

URL: https://{{Account_in_URL}}.restlets.api.netsuite.com/app/site/hosting/restlet.nl?script={{RestletID}}&deploy=1

Authorization → Type: OAuth 1.0

Signature Method: HMAC-SHA256
Consumer Key: {{ConsumerToken}}
Consumer Secret: {{ConsumerSecret}}
Access Token: {{AccessToken}}
Token Secret: {{AccessSecret}}





```
Timestamp: {{Timestamp}}

Nonce: {{Nonce}}

Realm: {{Account}}

Headers

Authorization: {{Signature}}

Content-Type: application/json

Body
```

Pre-request Script

```
pm.sendRequest('https://cdnjs.cloudflare.com/ajax/libs/crypto-js/4.0.0/crypto-js.min
.js', (err, res) => {
    (new Function(res.text()))();
});
```



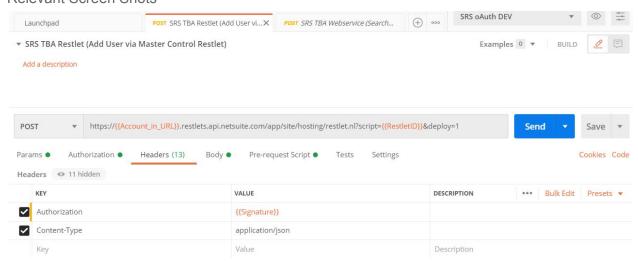


```
let account = pm.variables.replaceIn('{{Account}}');
let consumerKey = pm.variables.replaceIn('{{ConsumerToken}}');
let consumerSecret = pm.variables.replaceIn('{{ConsumerSecret}}');
let tokenId = pm.variables.replaceIn('{AccessToken}}');
let tokenSecret = pm.variables.replaceIn('{{AccessSecret}}');
console.log("account " + account);
console.log("consumerKey " + consumerKey);
console.log("consumerSecret " + consumerSecret);
console.log("tokenId " + tokenId);
console.log("tokenSecret " + tokenSecret);
let timestamp = new Date().getTime().toString().substring(0, 10);
let nonce = CryptoJS.lib.WordArray.random(10).toString();
let baseString = account + '&' + consumerKey + '&' + tokenId + '&' + nonce + '&' +
timestamp;
let key = consumerSecret + '&' + tokenSecret;
let signature = CryptoJS.HmacSHA256(baseString, key).toString(CryptoJS.enc.Base64);
pm.environment.set("Nonce", nonce);
pm.environment.set("Timestamp", timestamp);
pm.environment.set("Signature", signature);
console.log("Nonce " + nonce);
console.log("Timestamp " + timestamp);
console.log("Signature " + signature);
```



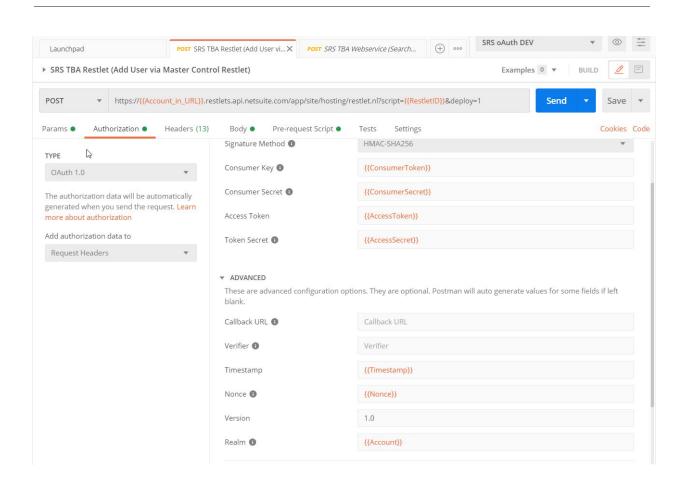


Relevant Screen Shots



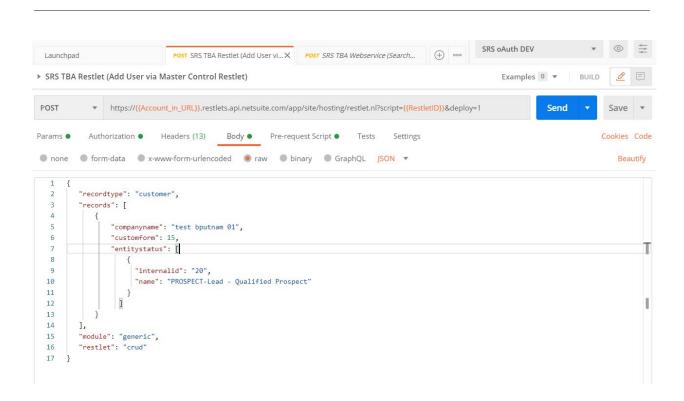






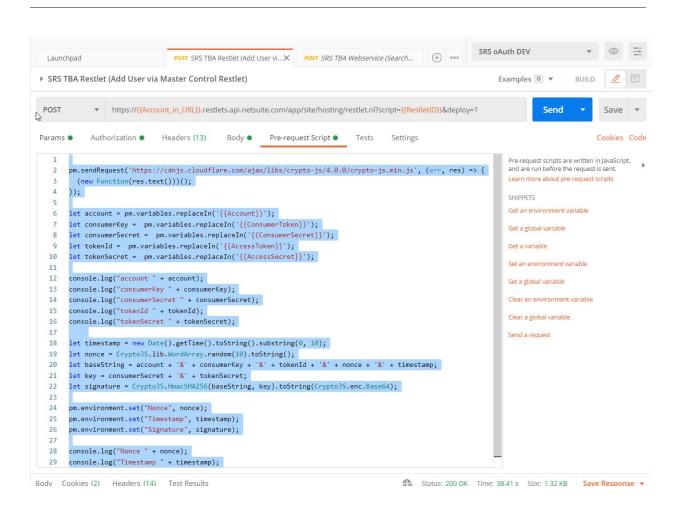












Environment Variables

| Name | SRS oAuth DEV |
|----------------|---|
| Account | 772390_SB3 |
| ConsumerToken | Create New one or use APP Setting "Restlet TBA SRS Consumer Token" → public |
| ConsumerSecret | Create New one or use APP Setting "Restlet TBA SRS Consumer Token" → secret |
| AccessToken | Create New one or use APP Setting "oAuth TBA Access Token" → public |





| AccessSecret | Create New one or use APP Setting "oAuth TBA Access Token" → secret |
|----------------|---|
| Nonce | |
| Signature | |
| Timestamp | |
| Version | 2018_1 |
| Account_in_URL | 772390-sb3 |
| RestletID | 690 |

Screen Shot

| vironment Name | | | | |
|-----------------------|----------------------|---------------------|--|--|
| SRS oAuth DEV | | | | |
| A. | | | | |
| VARIABLE | INITIAL VALUE ① | CURRENT VALUE 1 | | |
| Account | 772390_SB3 | 772390_SB3 | | |
| ConsumerToken | cc38e15a4b1901fe2bd2 | cc38e15a4b1901fe2bd | | |
| ConsumerSecret | c72e8baaa1d38efe6520 | c72e8baaa1d38efe652 | | |
| ✓ AccessToken | 97bdc814ba467b255f9 | 97bdc814ba467b255f9 | | |
| ✓ AccessSecret | 1d5da6c111fb23008cb | 1d5da6c111fb23008cb | | |
| ✓ Nonce | | | | |
| Signature | | | | |
| ✓ Timestamp | | | | |
| Version | 2018_1 | 2018_1 | | |
| ✓ Account_in_URL | 772390-sb3 | 772390-sb3 | | |
| ✓ RestletID | 690 | 690 | | |
| Add a new variable | | | | |



PHP Setup

Articles used to help setup PHP are as follows

- 1. Stack Overflow: Oauth implementation in netsuite using php
- 2. Answer Id: 24917 Downloading the PHP Toolkit
- 3. Tomual: Token-based Authentication with Netsuite's PHP Toolkit

TBA Restlet via PHP

```
Use RestletTBAExample.php to execute restlets. Example parameters needed to be set up: define("NETSUITE_SCRIPT_ID", '690'); define("NETSUITE_DEPLOY_ID", '1'); define("NETSUITE_ACCOUNT", '772390_SB3'); define("NETSUITE_ACCOUNT_INURL", '772390-sb3'); define("NETSUITE_CONSUMER_KEY", 'cc38e15a4b1901fe2bd215f4cc823d'); define("NETSUITE_CONSUMER_SECRET", 'c72e8baaa1d38efe652020db31d'); define("NETSUITE_TOKEN_ID", '97bdc814ba467b255f9a97bc3e4e611a25'); define("NETSUITE_TOKEN_SECRET", '1d5da6c111fb23008cb8273fc3665c');
```

This example was created using <u>Stack Overflow: Oauth implementation in netsuite using php</u>, and upgraded to use SHA256 encoding algorithm.

TBA Webservice via PHP

WebserviceTBAExample.php uses Netsuite's 2020 1 PHP Toolkit.



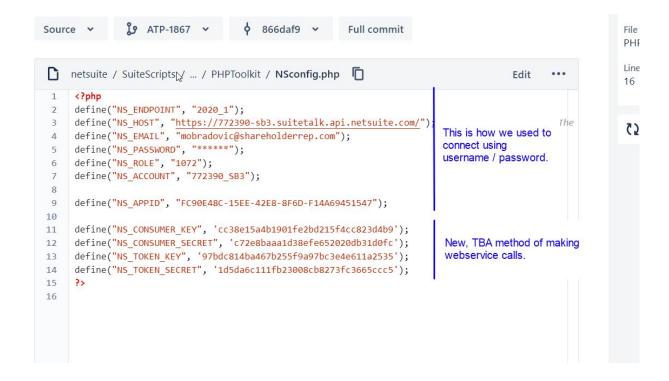


PHP 2020_1 toolkit provided in <u>Answer Id: 24917 Downloading the PHP Toolkit</u> was updated using <u>Tomual: Token-based Authentication with Netsuite's PHP Toolkit</u> instructions.



NStba.php is an addon Netsuite TBA file (provided by tomual) that generates signature and allows for Webservice TBA calls.

NSPHPClient.php was enhanced to use NStba.php and detect if NSConfig.php contains Consumer and Access Authentication Tokens.







Deprecating Username Password for Webservice

After TBA has been implemented, username password can be deprecated by unchecking Integration→ User Credentials.

