



PingFederate[®]

**REST Password Credential
Validator 1.0**

User Guide

This document is proprietary and not for general publication. It may be provided ad hoc for informational purposes only, and the information herein is subject to change without notice. Ping Identity does not provide any warranties and specifically disclaims any liability in connection with this document.

Note that Ping Identity may not provide support for any sample configurations provided in this document. The variability inherent among security environments prevents full testing and support for all possible platform configurations. If you need special assistance or would like to inquire about implementation or support programs, please contact Ping Identity Global Client Services (support.pingidentity.com).

Contact Information

1001 17th Street
Suite 100
Denver, CO 80202
U.S.A.

Phone: 877.898.2905 (+1 303.468.2882 outside North America)
Fax: 303.468.2909
E-mail: info@pingidentity.com
Web Site: <http://www.pingidentity.com>

© 2015 Ping Identity Corporation. All rights reserved.

August, 2015

Introduction.....	4
System Requirements.....	4
Installation.....	4
Configuring an REST PCV Instance	4
Debugging.....	7

Introduction

The PingFederate REST Password Credential Validator makes REST based web service calls to authenticate a user and get additional information about that user. The input to the PCV needs to be JSON in the username field. The password field is ignored.

System Requirements

This version of the adapter requires the following:

- PingFederate 7.3.0.5 (or higher)
- JDK 1.7.0_51 or higher

Installation

1. From the /dist directory in the ZIP file, copy the following file to the <pf_install>/pingfederate/server/default/deploy directory:

json-simple-1.1.1.jar
pf-rest-password-credential-validator-1.0.jar

2. Start or restart PingFederate.

Configuring an REST PCV Instance

1. Log on to the PingFederate administrative console and click **Password Credential Validators** under Server Configuration -> Authentication.
2. Select **Create New Instance...**
3. Select **REST Password Credential Validator 1.0.0** as the adapter type, fill in the Instance Name and Instance Id fields and click **Next**.

Instance Name	<input type="text" value="RESTPCV"/>	*
Instance Id	<input type="text" value="RESTPCV"/>	*
Type	<input type="text" value="REST Password Credential Validator 1.0"/>	*
Parent Instance	<input type="text" value="None"/>	

- Specify all the required fields.

REST Password Credential Validator 1.0

HEADERS (Headers that need to be set and passed along with the response.)

HEADER NAME	HEADER VALUE	Action
Add a new row to 'Headers'		

FIELD NAME	FIELD VALUE	DESCRIPTION
REST WEB SERVICE URL	<input type="text" value="https://localhost:9031/mock-rest-service/rest/User/Authenticate"/>	* The REST web service URL.
SUCCESS RESPONSE VALUE	<input type="text" value="200"/>	* The success response value (e.g., 200, 201).

- Click **Next**.
- Click **PSEUDONYM** for userid and click **Next**.
- Review the configuration and select **Done** and finally **Save**.

8. Wherever this PCV is used the contract needs to be extended using “flattened” names from the JSON response. For example, if the response from the REST service looks like:

```
{
  -response: {
    userid: "jdoe"
    -clientReturnHeader: {
      returnCode: "01022"
      returnMessage: "Out Of Band verification req
      returnStatus: "W"
    }
  }
  -account: {
    -mosaicRewards: {
      currencyType: ""
      currentTotalPointsAvailable: 0
    }
    pendingTransactionLoadSuccess: true
    pendingTransactionTotal: 0
    postedTransactionLoadSuccess: true
    -profile: {
      firstName: "John"
      lastName: "Doe"
      -email: {
        bounceCode: "V"
        bounceType: ""
        email: "john.doe@example.com"
      }
    }
    -phoneList: [1]
    -0: {
```

You can extend your contract with `response.userid` and `response.account.profile.email.email`. Here is an example in an HTML Form Adapter configuration:

EXTEND THE CONTRACT	ACTION
response.account.profile.email.email	Edit / Delete
response.account.profile.firstName	Edit / Delete
response.account.profile.lastName	Edit / Delete
response.clientReturnHeader.returnCode	Edit / Delete
response.clientReturnHeader.returnMessage	Edit / Delete
response.clientReturnHeader.returnStatus	Edit / Delete
response.userid	Edit / Delete

Debugging

To set the log level for this adapter (what appears in server.log) one would update the following file:

```
<PingFederateInstall>/server/default/conf/log4j.xml
```

And add the following entry into

```
<logger name="com.pingidentity.clientservices">  
    <level value="DEBUG"/>  
</logger>
```

Detailed training on using Log4j in Ping Federate can be found at:

- <https://www.pingidentity.com/support/training-center/index.cfm/configuring-log-files-using-log4j?id=1080675847001>