PingAccess®

Tomcat JWT Integration Kit v1.2.0

User Guide

PingAccess Tomcat JWT Integration Kit User Guide
Version 1.2.0
November 2017

Ping Identity Corporation
1001 17th Street, Suite 100
Denver, CO 80202
USA

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

Trademarks

Disclaimer
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 (http://support.pingidentity.com).

Table of Contents

Purpose

The PingAccess Tomcat JWT Integration Kit provides a Tomcat valve that intercepts a JWT Identity Mapping from PingAccess, validates its signature against PingAccess JWKS and translates the claims in the JWT to HTTP request headers.

Prerequisites

This document assumes that you already have the following installed and configured:

- A functional PingAccess environment, version 4.3.*
- JDK version 7 or 8 pending PingAccess version
- Tomcat 8.* configured to use SSL/TLS HTTP/1.1 Connector
  - The install location of Tomcat will be referred to as CATALINA_HOME.
- Optional:
  - A Tomcat Realm configured for user lookup. JNDI Example:

```
<Realm className="org.apache.catalina.realm.JNDIRealm"
        connectionName="cn=dmanager"
        connectionPassword="xxxxxxxx"
        connectionURL="ldap://localhost:1389"
        userBase="ou=People,dc=example,dc=com"
        userSearch="(uid={0})"
        userPattern="uid={0},ou=People,dc=example,dc=com"
        userPassword="userPassword"
        userRoleName="isMemberOf"
        roleBase="ou=Groups,dc=example,dc=com"
        roleName="cn"
        roleSearch="(uniqueMember={0})"
        />
```

Installation

1. Stop the Tomcat server.
2. Extract the contents of *com.pingidentity.clientservices.product.tomcat.pf-tomcat-jwt-token-integration-kit-1.2.0.zip* into a temporary directory that can be removed after installation.
3. Change directories to the extracted zip.
4. Copy dist/pf-tomcat-jwt-token-integration-kit-1.2.0.jar to the CATALINA_HOME/lib directory.
5. Copy the files in dist/lib to the CATALINA_HOME/lib directory.
6. Open the sample file dist/conf/server.xml and copy the contents to the clipboard.
7. Edit the file CATALINA_HOME/conf/server.xml:
   a. Find the <Host> section.
   b. Paste the contents of the clipboard between <Host></Host>.
   c. Modify the values as noted in the sample server.xml file. Note that values must match the PingAccess JWT Identity Mapping that will be sent to the application container. Example:

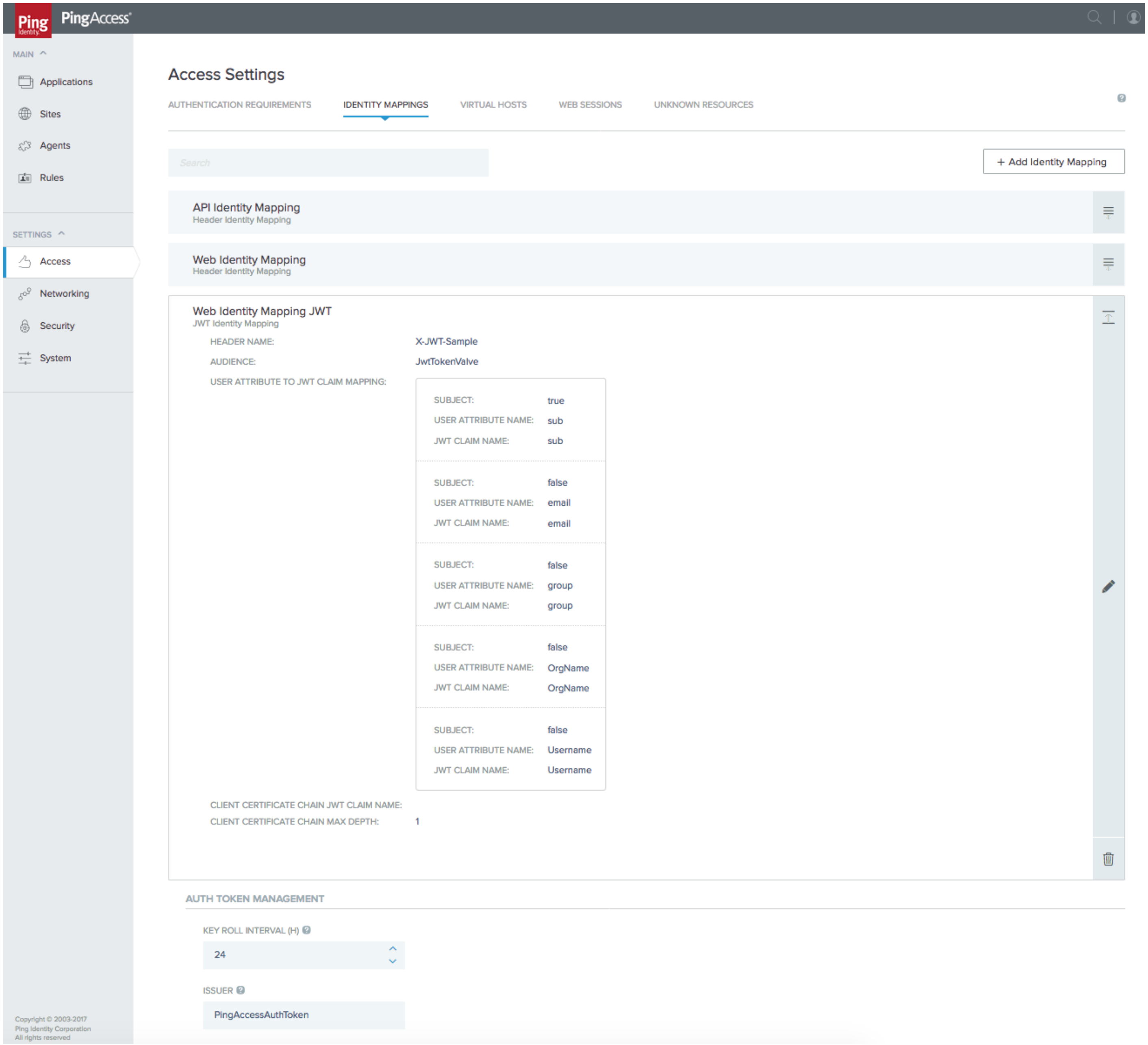
```
<Host …>
        <Valve className="com.pingidentity.professionalservices.tomcat.JwtTokenValve"
                   jwksEndpoint="https://localhost:3000/pa/authtoken/JWKS"
                   jwtHeader="X-JWT-Sample"
                   jwtIssuer="PingAccessAuthToken"
                   jwtAudience="JwtTokenValve"
                   jwtRoleAttribName="group"
                   jwtClockSkew="30"
                   jwtCacheRefreshRate="3600"
                   createPrincipalFromJWT="true"
                   enableDebug="true" />

</Host>
```

8. Start the Tomcat server.

Testing

The PingAccess QuickStart application is not required for testing. It is used in this document for illustrative purposes only. Similar steps for configuring PingAccess could be done for any application that is deployed to Tomcat and protected by PingAccess.

1. Deploy the PingAccess Quickstart v4.3.0.
    a. Follow the instructions of the Quickstart application with the following changes:
        i. Deploy PingAccessQuickStart.war to CATALINA_HOME/webapps
        ii. Rename CATALINA_HOME/webapps/PingAccessQuickStart.war to CATALINA_HOME/webapps/PingAccessQuickStart
        iii. Supply the Tomcat host and port values during the configuration of the Quickstart script
2. Log into the PingAccess administration console.
3. Add a new JWT Identity Mapping in Access -> Identity Mappings.
    a. Configure the JWT Identity Mapping as necessary.
    b. The header, audience and issuer values defined in PingAccess should be the values used to configure the Tomcat valve in CATALINA_HOME/conf/server.xml.



Access Settings

API Identity Mapping
Header Identity Mapping

Web Identity Mapping
Header Identity Mapping

Web Identity Mapping JWT
JWT Identity Mapping

HEADER NAME: X-JWT-Sample
AUDIENCE: JwtTokenValve
USER ATTRIBUTE TO JWT CLAIM MAPPING:

SUBJECT:	true
USER ATTRIBUTE NAME:	sub
JWT CLAIM NAME:	sub
SUBJECT:	false
USER ATTRIBUTE NAME:	email
JWT CLAIM NAME:	email
SUBJECT:	false
USER ATTRIBUTE NAME:	group
JWT CLAIM NAME:	group
SUBJECT:	false
USER ATTRIBUTE NAME:	OrgName
JWT CLAIM NAME:	OrgName
SUBJECT:	false
USER ATTRIBUTE NAME:	Username
JWT CLAIM NAME:	Username

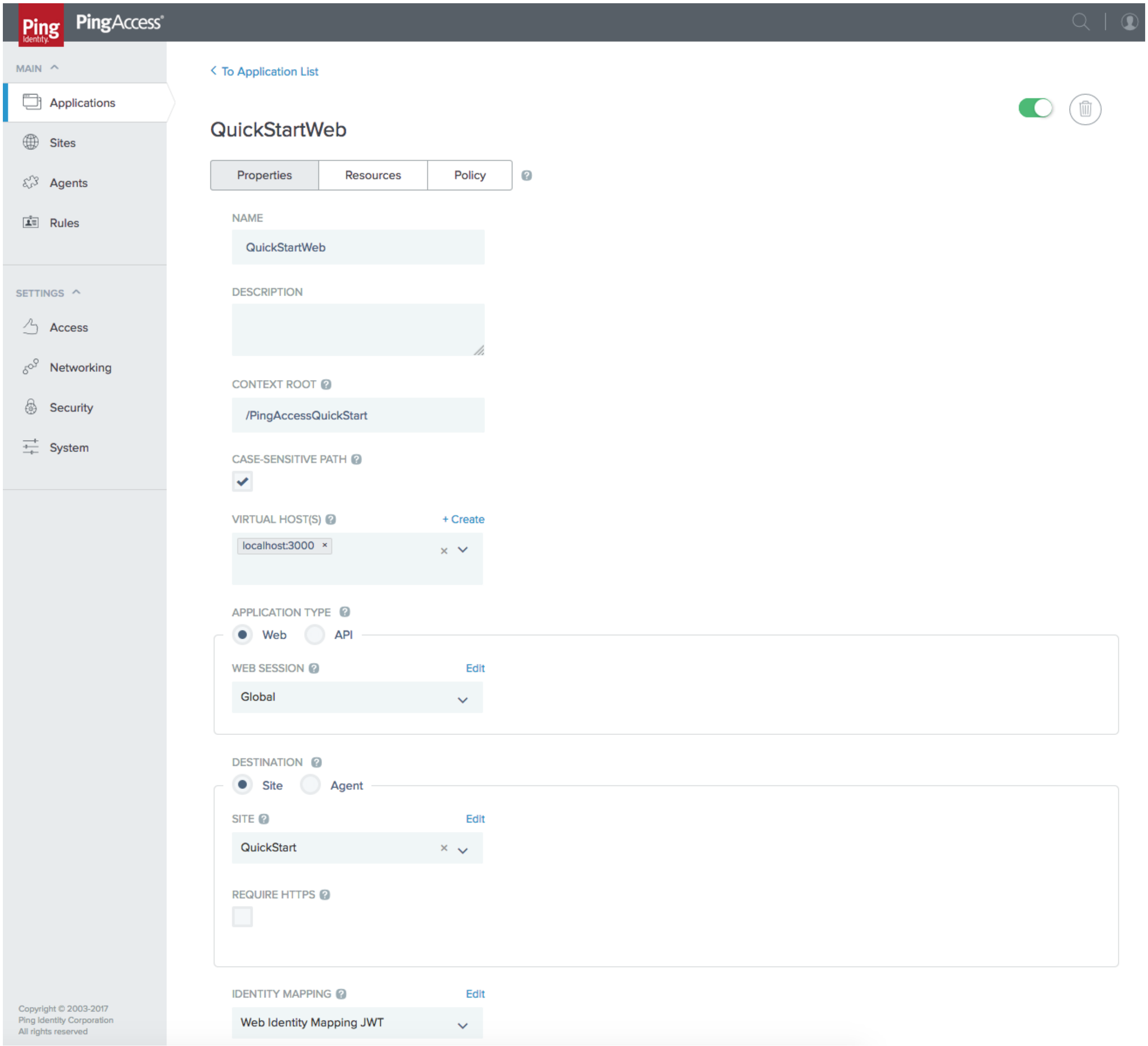
CLIENT CERTIFICATE CHAIN JWT CLAIM NAME:
CLIENT CERTIFICATE CHAIN MAX DEPTH: 1

AUTH TOKEN MANAGEMENT

KEY ROLL INTERVAL (H): 24

ISSUER: PingAccessAuthToken

4. Configure the QuickStartWeb application to use the newly created JWT Identity Mapping in Applications -> QuickStartWeb -> Edit Properties



QuickStartWeb

NAME: QuickStartWeb

DESCRIPTION:

CONTEXT ROOT: /PingAccessQuickStart

CASE-SENSITIVE PATH: ☑

VIRTUAL HOST(S): localhost:3000

APPLICATION TYPE: ● Web ○ API

WEB SESSION: Global

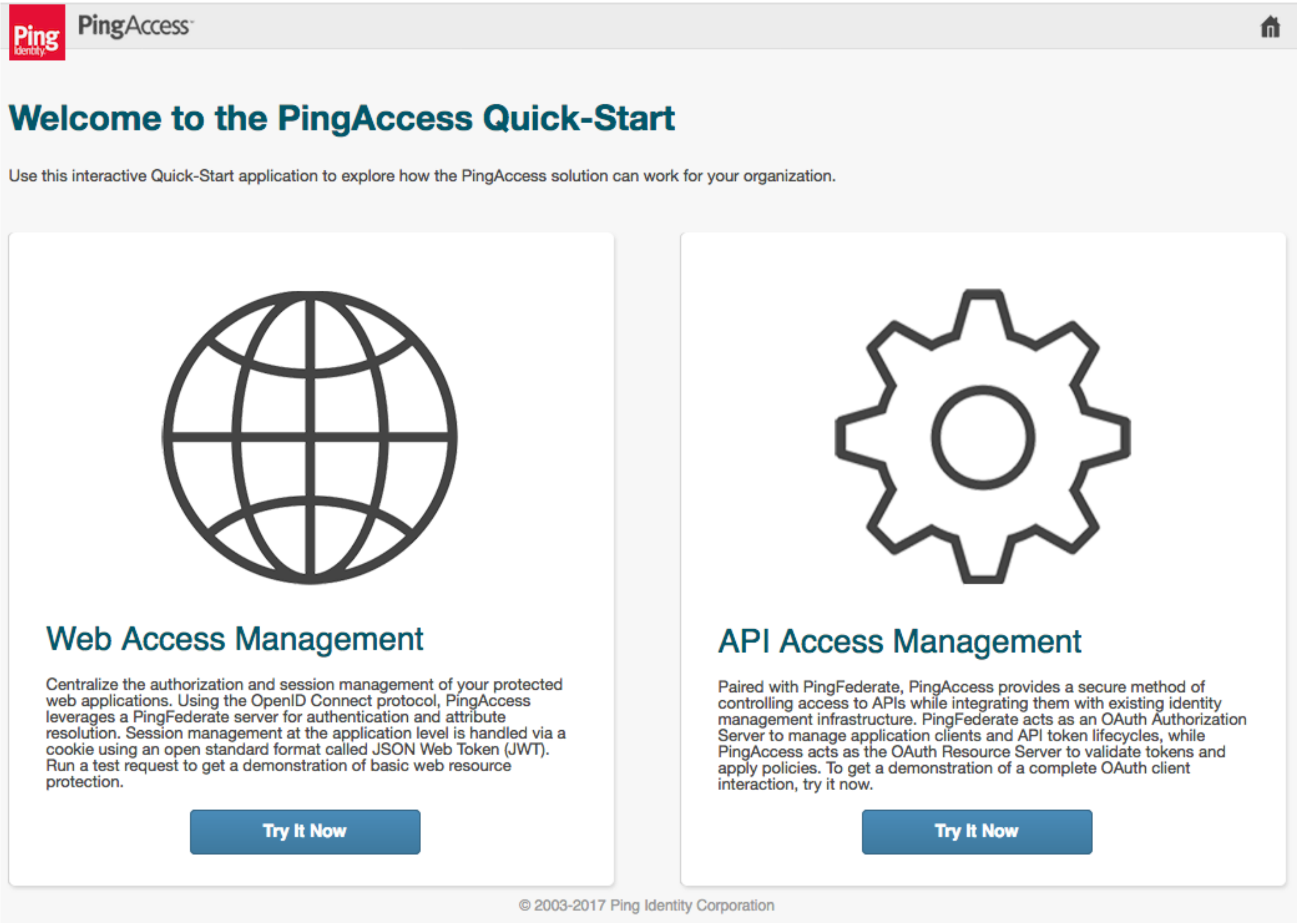
DESTINATION: ● Site ○ Agent

SITE: QuickStart

REQUIRE HTTPS: □

IDENTITY MAPPING: Web Identity Mapping JWT

5. Open the QuickStart application.



Welcome to the PingAccess Quick-Start

Use this interactive Quick-Start application to explore how the PingAccess solution can work for your organization.

Web Access Management

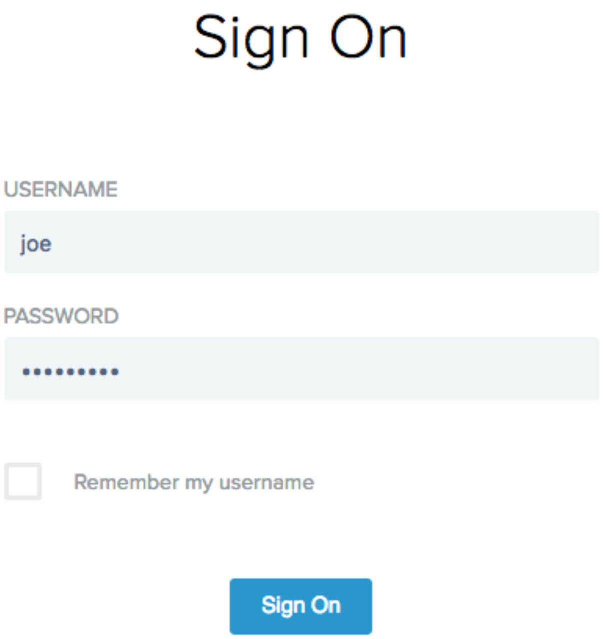
Centralize the authorization and session management of your protected web applications. Using the OpenID Connect protocol, PingAccess leverages a PingFederate server for authentication and attribute resolution. Session management at the application level is handled via a cookie using an open standard format called JSON Web Token (JWT). Run a test request to get a demonstration of basic web resource protection.

API Access Management

Paired with PingFederate, PingAccess provides a secure method of controlling access to APIs while integrating them with existing identity management infrastructure. PingFederate acts as an OAuth Authorization Server to manage application clients and API token lifecycles, while PingAccess acts as the OAuth Resource Server to validate tokens and apply policies. To get a demonstration of a complete OAuth client interaction, try it now.

© 2003-2017 Ping Identity Corporation

6. Select the Web Access Management application and log in using the credentials joe/2Federate.



Sign On

USERNAME

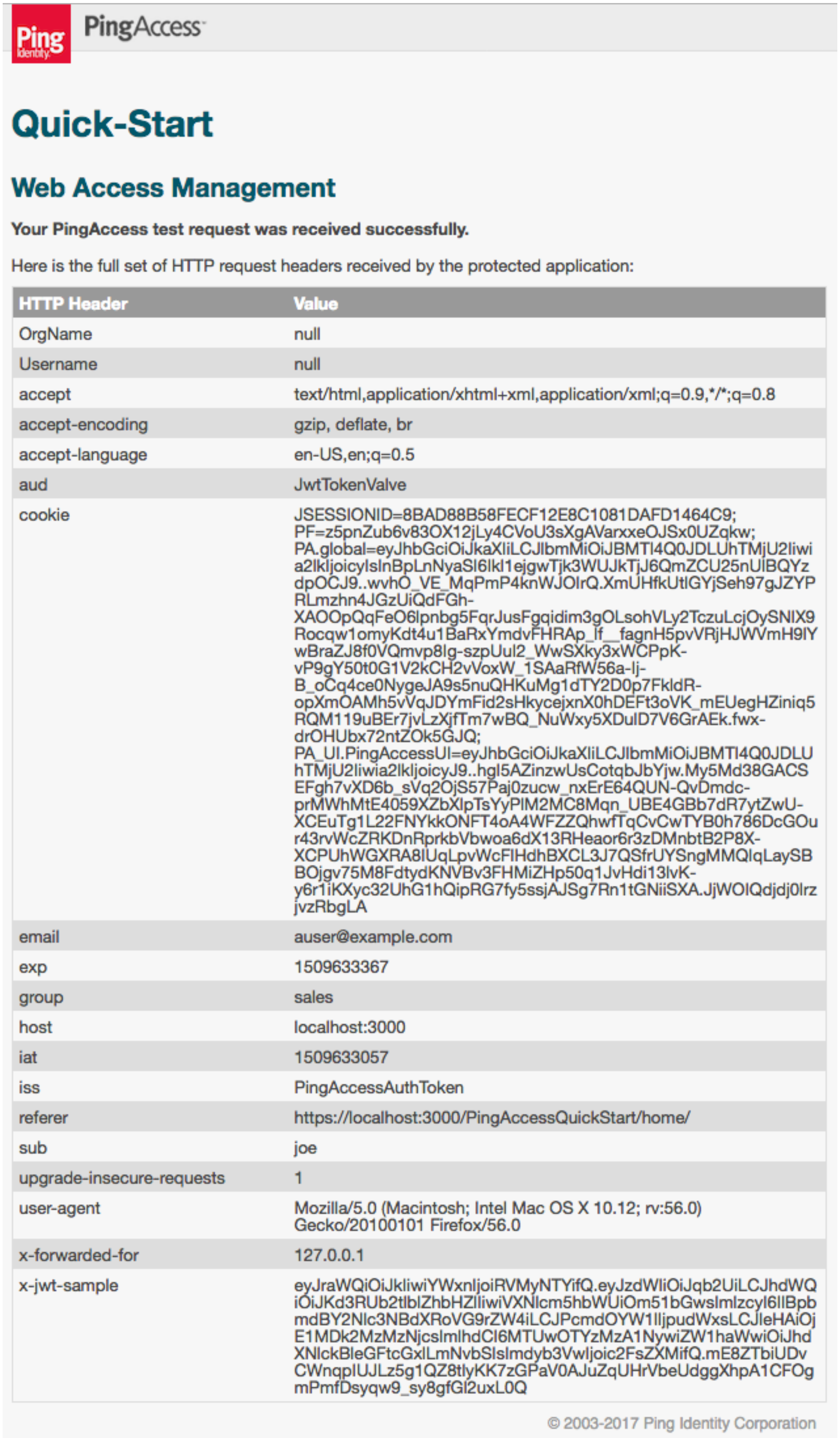
joe

PASSWORD

••••••••••

□ Remember my username

7. Confirm the JWT claims configured in PingAccess have been translated to HTTP request headers.



Quick-Start

Web Access Management

Your PingAccess test request was received successfully.

Here is the full set of HTTP request headers received by the protected application:

HTTP Header	Value
OrgName	null
Username	null
accept	text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
accept-encoding	gzip, deflate, br
accept-language	en-US,en;q=0.5
aud	JwtTokenValve
cookie	JSESSIONID=8BAD88B58FECF12E8C1081DAFD1464C9; PF=z5pnZub6v83OX12jLy4CVoU3sXgAVarxxeOJSx0UZqkw; PA.global=eyJhbGciOiJkaXIiLCJlbmMiOiJBMTI4Q0JDLUhTMjU2Iiwia2lkIjoicyIsInBpLnNyaSI6Ikl1ejgwTjk3WUJkTjJ6QmZCU25nUlBQYzdpOCJ9..wvhO_VE_MqPmP4knWJOIrQ.XmUHfkUtlGYjSeh97gJZYPRLmzhn4JGzUiQdFGh-XAOOpQqFeO6lpnbg5FqrJusFgqidim3gOLsohVLy2TczuLcjOySNIX9Rocqw1omyKdt4u1BaRxYmdvFHRAp_lf__fagnH5pvVRjHJWVmH9lYwBraZJ8f0VQmvp8lg-szpUul2_WwSXky3xWCPpK-vP9gY50t0G1V2kCH2vVoxW_1SAaRfW56a-Ij-B_oCq4ce0NygeJA9s5nuQHKuMg1dTY2D0p7FkldR-opXmOAMh5vVqJDYmFid2sHkycejxnX0hDEFt3oVK_mEUegHZiniq5RQM119uBEr7jvLzXjfTm7wBQ_NuWxy5XDuID7V6GrAEk.fwx-drOHUbx72ntZOk5GJQ; PA_UI.PingAccessUI=eyJhbGciOiJkaXIiLCJlbmMiOiJBMTI4Q0JDLUhTMjU2Iiwia2lkIjoicyJ9..hgI5AZinzwUsCotqbJbYjw.My5Md38GACSEFgh7vXD6b_sVq2OjS57Paj0zucw_nxErE64QUN-QvDmdc-prMWhMtE4059XZbXlpTsYyPlM2MC8Mqn_UBE4GBb7dR7ytZwU-XCEuTg1L22FNYkkONFT4oA4WFZZQhwfTqCvCwTYB0h786DcGOur43rvWcZRKDnRprkbVbwoa6dX13RHeaor6r3zDMnbtB2P8X-XCPUhWGXRA8IUqLpvWcFlHdhBXCL3J7QSfrUYSngMMQlqLaySBBOjgv75M8FdtydKNVBv3FHMiZHp50q1JvHdi13lvK-y6r1iKXyc32UhG1hQipRG7fy5ssjAJSg7Rn1tGNiiSXA.JjWOIQdjdj0lrzjvzRbgLA
email	auser@example.com
exp	1509633367
group	sales
host	localhost:3000
iat	1509633057
iss	PingAccessAuthToken
referer	https://localhost:3000/PingAccessQuickStart/home/
sub	joe
upgrade-insecure-requests	1
user-agent	Mozilla/5.0 (Macintosh; Intel Mac OS X 10.12; rv:56.0) Gecko/20100101 Firefox/56.0
x-forwarded-for	127.0.0.1
x-jwt-sample	eyJraWQiOiJkIiwiYWxnIjoiRVMyNTYifQ.eyJzdWIiOiJqb2UiLCJhdWQiOiJKd3RUb2tlblZhbHZlIiwiVXNlcm5hbWUiOm51bGwsImlzcyI6IlBpbmdBY2Nlc3NBdXRoVG9rZW4iLCJPcmdOYW1lIjpudWxsLCJleHAiOjE1MDk2MzMzNjcsImlhdCI6MTUwOTYzMzA1NywiZW1haWwiOiJhdXNlckBleGFtcGxlLmNvbSIsImdyb3VwIjoic2FsZXMifQ.mE8ZTbiUDvCWnqpIUJLz5g1QZ8tlyKK7zGPaV0AJuZqUHrVbeUdggXhpA1CFOgmPmfDsyqw9_sy8gfGl2uxL0Q

If debug logging was enabled, similar statements should be in the CATALINA_HOME/logs/catalina.out log file:

```
02-Nov-2017 10:30:59.628 INFO [http-nio-8443-exec-6]
com.pingidentity.professionalservices.tomcat.JwtTokenValve.debug JWT validation succeeded!
{"sub":"joe","aud":"JwtTokenValve","Username":null,"iss":"PingAccessAuthToken","OrgName":nu
ll,"exp":1509633367,"iat":1509633057,"email":"auser@example.com","group":"sales"}
02-Nov-2017 10:30:59.629 INFO [http-nio-8443-exec-6]
com.pingidentity.professionalservices.tomcat.JwtTokenValve.debug Translating JWT claim
(sub) to HTTP request header
02-Nov-2017 10:30:59.629 INFO [http-nio-8443-exec-6]
com.pingidentity.professionalservices.tomcat.JwtTokenValve.debug Translating JWT claim
(aud) to HTTP request header
02-Nov-2017 10:30:59.629 INFO [http-nio-8443-exec-6]
com.pingidentity.professionalservices.tomcat.JwtTokenValve.debug Translating JWT claim
(Username) to HTTP request header
02-Nov-2017 10:30:59.629 INFO [http-nio-8443-exec-6]
com.pingidentity.professionalservices.tomcat.JwtTokenValve.debug Translating JWT claim
(iss) to HTTP request header
02-Nov-2017 10:30:59.629 INFO [http-nio-8443-exec-6]
com.pingidentity.professionalservices.tomcat.JwtTokenValve.debug Translating JWT claim
(OrgName) to HTTP request header
02-Nov-2017 10:30:59.629 INFO [http-nio-8443-exec-6]
com.pingidentity.professionalservices.tomcat.JwtTokenValve.debug Translating JWT claim
(exp) to HTTP request header
02-Nov-2017 10:30:59.629 INFO [http-nio-8443-exec-6]
com.pingidentity.professionalservices.tomcat.JwtTokenValve.debug Translating JWT claim
(iat) to HTTP request header
02-Nov-2017 10:30:59.629 INFO [http-nio-8443-exec-6]
com.pingidentity.professionalservices.tomcat.JwtTokenValve.debug Translating JWT claim
(email) to HTTP request header
02-Nov-2017 10:30:59.629 INFO [http-nio-8443-exec-6]
com.pingidentity.professionalservices.tomcat.JwtTokenValve.debug Translating JWT claim
(group) to HTTP request header
```