



# PS Demo Documentation

*1.0 (alpha)*

Nov 04, 2019

Pulse Secure, LLC  
2700 Zanker Road, Suite 200  
San Jose, CA 95134  
<https://www.pulsesecure.net>

© 2019 by Pulse Secure, LLC. All rights reserved

**CONTENTS:**

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>PSdemo Release Notes</b>                  | <b>1</b>  |
| 1.1      | Introduction . . . . .                       | 1         |
| 1.2      | Hardware Platforms . . . . .                 | 1         |
| <b>2</b> | <b>PSdemo Getting Started Guide</b>          | <b>3</b>  |
| 2.1      | Introduction . . . . .                       | 3         |
| 2.2      | Chapter 1 . . . . .                          | 3         |
| 2.3      | Chapter 2 . . . . .                          | 3         |
| <b>3</b> | <b>PSdemo Developer’s Guide</b>              | <b>5</b>  |
| 3.1      | Introduction . . . . .                       | 5         |
| 3.2      | API for some functionality . . . . .         | 5         |
| 3.3      | Example Code . . . . .                       | 13        |
| <b>4</b> | <b>PSdemo User Guide</b>                     | <b>15</b> |
| 4.1      | Revision History . . . . .                   | 15        |
| 4.2      | Introduction . . . . .                       | 15        |
| 4.3      | Section A . . . . .                          | 15        |
| 4.4      | Section B . . . . .                          | 16        |
| <b>5</b> | <b>PSdemo Frequently Asked Questions</b>     | <b>17</b> |
| 5.1      | What is abc? . . . . .                       | 17        |
| 5.2      | How do we setup pqr? . . . . .               | 17        |
| 5.3      | Is abk supported? . . . . .                  | 17        |
| 5.4      | What about mno? Can a user do pqr? . . . . . | 17        |
| <b>6</b> | <b>Indices and tables</b>                    | <b>19</b> |



## PSDEMO RELEASE NOTES

**Warning:** This is a sample release note only for demonstration purposes.

### Version 1.0

## 1.1 Introduction

This document is the release notes for the product PSdemo Release 9.1R2. This document contains information about what is included in this software release: supported features, feature changes, unsupported features, and known issues. If the information in the release notes differs from the information found in the documentation set, follow the release notes.

## 1.2 Hardware Platforms

You can install and use this software version on the following hardware platforms: • PSA300, PSA3000, PSA5000, PSA7000f, PSA7000c

To download software for these hardware platforms, go to: <https://support.pulsesecure.net/> Virtual Appliance Editions  
This software version is available for the following virtual appliance editions: • Virtual Pulse Secure Appliance (PSA-V)

---

**Note:** From 9.1R1 release onwards, VA-DTE is not supported.

From 9.0R1 release, Pulse Secure has begun the End-of-Life (EOL) process for the VA-SPE virtual appliance. In its place, Pulse Secure has launched the new PSA-V series of virtual appliances designed for use in the data center or with cloud services such as Microsoft Azure and Amazon AWS.

---

The following table lists the virtual appliance systems qualified with this release.

### 1.2.1 Platform Qualified System

#### VMware

- HP ProLiant DL380 G5 with Intel(R) Xeon(R) CPU
- ESXi 6.7

- CentOS 6.6 with Kernel `cst-kvm 2.6.32-504.el6.x86_64`
- QEMU/KVM v1.4.0

**Linux Server Release 6.4 on an Intel Xeon CPU L5640 @ 2.27GHz**

- 24GB memory in host
- Allocation for virtual appliance ( KVM, Hyper-V)
  - 4vCPU,
  - 4GB memory,
  - 40GB disk space
- Standard DS2 V2 (2 Core, 2 NICs)

## PSDEMO GETTING STARTED GUIDE

TBD

- *Introduction*
- *Chapter 1*
- *Chapter 2*

### 2.1 Introduction

TBD

### 2.2 Chapter 1

TBD

### 2.3 Chapter 2





## PSDEMO DEVELOPER'S GUIDE

TBD

- *Introduction*
- *API for some functionality*
  - *Pet Store APIs*
    - \* *pet*
    - \* *store*
    - \* *user*
- *Example Code*
  - *Using API for xyz use case*
  - *Using API for mno use case*

---

### 3.1 Introduction

TBD

### 3.2 API for some functionality

TBD

#### 3.2.1 Pet Store APIs

**pet**

---

**GET /pet/{petId}**

Find pet by ID

- **Description:** Returns a single pet

- **Produces:** ['application/json', 'application/xml']

### Parameters

| Name  | Position | Description         | Type    |
|-------|----------|---------------------|---------|
| petId | path     | ID of pet to return | integer |

### Responses

200 - *successful operation*

400 - *Invalid ID supplied*

404 - *Pet not found*

---

## POST /pet/{petId}

Updates a pet in the store with form data

- **Description:**
- **Consumes:** ['application/x-www-form-urlencoded']
- **Produces:** ['application/json', 'application/xml']

### Parameters

| Name   | Position | Description                        | Type    |
|--------|----------|------------------------------------|---------|
| petId  | path     | ID of pet that needs to be updated | integer |
| name   | formData | Updated name of the pet            | string  |
| status | formData | Updated status of the pet          | string  |

### Responses

405 - *Invalid input*

---

## DELETE /pet/{petId}

Deletes a pet

- **Description:**
- **Produces:** ['application/json', 'application/xml']

### Parameters

| Name    | Position | Description      | Type    |
|---------|----------|------------------|---------|
| api_key | header   |                  | string  |
| petId   | path     | Pet id to delete | integer |

### Responses

400 - *Invalid ID supplied*

404 - *Pet not found*

---

---

**POST /pet/{petId}/uploadImage**

uploads an image

- **Description:**
- **Consumes:** ['multipart/form-data']
- **Produces:** ['application/json']

**Parameters**

| Name               | Position | Description                       | Type    |
|--------------------|----------|-----------------------------------|---------|
| petId              | path     | ID of pet to update               | integer |
| additionalMetadata | formData | Additional data to pass to server | string  |
| file               | formData | file to upload                    | file    |

**Responses**

200 - successful operation

---

---

**POST /pet**

Add a new pet to the store

- **Description:**
- **Consumes:** ['application/json', 'application/xml']
- **Produces:** ['application/json', 'application/xml']

**Parameters**

| Name | Position | Description                                    | Type |
|------|----------|--|------|
| body | body     | Pet object that needs to be added to the store |      |

**Responses**

405 - Invalid input

---

---

**PUT /pet**

Update an existing pet

- **Description:**
- **Consumes:** ['application/json', 'application/xml']
- **Produces:** ['application/json', 'application/xml']

**Parameters**

| Name | Position | Description                                    | Type |
|------|----------|--|------|
| body | body     | Pet object that needs to be added to the store |      |

**Responses**

*400 - Invalid ID supplied*

*404 - Pet not found*

*405 - Validation exception*

---

### GET /pet/findByStatus

Finds Pets by status

- **Description:** Multiple status values can be provided with comma separated strings
- **Produces:** ['application/json', 'application/xml']

#### Parameters

| Name   | Position | Description   | Type  |
|--------|----------|---|-------|
| status | query    | Status values that need to be considered for filter | array |

#### Responses

*200 - successful operation*

*400 - Invalid status value*

---

### GET /pet/findByTags

Finds Pets by tags

- **Description:** Multiple tags can be provided with comma separated strings. Use tag1, tag2, tag3 for testing.
- **Produces:** ['application/json', 'application/xml']

#### Parameters

| Name | Position | Description       | Type  |
|------|----------|-------------------|-------|
| tags | query    | Tags to filter by | array |

#### Responses

*200 - successful operation*

*400 - Invalid tag value*

---

## store

---

### GET /store/inventory

Returns pet inventories by status

- **Description:** Returns a map of status codes to quantities
- **Produces:** ['application/json']

## Parameters

| Name | Position | Description | Type |
|------|----------|-------------|------|
|------|----------|-------------|------|

## Responses

200 - *successful operation*

---

### GET /store/order/{orderId}

Find purchase order by ID

- **Description:** For valid response try integer IDs with value  $\geq 1$  and  $\leq 10$ . Other values will generated exceptions
- **Produces:** ['application/json', 'application/xml']

## Parameters

| Name    | Position | Description                        | Type    |
|---------|----------|------------------------------------|---------|
| orderId | path     | ID of pet that needs to be fetched | integer |

## Responses

200 - *successful operation*

400 - *Invalid ID supplied*

404 - *Order not found*

---

### DELETE /store/order/{orderId}

Delete purchase order by ID

- **Description:** For valid response try integer IDs with positive integer value. Negative or non-integer values will generate API errors
- **Produces:** ['application/json', 'application/xml']

## Parameters

| Name    | Position | Description                              | Type    |
|---------|----------|--|---------|
| orderId | path     | ID of the order that needs to be deleted | integer |

## Responses

400 - *Invalid ID supplied*

404 - *Order not found*

---

### POST /store/order

Place an order for a pet

- **Description:**

- **Consumes:** ['application/json']
- **Produces:** ['application/json', 'application/xml']

### Parameters

| Name | Position | Description                         | Type |
|------|----------|-------------------------------------|------|
| body | body     | order placed for purchasing the pet |      |

### Responses

200 - *successful operation*

400 - *Invalid Order*

---

## user

---

### GET /user/{username}

Get user by user name

- **Description:**
- **Produces:** ['application/json', 'application/xml']

### Parameters

| Name     | Position | Description   | Type   |
|----------|----------|---|--------|
| username | path     | The name that needs to be fetched. Use user1 for testing. | string |

### Responses

200 - *successful operation*

400 - *Invalid username supplied*

404 - *User not found*

---

---

### PUT /user/{username}

Updated user

- **Description:** This can only be done by the logged in user.
- **Consumes:** ['application/json']
- **Produces:** ['application/json', 'application/xml']

### Parameters

| Name     | Position | Description                  | Type   |
|----------|----------|------------------------------|--------|
| username | path     | name that need to be updated | string |
| body     | body     | Updated user object          |        |

### Responses

400 - *Invalid user supplied*

---

---

404 - User not found

---

---

### DELETE /user/{username}

Delete user

- **Description:** This can only be done by the logged in user.
- **Produces:** ['application/json', 'application/xml']

#### Parameters

| Name     | Position | Description                       | Type   |
|----------|----------|-----------------------------------|--------|
| username | path     | The name that needs to be deleted | string |

#### Responses

400 - Invalid username supplied

404 - User not found

---

---

### GET /user/login

Logs user into the system

- **Description:**
- **Produces:** ['application/json', 'application/xml']

#### Parameters

| Name     | Position | Description                          | Type   |
|----------|----------|--------------------------------------|--------|
| username | query    | The user name for login              | string |
| password | query    | The password for login in clear text | string |

#### Responses

200 - successful operation

400 - Invalid username/password supplied

---

---

### GET /user/logout

Logs out current logged in user session

- **Description:**
- **Produces:** ['application/json', 'application/xml']

#### Parameters

| Name | Position | Description | Type |
|------|----------|-------------|------|
|------|----------|-------------|------|

#### Responses

*default - successful operation*

---

### POST /user

Create user

- **Description:** This can only be done by the logged in user.
- **Consumes:** ['application/json']
- **Produces:** ['application/json', 'application/xml']

#### Parameters

| Name | Position | Description         | Type |
|------|----------|---------------------|------|
| body | body     | Created user object |      |

#### Responses

*default - successful operation*

---

### POST /user/createWithArray

Creates list of users with given input array

- **Description:**
- **Consumes:** ['application/json']
- **Produces:** ['application/json', 'application/xml']

#### Parameters

| Name | Position | Description         | Type |
|------|----------|---------------------|------|
| body | body     | List of user object |      |

#### Responses

*default - successful operation*

---

### POST /user/createWithList

Creates list of users with given input array

- **Description:**
- **Consumes:** ['application/json']
- **Produces:** ['application/json', 'application/xml']

#### Parameters

| Name | Position | Description         | Type |
|------|----------|---------------------|------|
| body | body     | List of user object |      |

#### Responses



*default - successful operation*

---

## 3.3 Example Code

Perform the setup using the following commands

```
bash xyz.py -o <outputfile> -d <inputdir> -t <type>
```

### 3.3.1 Using API for xyz use case

### 3.3.2 Using API for mno use case



## PSDEMO USER GUIDE

This is the user guide for Pulse Secure product codenamed PSdemo.

- *Revision History*
- *Introduction*
- *Section A*
  - *Part 1*
  - *Part 2*
- *Section B*
  - *How to do xyz*
  - *How to setup mno*

---

### 4.1 Revision History

TBD

### 4.2 Introduction

This is the introduction of ...

### 4.3 Section A

TBD

#### **4.3.1 Part 1**

#### **4.3.2 Part 2**

### **4.4 Section B**

#### **4.4.1 How to do xyz**

#### **4.4.2 How to setup mno**

## PSDEMO FREQUENTLY ASKED QUESTIONS

TBD

- *What is abc?*
- *How do we setup pqr?*
- *Is abk supported?*
- *What about mno? Can a user do pqr?*

### 5.1 What is abc?

Answer to What is abc. . . .

### 5.2 How do we setup pqr?

Detailed instructions on how to setup pqr. . . .

### 5.3 Is abk supported?

Insights on backward compatibility and support matrix or *links to the same*.

### 5.4 What about mno? Can a user do pqr?

Ans



## INDICES AND TABLES

- `genindex`
- `modindex`
- `search`