

Version: 4.8

Published date: September 2025

Copyright © 2025 by ESW. All rights reserved. Information contained herein is subject to change without notice.

JWKS Authentication Integration



**SFCC - ESW Cartridge**

+

Contents

[**1** **ESW JWKS Integration** 2](#_Toc189588919)

* 1. [JWKS Configuration 3](#_Toc189588920)
  2. [Work Flow 3](#_Toc189588921)

# **ESW JWKS Integration**

ESW SFCC Cartridge provides the capability to validate webhooks using ESW’s public key, simplifying cartridge configuration for retailers. The cartridge supports basic authentication as well however JWKS eliminates the need to manage authentication credentials.

* 1. JWKS Configuration

To use this feature, retailer needs to make sure that he has custom attributes, services and the jobs are configured correctly in the BM. Make sure to import following from the cartridge:

**Custom Objects:**

* \link\_eshopworld\Metadata\meta\customobjects.xml

**Jobs**:

* \link\_eshopworld\Metadata\jobs.xml

**Services:**

* \link\_eshopworld\Metadata\services.xml

The following configurations should be imported and once the import is completed the following data will be available in the BM.

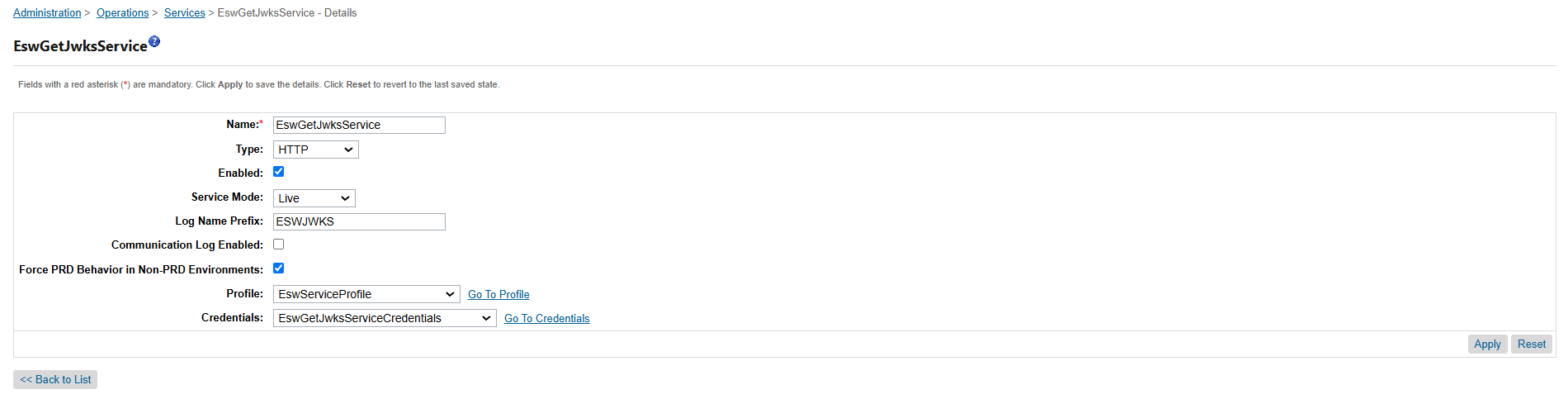
**Job:**

* The job “eswGetJwksJob**”** will be added

A screenshot of a computer

Description automatically generated

**Service:**

The service “EswGetJwksService” will be added 

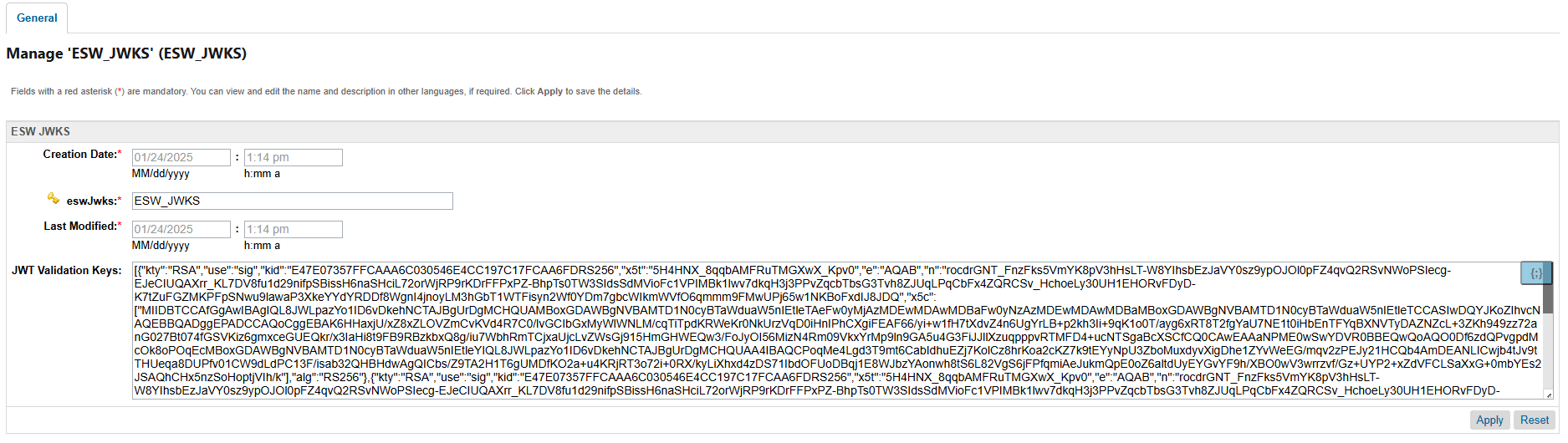
**Custom object**

The ESW\_JWKScustom object will be added.

* 1. Work Flow

**Fetching Keys**

The **eswGetJwksJob** is responsible to fetch the JWKS from ESW. The job relies on the service EswGetJwksService. Once the job is executed, the job fetches the keys from ESW and populates the custom object in SFCC to store the public keys locally in SFCC.

****

**Validation**

* Upon each webhook call from ESW, the cartridge checks for the JWKS authentication header (esw-authorization) in the request and tries to validate the public keys.
* If the validation fails, the cartridge retrieves the keys again using the EswGetJwksService and attempts validation once more.
* If the keys fail validation a second time, the cartridge checks the presence of a basic authentication header, if enabled, and attempts basic auth validation.
* If the basic auth fails as well or its not enabled, the cartridge generates and sends an error response for the webhook due to failed authentication.

**List of Supported WebHooks:**

* Order Confirmation
* Inventory Webhook
* Order Return
* Appeasements
* Cancellation