|  |
| --- |
|  |
| Queue-It Integration |
|  |

|  |
| --- |
| Queue-It integration components and SFRA implementation |



# Contents

[Contents](#_Toc4496575)

[Intended Audience](#_Toc4496576)

[Revision History](#_Toc4496577)

[Overview](#_Toc4496578)

[Component Overview](#_Toc4496579)

[Functional Overview](#_Toc4496580)

[Compatibility](#_Toc4496581)

[Integration Cartridges](#_Toc4496582)

[Integration with SG Controllers or SFRA](#_Toc4496583)

[SFCC installation](#_Toc4496584)

[Configuring Queue It IN QUEUE IT PORTAL](#_Toc4496585)

[Configuring QUEUE IT IN SFCC](#_Toc4496586)

[Testing QUEUE IT ON THE STOREFRONT](#_Toc4496587)

# Intended Audience

This document is intended for technical personnel who will be setting up and integrating Queue It with the Salesforce Commerce Cloud SiteGenesis or SFRA Storefront.

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Notes | Author |
| 03/17/2019 | Version 19.3.0.0 | Initial release | Mark Barnum |

# Overview

Queue-it is a waiting room platform developed to prevent website overload and protect your online business, while providing a fair user experience.

[www.Queue-It.com](http://www.Queue-It.com).

# Component Overview

## Functional Overview

The functionality provided by this integration component allows retailers with web stores on the Salesforce Commerce Cloud platform to quickly integrate a waiting room mechanism on their site for high traffic / pop up sales traffic flood mitigation.

Once configured, this cartridge will route extraneous traffic to the Queue It waiting room for processing.

## Compatibility

Compatible Site Genesis version 18.10

## Integration Cartridges

**Cartridge name**

* *int\_queueit\_sfra*

**Controllers**

* request.js

**Scripts**

* *httpContextProvider.js*
* *queueit\_knownuserv3\_skd.js*
* *QueueIt.js*

**Cartridge name**

* *int\_queueit\_controllers*

**Scripts**

* *httpContextProvider.js*
* *queueit\_knownuserv3\_skd.js*
* *QueueIt.js*

**Cartridge name**

* *int\_queueit\_pipelines*

**Pipelines**

* OnRequest.xml

**Scripts**

* *httpContextProvider.js*
* *queueit\_knownuserv3\_skd.js*
* *QueueIt.js*
* *QueueItExecute.ds*

**Metadata**

* *queueit\_metadata.xml*

# Integration with SG Controllers or SFRA

## SFCC installation

1. **Import the appropriate int\_queueit\_controllers, int\_queueit\_pipelines, or int\_queueit\_sfra cartridge into the Salesforce Commerce Cloud Studio Workspace:**
   1. IMPORTANT: ONLY IMPORT THE CARTRIDGE COMPATIBLE WITH YOUR CODEBASE. SFRA builds will run with int\_queueit\_sfra cartridge ONLY! A controller based installation will utilize int\_queueit\_controllers ONLY, and a pipeline based implementation will utilize int\_queueit\_pipelines ONLY.
   2. Open Salesforce Commerce Cloud Studio
   3. Click File -> Import -> General -> File System
   4. Browse to the "int\_queueit\_controllers", “int\_queueit\_pipelines” or “int\_queueit\_sfra” directory directory
   5. Click Finish.
2. **Import the metadata into the Salesforce Commerce Cloud Business Manager on the Staging Server**
   1. Log into the Salesforce Commerce Cloud Business Manager for the Staging Server.
   2. Click Administration -> Site Development -> Import/Export
   3. Under Import & Export Files Click Upload
   4. Click Upload
   5. Browse for the QueueIt\_metadata.xml file and Click Upload
   6. Click the "<< Back" button at the bottom of the page.
   7. Under Meta Data Click Import
   8. Select the queueit\_metadata.xml file
   9. Click Next
   10. When validation is complete, click Import.
3. **ADVANCED / PERFORMANCE CONSIDERATION: Change SFCC Request Pre-filter**
   * 1. The base implementation of Queue-It utilizes a filtering mechanism to limit the number of times logic is executed to only look at the following requests: Search-Show, Product-Start, AddToCart.
     2. If your queuing needs are more SPECIFIC, you will WANT to update the code to limit the scope of the logic checks on line 31 of QueueIt.js to more actively filter as this code is called on every page request and can be a performance bottleneck.
4. **SFRA : Skip to next section**
5. **SG CONTROLERS ONLY: Override onrequest.js to include Queue It**
   1. Open onRequest.js in your controller cartridge
   2. Add the following line at the top under var Status = require…

**var** QueueIt = require('int\_queueit\_controllers/cartridge/scripts/QueueIt.js');

* 1. Add the following right after onRequest = function() …

QueueIt.Start();

1. **SG CONTROLERS and SFRA ONLY: Override Cart.js to add queue it logic**
   1. Open cartridge/controllers/Cart.js
   2. Find addProduct
   3. At the top of addProduct function, add the following code:

**if** (session.custom.ajaxredirecturl) {

**const** qtemplate = '/queueit/queueiterror.isml';

**const** queueItHeaderName = 'X-SF-CC-QUEUEIT';

response.addHttpHeader(queueItHeaderName, session.custom.ajaxredirecturl );

app.getView().render(qtemplate);

}

* 1. For SG Controllers: Make sure this code is BEFORE the var cart = … line.
  2. For SFRA: Make sure this code is BEFORE the var BasketMgr = … line.

1. **SG PIPELINES ONLY: Validate OnRequest Pipeline**
   1. By default, the implementation for pipelines OVERRIDES the OnRequest pipeline. If your implementation already had logic in OnRequest Pipeline, you may want to add script node reference to QueueItExecute.ds to current Pipeline and remove OnRequest in int\_queueit\_pipelines

## Configuring Queue It IN QUEUE IT PORTAL

There are detailed help documents at <TBD> or work with your sales contact to configure your queue it portal correctly.

1. Log into <companyid>.go.queue-it.net. Note: if you are unsure of your company id or credentials contact queue-it customer service.
2. Secret Key is found at Account -> Settings -> Integration Tab
3. Configure your Integration -> actions and triggers based on your desired queue-it integration points. See help documentation within the queueit portal for guidance.
4. Publish your changes using the Publish Now Button
5. Copy your configuration from http://<companyid>.queue-it.net/status/integrationconfig/<companyid>. **You will need this data in the next step.**

## Configuring QUEUE IT IN SFCC

1. Append int\_queueit\_controllers, int\_queueit\_pipelines, or int\_queueit\_sfra to the Effective Cartridge Path.
2. Update Site Preferences
   1. Navigate to Site -> Site Preferences -> Custom Preferences
   2. Select QueueIt
   3. Enable queue it by selecting Yes for Queue It Enabled
   4. Enter in your queue it secret key from the prior section
   5. Enter your queue it customer id from the prior section
3. Update queueit custom object to reflect configuration in queue it portal
   1. Navigate to site -> Custom Objects – Custom Object Editor
   2. Select queueit from the drop down
   3. Click New
   4. Id = 1
   5. Queue It Data = Paste from the copy of the configuration above (http://<companyid>.queue-it.net/status/integrationconfig/<companyid>)
   6. Click Apply

## Testing QUEUE IT ON THE STOREFRONT

1. Navigate to your storefront
2. Navigate to the configured page(s) on your storefront
3. You should be sent to a queue
4. If you click the button you will go to the storefront with access. More details on testing and behavior of the queue it service are in the whitepaper in the queue it portal