

**RADWARE BOT MANAGER - INTEGRATION GUIDE**

1. **Summary**

Shieldsquare protects online properties i.e. website, mobile app, and APIs from bot attack without impacting the genuine traffic. A user can select action against that traffic based on IP, Bot category, URL, Country, ISP etc. Shieldsquare enables dashboard to get the visualizations of the source of the traffic and action taken against them.

This document talks about the installation process of Shieldsquare Salesforce Commerce Link Cartridge, which enables you to integrate Shieldsquare protection API within a Salesforce Commerce storefront.

1. **Component Overview**
   1. **Functional Overview**

Shieldsquare is enabling below feature list in initial release of SFCC plugin.

* + - Enrich dashboard to get visibility on attack source in realtime
    - Take custom action based on bot category
    - Customize CAPTCHA and Block page to match your brand design
  1. **Use Cases- Shieldsquare Bot Mitigation and Management**

|  |  |
| --- | --- |
| **Use Case Statement** | **Result** |
| A user visits the merchant store | Shieldsqaure will intercept all the requests coming to the merchant store. Once request reaches to Shieldsquare bot engine it will do a check and qualifies them for the actions. If engine doesn't suspect anything it will allow otherwise it will react according to the response selected for each Bot categories, ex- Block, Challenge CAPTCHA, Redirect Loop, Drop Request etc. |

* 1. **Test Cases**

|  |  |  |
| --- | --- | --- |
| **Test Cases** | **Passing Criteria** | **Status** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

* 1. **Constraints**

To enable Shieldsquare Bot protection solution both the party should have agreement signed, only enabling the cartridge will not grant the use of Shieldsquare product offerings. Please write to us @ [support@shieldsquare.com](mailto:support@shieldsquare.com).

* 1. **Compatibility and Support**

Shieldsquare LINK cartridge is compatible with

URL:https://www.demandware.com/link-marketplace/shieldsquare

* 1. **Privacy and Terms**

Once you singup with Shieldsquare you are bound with the [terms of service](https://www.shieldsquare.com/terms-conditions/) and [policy](https://www.shieldsquare.com/privacy/). Please send mail for any query related to term of service and policy @ [support@shieldsquare.com](mailto:support@shieldsquare.com).

1. **Implementation Guide**
   1. **Prerequisites**

User must create an account at Shieldsquare using the signup form.

* **Signup:** <https://www.shieldsquare.com/signup>

Once your account is created you will get the Sandbox ID, Production ID and API key in Subscriber details page

* **Subscriber details**: <https://portal.radwarebotmanager.com/subscriber-id-details>

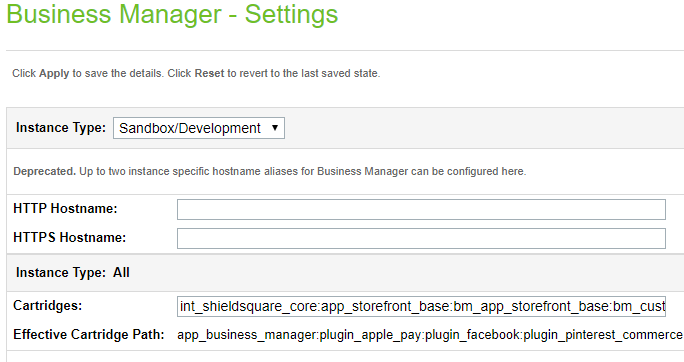
1. **Sandbox ID:** If you are integrating for the very first time, you should use the Sandbox Id, as it will enable you to verify the integration. This Id has been specifically provided for testing purposes and is to be used only in test environments
2. **Production ID:** The Production Id should only be used in the Production Environment, once it is verified that the integration is successful
   1. **Clone**

* Clone repository to their local machine.
* After cloning you will find the int\_shieldsquare\_core and int\_shieldsquare\_sfra cartridges.
* The cartridge needs to be imported and uploaded onto the instance
  1. **Business Manager Path Configuration**

Navigate to the following path Sites 🡪 Manage Sites 🡪Merchant Site 🡪 Settings

Add the cartridge names to Cartridges path as shown in screenshot below:

int\_shieldsquare\_core:app\_storefront\_base:bm\_app\_storefront\_base

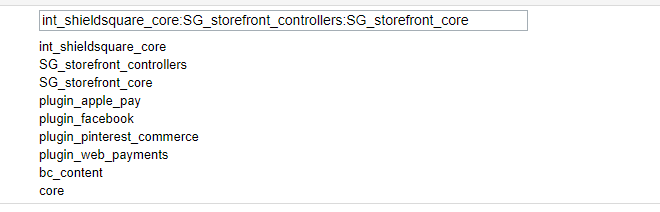


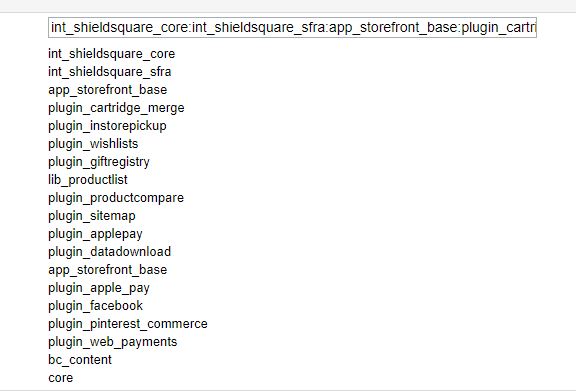
* 1. **Cartridge path Configuration**

Navigate to the following path Sites 🡪 Manage Sites 🡪Merchant Site 🡪 Settings

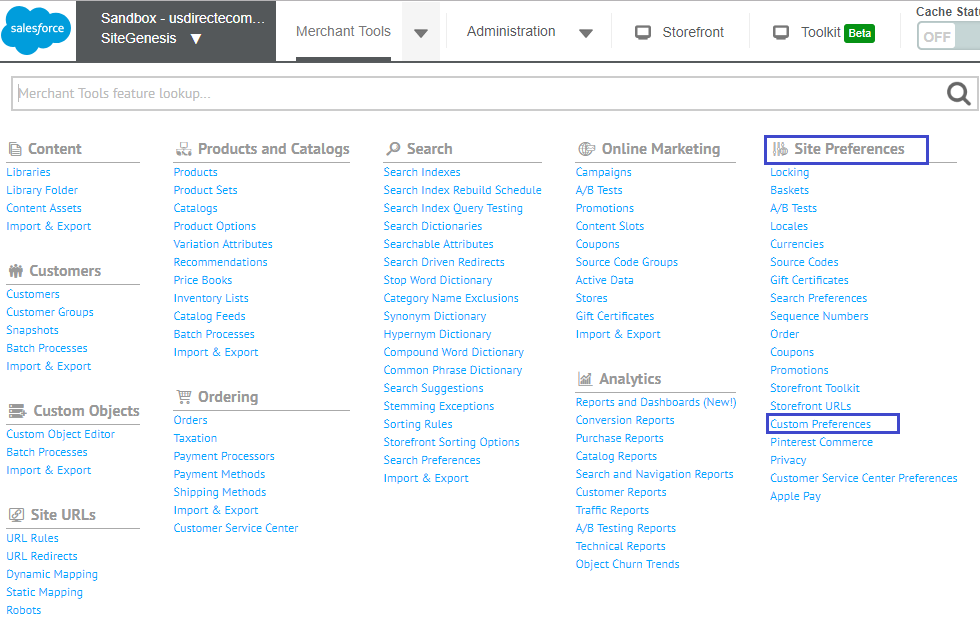
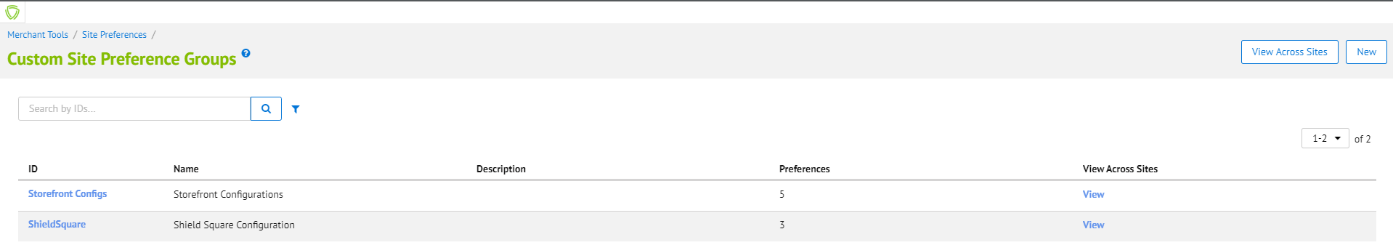
Add the cartridge names to Cartridges path as shown in screenshot below:

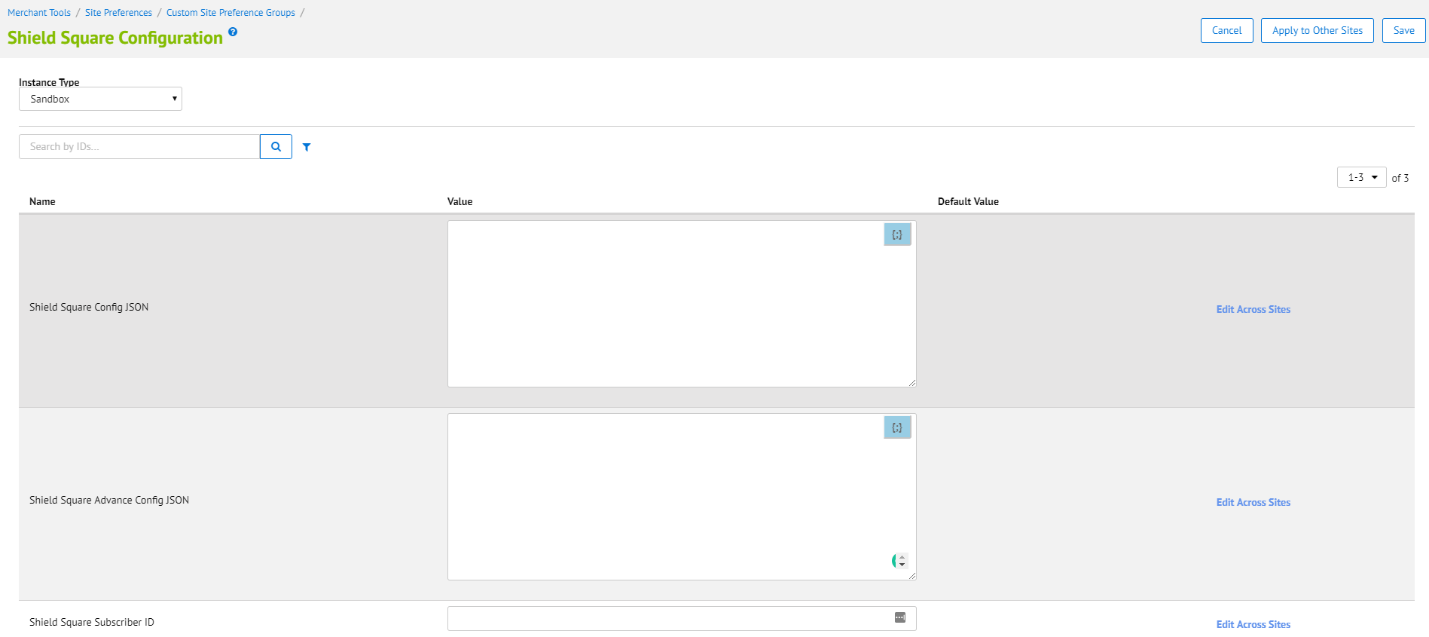
* + 1. **Controller Version:** int\_shieldsquare\_core:SG\_storefront\_controllers:SG\_storefront\_core

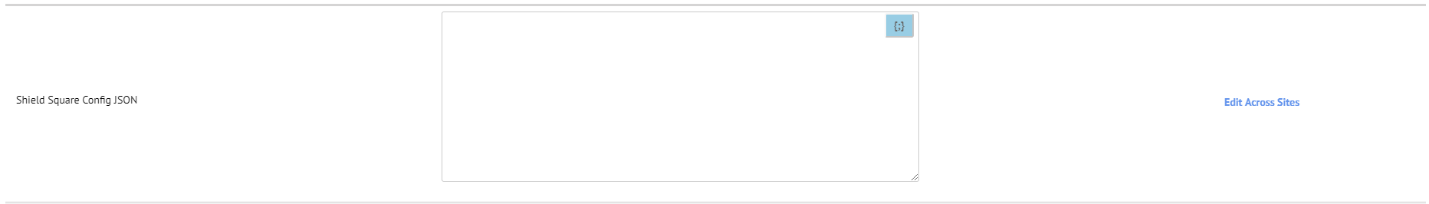
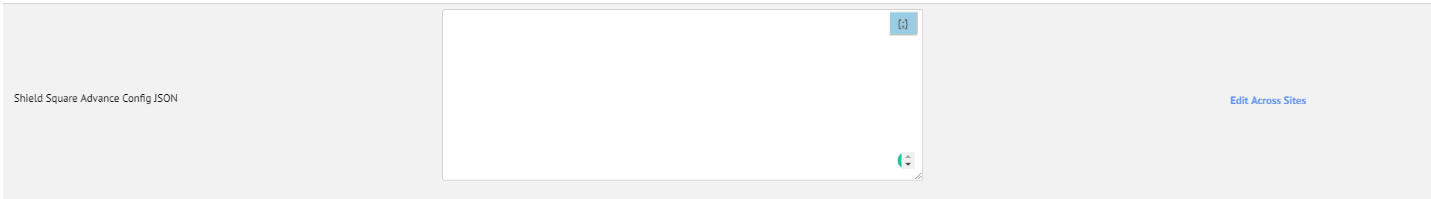
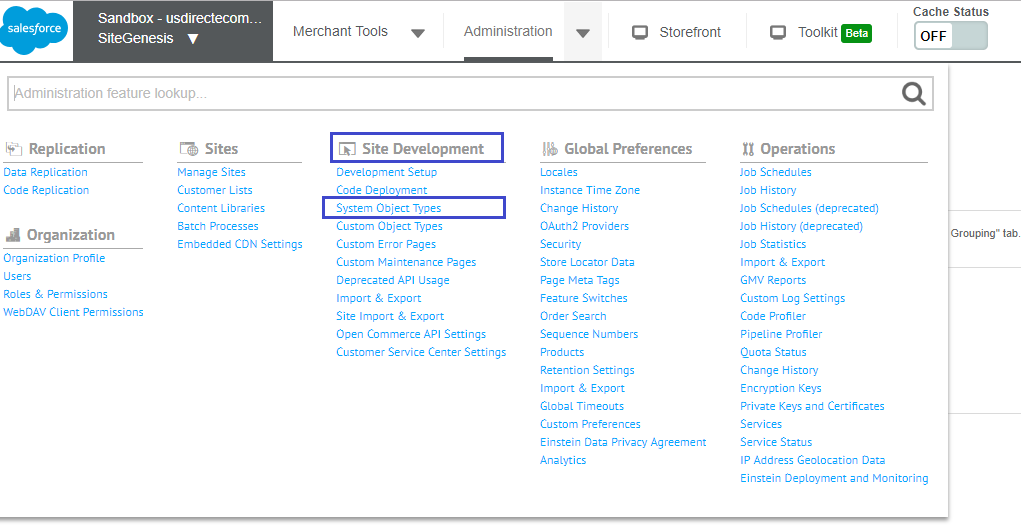
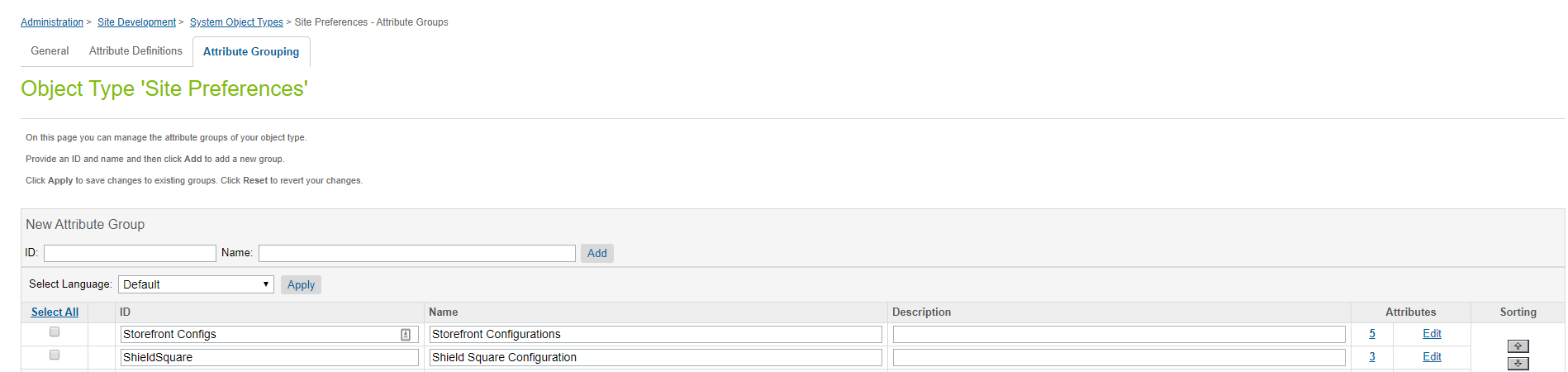
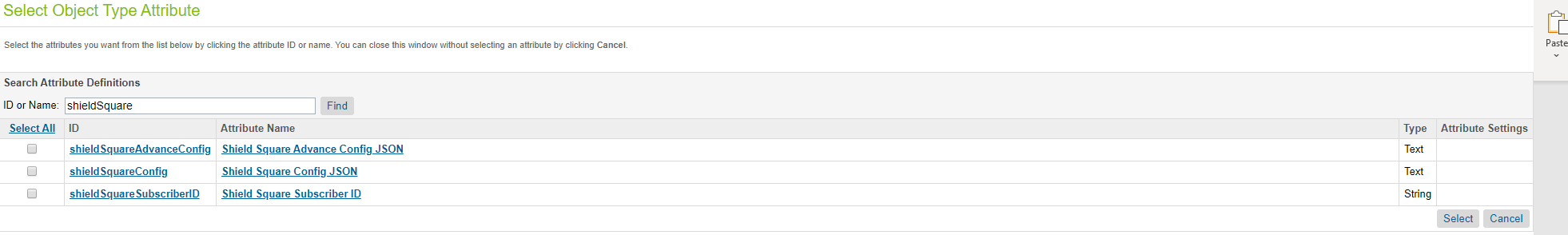


* + 1. **SFRA version:** int\_shieldsquare\_core:int\_shieldsquare\_sfra:app\_storefront\_base:plugin\_cartridge\_merge:plugin\_instorepickup:plugin\_wishlists:plugin\_giftregistry:lib\_productlist:plugin\_productcompare:plugin\_sitemap:plugin\_applepay:plugin\_datadownload:app\_storefront\_base
  1. **Importing Metadata**
* Go to **Metadata** in your downloaded folder. Rename the **SiteGenesis** folder to your **site id**.
* To import Sitesdata, Zip the folder “sitesdata”. And Import sitesdata.zip into your instance (Meta Data, Jobs and services)
* Assign the **int\_shieldsquare\_core / int\_shieldsquare\_sfra** cartridge to your site’s cartridge path like this. It can be placed anywhere in the cartridge path.
* User can now make use of the cartridge

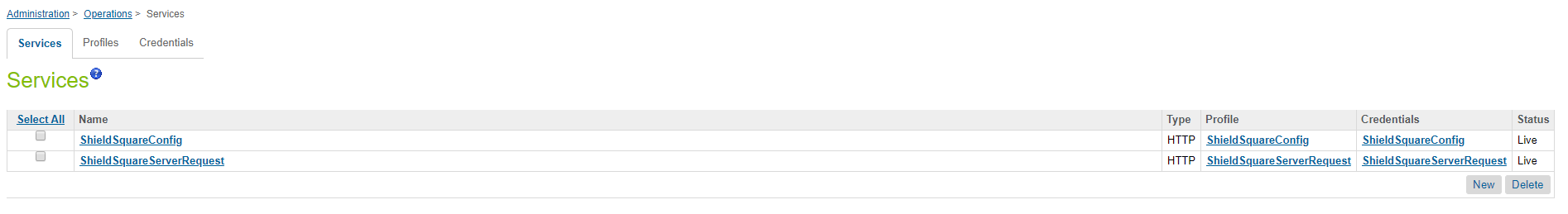
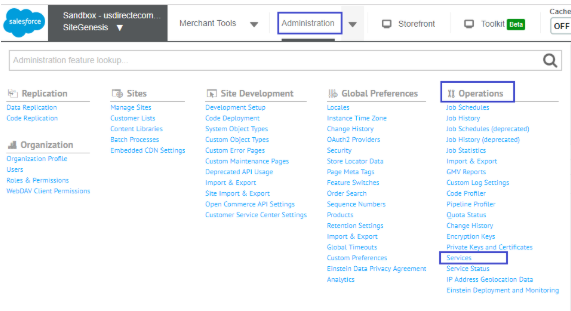
1. **Site Configurations**
   1. **Custom Preferences**

* Navigate to (Merchant Tools > Site Preferences > Custom Preferences)
* Custom Preferences are used for several configurations of Shieldsquare cartridge as explained below.



* **Shield Square Config JSON-** This initial configuration can be downloaded from ShieldSquare portal. Download salesforce connector from portal, you will find ss2\_config.json.
* **Shieldsquare Advance Config JSON-**
* This configuration will update on every 5 min. We have created one job called ShieldSquareUpdateConfig
* Using this job this custom preference will update on every 5 min. This custom preference will used to build Shield Square request
* **Shieldsquare Subscriber ID-** Subscription ID will be to find from shiedsquare portal (https://portal.radwarebotmanager.com/subscriber-id-details). This subscription id is used in shieldsquare request.
  1. **System Object Types**
* Go to > Administration > Site Development > System Object Types > Site Preferences
  1. **Attribute Grouping**
* Click on Attribute Grouping tab to view/update the Site Preference Groups ID and Name
  1. **Attribute Definitions**
  2. **Services**

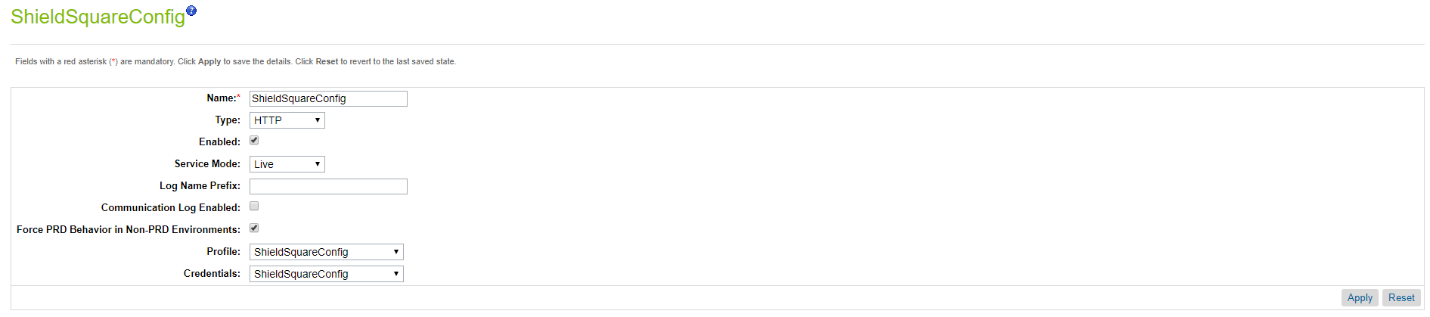
(Administration > Operations > Services)

There are two services used for Shieldsquare requests. To find the services, follow the steps given below. Administration > Operations > Services and search for keyword “Shieldsquare”. it will show below two services.

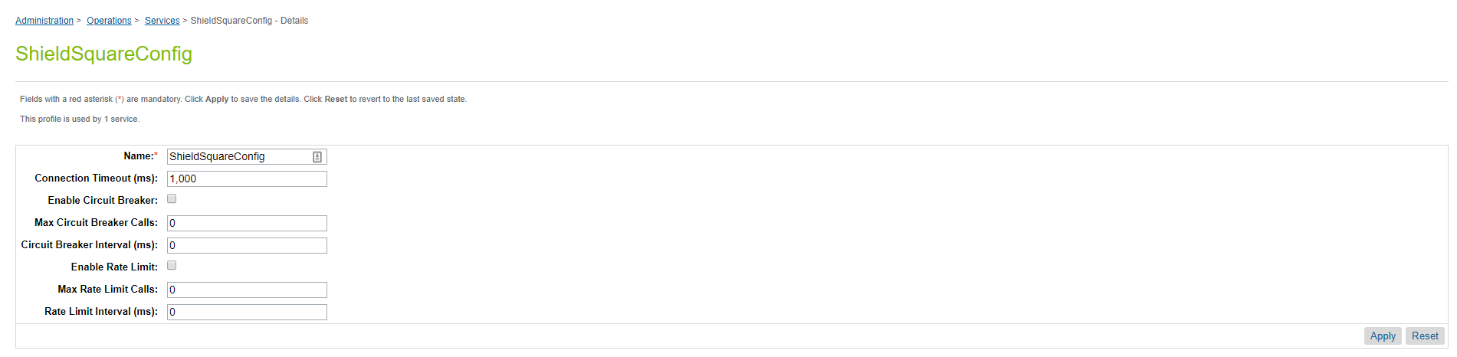
Every Service will have their Credentials configured. User can find Credentials under “Credentials” tab as shown in above screenshot.

* **ShieldSquareConfig**

This is ShieldSquareConfig service configured as per below image.

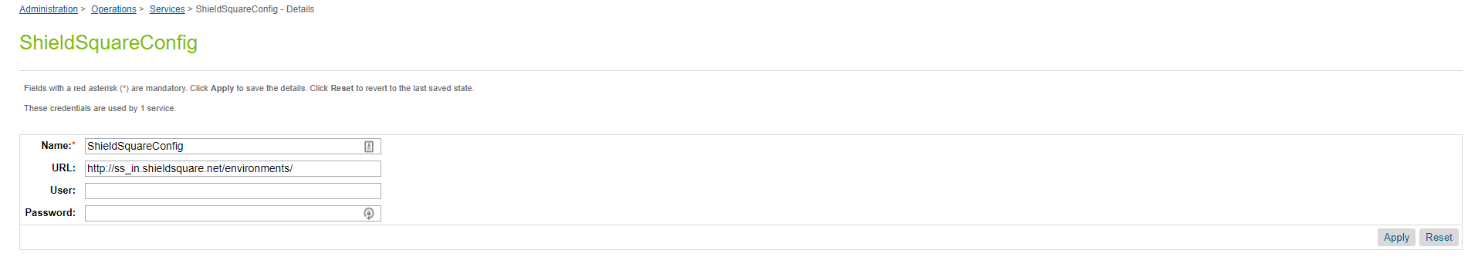


* **ShieldSquareConfig**

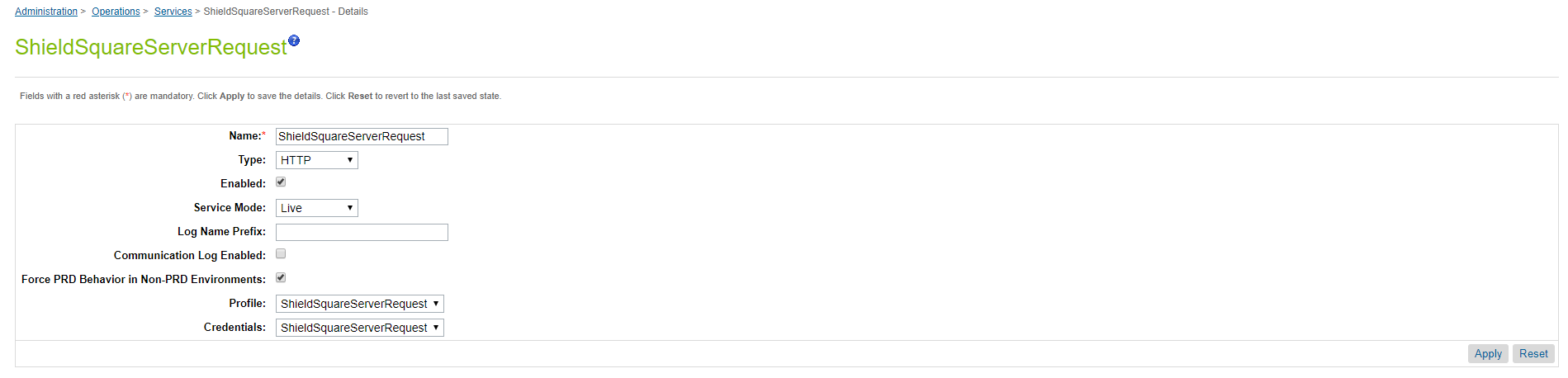
This is ShieldSquareConfig profile used in service

* **ShieldSquareConfig**

This is ShieldSquareConfig used in ShieldSquareConfig service.

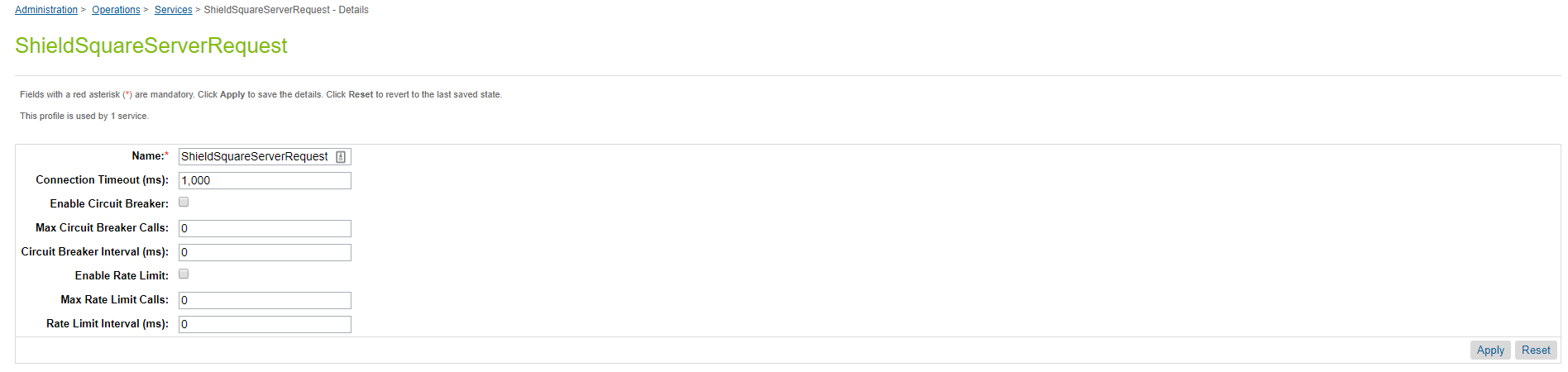
The URL will be provided by Shieldsquare

* **ShieldSquareServerRequest**

This is ShieldSquareServerRequest configured as per below image.

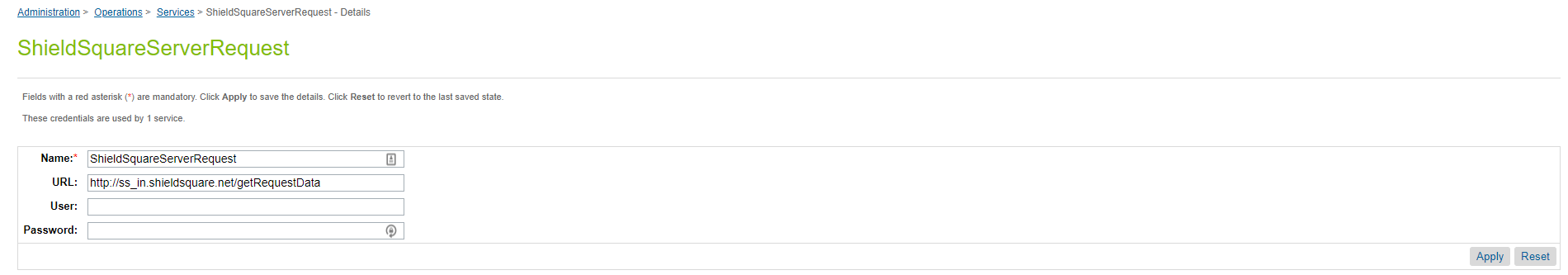
* **ShieldSquareServerRequest**

This is ShieldSquareServerRequest used in ShieldSquareServerRequest.

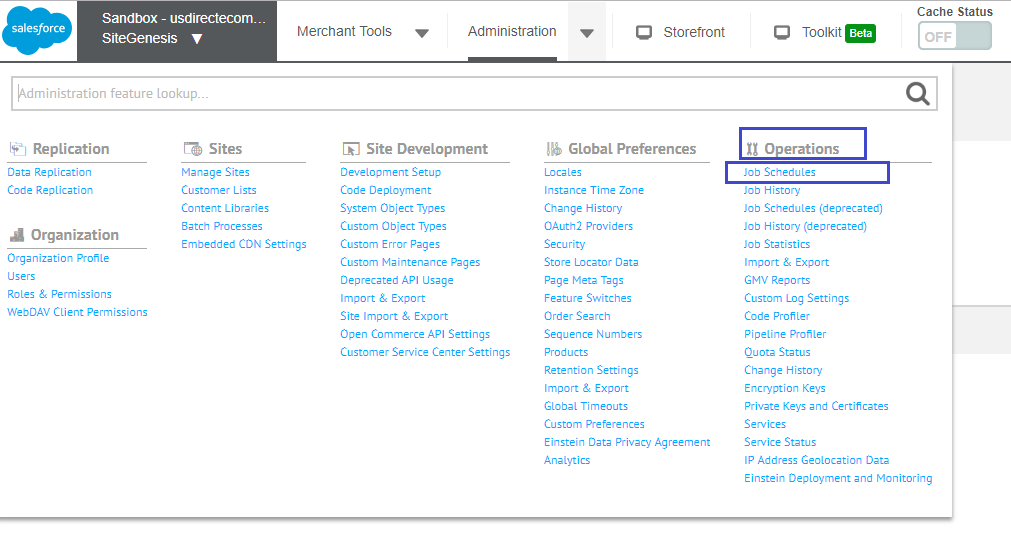
The URL will be provided by Shieldsquare.

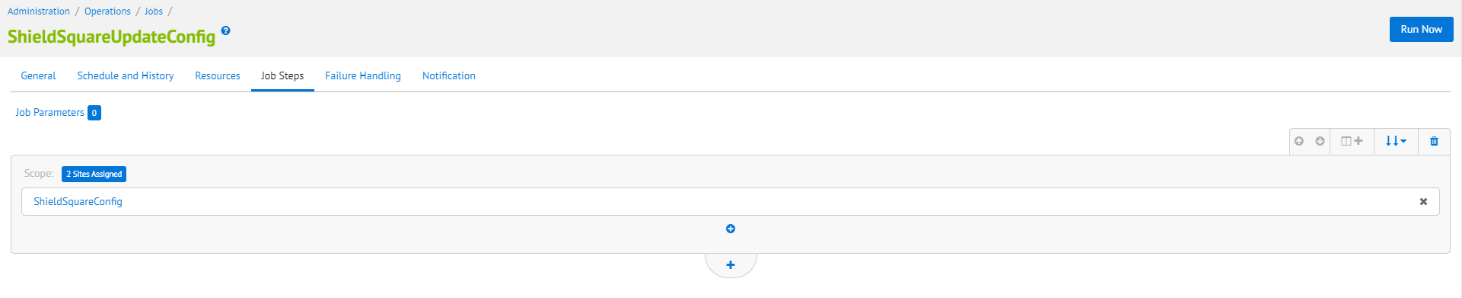
* **ShieldSquareServerRequest credentials**

This is ShieldSquareServerRequest used in ShieldSquareServerRequest.

The URL will be provided by Shieldsquare.

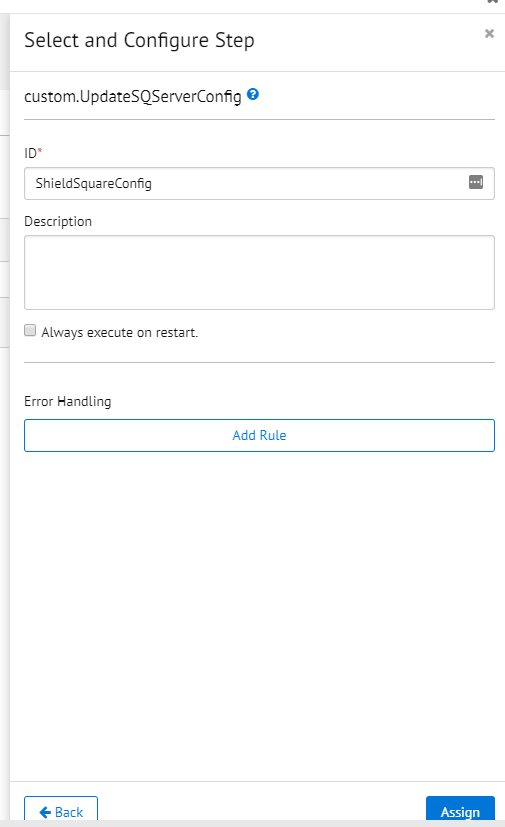
* 1. **Job Schedules**
* Administration > Operations > Job Schedules)



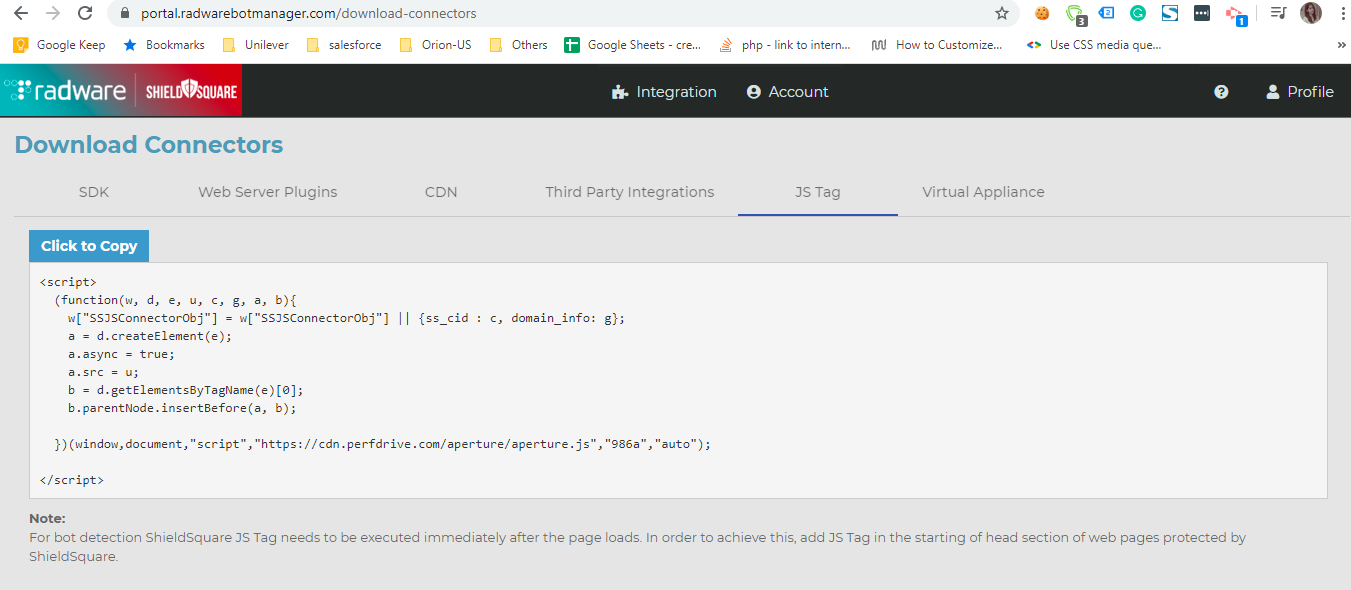
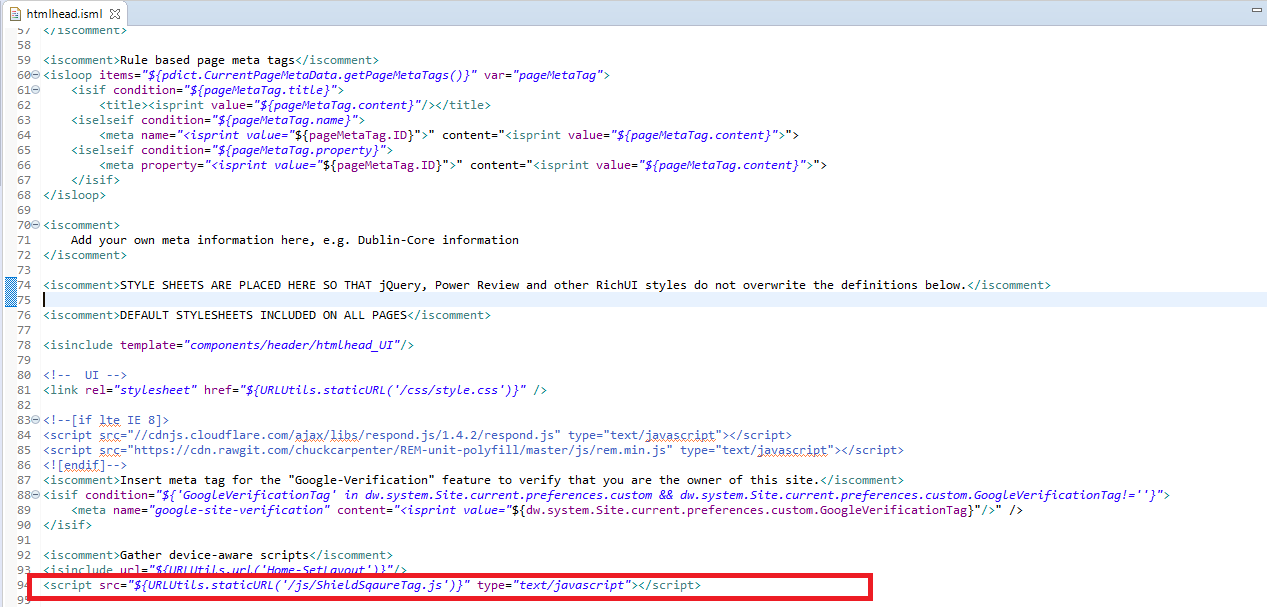
* Shieldsquare makes use of 1 job as shown below.
* **ShieldSquareUpdateConfig-** This job is used to update configuration in Shieldsquare.

Once the configuration section opens,

* select **custom.UpdateSQServerConfig**
* configure ID , **custom.UpdateSQServerConfig,**



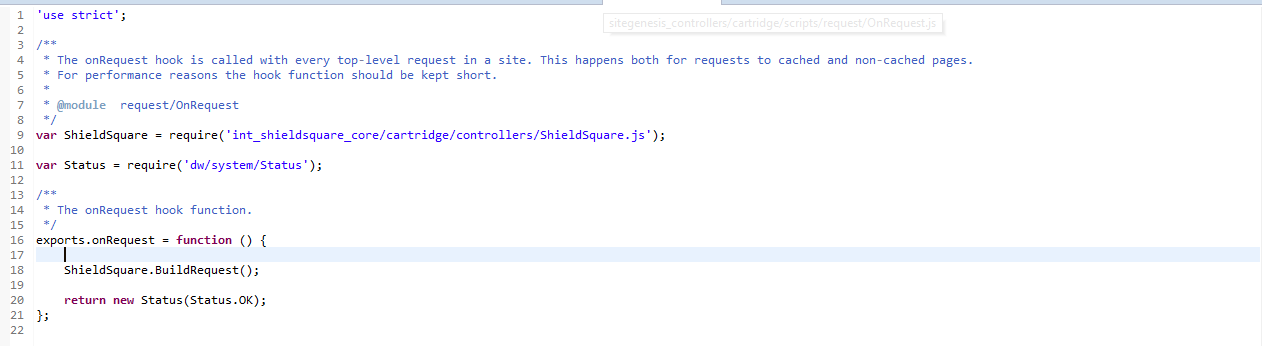
1. **Integrate the Shieldsquare JS Tag to your code**

* To Integrate JS connector, log In Shieldsqaure portal and got to download connector page 🡪JS Tag and copy the JS snippet (<https://portal.radwarebotmanager.com/download-connectors>)
* Create js file in your project.
* Add js file in htmlhead.isml as below

1. **Customized Code**
   1. **For Controller**
      1. **OnRequest.js**

Path:

SiteGenesis\_controllers/cartridge/controllers/OnRequest.js

Below code is added to OnRequest.js.

var ShieldSquare = require('int\_shieldsquare\_core/cartridge/controllers/ShieldSquare.js')

Call this function inside export.onRequest

ShieldSquare.BuildRequest();

* 1. **For SFRA**
     1. **package.json**

Path:

int\_shieldsquare\_sfra/package.json

Below code is added to package.json

{

./hooks.json

}

* + 1. **hooks.json**

Path:

int\_shieldsquare\_sfra/hooks.json

Below code is added to hooks.json

{

"hooks": [

{

"name": "dw.system.request.onRequest",

"script": "./cartridge/scripts/hooks/OnRequest.js"

}

]

}

* + 1. **htmlhead.isml**

Path:

int\_shieldsquare\_sfra/cartridge/templates/default/common/htmlHead.isml

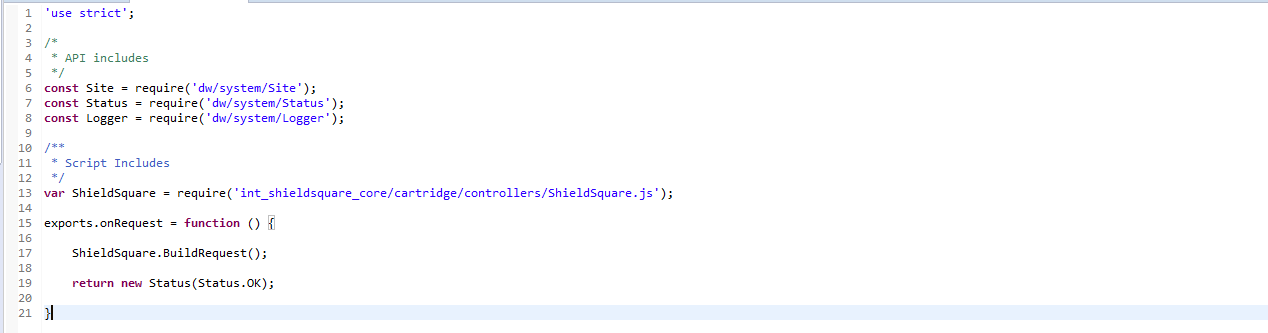
Below code is added to htmlhead.isml.

<script src="${URLUtils.staticURL('/js/ShieldSqaureTag.js')}" type="text/javascript"></script>

* + 1. **OnRequest.js**

Path:

int\_shieldsquare\_sfra/cartridge/scripts/hooks/OnRequest.js

Below code is added to OnRequest.js.

var ShieldSquare = require('int\_shieldsquare\_core/cartridge/controllers/ShieldSquare.js');

exports.onRequest = function () {

ShieldSquare.BuildRequest();

return new Status(Status.OK);

}

1. **Operation and Maintenance**
   1. **Availability**

High availability with auto scale infrastructure design, Shieldsquare has uptime of 99.99%. If Shieldsquare endpoint goes down, Shieldsquare cartridge will not take actions to your requests and it will not break your pages.

* 1. **Support**

Drop us mail @ [support@shieldsqaure.com](mailto:support@shieldsqaure.com) for the SFCC integration or submit your request to via [contact us](https://www.shieldsquare.com/contact/) page.

1. **Release History**

|  |  |  |
| --- | --- | --- |
| **Version** | **Date** | **Changes** |
|  |  |  |