SumTotal Talent Acquisition

Background Check Connector Administrator Guide

Last Updated: August 2020

Legal Notice

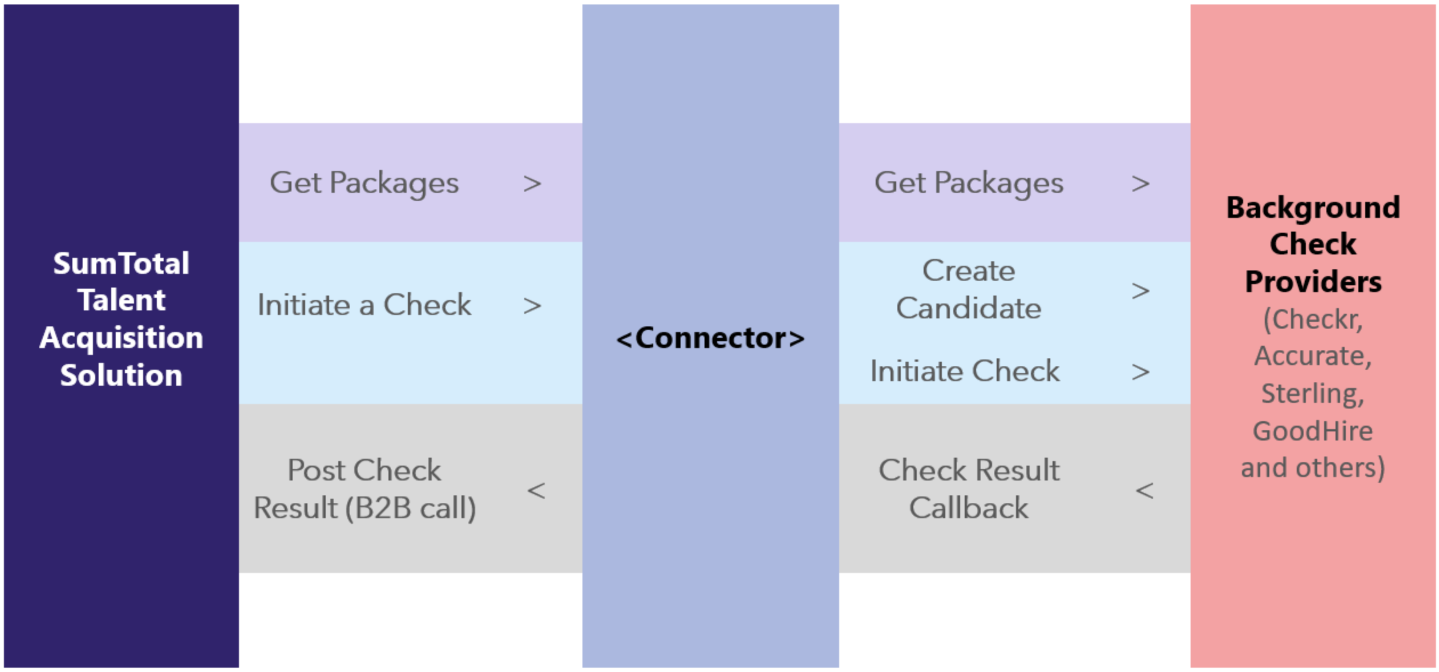
Publication Date: February 2020

Printed and reproduced in the United States of America
SumTotal Systems, LLC
107 Northeastern Blvd., Nashua, NH 03062 USA

Content

Introduction

SumTotal's Background Check Connector Service provides the ability to integrate various Background Check service providers into the SumTotal application.



Prerequisites

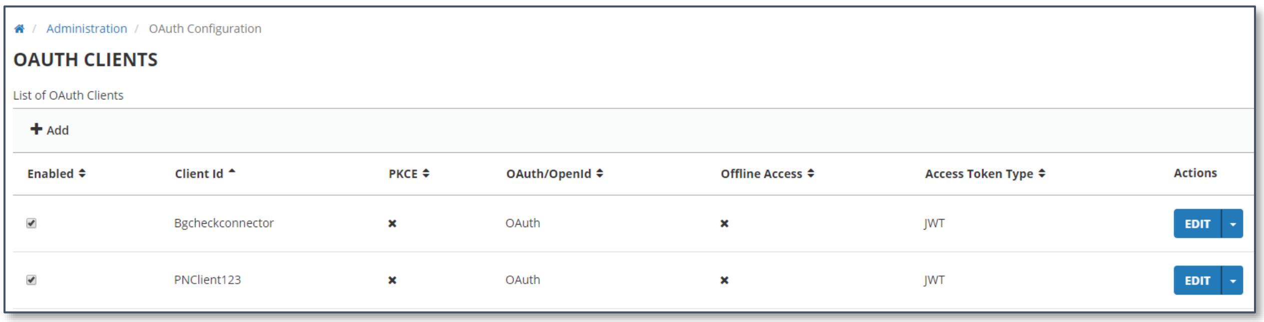
1. SumTotal Recruiting Module License
2. License for your background check vendor of choice
3. Internal development team, agreement with SumTotal Professional Services, or a certified Partner prepared to integrate your vendor using the connector service.

Setup Guide

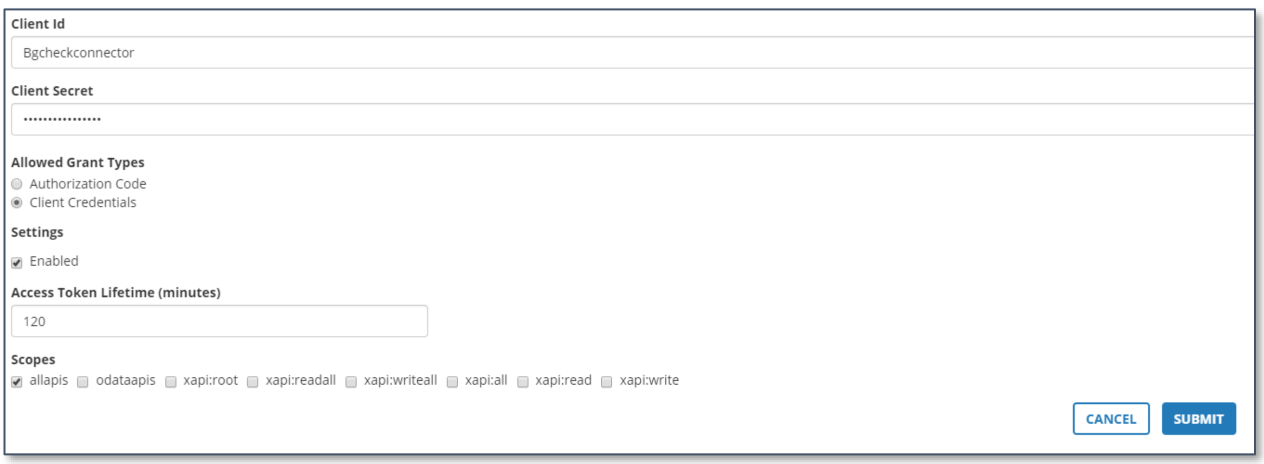
Before you can configure the connector to work with your background check provider, you must create an OAuth Client.

OAuth Configuration

1. Log in in as an Administrator to your SumTotal Talent Acquisition Site.
2. Click the **Administration** icon in the header menu.
3. Navigate to **System** > **Configuration** > **Technical Configuration** > **OAuth Configuration**.



4. Click **Add** to create a new OAuth Client.



   a. Enter desired **Client ID**.
   b. Enter secure **Client Secret**.
   c. Under **Allowed Grant Types,** select **Client Credentials**.
   d. Select desired **Scopes**:
      i. **allapis** - Access SumTotal Rest APIs

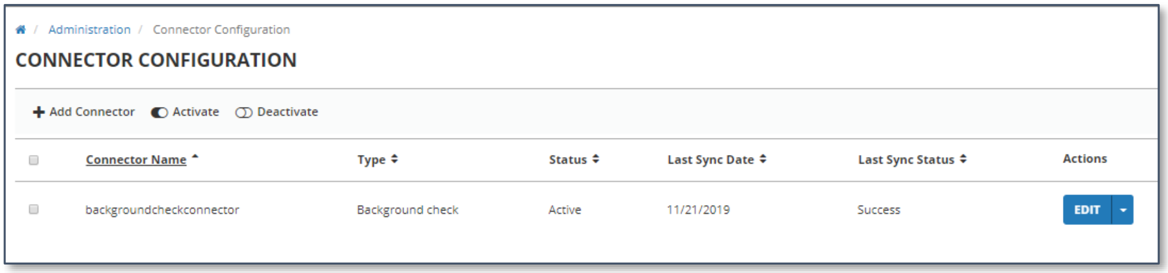
Note: *If you use this client for other purposes, you may have additional scopes checked.*

   e. Click **Submit.**

Connector Configuration

Once you've setup your OAuth configuration above, set your connector configurations.

1. Log in as an Administrator to your SumTotal Talent Acquisition Site.
2. Click the **Administration** icon in the header menu.
3. Navigate to **System** > **Configuration** > **Technical Configuration** > **Connector Configuration.**



4. Click **Add Connector**.

5. Under **Connector Configuration**, enter the following configurations:
   a. **Connector Name:** Connector Id which was configured in the connector service project.
   b. **Connector Secret Key:** Connector Id which was configured in the connector service project.
   c. **Type:** This is the type of connecter being used. Select **Background Check**.
   d. **Connector Host Name**: This is the base URL endpoint for all the background check api's. This is provided by your vendor/development team.
   e. Check the **Active** checkbox
6. Under **OAuth Configuration**:
   a. Select the OAuth client in the dropdown menu that you configured above.
7. Click **Save.**

Note: *Connector Host Name field needs complete route URL of the connector hosted environment ex: https://connectorhostedbaseurl/ControllerName (where the REST API call implemented in connector).*

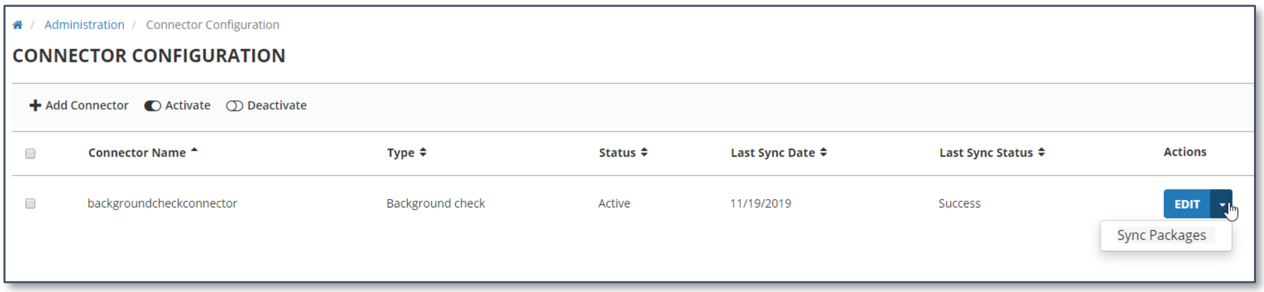
SumTotal Talent Acquisition also supports integration of multiple background check vendors with the current implementation of connector service solution but each connector service should have a different hosted environment.

Note: *Multiple vendors using one hosted connector service is not supported. To avoid overly complex integrations, each vendor needs one hosted connector service.*

Using Checks in the Application

After setting up the OAuth Configuration and Connector Configurations above, you need to syncronize the various background check packages available from your vendor.

The **Sync Packages** action brings those packages into the local system, making them available when working with candidates in Recruiting. Click the **Sync Packages** action to send a request from the application to the connector service endpoint (`GetPackages`) and from there the connector will send a request to your vendor site and returns all the available packages.



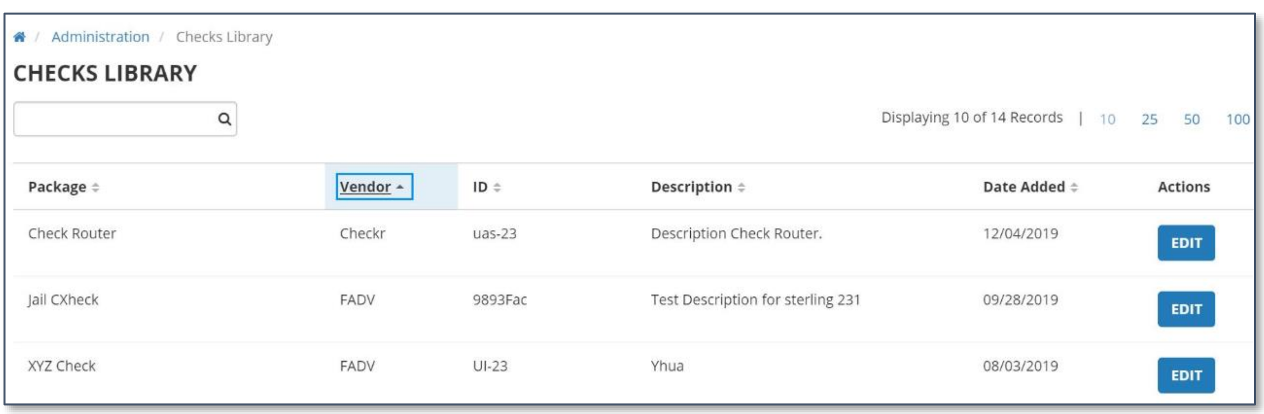
***Note:** To set up connector configurations, your account needs to the Manage Connector Configuration permission.*

Checks Library

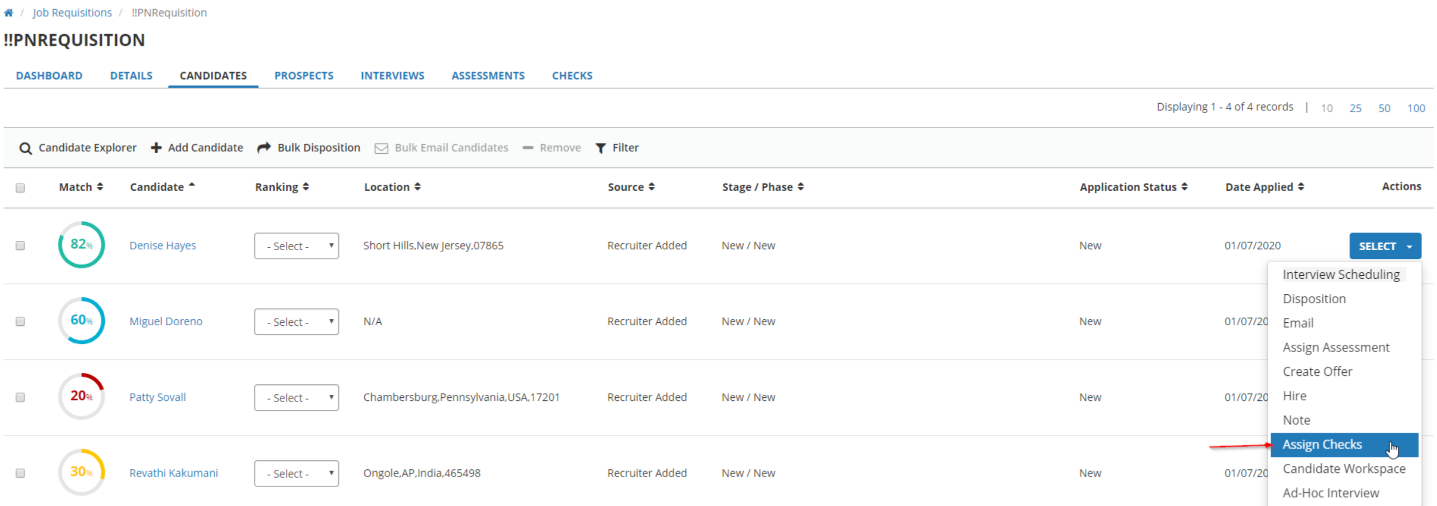
After packages are synchronized on the **Connector Configuration** page, available packages from the vendor import into the SumTotal application and become available in the **Checks Library**.

To access the Checks Library:

1. Click the **Administration** icon in the header menu.
2. Navigate to **Recruiting** > **Checks Library**.

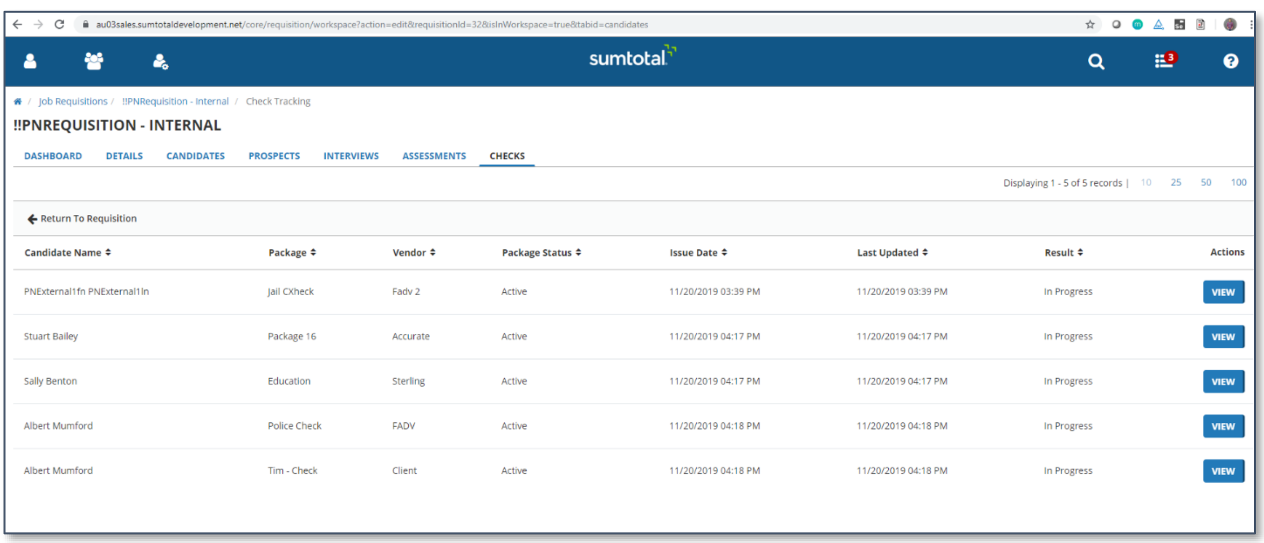


Once a recruiter assigns a package to the selected candidate, the system will send an initiate check request to the connector service endpoint (`InitiateCheck`), which will then create a check order at the vendor side, and then the connector returns an orderId.



Once the vendor returns the status of the candidate check to the connector service (through call back/webhook), that status will be returned to SumTotal REST API through B2B call-back (`UpdateBackgroundCheck`).

Recruiters can track the status and results of the check in the **Checks** tab of a requisition. Clicking **View** will open the vendor site with the related check results. The recruiter may require appropriate credentials with the vendor site, depending on the vendor's configuration.



Connector API Setup Guide

SumTotal requires two basic REST API calls from the connector service to integrate a background check provider with Recruiting. Specifically, these REST API calls are used to:

- Get the available packages from the vendor
- Initiate a check for a candidate
- Get the status of an assigned check

SumTotal suports hash-based authentication for the APIs implemented in the connector solution. For more information on authentication and REST APIs:

- Check out the Github organization and template project.
- Review the Swagger documentation in your site. (https://<YourSumTotalSiteURL>/apis/swagger)

FieldsMapping in SumTotal

Request Parameters

Connector API Name	Field Name	Description
`InitiateCheck`	`CandidateId`	The unique identifier for the package the candidate will be screened against Required
	See [Candidate Objects](#) for a list of candidate objects.	Candidate details. If no candidateId then candidate object is required with some mandatory fields: ○ Email ○ FirstName ○ Lastname
	PackageId	The Package Id of the Provider
	PackageName	The Package Name of the Provider
	PositionApplied	`The position that Candidate has applied for`
	PositionLocations	
	StatusNotificationUrl	The CallBack URL for the employer to view the background check
	ClientReferences	Client References
	QuoteBacks	Quote Backs
	Locale	Package Position Locale
	AdditionalAttributes	Additional Attributes
	CandidateType	Candidate Type
	Documents	Documents if there are any
	UserDefinedFields	User Defined Fields
	BillingAccount	The Requesting Account /chargeId
	ClientReference	Client specified codes
	ConnectorId	Connector Id for sync
Call Back (B2B)	applicantId	Applicant Id of the candidate *Required field*
	vendorPackageId	Vendor Package Id
	vendorCandidateId	Candidate id in vendor side
	vendor	Vendor name
	checkIssueDate	Check issue date
	checkModifiedDate	Check modified date
	checkResult	Status of the check
	ReportURL	Report URL of the check

Candidate Object

- Email
- Date of birth
- Gender
- PersonNameDetails
  - FirstName
  - LastName
  - MiddleName
  - Suffix
  - Type
- PersonAddresses(list)
  - City
  - State
  - Country
  - ZipCode
  - StreetAddress
  - StartAt
  - EndAt
- PersonPhones(list)
  - PhoneNumber
  - CountryCode
  - Extension
  - Type
- PersonIdentities(list)
  - Type
  - IdNumber
  - IssueDate
  - ExpiryDate
  - IssuedCountry
- PersonLicenseDetailsList
  - Type
  - LicenseNumber
  - IssueDate
  - ExpiryDate
  - IssuedCountry
  - IssuedState
- PersonEmploymentDetailsList
  - CompanyName
  - Title
  - StartAt
  - EndAt
  - StreetAddress
  - State
  - City
  - EmployerContactEmail
  - EmployerContactPhone
  - IsCurrentlyEmployed
  - EmployeeId
- PersonEducationDetailsList
  - InstitutionName
  - Degree
  - DegreeType
  - ZipCode
  - StreetAddress
  - State
  - City
  - StudyMajor
  - StartYear
  - EndYear
  - InstitutionContactEmail
  - InstitutionContactPhone
- PersonReferences(list)
  - FirstName
  - LastName
  - ContactEmail
  - ContactPhone

Response Parameters

API Name	Field Name	Description
InitiateCheck	PackageId	The Package Id which is assigned to the candidate
	ApplicantId	The unique id for each candidate invitation
	CandidateId	The CandidateId property gets/sets the value of the Candidate Id.
	OrderId	The unique id for each screening
	HttpStatusCode	Response status
Get Packages	PackageName	Package name
	PackageId	Package Id from the vendor
	PackageDescription	Package description from connector
	Vendor	Vendor Name
	PackageStatus	Package status.
	PackageSubcomponents	Package sub components
	PackageRequiredFields	Package required fields
Call Back(B2B)	Http status codes	Response status codes

Technical Guide to Implement Connector Service

Use the following instructions to configure your background check connector service.

1. Grab the template solution from SumTotal's github repository: https://github.com/sumtotal-services
   a. Your development team should be familiar with asp.net core, which the template solution was built with.
2. The template solution was implemented with the complete authentication mechanism (between the SumTotal application and the connector) – which means you do not need to set up the authentication part while building connector service.
3. Open the **appsetting.json** file.

```
"Settings": {
  "ConnectorId": "backgroundcheckconnector",
  "ConnectorSecret": "BackgroundCheckConnector",
  "SumtOAuthSettings": {
    "Authority": "https://localhost:44330/",//Authentication Url
    "Scope": "allapis",
    "ClientId": "Bgcheckconnector",
    "ClientSecret": "Bgcheckconnector"
  },
  "Vendors": [
    {
      "VendorId": "sterlingConnector",//here sterling is the vendor name
      "VendorOAuthSettings": {
        "SumtCallBackUrl": "{Host}/sampleTemplate/restApi/updateChecks",
        "VendorOauth": "https://sampleTemplate.com/auth/token",
        "VendorUser": "vendorClient",
        "VendorPassword": "vendorClientSecret",
        "BaseUrl": "https://SampleTemplateSterlingUrl.com/"
      }
    }
  ]
},
```

   a. Enter your connector details:
      i. **ConnectorID:** Unique Name given to the connetcor service.
      ii. **ConnectorSecret:** Connector servcie secret key which is used for authentictaion purpose.
      iii. **SumtOAuthSettings:** OAuth settings which were provided by SumTotal.
      iv. **VendorOAuthSettings:** OAuth settings which were provided by Vendor.
4. Open the **defaultMapper.cs** file.

5. Configure the REST API Controller, which will handle the business calls between the vendor and the application. The template solution has a basic structure for those API calls, but will need to be customized to work with your specific vendor:
    a. Fields will need to be mapped properly for requests and responses of the API Calls, implemented in defaultMapper.cs
    b. Data validations will need to be set up in accordance with your vendor's documentation.
6. The template solution also provides a default logging framework to log info, errors, and warnings. In appsettings.json, update the URL where you want the logs placed.
7. Add OAuth ClientId as vendorUser, OAuth clientSecret as vendorPassword and CallBackUrl as SumtCallBackUrl.

   These OAuth details should be same as we configured in Vendor site.

```
"VendorOAuthSettings": {
  "SumtCallBackUrl": "{Host}/sampleTemplate/restApi/updateChecks",
  "VendorOauth": "https://sampleTemplate.com/auth/token",
  "VendorUser": "vendorClient",
  "VendorPassword": "vendorClientSecret",
  "BaseUrl": "https://SampleTemplateSterlingUrl.com/"
}
```

8. Add ClientId, ClientSecret and Authority with the details which we configured in OAuthConfiguration page.

```
"SumtOAuthSettings": {
  "Authority": "https://localhost:44330/",//Authentication Url
  "Scope": "allapis",
  "ClientId": "Bgcheckconnector",
  "ClientSecret": "Bgcheckconnector"
},
```

9. Update the **defaultMapper.cs** file to map the properties the CallBackURL is expecting. Refer to the FieldsMapping in SumTotal section above.