ImpactControlSco Front-End Developer Documentation

Introduction

ImpactControlSco is the external interface for the SCO vendors to use the My-Scan Impact CTRL environment. By providing a clean and simple interface it gives retailers the ability to decrease shrinkage in their store with minimal adjustments to POS/SCO. Not only does ImpactControlSco supports static rescan triggers, it also supports the AI/ML functionality of Impact CTRL. All within the same API.

For more information about the exact requests and responses and HTTP codes, see the Swagger documentation available when on the URI of the ImpactControlSco.

Calling ImpactControlSco

Calls to the ImpactControlSco have the following URI format:

{BaseURI}/{APIversion}/{Controller}/{Command}

BaseURI: the protocol, server and port, for example: "http://localhost:6996"

APIVersion: the version of the API, for example "v1" or "v2"

Controller: can be "CheckForRescan", "RescanResult", "PaymentConfirmation" or "HealthCheck"

Command: see the Swagger documentation for available commands.

All request headers should have Content-Type : "application/json". All responses are in json format.

When errors occur, the response will contain error information as specified in RFC-7807.

Example scenarios

No Rescan	POST APIversion/CheckForRescan
	PUT APIversion/PaymentConfirmation
Rescan by SCO vendors	POST APIversion/CheckForRescan
	Rescan Calls
	PUT APIversion/RescanResult
	PUT APIversion/PaymentConfirmation
Rescan by RescanService	POST APIversion/CheckForRescan
	common/AvailableRescans
	partial/Start
	partial/Add
	partial/Finish
	GET APIversion/RescanResult
	PUT APIversion/PaymentConfirmation

POST CheckForRescan

- It is considered the starting point of all scenarios executed on the ImpactControlSCO APIs.
- It receives the shopping trip details in the body and decides whether rescan is required or not.

- It uses **External Ticket ID** to cache ticket details and is being used in Get RescanResult as (**BasketId**).

API:

End point	Request Body	Response Body
APIversion/CheckForRescan	**CheckBasket**: CheckBasket object **ScoInformation**: ScoInformation object	**200** **CustomerId** : string **StoreId** : string **SelectedForRescan** : bool **FullRescan** : bool **400** - ValidationProblemDetails **500** - ProblemDetails

CheckBasket object contains all Basket Details, and SCOInformation contains POS Number (ScoId) & Employee ID (AuditorId)

Object	Attributes
ScoInformation	**ScoId** : int **AuditorId** : string
CheckBasket	**Id** : string **StoreId** : string **CustomerId** : string **TripStartTime** : string **TripEndTime** : string **SelectedItems** : List<ScanableItem> **Language** : string - optional **RemovedItems**: List<ScanableItem> - optional **Attributes** : IEnumerable<Attribute> - optional
ScanableItem	**ScanTimeStamp** : string **Barcode** : string **ItemId** : string **Quantity** : int **Price** : Price **Department** : string - optional **ShortDescription** : string - optional **ItemAttributes** : IEnumerable<Attribute> - optional
Price	**GrossPrice** : double **UnitPrice** : double **ItemDiscount** : double **LiquidationDiscount** : double **Deposit** : double
Attribute	**Name** : string **Value** : string

The response indicates if the specified customer is selected for a rescan (SelectedForRescan = true) or not (SelectedForRescan = false). If SelectedForRescan is true, the FullRescan flag determines a partial (FullRescan = false) or full (FullRescan = true) rescan.

PUT RescanResult

In case of **Rescan is required and managed by SCO vendors** , after finishing rescan a call to RescanResult is required to upload the resulting rescan ticket ,

This endpoint is used to report back what happened in the rescan when it is preformed on the SCO.

API:

End point	Request Body	Response

APIversion/RescanResult	• **ResultBasket**: ResultBasket object • **RescanInformation**: RescanInformation object • **ScoInformation**: ScoInformation object	• **204** - No content • **400** - ValidationProblemDetails • **500** - ProblemDetails

Object	Attributes
ResultBasket	• **Id** : string • **StoreId** : string • **CustomerId** : string • **Items** : ICollection<ScanableItem> • **Attributes** : ICollection<Attribute>
RescanInformation	• **RescanType** : Enum (None =0, Partial =1, Full=2)

GET RescanResult

In case of **Rescan is required and managed by Rescan Service** , after finishing rescan a call for RescanResult is required to download the resulting rescan ticket to the POS/SCO.

This endpoint is used to get the ticket which user is going to pay for.

API:

End point	Query String	Response

APIversion/RescanResult	• **BasketId** : string	• **200**
		  • **ResultBasket** : ResultBasket object
		  • **RescanInformation** : RescanInformation object
		  • **ScoInformation** : ScoInformation object
		• **400** - ValidationProblemDetails
		• **500** - ProblemDetails

PUT ConfirmPayment

After **Rescan is done successfully** this endpoint is used to confirm that the ticket is paid.

It is considered the end point of all scenarios executed on the ImpactControlSCO APIs, it also writes the pos ticket

API:

End point	Request Body	Response
APIversion/ConfirmPayment	• **BasketId** : string • **StoreId** : string	• **204** - No Content • **400** - ValidationProblemDetails • **500** - ProblemDetails

GET HealthCheck

As ImpactControlSco internally calls Business Server (BS) and Ticket integration service (TIS) , so we need to check the availability for the 2 services to make sure ImpactControlSco can work fine.

API:

End point	Response
/HealthCheck	• **200** - string

The response represents the status for ImpactControlSco, BS and TIS status