



# LPaaS API Manual

## Table of Contents

1. Introduction to LPaaS.....	5
2. API's Workflow .....	5
2.1. POS API Sequence/Purpose .....	5
2.2. Ecommerce API Sequence/Purpose .....	7
3. API Details .....	8
3.1. Security .....	9
3.1.1. Generate Security Token.....	9
3.2. Registration of Member.....	12
3.2.1. IsCustomer.....	12
3.2.2. Register Customer.....	13
3.2.3. Update Member Profile .....	20
3.2.4. Search Member .....	23
3.2.5. Save Children Details .....	29
3.3. Accrual of Loyalty Points.....	30
3.3.1. Accrue Loyalty Points to a Member's Account .....	30
3.3.2. Flat Accrual to Member's Account .....	35
3.4. Redemption of Loyalty Points .....	36
3.4.1. Get Available Points of a Member.....	36
3.4.2. Check Availability for Easy Points Redemption .....	39
3.4.3. ConfirmOTP .....	44
3.4.4. Resend OTP.....	45
3.4.5. Release Redemption Points .....	47
3.4.6. Confirm Redemption against Code Received.....	49
3.4.7. Flat Redemption from Member's account .....	54
3.5. Issuance and Redemption of Coupons .....	55
3.5.1. Issue Coupon .....	56
3.5.2. Get LPaaS Campaign Available Coupon.....	58
3.5.3. Check Coupon Availability .....	61
3.5.4. Redeem Coupon .....	65
3.5.5. Coupon OTP.....	68
3.5.6. Resend Coupon OTP.....	70
3.5.7. Use Coupon .....	71
3.5.8. Unblock Coupon .....	73
3.5.9. Update Coupon Status .....	74
3.6. Issuance and Redemption of Gift Voucher (GV) .....	76

3.6.1.	Issue GV .....	76
3.6.2.	Get Available GV.....	80
3.6.3.	Check Availability of GV .....	82
3.6.4.	Redeem GV .....	85
3.6.5.	GV OTP .....	87
3.6.6.	Use GV .....	89
3.6.7.	Unblock GV .....	92
3.6.8.	Rollback GV.....	93
3.6.9.	Update GV Status .....	95
3.7.	Accruals based on SKUs .....	97
3.7.1.	GetSKUBillDetails .....	97
3.7.2.	SaveSKUBillDetails .....	102
3.8.	Ecommerce Customer Journey .....	105
3.8.1.	Customer Signup .....	105
3.8.2.	Customer Login.....	110
3.8.3.	Customer Profile.....	111
3.8.4.	Forgot Password.....	114
3.8.5.	Update Password .....	115
3.8.6.	Member Address Detail .....	117
3.8.7.	Get Address Detail .....	119
3.8.8.	Save SKU Bill Details.....	121
3.8.9.	Lookup .....	127
3.8.10.	Order Status Update .....	129
3.9.	Miscellaneous.....	131
3.9.1.	Easy Points Refund .....	131
3.9.2.	Get Tier Details .....	134
3.9.3.	Assign Tier .....	136
3.9.4.	Get Customer Transaction Details .....	138
3.9.5.	Get Referral Code .....	142
3.9.6.	Update Bill Number.....	144
3.9.7.	Generate OTP .....	145
3.9.8.	Validate OTP .....	147
4.	List of Return Codes .....	148
5.	Physical Store Customer Lifecycle .....	152
5.1.	Member Enrolment Flow .....	152
5.2.	View Customer Widget .....	153

<b>5.3.</b>	<b>Point Redemption Flow .....</b>	<b>154</b>
5.3.1.	Point as a Discount - Impact/Concerns .....	155
5.3.2.	OTP delivery Process:.....	156
<b>5.4.</b>	<b>Coupon Redemption Flow .....</b>	<b>156</b>
<b>5.5.</b>	<b>Sales Return.....</b>	<b>158</b>
5.5.1.	Customer is carrying the invoice .....	158
5.5.2.	Invoice is not available.....	158
5.5.3.	Non loyalty Customer .....	158
<b>5.6.</b>	<b>Offline Stores/Internet not working.....</b>	<b>158</b>
<b>5.7.</b>	<b>GV Issuance Flow.....</b>	<b>160</b>
<b>5.8.</b>	<b>GV Redemption Flow .....</b>	<b>160</b>
<b>6.</b>	<b>Widgets.....</b>	<b>161</b>
6.1.	Redeem Points .....	162
6.1.1.	Input Parameters for Redeem Point Widget URL .....	163
6.1.2.	Sample Request/Response for the Redeem Point widget in GetWorkflowWidget API.....	164
6.1.3.	Redeem Point Widget Screen .....	165
6.2.	Redeem Coupons Widget .....	165
6.2.1.	Input Parameters for Redeem Coupon Widget URL .....	167
6.2.2.	Sample Request/Response for the Redeem Coupon widget .....	167
6.2.3.	Redeem Coupon Widget Screen .....	168
6.3.	Add/Edit Customer Widget.....	169
6.3.1.	Sample Request & Response for AEC customer widget .....	171
6.4.	View Customer Widget .....	174
6.4.1.	Input Parameters for View Customer widget .....	174
6.4.2.	Sample Request & Response for the View Customer Widget .....	175
6.4.3.	Customer Widget Screen .....	176
<b>7.</b>	<b>Ecommerce Workflows .....</b>	<b>177</b>
7.1.	Registration Flow .....	177
7.2.	Login Flow.....	178
7.3.	My Account Flow.....	179
7.4.	Redemption Flow .....	180
7.5.	Accrual Flow .....	181

## 1. Introduction to LPaaS

LPaaS (Loyalty Program as a Service) is EasyRewardz's proprietary service delivery platform that offers the capability to drive loyalty programs for a partner firm. The end users for the partner firm can be their Customers, Employees or Channel Partners. Using LPaaS, partners can configure rules for Accrual and Redemption of Loyalty points, Issuance of Coupons and Gift Vouchers (GVs) and much more.

The various functionalities of LPaaS are exposed via a set of APIs. A partner firm requires to integrate these APIs with their systems to start using the services offered. The list of APIs required for integration and web methods exposed by these APIs are covered in the following sections.

The LPaaS REST are available in JSON/JSON format.

## 2. API's Workflow

### 2.1. POS API Sequence/Purpose

The below API sequence to be used by POS Integrating Partner

Integration Event	API's Involved	Purpose
Member Enrolment	IsCustomer	To search a customer into ER DB
	RegisterEasyAccount	To register a new customer or update a Member.

	UpdateMemberProfile	To update an existing user profile
Invoice Push	SaveSKUBillDetails	This API pushes the SKU level transaction data into ER database.
Activity Data Flow	EasyPointAccrualEOSS	To push activity data.
Point Redemption	CustomerAvailablePoints	To retrieve the existing points of a member
	CheckForEasyPointsRedemption	To block the loyalty points that are requested to be redeemed.
	ConfirmOTP	To validate the OTP triggered from the invocation of the wsCheckForEasyPointsRedemption method.
	ConfirmEasyPointsRedemption	To deduct the points that are being blocked by CheckForEasyPointRedemption
	ResendOTP	To resend the same OTP which was triggered from CheckforeasyPointRedemption.
	ReleaseRedemptionPoints	To release the points blocked during the wsCheckForEasyPointsRedemption method.
Coupon Issuance	IssueCoupon	To issue coupon
Coupon Redemption	GetAvailableCoupons	To get all coupons issued to a customer.
	CheckCouponAvailability	To check the availability of a specific coupon for a member.
	RedeemCoupon	To Redeem the coupon in any transaction.
	CouponOTP	To trigger an OTP to the redeeming member for validation.
	UseCoupon	Application of Coupon in the Bill
	ResendCouponOTP	To resend the coupon OTP in case the previous OTP sent by 'CouponOTP' API is not received or expired.
	UnBlockCoupon	Unblocks a coupon blocked using RedeemCoupon
GV Issuance	IssueGV	To issue GV
GV Redemption	GetAvailableGVs	To get all the issued GV's for a customer.
	CheckGVAvailabilty	To verify GV code and its status before application
	RedeemGV	To verify and block GV for redemption against a transaction
	GVOTP	To trigger an OTP for GV redemption
	UseGV	To confirm redemption of GV amount as applied in the bill

	UnBlockGV	Unblocks a blocked GV in case a customer aborts a transaction.
	RollbackGV	To rollback the amount to GV
Bill Void/Cancel	EasyPointRefund	It refunds the transaction recorded in ER
View Widget	GetWorkflowWidget	To show single view of customer in CRM

## 2.2. Ecommerce API Sequence/Purpose

The below API sequence to be used by Ecommerce Integrating Partner

Ecommerce Screen	API's Involved	Purpose
Signup	Generate OTP	To generate OTP for mobile authorization during signup
Signup	Validate OTP	To generate OTP for mobile authorization during signup
Signup	IsCustomer	To search a customer into ER DB.
Signup	RegisterEasyAccount	To register a new customer or update a member
MyAccounts/Signup	UpdateMemberProfile	To update an existing customer data.
Order Completion	SaveSKUBillDetails	This API pushes the SKU level transaction data into ER database.
Post Order Delivery	OrderStatusUpdate	To update Item Status
Customer Engagement Activity	EasyPointAccrualEOSS	To issue points on activity data like Spin a wheel, Product review and feedback
Cart Page/Point Redemption	CustomerAvailablePoints	To retrieve the existing points of a member
Cart Page/Point Redemption	CheckForEasyPointsRedemption	To block the loyalty points that are requested to be redeemed.
Cart Page/Point Redemption	ConfirmOTP	To validate the OTP triggered from the invocation of the CheckForEasyPointsRedemption method.
Cart Page/Point Redemption	ConfirmEasyPointsRedemption	To deduct the points that are being blocked by CheckForEasyPointRedemption
Cart Page/Point Redemption	ResendOTP	To resend the same OTP which was triggered from CheckforeasyPointRedemption.
Cart Page/Point Redemption	ReleaseRedemptionPoints	To release the points blocked during the CheckForEasyPointsRedemption method.
Cart Page/Coupon Redemption	CheckCouponAvailability	To check the status a specific coupon for a member.
Cart Page/Coupon Redemption	RedeemCoupon	The API blocks the coupon and Triggers OTP if applicable for Campaign

Cart Page/Coupon Redemption	CouponOTP	To validation an OTP to the redeeming member for validation.
Cart Page/Coupon Redemption	UseCoupon	Application of Coupon in the Bill
Cart Page/Coupon Redemption	ResendCouponOTP	To resend the coupon OTP in case the previous OTP sent by 'CouponOTP' API is not received.
Cart Page/Coupon Redemption	UnBlockCoupon	Unblocks a coupon blocked using RedeemCoupon
Payment Page/GV Redemption	CheckGVAvailallity	To check the status a specific GV for a member.
Payment Page/GV Redemption	RedeemGV	The API blocks the GV and Triggers OTP if applicable for GV
Payment Page/GV Redemption	GVOTP	To validation an OTP to the redeeming member for validation.
Payment Page/GV Redemption	UseGV	Application of GV in the Bill
Payment Page/GV Redemption	ResendCouponOTP	To resend the OTP in case the previous OTP sent by 'GVOTP' API is not received.
Payment Page/GV Redemption	UnblockGV	Unblocks a GV blocked using RedeemGV
Payment Page/GV Redemption	RollbackGV	Rollback a GV post UseGV
MyAccounts/Point Ledger	GetCustomerTransactionDetails	To Show Point Credit and debit Entries across all channels
MyAccounts/Omni Channel Transaction Display	GetCustomerTransactionDetails	To show all omni channel transactions
MyAccounts/Provisional Point Display	Lookup	To show provisional points
My Accounts/Coupons	GetLPaaSACampaignAvailableCoupon	To get all coupons issued to a customer.
Login Management	CustomerLogin	To validate customer email/mobile and password combination is valid or not
Password Management	forgotPassword	This api trigger OTP on customer email which can be used to login again
Password Management	updatePassword	To update the existing password
Address Management	memberAddressDetail	To store the address details
Address Management	getAddressDetail	to get the address details
Ecommerce Custom	customerSignUp	To sign up customer data along with password and address
Ecommerce Custom	CustomerProfile	to show customer profile data along with communication subscription

### 3. API Details

The process of Integration follows a workflow that will be detailed out in this section.

The Base URL for API Details in the APIs will be replaced by details shared by ER Integration team



### 3.1. Security

#### 3.1.1. Generate Security Token

**Method Name and Signature:** wsGenerateSecurityToken (Varchar strRequest)

**API URL (REST):** [{{BaseUrl}}/GenerateSecurityToken](#)

**Web Method Description:** Partners would be required to call the **wsGenerateSecurityToken** to generate Security Token. A Security token is unique to a program/instance and is used to identify a program/instance for transferring information with an authenticated end-point.

**Approach:** Purpose of generating Security token is to ensure the authentic communication for all the other methods. Generate the security token once and use the token in all other methods.

**Pre-Condition:** It is recommended that this method is called once with new session.

**Post-Condition:** None

**Input Parameters:** [Fields marked with (\*) are mandatory]

Field	Mandatory	Field Type	Field Length	Field Description
UserName	M	Varchar	64	This is a unique login id given to a vendor for using the web services which needs to be sent along with web service at the beginning of each new call to access Member Detail.
UserPassword	country	Varchar	32	User Password to be sent along with the Login Id
DevId	M	GUID	16	This will be a pre-generated unique code used only during development & shared with every vendor during Vendor Sign up to know who is calling the services & how many times has it been called from a given source.
AppId	M	GUID	16	This will be a pre-generated unique code used for production & shared with every vendor during Vendor Sign up so as to know who is calling the services & how many times has it been called from a given source
ProgramCode	M	Varchar	64	Loyalty Program Code. It is mandatory for Third Party API

### Token Generation Internal Logic

1. Based on UserName, ProgramCode, AppId, DevId and Internal GUID token is generated
2. Rijndael Algorithm is used to encrypt the Token
3. According to the time defined in CMS token expiry time is generated and once expired it will follow step1, once new request is raised.

### Return Parameters:

Parameter Name	Description	Data Type	Length
ReturnCode	Return Code indicating the status	Number	4
SecurityToken	Generated Security Token (in case of success)	Varchar	512
ReturnMessage	Return Message (in case of failure)	Varchar	512

### Error Codes:

Error Code	Description
0	Success
125	Input parameters are not provided correctly.
246	Failed in generating security token.
104	Do not have permissions to access this method.
248	Invalid user name or password.
249	Invalid appid.
250	Invalid devid.
251	Invalid vendor code supplied.
252	Invalid loyalty program.
319	Loyalty program does not exist.

### Sample:

Type	JSON
------	------

Request	<pre>{   "UserName": "IntegrationProd",   "UserPassword": "IntegrationProd123",   "DevId": "4b474ea9-e4ce-4fce-a7a1-74461dafed26",   "Appld": "828236dd-badd-4238-b0e8-a69ac653be1c",   "ProgramCode": "IntegrationProd" }</pre>
Response {Success}	<pre>{   "ReturnCode": 0,   "Token":   "7FUYCvpq0+gEFD56ivjg44il07dZWNQqAFZ2c/IIWsE7hwdhQz8XISdcDXniggq7iRTFRK   A+0Cs7D6lwgzRLsB2gTYWXOnZANijFZyE/eO6M24uoF0Wiyx13muxugJ61QNb6cAQIT   U/oU5KflR9w4YuXb3ongARO4THLRrf96IZa2Ugs7sVfhQfYZA27/Jq" }</pre>
Response {Error}	<pre>&lt;Response&gt;   &lt;ReturnCode&gt;248&lt;/ReturnCode&gt;   &lt;ReturnMessage&gt;Failed in generating security token. &lt;/ReturnMessage&gt; &lt;/Response&gt;</pre>

### Guidelines on Generation of Security Token and Its Usage

There are three approaches for generation of security token and its usage.

#### **1. Generating Security Token on every API call (Recommended)**

From security perspective, it is recommended that a partner generates a security token every time they call an API. The persistence time (in minutes) of a security token is configurable from Third party CMS application. The expiry time can be configured as maximum time understood to be taken for an API call to be executed without the risk of the security token getting expired in between.

#### **2. Generating Security Token once per day**

As an alternative to approach 1 above, when partner is not able generate security token for every API call due an unavoidable reason, it is advised to generate Security Token at least once per day.

To follow this approach, partner would be required to store the security token in their database/configuration file along with the timestamp when the token was generated. Before making an API call, partner would retrieve the security token from their database/configuration file and check if the token hasn't expired by comparing the current timestamp with the timestamp of token generation. If the

difference between the two timestamps comes greater than 1 day, then a new Security Token must be generated by the partner.

In case ReturnCode is 105 or 124, token has to be regenerated.

### 3. Getting Security Token from EasyRewardz

As a last resort, if the partner is unable to follow any of the above two approaches, then the partner must get the Security Token from EasyRewardz and save it in their database/configuration file.

## 3.2. Registration of Member

Customer Enrolment process is key to Loyalty Program Integration. The data captured in the Enrolment process provides brand an opportunity to communicate to customer during customer life cycle.

This section describes the APIs used for registering/enrolling a member to a Loyalty program.

### 3.2.1. IsCustomer

**Method Name and Signature:** - wslsCustomer (Varchar strRequest)

**API URL (REST):** - [{{BaseUrl}}/IsCustomer](#)

**Method Description:** - The purpose of this API is to fetch the details of the customer.

**Input Parameter:** -

Field	Mandatory	Field Type	Field Length	Field Description
SecurityToken	M	Varchar	512	This is a unique token Id used to authenticate the request.
EasyId	M	Varchar	32	Unique Identifier of Member
UserName	M	Varchar	64	Partner User Name
CountryCode	M	Varchar	64	Country Code

**Sample:**

Request	<pre>{   "SecurityToken":     "7FUYCvpq0+gEFD56ivjg44il07dZWNQQaFZ2c/IIWsE7hwdhQz8XISdcDXniggq7iRTFRKA+0Cs     7D6lwgzRLsB2gTYWXOnZANijFZyE/eO6M24uoF0Wiyx13muxugJ61QNb6cAQITU/oU5KfIR9     w4YuXb3ongARO4THLRrf96IZa2Ugs7svfhQfYZA27/Jq",   "EasyId": "9634098384",   "UserName": "vaibhav@easyrewardz.com",   "CountryCode": "91" }</pre>
Response	<pre>{   "ReturnCode": 0,   "ReturnMessage": "Success",   "FirstName": "Vaibhav",   "LastName": "", }</pre>

```

    "Email": "vaibhav@easyrewardz.com",
    "Mobile": "9634098384",
    "ClientID": "",
    "DateOfBirth": "",
    "AvailablePoints": 4545.0000,
    "MembershipCardNumber": "",
    "CurrentTier": "",
    "TotalVisits": 50,
    "TierDiscountType": "",
    "TierDiscount": 0.0,
    "CustomerType": "Loyalty",
    "ReferralCode": "FW3I3N9Q",
    "MobileCountryCode": "91",
    "TotalSpends": 1170360.0000
  }

```

### 3.2.2. Register Customer

**Method Name and Signature:** wsRegisterEasyAccount (Varchar strRequest)

**API URL (REST):** [{{BaseUrl}}/RegisterEasyAccount](#)

**Method Name and Description:** This method is required to register a member into the Loyalty program. Partners will be required to call this API and pass in the mandatory parameters to successfully enrol/register a member. The required fields for this method can be configured in LPaaS based on the required fields in POS/Portal.

**Approach:**

LPaaS supports following ways of enrolment process

- Mobile number
- Membership card number
- Mobile number and Membership card number

A Membership Number can either be a pre-generated series that a Partner shares with us, or it can be generated from LPaaS. These Membership numbers can then be used for enrolment.

**Pre-Condition:**

Following conditions should have been met before using this method for enrolling a member.

- Security Token must have been generated
- Partner outlet must have been configured in LPaaS (each outlet – POS store, web channel, mobile app etc. - must have a unique code configured in LPaaS)
- If Country Code is being passed, then it must already be configured in LPaaS.
- If Customer Type Code is being passed, then it must already be configured in LPaaS.

**Post-Condition:**

Member will be enrolled in LPaaS and will be able to accrue or redeem points based on his/her eligibility.

**Best Implementation Practice:**

- For a new customer and existing Customer API should push Demographic Data (First Name, Last Name, Gender, Email ID, DOB, DOA), The list of fields should be configurable and control should be there to make the fields mandatory and non-Mandatory
- Register API should push Country Code to support all international Customer. Country Code should be part of all APIs
- Register API should push Tier Information to support Paid Membership Program
- Register API should be able to Push Email and SMS Subscribe Information (opt In Information)
- Register API should be able to Support to push different Customer Type and Channel Information.
- Depending upon Customer Unique Identifier Mobile, Membership Number, Client ID POS Should support Registration of Customer. Customer Unique Identifier for a Brand should be configurable from POS Screen

#### Input Parameters:

Fields	Mandatory	Field Type	Field Length	Field Description
SecurityToken	M	Varchar	512	This is a unique token Id used to authenticate the request.
FirstName	O	Varchar	32	First Name of the member
LastName	O	Varchar	32	Last name of the member
DateOfBirth	O	DateTime		The date of birth of the member. It should be in (DD MMM YYYY) format
PinCode	O	Varchar	6	The address pin code of the member.
EmailId	O	Varchar	128	Email id of the member. Should be unique
MobileNo	M	Varchar	32	Enrolling Member's Mobile number
UserName	M	Varchar	64	Partner's User Name
Gender	O	Varchar	32	Gender of the Member (Male, Female, M or F)
MemberShipCardNumber	M/O	Varchar	64	Membership Card Number (if the program supports MemberShipCard)
ServicePersonNo	O	Varchar	64	Service Person Number
EmailSubscribe	O	Boolean	1	Flat to capture if customer has opted for Email notifications
SMSSubscribe	O	Boolean	1	Flat to capture if customer has opted for SMS notifications
TierExpiryDate	O	DateTime		Tier Expiry Date
StoreCode	M	Varchar	64	Store Code used to enroll member. Store Code must already be configured in LPaaS before calling this method.
Address1	O	Varchar	64	Address of the Member – 1
Address2	O	Varchar	64	Address of the Member – 2

<b>ReferralCode</b>	O	Varchar	64	Referral Code of the existing Member if the enrolling Member is being referred by an existing member. Referral points may be given to both the enrolling and referring members (based on Referral Offer configured in LPaaS).
<b>EasyId</b>	M	Varchar	32	Unique Identifier of Member
<b>TierName</b>	O	Varchar	64	Tier Name for Tier Based Program
<b>Remarks</b>	O	Varchar	512	Any other Remarks
<b>Anniversary Date</b>	O	DateTime		Anniversary Date
<b>CompanyName</b>	O	Varchar	64	Company Name
<b>PANNo</b>	O	Varchar	32	PAN Number
<b>ClientCustomerID</b>	M/O	Varchar	128	Unique identifier of member apart from mobile/membershipnumber
<b>EnrollDate</b>	O	DateTime		Enrollment Date
<b>CustomerTypeCode</b>	O	Varchar	32	It is the Customer Type of Member. Customer Type must be configured in LPaaS before passing in this field. Sample values for Customer Type code are Loyalty, NonLoyalty etc.
<b>ChannelCode</b>	M	Varchar	64	It is the Channel used for enrolling a Member. The channel must be configured in LPaaS before passing in this field. Sample values for channel are Online, Offline etc.
<b>AgeGroup</b>	O	Varchar	128	Age Group
<b>CountryCode</b>	M	Varchar	64	Country Code
<b>NumberOfChildren</b>	O	Number		Number of Children

**Note:**

1. The field "MemberShipCardNumber" is mandatory if Loyalty program supports membership card.
2. If member already exists in the system profile details are updated automatically when update existing member is enabled for a program.
3. If Tracking is enabled for the Program, then member is also enrolled in EasyRewardz and Program is added to member's dashboard in EasyRewardz.

**Return Parameters:**

Parameter Name	Description	Data Type	Length
ReturnCode	Return Code indicating the status	Number	4
MemberShipNumber	MemberShipCard Number	Varchar	16

BonusPoints	Bonus Points Earned (in case of Enroll Bonus)	Number	6
TPEnrollStatus	Enroll Status in Third Party (1→ success, 0 → failure)	Number	1
REnrollStatus	Enroll Status in EasyRewardz (1→ success, 0 → failure)	Number	1
ProgramStatus	Program Creation Status of Member Dashboard in EasyRewardz (1→ success, 0 → failure)	Number	1
ReturnMessage	Return Message	Varchar	512

**Error Codes:**

Error Code	Description
0	Success
125	Input parameters are not provided correctly.
247	Input parameter Membership card number missed.
186	Membership card number verification process failed.
187	Unable to generate membership card no. please try after some time.
229	Invalid Policy
200	User does not have permission under this store.
199	User does not exist.
222	Unauthenticated user.
223	Unauthenticated store.
224	Policy already claimed.
227	Your Policy has been deactivated.
225	Your Policy has expired.
228	Policy not started yet.
229	Invalid Policy
285	Offer does not exist.





198	Store code does not exist
315	Failed to update membership card details.
318	Failed to accrue points to member.
316	Failed to insert membership card details.
317	Membership card not exists.
261	Bill id already exists.
129	Cannot accrue as wallet does not have enough easyPoints.
264	Activity does not exist
262	User name does not exist.
263	Store does not exist.
299	Failed to register.
212	Member already exists.
124	Security token verification failed.
104	Do not have permissions to access this method.
105	Security token expired.
139	Cannot accrue, please try later
268	Invalid mobile number.
269	Invalid email id.
270	Invalid pincode.
271	Invalid date of birth.
273	Invalid lastname.
272	Invalid firstname.
274	Invalid membership card no.
275	Invalid issue card.
276	Invalid card value.

267	Easy pin should be between 6-12 letters and should include 1 number and 1 alpha Varcharacter.
207	Email id already exists.
139	Cannot accrue, please try later

**Sample Request and Response:**

Type	JSON
Request	<pre>{   "UserName": "vaibhav@easyrewardz.com",   "SecurityToken":     "7FUYCvpq0+gEFD56ivjg44il07dZWNQQaFZ2c/IIWsE7hwdhQz8XISdcDXniggq7     iRTFRKA+0Cs7D6lwgzRLsB2gTYWXOnZANijFZyE/eO6M24uoF0Wiyx13muxugJ     61QNb6cAQITU/oU5KfIR9w4YuXb3ongARO4THLRrf96IZa2Ugs7sVfhQfYZA27/J     q",   "FirstName": "Vikram",   "LastName": "Singh ",   "DateOfBirth": "22 Feb 1995",   "PinCode": "",   "EmailId": "test568@gmail.com",   "MobileNo": "9045372162",   "EasyPin": "",   "Gender": "",   "MembershipCardNumber": "",   "StoreCode": "DummyStore",   "AnniversaryDate": "",   "AssignMembershipCard": "0",   "CCIPolicyNo": "",   "Address1": "",   "Address2": "",   "EmailSubscribe": [     "",     ""   ] }</pre>



	<pre>], "SMSSubscribe": "", "EasyPinTypeId": "", "ReferralCode": "", "InvitedBy": "", "ProfileThumbnailImagePath": "", "EnrollDate": "", "CompanyName": "", "ClientCustomerID": "", "LoyaltyID": "", "PANNo": "", "DeviceID": "", "ChildGender": "", "ChildName": "", "ChildDOB": "", "ExpectedDateDelivery": "", "Twin": "", "Remarks": "", "ChannelCode": "Offline", "CustomerTypeCode": "Loyalty", "TierName": "", "TierExpiryDate": "" }</pre>
Response	<pre>{   "ReturnCode": "0",   "SmsStatus": "0",   "EmailStatus": "0",   "TPEnrollStatus": "1",   "EREnrollStatus": "0",   "ProgramStatus": "0",   "MemberShipNumber": null,</pre>

	<pre> "BonusPoints": "0", "CCIPoints": "0", "ReturnMessage": "Success.", "OTP": null, "EnrollStatus": "1", "ProfileStatus": "-1", &gt;LoginStatus": "0", "UpdateProfileFlagSet": "false" } </pre>
Return Code	<pre> &lt;Response&gt; &lt;ReturnCode&gt;212&lt;/ReturnCode&gt; &lt;ReturnMessage&gt;Member already exist. &lt;/ReturnMessage&gt; &lt;/Response&gt; </pre>

### 3.2.3. Update Member Profile

**Method Name and Signature:** wsUpdateMemberProfile (Varchar strRequest)

**API URL (REST):** [{{BaseUrl}}/UpdateMemberProfile](#)

**Method Description:** This method is used to update the member's profile.

**Approach:** There are scenarios where a customer is enrolled with minimum required details and later customer is encouraged to share his/her profile details. Also, a customer may choose to update his existing details in the program. In such cases, the enrolled member's details, except the primary identifier (mobile number or membership number), can be updated using this API.

**Best Implementation Practice:**


Based on Customer Unique Identifier Member profile update should be done. Unique Identifier can't be updated. Updating Mobile Number should be a process need to be defined with Integration Team.

**Pre-Condition:** Customer should be enrolled in the program.

**Post-Condition:** Customer details are updated in the program.

**Input Parameters:**

Field	Mandatory	Field Type	Field Length	Field Description
-------	-----------	------------	--------------	-------------------

<b>SecurityToken</b>	M	Varchar	512	This is a unique token Id used to authenticate the request.
<b>FirstName</b>	O	Varchar	128	First Name of the member
<b>LastName</b>	O	Varchar	128	Last name of the member
<b>DateOfBirth</b>	O	DateTime		The date of birth of the member. It should be in (DD MMM YYYY) format
<b>PinCode</b>	O	Varchar	6	The address pin code of the member.
<b>EmailId</b>	O	Varchar	128	Email id of the member. Should be unique
<b>MobileNo</b>	M	Varchar	32	Mobile no of the member
<b>UserName</b>	M	Varchar	64	Partner User Name
<b>Gender</b>	O	Varchar	32	Gender of the member (male, female, m or f)
<b>Address1</b>	O	Varchar	64	Address of the member
<b>Address2</b>	O	Varchar	64	Address of the member
<b>AnniversaryDate</b>	O	DateTime		AnniversaryDate of the member (DD MMM YYYY)
<b>StoreCode</b>	M	Varchar	64	Store Code
<b>ChildName</b>	O	Varchar	128	
<b>ChildGender</b>	O	Varchar	32	
<b>ChildDOB</b>	O	DateTime		
<b>SMSSubscribe</b>	O	Boolean	1	
<b>EmailSubscribe</b>	O	Boolean	1	
<b>ProfileThumbnailImagePath</b>	O	Varchar	256	
<b>CompanyName</b>	O	Varchar	64	
<b>PANNo</b>	O	Varchar	128	
<b>ClientCustomerID</b> 	M/O	Varchar	128	
<b>MemberShipCardNumber</b>	M/O	Varchar	64	
<b>ExpectedDateDelivery</b>	O	DateTime		
<b>Twin</b>	O	Varchar	6	

Remarks	O	Varchar	512	
ReferralCode	O	Varchar	64	
NumberOfChildren	O	Integer		
CustomerTypeCode	O	Varchar	64	
MigratedSpends	O	Decimal		
MigratedVisits	O	Long		
LastMigratedDate	O	DateTime		
CountryCode	M	Varchar	64	

**Return Parameters:**

Parameter Name	Description	Data Type	Length
ReturnCode	Return Code indicating the status	Number	4
ReturnMessage	Return Message	Varchar	512

**Sample:**

Type	JSON
Request	<pre>{   "SecurityToken":     "7FUYCvpq0+gEFD56ivjg44il07dZWNQqAFZ2c/IIWsE7hwdhQz8XISdcDXnigg     q7iRTFRKA+0Cs7D6lwgzRLsB2gTYWXOnZANijFZyE/eO6M24uoF0Wiyx13mu     xugJ61QNb6cAQITU/oU5KflR9w4YuXb3ongARO4THLRf96IZa2Ugs7sVfhQfY     ZA27/Jq",   "FirstName": "Tarak",   "LastName": "Mehta Me",   "DateOfBirth": "02 Mar 1994",   "MobileNo": "7865439765",   "Address1": "Test123",   "Address2": "Test321",   "PinCode": "",   "EmailId": "TestQ12@gmail.com",   "Gender": "M",   "StoreCode": "DummyStore",   "AnniversaryDate": "12 Jun 2016",   "UserName": "vaibhav@easyrewardz.com",   "ChannelCode": "Offline",   "CustomerTypeCode": "Loyalty",   "ReferralCode": "",   "NumberOfChildren": "" }</pre>

	<pre>"CountryCode": "91" }</pre>
Response - Success	<pre>{   "ReturnCode": "0",   "ReturnMessage": "Success.",   "ProfileUpdatedStatus": "1",   "CustomerTypeId": "1",   "UpdateProfileFlagSet": "true" }</pre>
Response - Error	<pre>&lt;Response&gt; &lt;ReturnCode&gt;124&lt;/ReturnCode&gt; &lt;ReturnMessage&gt;Security token verification failed. &lt;/ReturnMessage&gt; &lt;/Response&gt;</pre>

### 3.2.4. Search Member

**Method Name and Signature:** ~~wsSearchMember~~ (Varchar strRequest)

This method is deprecated method. IsCustomer API should be used instead of the API

**API URL (REST):** [{{BaseUrl}}/SearchMember](#)

**Web Method Description:** This method is required to search and retrieve member details. Partner will be required to call this API to know if a member already exists in the system or not.

**Approach:** The method accepts the Member ID (mobile or membership no) plus additionally first name, last name, membership no & date of birth. It returns all details of the member (including profile, transactions & points).

**Best Implementation Practice:**

1. POS should integrate the API to get the decision If member exists or not
2. If the member doesn't exist, POS will call Register API
3. POS should be able to search using below parameters
  - a. First Name
  - b. Last Name
  - c. Email ID
  - d. Mobile Number
  - e. Membership Number (for Card Based Loyalty Program)
  - f. Client ID (Custom Unique Identifier)

**Pre-Condition:** None

**Post-Condition:** None

**Input Parameters:**

Field	Mandatory	Field Type	Field Length	Field Description
<b>SecurityToken</b>	M	Varchar	512	This is a unique token Id used to authenticate the request.
<b>PageSize</b>	O	Integer		Number of records to be displayed in the page
<b>PageNumber</b>	O	Integer		Which page to be displayed
<b>SortExpression</b>	O	Varchar	64	Expression to sort the result
<b>SortDirection</b>	O	Varchar	64	Sort Order of the expression (desc or asc)
<b>EasyId</b>	M	Varchar	32	Unique Identifier of the Member
<b>MobileNo</b>	O	Varchar	32	Mobile Number
<b>CountryCode</b>	M	Varchar	64	It is the country code. To be used for support of international numbers.
<b>FirstName</b>	O	Varchar	128	First Name of the member
<b>LastName</b>	O	Varchar	128	Last Name of the Member

**Return Parameters:**

Returns Array of Member Response with following parameters.

Parameter Name	Description	Data Type	Length
ReturnCode	Return Code	Number	4
FirstName	First Name of the member	Varchar	50
LastName	Last name of the member	Varchar	50
Email	Email of the member	Varchar	128
Mobile	Mobile number of the member	Number	10
ClientID	Client ID is the unique identifier of the Member in Partner system	Varchar	64
DateOfBirth	DateOfBirth of the member	Varchar	11
RecordCount	Record count of members matching the search criteria	Number	4



Title	Title of the member	Varchar	30
AccrualPoints	Points available at member	Number	6
MembershipCardNumber	MembershipCardNumber of the member	Varchar	16
ReturnMessage	Return Message	Varchar	512
CurrentTier	Current Tier of the Member	Varchar	50
StoreCustomerID			
EndDate		Varchar	11
EnrollDate	Enroll Date in DD MMM YYYY Format	Varchar	11
TierEndDate	Tier End Date in DD MMM YYYY Format	Varchar	11
TierStartDate	Tier Start Date in DD MMM YYYY Format	Varchar	11
TotalPointsRedeemed	Total Points Redeemed till now		
TotalSpends	Total Spend till date		
Address1	Address of the Member – 1	Varchar	64
Address2	Address of the Member – 2	Varchar	64
CustomerType	Customer Type of the Member	Varchar	128
TotalVisits	Total Visit till date	Number	
TotalPointsAccrued	Total Points accrued for the Member	Number	
TotalPointsLapsed	Total Point Lapsed for the Member	Number	
ReferralPoints	Points earned via Referral Process	Number	
ReferredCount	Number of Referrals	Number	
RemainingReferrals	No of Referral Pending	Number	
ReferralCode	Referral Code of the Member	Number	
MobileCountryCode	Country Code	Varchar	3
PinCode	Pin Code	Number	6
Gender	Gender	Varchar	10

ChildName	Name of the Child	Varchar	
AnniversaryDate	Anniversary Date	Varchar	11
ChildDOB	Child's DOB	Varchar	11
MigratedVisits	Migrated Visit	Number	
MigratedSpends	Migrated Spends	Number	
LastMigrationDate	Last Migration Date	Varchar	11
LifeTimeATV	Lifetime Average Transaction Value	Number	

#### Error Codes:

Error Code	Description
0	Success
172	No results found.
104	Do not have permissions to access this method.
105	Security token expired.
124	Security token verification failed.
125	Input parameters are not provided correctly.
171	Enter at least one fields
271	Invalid date of birth.
268	Invalid mobile number.
274	Invalid membership card no.

#### Sample:

Type	JSON
Request	<pre>{   "EasyId": "9634098384",   "UserName": "vaibhav@easyrewardz.com",   "SecurityToken":     "7FUyCvpq0+gEFD56ivjg44il07dZWNQQaFZ2c/IIWsE7hwdhQz8XISdcDXnig"</pre>



	<pre>gq7iRTFRKA+0Cs7D6lwgzRLsB2gTYWXOnZANijFZyE/eO6M24uoF0Wiyx13 muxugJ61QNb6cAQITU/oU5KfIR9w4YuXb3ongARO4THLRrf96IZa2Ugs7sVf hQfYZA27/Jq",    "FirstName": "",   "LastName": "",   "DateOfBirth": "",   "MobileNo": "",   "PageSize": "10",   "PageNumber": "1",   "SortExpression": "LastUpdated",   "SortDirection": "desc",   "UTCOffset": "0",   "CultureName": "en-US",   "MembershipCardNumber": "" }</pre>
Response	<pre>{   "MemberResponse": {     "ReturnCode": "0",     "FirstName": "Vaibhav",     "LastName": null,     "Email": "vaibhav@easyrewardz.com",     "Mobile": "9634098384",     "ClientID": null,     "DateOfBirth": null,     "RecordCount": "1",     "Title": null,     "AccrualPoints": "29",     "MembershipCardNumber": null,     "ReturnMessage": "Success.",     "CurrentTier": null,     "StoreCustomerId": "472",     "EndDate": "0001-01-01T00:00:00",</pre>



```
"EnrollDate": "2020-01-06T17:54:44.13",  
"TierEndDate": "0001-01-01T00:00:00",  
"TierStartDate": "0001-01-01T00:00:00",  
"TotalPointsRedeemed": "0",  
"TotalSpends": "65700.00",  
"Address1": null,  
"Address2": null,  
"CustomerType": "Loyalty",  
"TotalVisits": "21",  
"TotalPointsAccrued": "29",  
"TotalPointsLapsed": "0",  
"ReferralPoints": "0",  
"ReferredCount": "0",  
"RemainingReferrals": "120",  
"ReferralCode": "FW3I3N9Q",  
"MobileCountryCode": "91",  
"PinCode": null,  
"Gender": null,  
"ChildName": null,  
"AnniversaryDate": null,  
"ChildDOB": null,  
"MigratedVisits": "0",  
"MigratedSpends": "0.00",  
"LastMigrationDate": null,  
"LifeTimeATV": "3128.57",  
"UAEResident": null,  
"Nationality": null,  
"EmirateResidence": null,  
"CompanyName": null,  
"PANNo": null  
}
```

	}
	<pre> &lt;MemberResponse&gt;   &lt;ReturnCode&gt;172&lt;/ReturnCode&gt;   &lt;ReturnMessage&gt;No results found.&lt;/ReturnMessage&gt; &lt;/ MemberResponse &gt; </pre>

### 3.2.5. Save Children Details

**Method Name and Signature:** SaveChildrenDetails (Varchar strRequest)

**API URL (REST):** [{{BaseUrl}}/SaveChildrenDetails](#)

**Method Name and Description:** This method is required save the children details for an enrolled member.

**Approach:** For programs where children details are required to be captured, this API can be used to add the children details.

**Pre-Condition:**

Following conditions should have been met before using this method for enrolling a member.

- Security Token must have been generated
- Member whose children's details are being added should be enrolled in the Loyalty program

**Post-Condition:**

Children details will be saved against the Loyalty member's ID.

**Input Parameters:**

Field	Mandatory	Field Type	Field Length	Field Description
ChildDOB	M	DateTime		Child's Date of Birth
ChildGender	M	Varchar	32	Gender of Children
ChildName	M	Varchar	128	Name of children
CountryCode	M	Varchar	64	Country code
CustomerMemberID	M	Varchar	32	Customer Member ID
NumberOfChildren	M	Integer		
SecurityToken	M	Varchar	512	Security Token Code
StoreCode	M	Varchar	64	Store Code
UserName	M	Varchar	64	Partner User Name

### 3.3. Accrual of Loyalty Points

#### 3.3.1. Accrue Loyalty Points to a Member's Account

**Method Name and Signature:** wsEasyPointsAccrualEOSS (Varchar strRequest)

**API URL (REST):** [{{BaseUrl}}/EasyPointsAccrualEOSS](#)

**Method Name and Description:** This method is required to accrue points to Member's account against the purchase made. Partner can pass the EOSS amount and Non EOSS Amount in this method based on the items on the order.

**Approach:** This method needs to be called after the invoice is generated and saved in POS. LPaaS will auto select the best offers for point accrual based on Store/Amount/Tier etc. and Member attributes. LPaaS will calculate the points for the transaction and updated the member's points total accordingly.

**Pre-Condition:** Following conditions should have been met before using this method for enrolling a member.

- Security Token must have been generated
- Partner outlet must have been configured in LPaaS (each outlet – POS store, web channel, mobile app etc. - must have a unique code configured in LPaaS)
- The transaction needs to be complete at the POS/Transaction system with payment posted for the transaction, so that the Transaction ID generated is permanent.

**Post-Condition:** Members accrue points for the transaction based on the offer applied. Depending on the program structure, the member can get upgraded to a higher tier basis the transaction amount accrued. Also, any bonus offers, or coupons linked to this transaction offer is also triggered.

**Input Parameters:**

Field	Mandatory	Field Type	Field Length	Field Description
EasyId	M	Varchar	32	EasyId of the member
SecurityToken	M	Varchar	512	This is a unique Token Id used to authenticate the request.
StoreCode	M	Varchar	64	Store Code used for Transaction. Store Code must already be configured in LPaaS before calling this method.
TransactionCode	M	Varchar	64	
Amount	M	Decimal		Transaction Amount is mandatory for Transaction Based Offer. Amount must be equal to EOSS Amount + NONEOSSAmount
TransactionDate	M	DateTime		Date of Transaction in DD MMM YYYY format
ActivityCode	M/O	Varchar	64	Offer Code. This code will define point accrual rule. This needs to be a configurable parameter.

<b>TransactionDescription</b>	O	Varchar	128	Transaction details of the purchase made.
<b>UserName</b>	M	Varchar	64	Partner's User Name
<b>ActivityName</b>	O	Varchar	64	
<b>Quantity</b>	O	Float		
<b>EossAmount</b>	M	Decimal		This is EOSS (End of Season Sale) Amount
<b>NONEOSSAmount</b>	M	Decimal		This is Non EOSS (End of Season Sale) Amount
<b>BillTime</b>	O	TimeSpan		
<b>CustomerTypeCode</b>	O	Varchar	64	It is the Customer Type of Member. Customer Type must be configured in LPaaS before passing in this field. Sample values for Customer Type code are Loyalty, NonLoyalty etc.
<b>ChannelCode</b>	O	Varchar	64	It is the Channel used for transaction. The channel must be configured in LPaaS before passing in this field. Sample values for channel are Online, Offline etc.
<b>ClientCustomerID</b>	M/O	Varchar	128	<b>Client Customer ID</b>
<b>MemberShipCardNumber</b>	M/O	Varchar	64	<b>MemberShip Card Number</b>
<b>CountryCode</b>	M	Varchar	64	It is the country code. It is used to support of international numbers. Country Code must be configured in LPaaS before being passed into this field.

**Note:**

- If the EasyId provided in this method does not exist and it is mobile number, then first the Registration will be done & then accrual will happen for the member.
- **EasyId in all methods is either Membership Card Number if program supports Membership or it is Mobile Number.**
- **EOSS and NONEOSS Amount calculation** - The Sum of EOSSAmount & NONEOSSAmount needs to total to Amount. If in a transaction there are discounted items, then their Net Payable will be reflected in EOSSAmount & the rest of the non-discounted items' Net Payable will be reflected in NONEOSSAmount. The calculation of EOSS and NONEOSS amount will be done at POS end and will be sent in the API by POS.  
For e.g. If Bill value = INR2000, price of discounted items = INR1500, and price of non-discounted items = INR500, then  
Amount will be INR2000,  
EOSSAmount will be INR1500, and  
NONEOSSAmount will be INR500
- **Tax Calculation** - It depends solely on a Brand if it wants to enable accruals on a bill value including (or excluding) the tax component. If the accruals are to be done including tax amount in total bill value, then POS would send bill amount including tax in Amount field. If tax is not to be considered for Accrual, then POS will

have to remove the tax component from the bill and send the remaining amount in Amount field. POS would again be required to break the Amount into EOSS and NONEOSS Amount.

For e.g. If Bill Value = INR1000, Tax = INR100, then

If accruals to be done including tax, the Amount will be INR1000

If accruals to be done excluding tax, the Amount will be INR900

#### Return Parameters:

Parameter Name	Description	Data Type	Length
ReturnCode	ReturnCode indicating the status	Number	4
AccrualId	AccrualId for reference	Varchar	30
PointsEarned	Points Earned by the member	Number	6
SMSStatus			
EmailStatus			
BonusPoints	Bonus Points Earned by the member	Number	6
CCIPoints			
RedemptionID	Redemption Id for reference		
TPEnrollStatus	Enroll Status in Third Party (1→ success, 0 → failure)	Number	1
REnrollStatus	Enroll Status in EasyRewardz (1→ success, 0 → failure)	Number	1
ProgramStatus	Program Creation Status of Member Dashboard in EasyRewardz (1→ success, 0 → failure)	Number	1
EOSSPoints	Points Earned on EOSS Amount	Number	6
NOEOSSPoints	Points Earned on NON EOSS Amount	Number	3
ReturnMessage	Return Message	Varchar	512

#### Error Codes:

Error Code	Description
0	Success
124	Security token verification failed.
104	Do not have permissions to access this method.



256	Transaction date should be less than or equal to current date.
125	Input parameters are not provided correctly.
261	Bill id already exists.
253	Member Id does not exist.
301	Points are mandatory for activity-based accrual.
129	Cannot accrue as wallet does not have enough easyPoints.
170	Offer does not exist in this range.
302	Cannot accrue easypoints as the amount is less.
264	Activity does not exist.
262	User name does not exist.
263	Store does not exist.
285	Offer doesnot exists.
254	Invalid easyId.
255	Invalid date.
320	Invalid Price
321	Invalid points

**Sample:**

Type	JSON
Request	<pre>{   "EasyId": "9634098384",   "SecurityToken":     "7FUYCvpq0+gEFD56ivjg44il07dZWNQqQaFZ2c/IIWsE7hwdhQz8XISdcDXniggq7iR     TFRKA+0Cs7D6lwgzRLsB2gTYWXOnZANijFZyE/eO6M24uoF0Wiyx13muxugJ61Q     Nb6cAQITU/oU5KfIR9w4YuXb3ongARO4THLRf96lZa2Ugs7sVfhQfYZA27/Jq",   "StoreCode": "DummyStore",   "TransactionCode": "Bill55",   "Amount": "1000",</pre>



	<pre>"TransactionDate": "11 Aug 2020", "ActivityCode": "DummyAccuralAX", "TransactionDescription": "Partner+Transaction+-+AX+-+DemoA+-+%5b%23Description%23%5d", "UserName": "vaibhav@easyrewardz.com", "EasyPoints": "0", "Activities": "", "EOSSAmount": "0", "NONEOSSAmount": "1000.00", "BillCount": "1", "BillCounter": "1", "TotalAmount": "0", "TotalPoints": "0", "BillTime": "", "CountryCode": "91", "MemberShipCardNumber": "" }</pre>
Response	<pre>{ "ReturnCode": "0", "AccrualId": "10701", "PointsEarned": "100", "SmsStatus": "0", "EmailStatus": "0", "TPEnrollStatus": "0", "EREnrollStatus": "0", "ProgramStatus": "0", "BonusPoints": "0", "CCIPoints": "0", "ReturnMessage": "Success.", "RedemptionId": "10701", "EOSSPoints": "0",</pre>

```
"NonEOSSPoints": "100",
"OfferModelId": "2"
}
```

### 3.3.2. Flat Accrual to Member's Account

**Method Name and Signature:** wsFlatEasyPointsAccrual (Varchar strRequest)

**API URL (REST):** [{{BaseUrl}}/FlatEasyPointsAccrual](#)

**Method Name and Description:** This method is used to accrue points to a member's account. This method differs from wsEasyPointsAccrualEOSS in a way that this method accrues points directly to a member's account without any validations.

**Approach:** This method needs to be called when a member (or members) need to be accrued points overriding any validations for accrual.

**Pre-Condition:**

**Post-Condition:**

**Input Parameters:**

Field	Mandatory	Field Type	Field Length	Field Description
SecurityToken	M	Varchar	512	This is a unique token Id used to authenticate the request.
UserName	M	Varchar	64	Partner's User Name
StoreCode	M	Varchar	64	Store Code used for Transaction. Store Code must already be configured in LPaaS before calling this method.
EasyId	M	Varchar	32	Easy ID of the member (This could be mobile number or the Membership Number)
MobileNo	O	Varchar	32	Mobile Number
EmailId	O	Varchar	128	Email Id
TransactionCode	O	Varchar	64	Transaction Code
TransactionDate	O	DateTime		Transaction Date
EasyPoints	O	Decimal	32	Points to be accrued to a Member's account
LapseEndDate	O	DateTime		Lapsation Date
TransactionDescription	O	Varchar	128	Transaction details of the purchase made.
FirstName	O	Varchar	128	First
LastName	O	Varchar	128	

DateOfBirth	O	DateTime		
PinCode	O	Varchar	6	
Gender	O	Varchar	32	
EligibleForTier	O	Boolean	1	
Amount	O	Decimal		Amount of purchase made by member
ClientCustomerID	M/O	Varchar	128	
LoyaltyID	O	Varchar	64	
CustomerTypeCode	O	Varchar	64	
CountryCode	M	Varchar	64	

#### Return Parameters:

Parameter Name	Description	Data Type	Length
ReturnCode	ReturnCode indicating the status	Number	4
AccrualId	AccrualId for reference	Varchar	50
PointsEarned	Points Earned by the member	Number	6
SmsStatus			
EmailStatus			
TPEnrollStatus	Enroll Status in Third Party (1→ success, 0 → failure)	Number	1
REnrollStatus	Enroll Status in EasyRewardz (1→ success, 0 → failure)	Number	1
ProgramStatus	Program Creation Status of Member Dashboard in EasyRewardz (1→ success, 0 → failure)	Number	1
BonusPoints	Bonus Points Earned by the member	Number	6
ReturnMessage	Return Message	Varchar	512
RedemptionId	Redemption Id for reference	Number	
EOSSPoints	Points Earned on EOSS Amount	Number	6
NonEOSSPoints	Points Earned on NON EOSS Amount	Number	3

### 3.4. Redemption of Loyalty Points

This section describes the APIs used to redeem a Loyalty point from a Member's account.

#### 3.4.1. Get Available Points of a Member

**Method Name and Signature:** wsCustomerAvailablePoints (Varchar strRequest)

**REST URL:** [{{BaseUrl}}/CustomerAvailablePoints](#)

**Method Description:** This method is required to retrieve the available points of a member.

**Approach:** This method is integrated in the transaction system to display the available points of a member. It helps the store staff to inform the member about their current balance and to confirm the correct points in the redemption request.

**Best Implementation Practice:**

- Point Rate for the Program will be coming in the response of the API. POS must read that information
- POS should be able support Multiple Wallet Point Balance. Wallets are differentiated using Activity Code parameter of the API

**Pre-Condition:** Member needs to be registered in LPaaS to get his/her available balance.

**Post-Condition:** The available points for a Member are retrieved.

**Input Parameters:**

Field	Mandatory	Field Type	Field Length	Field Description
ActivityCode	M	Varchar	64	Offer Code. (It will be used in multi wallet programs, to check points per wallet), If wrong activity code is passed, Balance across wallet will be displayed
StoreCode	M	Varchar	64	Store Code used for Transaction. Store Code must already be configured in LPaaS before calling this method.
EasyId	M	Varchar	32	EasyId of the member
SecurityToken	M	Varchar	512	This is a unique token Id used to authenticate the request.
BillAmount	M/O	Decimal		Transaction Amount is mandatory for Transaction Based Offer. Amount must be equal to EOSS Amount + NONEOSSAmount
CountryCode	M	Varchar	64	It is the country code. It is required to support international numbers. Country code needs to be configured in LPaaS before passing in this field.

**Return Parameters:**

Parameter Name	Description	Data Type	Length
ReturnCode	Return Code indicating the status of request	Number	4



AvailablePoints	Points available to a Member	Number	
ReferralPoints	Points earned via Referral	Number	
ReturnMessage	Return Message	Varchar	512
PartnerName	Name of the Partner	Varchar	128
EasyId	EasyId of the Member	Varchar	16
PointValue	Cash worth of points	Number	
PointRate	Rate which specifies the value of point in monetary terms. E.g. 1 point = 0.5 INR	Number	

**Error Codes:**

Error Code	Description
0	Success
253	Member Id does not exist.
124	Security token verification failed.
104	Do not have permissions to access this method.
105	Security token expired.
268	Invalid mobile number.
125	Input parameters are not provided correctly.

**Sample:**

Type	JSON
Request	<pre>{     "EasyId": "9634098384",     "SecurityToken":     "7FUyCvpq0+gEFD56ivjg44il07dZWNQqAFZ2c/IIWsE7hwdhQz8XISdcDXnig     gq7iRTFRKA+0Cs7D6lwgzRLsB2gTYWXOnZANijFZyE/eO6M24uoF0Wiyx13     muxugJ61QNb6cAQITU/oU5KfIR9w4YuXb3ongARO4THLRf96IZa2Ugs7sVf     hQfYZA27/Jq",     "ActivityCode": "",     "StoreCode": "DummyStore",</pre>

	<pre>"CountryCode": "91" }</pre>
Response	<pre>{   "ReturnCode": "0",   "AvailablePoints": "529",   "ReferralPoints": "0",   "ReturnMessage": "Success.",   "PartnerName": "INTEGRATION",   "EasyId": null,   "PointValue": "264.50",   "PointRate": "0.50" }</pre>
Return Message	<pre>&lt;Response&gt;   &lt;ReturnCode&gt;253&lt;/ReturnCode&gt;   &lt;ReturnMessage&gt;Member ID does not   exist.&lt;/ReturnMessage&gt; &lt;/Response&gt;</pre>

### 3.4.2. Check Availability for Easy Points Redemption

**Method Name and Signature:** wsCheckForEasyPointsRedemption (Varchar strRequest)

**API URL (REST):** [{{BaseUrl}}/CheckForEasyPointsRedemption](#)

**Method Description:** This method blocks the Loyalty points that are requested to be redeemed.

A user or store staff can request for specific points to be redeemed or pass 0, & bill value. If specific points are passed & member has same or more points & this is less than bill value, then the same is blocked. If 0 is passed, then all of member's points that are equal to the bill value or up-to the available points in the member's account if the same is less than bill value, is blocked.

**Approach:** Once this method is called and it is successful, points are blocked for a specified time (Blocking time is configurable, e.g. 20 minutes). This method is called before the save of bill. In some cases, POS or other system does not have actual bill (bill no. is generated post save of bill), in those cases temporary bill no. is passed, and temporary bill is updated to actual bill no. id the Confirm redemption method.

Irrespective of whether loyalty points are integrated in the POS/transaction system as a discount or a payment mode, this method is always called prior to bill save. The OTP generated during this call is triggered to the member's mobile no. registered during enrollment.

**Best Implementation Practice:**

1. Activity Code in this API will define redemption rule of the program.
2. This API just blocks the Point and ConfirmEasyPoint API deducts the point

**Pre-Condition:** Member/customer needs to know their available balance before they can decide on how many points they want to redeem. Search Member or Get available points method can be used to get Member available points.

**Post-Condition:** ConfirmOTP method is called with the OTP to continue with redemption Process. If Member decides not to redeem points, then Release redemption points should be called.

**Input Parameters:**

Field	Mandatory	Field Type	Field Length	Field Description
EasyId	M	Varchar	32	EasyId of the member
SecurityToken	M	Varchar	512	This is a unique token Id used to authenticate the request.
StoreCode	M	Varchar	64	Store Code used for Transaction. Store Code must already be configured in LPaaS before calling this method.
TransactionCode	M	Varchar	64	Transaction Code (Bill Number)
RedemptionDate	M	DateTime		RedemptionDate (DD MMM YYYY)
Amount	M	Decimal		This will be the purchase amount made by the member (in case of TransactionOffer).
RedemptionType	M	Varchar	32	This will indicate the type of the redemption using redemption type code mentioned as below: AIR- AR, HOTEL - HT, MOVIE- MV, BUS, BS, ANY TRANSACTIONAL - TL, PRODUCT – PD
EasyPoints	M	Decimal	32	Points to be redeemed (in case of Activity Based Offer)
ActivityCode	M	Varchar	64	Offer Code
Activities	O			This will be the selected activities by the member if the offer is ActivityBased, used to store the selected activities by the member.
TransactionDescription	O	Varchar	128	This will be used to save the transaction details of the redemption
UserName	M	Varchar	64	Partner User Name



<b>CountryCode</b>	M	Varchar	64	It is the country code. To be used for support of international numbers.
--------------------	---	---------	----	--------------------------------------------------------------------------

#### Return Parameters:

Parameter Name	Description	Data Type	Length
ReturnCode	ReturnCode indicating the status	Number	4
EasyPoints	Points available at member	Number	
CashWorthPoints	Amount equivalent to member points	Number	
BalanceToPay	Balance Amount need to pay by the member	Number	
BillAmountPoints	Total equivalent points to the bill amount	Number	
IsPointsPlusCash	Is Points Plus Cash enabled (true/false) for the program	Varchar	6
IsRedeemWithOutOTP	This flag informs if redemption without OTP is allowed ( If 1 = Without OTP redemption will happen , For 0 Redemption will happen with OTP	Number	
ReturnMessage	Return Message	Varchar	512
BillNo	Bill Number	Varchar	64

#### Error Codes:

Error Code	Description
0	Success
125	Input parameters are not provided correctly.
168	Invalid offer.
174	Member does not have enough points.
257	Points are mandatory for activity-based redemption.
175	Requested points less than bill amount points
168	Invalid offer.

169	Amount should be greater than transaction unit
170	Offer does not exist in this range.
258	Amount should be greater than zero.
259	Cannot redeem easyPoints as the amount is less.
260	Redemption is already done for this bill id.
261	Bill id already exists.
262	User name does not exist.
263	Store does not exist.
264	Activity does not exist.
253	Member ID does not exist.
173	Cannot redeem as customer does not have enough easyPoints.
124	Security token verification failed.
104	Do not have permissions to access this method.
105	Security token expired.
265	Invalid customer id.
255	Invalid date.
266	Redemption date should be current date.
320	Invalid Price.
321	Invalid points.

**Sample:**

Type	JSON
Request	<pre>{   "EasyId": "9634098384",   "SecurityToken":     "7FUYCvpq0+gEFD56ivjg44il07dZWNQqAFZ2c/IIWsE7hwdhQz8XISdcDXni     ggq7iRTFRKA+0Cs7D6lwgzRLsB2gTYWXOnZANijFZyE/eO6M24uoF0Wiyx1"</pre>



	<pre>3muxugJ61QNb6cAQITU/oU5KfIR9w4YuXb3ongARO4THLRf96lZa2Ugs7s VfhQfYZA27/Jq",      "StoreCode": "DummyStore",     "TransactionCode": "Old1234567",     "RedemptionDate": "01 Jul 2020",     "Amount": "1000",     "RedemptionType": "PD",     "EasyPoints": "0",     "ActivityCode": "DummyRedemption",     "TransactionDescription": "",     "UserName": "vaibhav@easyrewardz.com",     "Activities": "",     "CountryCode": "91"  }</pre>
Response - Success	<pre>{     "ReturnCode": "0",     "EasyPoints": "529",     "CashWorthPoints": "529",     "BalanceToPay": "471",     "BillAmountPoints": "1000",     "IsPointsPlusCash": "true",     "ReturnMessage": "Success.",     "Vouchers": null,     "BillNo": "Old1234567",     "IsRedeemWithOutOTP": "false" }</pre>
Response - Error	<pre>&lt;Response&gt;     &lt;ReturnCode&gt;263&lt;/ReturnCode&gt;     &lt;ReturnMessage&gt;Store does not exist.&lt;/ReturnMessage&gt; &lt;/Response&gt;</pre>

### 3.4.3. ConfirmOTP

**Method Name and Signature:** wsConfirmOTP (Varchar strRequest)

**API URL (REST):** [{{BaseUrl}}/ConfirmOTP](#)

**Method Description:** This method is used to validate the OTP that is triggered to the member's registered mobile from the invocation of the **wsCheckForEasyPointsRedemption** method.

**Approach:** The OTP shared by the member redeeming his/her points for the transaction is validated by this method. If it successful, then the discount or payment due to loyalty points is applied to the transaction.

**Best Implementation Practice:**

1. Resend OTP is must to be implemented along with the API

**Pre-Condition:** This method is called after the **wsCheckForEasyPointsRedemption** method to continue with redemption process.

**Post-Condition:** The bill is saved applying the discount or payment with loyalty points, basis the mode in which the integration with transaction system/POS has happened.

**Input Parameters:**

Field	Mandatory	Field Type	Field Length	Field Description
EasyId	M	Varchar	32	EasyId of the member
SecurityToken	M	Varchar	512	This is a unique token Id used to authenticate the request.
SmsCode	M	Varchar	64	Sms code (OTP)
StoreCode	M	Varchar	64	Store Code used for Transaction. Store Code must already be configured in LPaaS before calling this method.
TransactionCode	M	Varchar	64	Transaction Code (Bill Number)
TransactionDate	O	DateTime		Transaction Date (DD MMM YYYY) If Transaction Date is not provided considered as todays date
UserName	M	Varchar	64	Partner User Name
CountryCode	M	Varchar	64	It is the country code. To be used for support of international numbers.

**Return Parameters:**

Parameter Name	Description	Data Type	Length
----------------	-------------	-----------	--------

ReturnCode	Return Code	Number	4
ReturnMessage	Return Message	Varchar	512

#### Sample:

Type	JSON
Request	<pre>{   "EasyId": "9634098384",   "SecurityToken":     "7FUyCvpq0+gEFD56ivjg44il07dZWNQqQaFZ2c/IIWsE7hwdhQz8XISdcDXniggq7     iRTFRKA+0Cs7D6lwgzRLsB2gTYWXOnZANijFZyE/eO6M24uoF0Wiyx13muxugJ     61QNb6cAQITU/oU5KfIR9w4YuXb3ongARO4THLRf96IZa2Ugs7sVfhQfYZA27/J     q",   "StoreCode": "DummyStore",   "TransactionCode": "Old003",   "SmsCode": "287148",   "UserName": "vaibhav@easyrewardz.com",   "TransactionDate": "02 Jul 2020",   "CountryCode": "91" }</pre>
Response - Success	<pre>{   "ReturnCode": "0",   "ReturnMessage": "Success." }</pre>
Response - Error	<pre>&lt;Response&gt;   &lt;ReturnCode&gt;365&lt;/ReturnCode&gt;   &lt;ReturnMessage&gt;Invalid OTP.&lt;/ReturnMessage&gt; &lt;/Response&gt;</pre>

#### 3.4.4. Resend OTP

**Method Name and Signature:** wsResendOTP (Varchar strRequest)

**API URL (REST):** [{{BaseUrl}}/ResendOTP](#)

**Method Description:** This method resends the same OTP which was triggered from CheckforeasyPointRedemption.

**Approach:** During a redemption process an OTP is triggered during **wsCheckForEasyPointsRedemption** method. The validation of this OTP is mandatory to complete the redemption. If the OTP is not delivered to the member, then this method is used to trigger a new OTP to the member.

**Pre-Condition:** The redemption is already initiated & **wsCheckForEasyPointsRedemption** is already invoked.

**Post-Condition:** OTP delivered to the member's account is confirmed in the ConfirmOTP method.

#### Input Parameters:

Parameter Name	Description	Data Type	Length
SecurityToken (*)	This is a unique token Id used to authenticate the request.	Varchar	512
StoreCode (*)	Store Code	Varchar	50
UserName (*)	Partner User Name	Varchar	64
TransactionCode (*)	Transaction Code/Bill No for which OTP needs to be sent again	Varchar	64
BillDate (*)	Bill Date (DD MMM YYYY)	Varchar	11
CountryCode	It is the country code. To be used for support of international numbers.	Varchar	3

#### Return Parameters:

Parameter Name	Description	Data Type	Length
ReturnCode	Return Code	Number	11
ReturnMessage	Return Message	Varchar	512

#### Error Codes:

Error Code	Description
0	Success

#### Sample:

Type	JSON
Request	{ "EasyId": "9634098384",

	<pre> "SecurityToken": "7FUyCvpq0+gEFD56ivjg44il07dZWNQqafZ2c/IIWsE7hwdhQz8XISdcDXnig gq7iRTFRKA+0Cs7D6lwgzRLsB2gTYWXOnZANijFZyE/eO6M24uoF0Wiyx13 muxugJ61QNb6cAQITU/oU5KfIR9w4YuXb3ongARO4THLRf96IZa2Ugs7sVf hQfYZA27/Jq",  "StoreCode": "DummyStore",  "TransactionCode": "Old003",  "SmsCode": "287148",  "UserName": "vaibhav@easyrewardz.com",  "TransactionDate": "02 Jul 2020",  "CountryCode": "91"  } </pre>
Response – Success	<pre> {    "ReturnCode": "0",    "ReturnMessage": "Success."  } </pre>

### 3.4.5. Release Redemption Points

**Method Name and Signature:** wsReleaseRedemptionPoints (Varchar strRequest)

**API URL (REST):** [{{BaseUrl}}/ReleaseRedemptionPoints](#)

**Method Description:** This method is used to release the points blocked during the wsCheckForEasyPointsRedemption method.

**Approach:** There are certain scenarios where a transaction which has points applied (mostly in case of discount) goes thru changes post application of loyalty points. This change can be addition of an item, removal of item or void bill. It is advisable that the points are released at such stage.

#### Best Implementation Practice:

1. This is an important API to call in all events where Point redemption could not be completed due to any reason for example Bill could not be saved, Item Addition/deletion

**Pre-Condition:** This method can only be called after wsCheckForEasyPointsRedemption method.

**Post-Condition:** Points that are blocked for the transaction is released. It is important to also clear the discount applied due to points, from the transaction.

#### Input Parameters:

Field	Mandatory	Field Type	Field Length	Field Description
<b>EasyId</b>	M	Varchar	32	EasyId of the member
<b>SecurityToken</b>	M	Varchar	512	This is a unique token Id used to authenticate the request.
<b>UserName</b>	M	Varchar	64	Partner User Name
<b>StoreCode</b>	M	Varchar	64	Store Code used for Transaction. Store Code must already be configured in LPaaS before calling this method.
<b>TransactionCode</b>	M	Varchar	64	Transaction Code (Bill Number)
<b>TransactionDate</b>	M	DateTime		Transaction Date (DD MMM YYYY)
<b>CountryCode</b>	M	Varchar	64	It is the country code. To be used for support of international numbers.

**Return Parameters:**

Parameter Name	Description	Data Type	Length
ReturnCode	Return Code	Number	4
ReturnMessage	Return Message	Varchar	512

**Sample:**

Type	JSON
Request	<pre>{   "EasyId": "9634098384",   "SecurityToken":     "7FUYCvpq0+gEFD56ivjg44il07dZWNQQaFZ2c/IIWsE7hwdhQz8XISdcDXnigg     q7iRTFRKA+0Cs7D6lwgzRLsB2gTYWXOnZANijFZyE/eO6M24uoF0Wiyx13mu     xugJ61QNb6cAQITU/oU5KfIR9w4YuXb3ongARO4THLRrf96iZa2Ugs7sVfhQfY     ZA27/Jq",   "StoreCode": "DummyStore",   "TransactionCode": "Old007",   "UserName": "vaibhav@easyrewardz.com",   "TransactionDate": "01 Jul 2020",   "CountryCode": "" }</pre>



	}
Response – Success	<pre>{   "ReturnCode": "0",   "ReturnMessage": "Success." }</pre>
Response – Error	<pre>&lt;Response&gt;   &lt;ReturnCode&gt;370&lt;/ReturnCode&gt;   &lt;ReturnMessage&gt;Transaction code does not   exist&lt;/ReturnMessage&gt; &lt;/Response&gt;</pre>

### 3.4.6. Confirm Redemption against Code Received

**Method Name and Signature:** wsConfirmEasyPointsRedemption (Varchar strRequest)

**API URL (REST):** [{{BaseUrl}}/ConfirmEasyPointsRedemption](#)

**Method Description:** This method confirms the points that are being redeemed for a transaction. Similar to AUTH & CAPTURE mechanism of payment processing, while wsCheckForEasyPointsRedemption AUTHorises a transaction, wsConfirmEasyPointsRedemption CAPTUREs a transaction.

**Approach:** Post Invoice generation (Bill save), this method confirms the redemption process. The same method also accepts the Permanent Bill ID that is generated for the transaction & updates the Bill id that was passed during wsCheckForEasyPointsRedemption. It also accepts the Paid amount of bill post applying the loyalty discount of payment via points. The Paid amount such passed into this method in three tags, excludes Amount paid by Points & any excluded tender (like Credit Note).

**Best Implementation Practice:**

1. Net Amount, EOSS Amount and Non EOSS Amount should be passed 0 here

**Pre-Condition:** The points should have been blocked & discount applied thru points or amount paid via points (if loyalty points is integrated as a tender & not discount). Bill should be saved or confirmed.

**Post-Condition:** Points will be deducted from members account after this method. If Accrual is set to TRUE for transactions where balance is paid in other modes, then member accrues points for these transactions. The member will accrue points on the amount that is left after the redemption has been applied.

**Input Parameters:**

Field	Mandatory	Field Type	Field Length	Field Description
-------	-----------	------------	--------------	-------------------

<b>EasyId</b>	M	Varchar	32	EasyId of the member
<b>SecurityToken</b>	M	Varchar	512	This is a unique token Id used to authenticate the request.
<b>TransactionCode</b>	M	Varchar	64	Transaction Code (Bill Number)
<b>RedemptionCode</b>	M	Varchar	64	This will be OTP generated by CheckForEasyPointsRedemption API and received on customer's mobile. OTP will be passed in this field
<b>UserName</b>	M	Varchar	64	Partner User Name
<b>EOSSAmount</b>	M	Decimal		Total EOSS Amount on the Bill
<b>NONEOSSAmount</b>	M	Decimal		Total NON EOSS Amount on the Bill
<b>NetAmount</b>	M	Decimal		Total Amount of Bill paid by Customer (amount eligible for points accrual). Net Amount should be equal to EOSSAmount + NONEOSS Amount)
<b>NewTransactionCode</b>	M	Varchar	64	If temporary bill Id is used in wsCheckForEasyPointsRedemption method, pass it in the field "TransactionCode" & the permanent Bill Id in this field. System will update the temporary Bill with permanent Bill. (If permanent bill is used in wsCheckForEasyPointsRedemption, pass this tag as blank)
<b>ChannelCode</b>	O	Varchar	64	Channel from where request is generated
<b>CountryCode</b>	M	Varchar	64	It is the country code. To be used for support of international numbers.

**Notes:**

- A member will accrue points on a redemption transaction if the settings for "Allow accruals while redemption" has been set as true. In this case, the member will accrue points on the amount that he/she is required to pay post the redemption has been applied. For e.g. if a customer has a total bill value of INR1000, and he redeems points worth of INR200, then customer pays an amount of INR800. These INR800 will then be eligible for accrual.
- Tax Calculation** - It depends solely on a Brand if it wants to enable accruals on a bill value including (or excluding) the tax component. As discussed in above point, if the flag for "Allow Redemption while accrual" is set as true, then the transaction becomes eligible for accrual. In this case, if the accruals are to be done including tax amount in net bill value, then POS would send bill amount including tax in NetAmount field. If tax is not to be considered for Accrual, then POS will have to remove the tax component from the bill and

send the remaining amount in NetAmount field. For e.g. If Bill Value = INR1000, Tax = INR100, Tax component is INR100, and customer redeems points worth INR200, then NetAmount becomes INR800. If accruals are to be done including tax component, then POS should send NetAmount as INR800; else if accruals are to be done excluding tax component, then POS should send NetAmount as INR700

#### Return Parameters:

Parameter Name	Description	Data Type	Length
ReturnCode	ReturnCode indicating the status	Number	4
PointsEarned	Points earned	Number	
SMSStatus	SMS Status		
EmailStatus	Email Status		
TPEnrollStatus	Enroll Status in Third Party (1→ success, 0 → failure)	Number	1
REnrollStatus	Enroll Status in EasyRewardz (1→ success, 0 → failure)	Number	1
ProgramStatus	Program Creation Status of Member Dashboard in EasyRewardz (1→ success, 0 → failure)	Number	1
BonusPoints	Bonus Points	Number	
RedemptionId	Redemption Id for reference	Number	
ReturnMessage	Return Message	Varchar	512
EOSSPoints	Points Earned on EOSS Amount	Number	6
NonEOSSPoints	Points Earned on NON EOSS Amount	Number	3
Vouchers			

#### Error Codes:

Error Code	Description
0	Success
253	Member Id does not exist.
301	Points are mandatory for activity based accrual.

261	Bill id already exists.
129	Cannot accrue as wallet does not have enough easyPoints.
170	Offer does not exist in this range.
302	Cannot accrue easyPoints as the amount is less.
264	Activity does not exist.
262	User name does not exist.
263	Store does not exist.
285	Offer does not exist.
287	Invalid user name.
288	No transaction has made with this bill id.
289	Customer does not have enough easyPoints to redeem.
290	Invalid security code.
265	InvalidCustomerId,
291	Security code already used.
124	Security token verification failed.
104	Do not have permissions to access this method.
105	Security token expired.
125	Input parameters are not provided correctly.

**Sample:**

Type	JSON
Request	<pre>{   "EasyId": "9634098384",   "SecurityToken":     "7FUyCvpq0+gEFD56ivjg44il07dZWNQqAFZ2c/IIWsE7hwdhQz8XISdcDXnigg     q7iRTFRKA+0Cs7D6lwgzRLsB2gTYWXOnZANijfZyE/eO6M24uoF0Wiyx13mu     xugJ61QNb6cAQITU/oU5KflR9w4YuXb3ongARO4THLRf96IZa2Ugs7sVfhQfY     ZA27/Jq",</pre>



	<pre>"TransactionCode": "Old003", "RedemptionCode": "287148", "UserName": "vaibhav@easyrewardz.com", "EOSS": "0", "NONEOSSAmount": "1500", "NetAmount": "1500", "CountryCode": "91", "NewTransactionCode": "New003" }</pre>
Response - Success	<pre>{   "ReturnCode": "0",   "PointsEarned": "0",   "SmsStatus": "0",   "EmailStatus": "0",   "TPEnrollStatus": "0",   "EREnrollStatus": "0",   "ProgramStatus": "0",   "BonusPoints": "0",   "CCIPoints": "0",   "ReturnMessage": "Success.",   "RedemptionId": "10216",   "EOSSPoints": "0",   "NonEOSSPoints": "0",   "Vouchers": null,   "OfferModelId": "0" }</pre>
Response - Error	<pre>&lt;Response&gt;  &lt;ReturnCode&gt;288&lt;/ReturnCode&gt;</pre>

```
<ReturnMessage>No transaction has made with this bill
id</ReturnMessage>

</Response>
```

### 3.4.7.Flat Redemption from Member's account

**Method Name and Signature:** wsFlatEasyPointsRedemption (Varchar strRequest)

**API URL (REST):** [{{BaseUrl}}/FlatEasyPointsRedemption](#)

**Method Name and Description:** This method is used to redeem points from a member's account. This method redeems points directly from a member's account without any validations.

**Approach:** This method needs to be called when points need to be redeemed from a member (or members) account without any validations.

**Pre-Condition:**

**Post-Condition:**

**Input Parameters:**

Field	Mandatory	Field Type	Field Length	Field Description
SecurityToken	M	Varchar	512	This is a unique token Id used to authenticate the request.
UserName	M	Varchar	64	Partner's User Name
StoreCode	M	Varchar	64	Store Code used to enroll member. Store Code must already be configured in LPaaS before calling this method.
EasyId	M	Varchar	32	EasyId of the member
EasyPoints	M	decimal	32	Points available at member
TransactionCode	O	Varchar	64	TransactionCode (Bill Number)
TransactionDate	O	DateTime		Transaction Date (DD MMM YYYY)
TransactionDescription	O	Varchar	128	This will be used to save the transaction details of the redemption.
ClientCustomerID	O	Varchar	128	-

<b>DestinationPoints</b>	O	decimal		-
<b>CountryCode</b>	M	Varchar	64	It is the country code. To be used for support of international numbers.
<b>EasyId</b>	M	Varchar	32	EasyId of the member
<b>SecurityToken</b>	M	Varchar	512	This is a unique token Id used to authenticate the request.
<b>UserName</b>	M	Varchar	64	Partner User Name
<b>StoreCode</b>	M	Varchar	64	Store Code used for Transaction. Store Code must already be configured in LPaaS before calling this method.
<b>TransactionCode</b>	M	Varchar	64	Transaction Code (Bill Number)
<b>TransactionDate</b>	M	DateTime		Transaction Date (DD MMM YYYY)
<b>TypeId</b>	O	Integer		-
<b>CountryCode</b>	M	Varchar	64	It is the country code. To be used for support of international numbers.

#### Return Parameters:

Parameter Name	Description	Data Type	Length
ReturnCode	Return Code of the Response	Number	4
RedemptionId	Redemption Id for reference	Number	
ReturnMessage	Return Message	Varchar	512
ERPassword			
StoreCode	Store Code used to enroll member. Store Code must already be configured in LPaaS before calling this method.	Varchar	50
BillNo	Bill Number	Varchar	64
Mobile	Mobile no of the member	Number	10
TotalRedeemPoints			
TotalBilledAmount			

### 3.5. Issuance and Redemption of Coupons

Any Code/Document which can be presented to the retailer to gain financial benefit in the form of discount on any product is called a coupon. Customers can get the coupons redeemed at the specific retail outlets to avail relevant discounts and rebates in shopping.

This section describes the APIs used to issue and redeem a Coupon to/from a Member's account.

### 3.5.1. Issue Coupon

**Method Name and Signature:** wsIssueCoupon (Varchar strRequest)

**API URL (REST):** [{{BaseUrl}}/IssueCoupon](#)

**Method Description:** This method is used to issue coupon.

**Approach:** There are certain scenarios where a transaction or activity has coupon offer associated. When this transaction or activity occurs then coupon is automatically issued to member account. For example, when a member enrolls to the program or does a first transaction or spends a minimum set amount, a coupon is issued to the member.

**Best Implementation Practice:**

1. This API is used by POS for scenario where Coupons have be issued at Store.

**Pre-Condition:** Coupon offer should exist.

**Post-Condition:** Coupon is issued to a customer.

**Input Parameters:**

Parameters	M/O	Type	Length	Decription
SecurityToken	M	Varchar	512	This is a unique token Id used to authenticate the request.
StoreCode	M	Varchar	64	Store Code from where coupon is issued. It must already be configured in LPaaS before calling this method.
UserName	M	Varchar	64	Partner's User Name, api username
MemberId	M	Varchar	32	Mobile no. or membership no. of customer to which coupon is issued
Date	M	DateTime		Date on which coupon is issued
CouponOfferCode	M	Varchar	64	Coupon Offer Code to issue
CouponCode	M	Varchar	64	Coupon code issued
ThirdParty	O	Varchar	64	Thirdparty coupon or not
SegmentCode	O	Varchar	64	Segment Code
TierCode	O	Varchar	64	TierCode
CategoryCode	O	Varchar	64	CategoryCode
Communicate	M	Varchar	64	To communicate or not
CountryCode	M	Varchar	64	It is the country code. To be used for support of international numbers
TotalPaidAmount	O	Decimal		
TransactionId	O	Varchar		
ChannelCode	O	Varchar	64	ChannelCode





CouponOfferTypeCode	O	Varchar	64	
---------------------	---	---------	----	--

**Return Parameters:**

Parameter Name	Description	Data Type	Length
ReturnCode	Return Code indicating the status	Number	4
ReturnMessage	Return Message	Varchar	512
CouponCode	Coupon Code issued to member	Varchar	32
ShortDescription	Description of coupon	Varchar	512
ValidDate	Validity date of Coupon code	Date	

**Sample:**

Type	JSON
Request	<pre>{   "SecurityToken":     "7FUyCvpq0+gEFD56ivjg44il07dZWNQQaFZ2c/IIWsE7hwdhQz8XISdcDXn     iggg7iRTFRKA+0Cs7D6lwgzRLsB2gTYWXOnZANiiFZyE/eO6M24uoF0Wiy     x13muxugJ61QNb6cAQITU/oU5KfIR9w4YuXb3ongARO4THLRrf96IZa2Ug     s7sVfhQfYZA27/Jq",   "MemberID": "9634098384",   "Date": "01 Jul 2020",   "StoreCode": "DummyStore",   "CouponOfferCode": "DummyFlat500",   "CouponCode": "",   "ThirdParty": "",   "SegmentCode": "",   "TierCode": "",   "CategoryCode": "",   "UserName": "vaibhav@easyrewardz.com",   "Communicate": "No",   "RecordID": "",   "OfferCode": "",   "CountryCode": "" }</pre>

	}
Response {Success}	{ "ReturnCode": 0, "CouponCode": "YUOMOK2C3R", "EasyProgramName": "", "EasyCodeClaimTypeName": "", "POSPromoAmount": "500.00", "ShortDescription": "", "ValidDate": "2022-07-01T23:59:59", "ReturnMessage": "Success" }
{Error}	<?JSON version="1.0"?>  <Response JSONNs:xsi="http://www.w3.org/2001/XMLSchema-instance" JSONNs:xsd="http://www.w3.org/2001/XMLSchema"> <ReturnCode>402</ReturnCode> <ReturnMessage>No Coupon Offer Exists</ReturnMessage> </Response>

### 3.5.2. Get LPaaS Campaign Available Coupon

**Method Name and Signature:** GetLPaaSCampaignAvailableCoupon (Varchar strRequest)

**API URL (REST):** [{{BaseUrl}}/GetLPaaSCampaignAvailableCoupon](#)

**Method Description:** This method is used to get all coupons issued to a customer.

**Approach:** There are certain scenarios where coupons are issued to a customer. One may want to view coupon status and details before redemption or just to view status. Then GetAvailableCoupon API can be used.

**Best Implementation Practice:**

1. This API will display Available Coupon Code Masked
2. Cashier should have flexibility to apply Coupon

**Pre-Condition:** Coupon should have been issued. IssueCoupon API is used for this purpose.

**Post-Condition:** One can redeem and use a coupon using RedeemCoupon API and UseCoupon API.

**Input Parameters:**

Parameter Name	M/O	Data Type	Length	Description
SecurityToken	M	Varchar	512	This is a unique token Id used to authenticate the request.
MobileNumber	M	Varchar	32	Mobile number of members
CountryCode	M	Varchar	64	It is the country code. To be used for support of international numbers.
CouponStatus	M	Varchar	32	Status of the Coupon. A coupon can have following statuses Generated – when coupon has been generated in Mapped – when coupon has been mapped to a customer Claimed – when coupon has been redeemed by a customer Expired – when coupon has been expired Blocked – when coupon is blocked for redemption
DayInterval	O	Number		
UserName	M	Varchar	64	Partner user name

**Return Parameters:**

Parameter Name	Description	Data Type	Length
ReturnCode	ReturnCode indicating the status	Number	4
ReturnMessage	Return Message	Varchar	512
IstCouponOfferList	This is parent tag that contains CouponOfferList tags		
CouponOfferList	Below tags are part of this tag		
CouponCode	Coupon Code	Varchar	32
CouponOfferName	Coupon Offer name	Varchar	64
CouponNarration	Narration text for Coupon	Varchar	
CouponStatus	Status of Coupon: Expired/Used/Blocked/Issued	Varchar	32
ExpiryDate	Coupon Expiry Date	Date	
POSPromoAmount	Discount amount to be applied	Varchar	



POSPromo	POS Promo- Configured in local engine	Varchar	128
----------	---------------------------------------	---------	-----

**Error Codes:**

Error Code	Description
0	Success
423	Invalid RequestID

**Sample:**

Type	JSON
Request	<pre>{   "SecurityToken":     "7FUYCvpq0+gEFD56ivjg44il07dZWNQQaFZ2c/IIWsE7hwd     hQz8XISdcDXniggq7iRTFRKA+0Cs7D6lwgzRLsB2gTYWXOn     ZANijFZyE/eO6M24uoF0Wiyx13muxugJ61QNb6cAQITU/o     U5KfIR9w4YuXb3ongARO4THLRrf96lZa2Ugs7sVfhQfYZA27     /Jq",   "MobileNumber": "9634098384",   "CouponStatus": "",   "DayInterval": "",   "UserName": "vaibhav@easyrewardz.com" }</pre>
Response {Success}	<pre>{   "ReturnCode": "0",   "ReturnMessage": "Success",   "IstCouponOfferList": {     "CouponOfferList": [       {         "CouponOfferName": "MFlatDiscount",         "CouponOfferCode": "MFlatDiscount",         "CouponCode": "MJA8LWO3H5",         "CouponNarration": "200 rs discount with MBA 2000",         "CouponStatus": "Used",         "Uses": null, </pre>



```
"ExpiryDate": "2021-08-31T23:59:59",
"POSPromoAmount": "200.0000",
"ClaimedDate": "6/4/2021 12:00:00 AM",
"IssuedDate": "2021-06-03T00:00:00",
"RedeemBrandList": null,
"POSPromo": "1000"
},
{
  "CouponOfferName": "SkechersOffer",
  "CouponOfferCode": "SkechersOffer",
  "CouponCode": "ERJG12A6CJ",
  "CouponNarration": null,
  "CouponStatus": "Expired",
  "Uses": null,
  "ExpiryDate": "2021-02-16T23:59:59",
  "POSPromoAmount": "15.0000",
  "ClaimedDate": null,
  "IssuedDate": "2020-11-09T00:00:00",
  "RedeemBrandList": null,
  "POSPromo": "1"
}
]
}
}
```

### 3.5.3. Check Coupon Availability

**Method Name and Signature:** wsCheckCouponAvailability (Varchar strRequest)

**API URL (REST):** [{{BaseUrl}}/CheckCouponAvailability](#)

**Method Description:** This method is used to check the availability of coupon for a member.

**Approach:** Before redeeming a coupon, a coupon's availability can be checked for a member.

**Pre-Condition:**

**Post-Condition:**

**Input Parameters:**

Parameter	M/O	Type	Length	Description
<b>SecurityToken</b>	M	Varchar	512	This is a unique token Id used to authenticate the request.
<b>UserName</b>	M	Varchar	64	Partner's User Name, api username
<b>MemberID</b>	M	Varchar	32	Mobile no. or membership no. of customer to which coupon is issued
<b>IsAlert</b>	M	Varchar	32	0,1 if alert has to be sent or not
<b>CouponCode</b>	M	Varchar	64	Coupon code whose availability to be checked
<b>CountryCode</b>	M	Varchar	64	It is the country code. To be used for support of international numbers

**Return Parameters:**

Parameter Name	Description	Data Type	Length
ResponseCode	Response Code indicating the status	Number	4
ResponseMessage	Message Response	Varchar	512
MemberID	Member ID	Varchar	16
CouponStatus	Status of Coupon: Expired/Used/Blocked/Issued	Varchar	32
CouponExpiryDate	Expiry date of the coupon		
CouponOfferName	Coupon Offer name	Varchar	64
CouponNarration	Narration text for Coupon	Varchar	
CouponValue	Coupon Value		
POSPromo	POS Promo Code linked to Coupon	Varchar	30
DiscountType	0- Percentage,	Varchar	30

	1- Amount, 2 - Basis Transaction Amount		
DiscountValue	10 or 20 etc. (consideration of this as percentage or Amount depends upon Discount Type)	Number	3
MaxDiscountValue	Maximum Discount available	Float	
ReturnMessage	Message Response	Varchar	512
AllowAccrual	Flag indicating if points accrual is allowed with this coupon	Varchar	1
AllowRedemption	Flag indicating if points redemption is allowed with this coupon	Varchar	1
OTPRquired	This flag will indicate if OTP required for coupon redemption. Call Coupon OTP method to validate OTP	Varchar	1
BinGroups	Bin Group Allowed with this coupon code (blank means all)	Varchar	50
EligibilityType	Eligibility Type = 0 , MinAmount will be On Bill level Eligibility Type =1, MinAmount will On Assortment Eligibility Type =2, None	Number	
MinAmount	Min Amount condition to be applied based on Eligibility type	Varchar	
MaxItemQty	Max Quantity of Items on which Discount to be applied	Number	
MinItemQty	Minimum Quantities of Item on which Discount will be applied	Number	

**Error Codes:**

Error Code	Description
0	Success

**Sample:**

Type	JSON
Request	<pre>{   "EasyId": "7865439765",   "MemberID": "7865439765",   "CouponCode": "YUOMHSJ5LB",   "SecurityToken":     "7FUYCvpq0+gEFD56ivjg44il07dZWNQQaFZ2c/IIWsE7hwdh     Qz8XISdcDXniggq7iRTFRKA+0Cs7D6lwgzRLsB2gTYWXOnZA"</pre>



	<pre>NijFZyE/eO6M24uoF0Wiyx13muxugJ61QNb6cAQITU/oU5K fIR9w4YuXb3ongARO4THLRrf96IZa2Ugs7sVfhQfYZA27/Jq", "IsAlert": "1", "UserName": "vaibhav@easyrewardz.com", "CountryCode": "91" }</pre>
Response	<pre>{    "ReturnCode": 0,    "ReturnMessage": "Success",    "MemberId": "9634098384",    "CouponStatus": "Claimed",    "CouponExpiryDate": "31 Dec 2022",    "CouponOfferName": "Test500",    "POSPromo": "TEST#500",    "CouponNarration": "Flat 500 Rs discount",    "DiscountType": "Amount",    "CouponValue": "500.00",    "MaxDiscountValue": "500",    "AllowAccrual": 1,    "ArrayBillDetails": [],    "DiscountValue": "500.0000",    "EligibilityType": "3",    "MinAmount": "0",    "SpecialMessage": "",    "MaxItemQty": "0",    "MinItemQty": "0",    "AppliedOn": "",    "AllowRedemption": false  }</pre>



### 3.5.4. Redeem Coupon

**Method Name and Signature:** wsRedeemCoupon (Varchar strRequest)

**API URL (REST):** [{{BaseUrl}}/RedeemCoupon](#)

**Method Description:** This method is used to use a coupon to apply the corresponding discount promotion tagged to the coupon, in the transaction.

**Approach:** Once this method is called the coupon is first validated & if the same is successful, then is blocked by LPaaS for use by this member in this transaction. The same also returns to the transaction system/POS all details including promotion code, that is needed to trigger the associated promotion in the transaction.

**Best Implementation Practice:**

1. Response of this API is very critical to be consumed by any POS Partner
2. If POS has promotion Engine , POS can drive the Coupon Redemption using the POS Promo coming in the API.
3. If POS is not having Promotion Engine, POS has to consume all Tags of Redeem Coupon API
4. POS is supposed to use IsAccrual and IsRedemption Flag in both Cases
5. POS integration approach has to be finalized between ER and POS before moving ahead

**Pre-Condition:** This method is called before prepayment, to apply the promotion & corresponding discount in the transaction.

**Post-Condition:** The promotion code returned by the response of this method is used to trigger the discount in in the transaction. If the promotion code is not to be triggered, then the Transaction Type can also use the Discount Type & Discount Value to trigger the discount.

**Input Parameters:**

Parameter	M/O	Type	Length	Description
SecurityToken	M	Varchar	512	This is a unique token Id used to authenticate the request.
StoreCode	M	Varchar	64	Store Code from where coupon to redeemed. It must already be configured in LPaaS before calling this method.
UserName	M	Varchar	64	Partner's User Name, api username
MemberId	M	Varchar	32	Mobile no. or membership no. of customer to which coupon is issued
Date	M	DateTime		Date on which coupon is redeemed
CouponCode	M	Varchar	64	Coupon Code redeemed
CountryCode	M	Varchar	64	It is the country code. To be used for support of international numbers

**Return Parameters:**

Parameter Name	Description	Data Type	Length
ReturnCode	Return Code indicating the status	Number	4
RequestID	Request ID for Redeem Coupon, This Request ID will be used in Use Coupon Method	Varchar	20
POSPromo	POS Promo Code linked to Coupon	Varchar	30
DiscountType	0 - Percentage, 1 - Amount, 2 - Basis Transaction Amount	Varchar	30
DiscountValue	10 or 20 etc. (consideration of this as percentage or Amount depends upon Discount Type)	Number	3
MaxDiscountValue	Maximum Discount available	Float	
ReturnMessage	Message Response	Varchar	512
AllowAccrual	Flag indicating if points accrual is allowed with this coupon	Varchar	1
AllowRedemption	Flag indicating if points redemption is allowed with this coupon	Varchar	1
OTPREquired	This flag will indicate if OTP required for coupon redemption. Call Coupon OTP method to validate OTP	Varchar	1
BinGroups	Bin Group Allowed with this coupon code (blank means all)	Varchar	50
EligibilityType	Eligibility Type = 0 , MinAmount will be On Bill level Eligibility Type =1, MinAmount will On Assortment Eligibility Type =2, None	Number	
MinAmount	Min Amount condition to be applied based on Eligibility type	Varchar	
MaxItemQty	Max Quantity of Items on which Discount to be applied	Number	
MinItemQty	Minimum Quantities of Item on which Discount will be applied	Number	

**Error Codes:**

Error Code	Description
0	Success
368	Invalid Coupon
438	Coupon already blocked
440	Coupon is expired
424	Invalid Request
436	Not Eligible Store
437	Coupon already claimed
439	Member is not registered
438	Coupon already blocked
441	Not same store
435	Not Same Member

**Sample:**

Type	JSON
Request	<pre>{   "SecurityToken":     "7FUYCvpq0+gEFD56ivjg44il07dZWNQQaFZ2c/IIWsE7h     wdhQz8XISdcDXniggq7iRTFRKA+0Cs7D6lwgzRLsB2gTYW     XOnZANijFZyE/eO6M24uoF0Wiyx13muxugJ61QNb6cA     QITU/oU5KflR9w4YuXb3ongARO4THLRf96lZa2Ugs7sVf     hQfYZA27/Jq",   "MemberID": "9634098384",   "Date": "10 Aug 2020",   "StoreCode": "DummyStore",   "CouponCode": "YUOM5V97L4",   "UserName": "vaibhav@easyrewardz.com",   "CountryCode": "91" }</pre>
Response	<pre>{   "ReturnCode": 0,   "ReturnMessage": "Success",   "RequestID": "002552",   "DiscountType": 1,   "DiscountValue": "500.00",   "MaxDiscountValue": "500.00",   "BalanceToPay": "0.00", }</pre>

```

"AllowAccrual": 1,
"AllowRedemption": 1,
"OTPRequired": 1,
"POSPromo": "Test#123",
"EligibilityType": 2,
"MinAmount": "0.00",
"PaymentTypes": null,
"PaymentSources": null,
"BinGroups": null,
"MaxItemQty": 0,
"MinItemQty": 0,
"SpecialMessage": "",
"IssuanceOnRedemption": 0,
"AppliedOn": ""
}

```

### 3.5.5. Coupon OTP

**Method Name and Signature:** wsCouponOTP (Varchar strRequest)

**API URL (REST):** [{{BaseUrl}}/CouponOTP](#)

**Method Description:** There are some coupons which require OTP. Such coupons codes return IsOTPRequired field value as TRUE in the wsRedeemCoupon method call. During the execution of wsRedeemCoupon method for such coupons, an OTP is triggered to the redeeming member & requires validation for application of the coupon in the bill.

**Approach:** If Coupon requires OTP validation, this method needs to be called by POS to confirm the validity of Member. If the Customer has EPIN, EPIN will be passed in the OTP field.

**Best Implementation Practice:**

1. Resend Coupon OTP has to be used along with the API

**Pre-Condition:** wsRedeemCoupon method needs to be called before calling this method. Redeem coupon will return a flag indicating if the Coupon requires OTP or not. Basis this field value, the POS needs to show a field to accept the OTP. Then, this method is invoked.

**Post-Condition:** Post successful validation of the coupon, proceed to applying the associated discount & proceed to saving the bill. Post bill save call the wsUseCoupon method to mark the use of the coupon in the bill.

**Input Parameters:**

Parameter	M/O	Type	Length	Description
SecurityToken	M	Varchar	512	This is a unique token Id used to authenticate the request.
RequestID	M	Varchar	64	Request ID in response of Redeem coupon
OTP	M	Varchar	6	OTP received by Customer to validate
MemberID	O	Varchar		Mobile no. or membership no. of customer to which coupon is issued

<b>CountryCode</b>	M	Varchar	64	It is the country code. To be used for support of international numbers
<b>UserName</b>	M	Varchar	64	Partner User Name

#### Return Parameters:

Parameter Name	Description	Data Type	Length
ReturnCode	ReturnCode indicating the status	Number	4
ReturnMessage	Return Message	Varchar	512

#### Error Codes:

Error Code	Description
0	Success
378	OTP Required
365	Invalid OTP

#### Sample:

Type	JSON
Request	<pre>{   "SecurityToken":     "7FUyCvpq0+gEFD56ivjg44il07dZWNQQaFZ2c/IIWsE7hwdhQz8XISdcDXn     iggq7iRTFRKA+0Cs7D6lwgzRLsB2gTYWXOnZANiiFZyE/eO6M24uoF0Wiy     x13muxugJ61QNb6cAQITU/oU5KflR9w4YuXb3ongARO4THLRf96IZa2Ug     s7sVfhQfYZA27/Jq",   "RequestID": "001999",   "OTP": "742968",   "UserName": "vaibhav@easyrewardz.com",   "MemberID": "9634098384",   "CountryCode": "91" }</pre>
Response	<pre>{   "ReturnCode": 0,   "ReturnMessage": "Success",   "OTP": "" }</pre>

### 3.5.6. Resend Coupon OTP

**Method Name and Signature:** wsResendCouponOTP (Varchar strRequest)

**API URL (REST):** [{{BaseUrl}}/ResendCouponOTP](#)

**Method Description:** This method is used to resend the coupon OTP, in case the previous OTP sent by wsCouponOTP is not received or has expired.

**Approach:** When an OTP for a coupon has not been received, this method can be used to resend the OTP again.

**Pre-Condition:**

**Post-Condition:**

**Input Parameters:**

Parameter	M/O	Type	Length	Description
SecurityToken	M	Varchar	512	This is a unique token Id used to authenticate the request.
MemberId	M	Varchar	32	Mobile no. or membership no. of customer to which coupon is issued
TransactionCode	M	Varchar	64	Bill no. on which coupon to redeemed
UserName	M	Varchar	64	Partner's User Name, api username
StoreCode	M	Varchar	64	Store code in which coupon is to redeem
BillDate	M	DateTime		date on which coupon is redeem
CountryCode	M	Varchar	64	It is the country code. To be used for support of international numbers

**Return Parameters:**

Parameter Name	Description	Data Type	Length
ReturnCode	Return Code indicating the status	Number	4
ReturnMessage	Return Message	Varchar	512

**Sample:**

Type	JSON
Request	{

	<pre> "SecurityToken": "7FUYCvpq0+gEFD56ivjg44il07dZWNQqAFZ2c/IIWsE7hwdhQz8XISdcDXnig gq7iRTFRKA+0Cs7D6lwgzRLsB2gTYWXOnZANijFZyE/eO6M24uoF0Wiyx13 muxugJ61QNb6cAQITU/oU5KfIR9w4YuXb3ongARO4THLRf96IZa2Ugs7sVf hQfYZA27/Jq", "MemberId": "9634098384", "TransactionCode": "YUOM7IHL4O", "UserName": "vaibhav@easyrewardz.com", "StoreCode": "Dummystore", "BillDate": "01 Jul 2020", "CountryCode": "91" } </pre>
Response - Success	<pre> {   "ReturnCode": "0",   "ReturnMessage": "Success." } </pre>

### 3.5.7. Use Coupon

**Method Name and Signature:** wsUseCoupon (Varchar strRequest)

**API URL (REST):** [{{BaseUrl}}/UseCoupon](#)

**Method Description:** This method confirms the use of the coupon in a bill.

**Approach:** This method is called once the bill is saved, in which the coupon is applied. This method marks the coupon as used.

#### Best Implementation Practice:

1. This is an important API to call in all events where Could redemption could not be completed due to any reason for example Bill could not be saved, Item Addition/deletion

**Pre-Condition:** wsRedeemCoupon method needs to be called prior to this. The Request ID generated by the wsRedeemCoupon method is passed to this method as input, to link the transactions across the two method calls. The bill needs to be saved also, so that the Paid amount & discount due to coupon applied is also settled.

**Post-Condition:** The coupon used in the bill is marked as used once, by the member. Depending on coupon offer properties, this will prevent for this particular coupon code from any future use.

#### Input Parameters:

Parameter	M/O	Type	Length	Description
<b>SecurityToken</b>	M	Varchar	512	This is a unique token Id used to authenticate the request.
<b>RequestId</b>	M	Varchar	64	Request ID in response of Redeem coupon

<b>CouponCode</b>	M	Varchar	64	Coupon Code redeemed
<b>BillNo</b>	M	Varchar	64	Bill no on which coupon is redeemed
<b>Discount</b>	M	Decimal		Discount applied against couponcode
<b>OTP</b>	M/O	Varchar	6	OTP to validate for coupon redemption
<b>TotalPaidAmount</b>	M	Decimal		Net Amount payed by Customer after discount
<b>UserName</b>	M	Varchar	64	Partner's User Name, api username
<b>CountryCode</b>	M	Varchar	64	It is the country code. To be used for support of international numbers

#### Return Parameters:

Parameter Name	Description	Data Type	Length
ReturnCode	ReturnCode indicating the status	Number	4
CouponCode	Coupon Code Used	Varchar	20
ReturnMessage	Return Message	Varchar	512

#### Error Codes:

Error Code	Description
0	Success
423	Invalid RequestID

#### Sample:

Type	JSON
Request	<pre>{   "SecurityToken":     "7FUyCvpq0+gEFD56ivjg44il07dZWNQQaFZ2c/IIWsE7hwdhQz8XISdcD     Xniggq7iRTFRKA+0Cs7D6lwgzRLsB2gTYWXOnZANiiFZyE/eO6M24uoF0     Wiyx13muxugJ61QNb6cAQITU/oU5KflR9w4YuXb3ongARO4THLRrf96lZ     a2Ugs7sVfhQfYZA27/Jq",   "RequestID": "001999",   "BillNo": "INV890789",   "Discount": "500",   "OTP": "742968",   "TotalPaidAmount": "1000",   "CouponCode": "YUOM7IHL4O",   "UserName": "vaibhav@easyrewardz.com", }</pre>



Response	<pre>{   "ReturnCode": 0,   "CouponCode": "YUOM7IHL4O",   "ReturnMessage": "Success" }</pre>
----------	----------------------------------------------------------------------------------------------

### 3.5.8. Unblock Coupon

**Method Name and Signature:** wsUnBlockCouponCode (Varchar strRequest)

**API URL (REST):** [{{BaseUrl}}/UnBlockCoupon](#)

**Method Description:** This method invalidates the block on the coupon code which is blocked for use by a prior call of wsRedeemCoupon method.

**Approach:** There are certain scenarios where a transaction which has coupon applied goes thru changes post application of coupon. This change can be addition of an item, removal of item or void bill. The coupon needs to be re-applied in such cases for right calculation of discount & distributing the same across items in the bill. In such cases this method is called to release the coupon code & make it available for re-application.

**Pre-Condition:** The coupon code to be unblocked needs to be prior blocked by wsRedeemCoupon method. It will accept the Request ID as an input, that is generated by the wsRedeemCoupon method. The request id links the two transactions & unblocks the coupon.

**Post-Condition:** Coupon code is unblocked after this call & is ready to be re-applied.

#### Input Parameters:

Parameter	M/O	Type	Length	Description
SecurityToken	M	Varchar	512	This is a unique token Id used to authenticate the request.
RequestID	M	Varchar		Request ID in response of Redeem coupon
CountryCode	M	Varchar	64	It is the country code. To be used for support of international numbers
UserName	M	Varchar	64	Partner's User Name, api username
CouponCode	M	Varchar		Coupon Code to unblock

#### Return Parameters:

Parameter Name	Description	Data Type	Length
ReturnCode	ReturnCode indicating the status	Number	4
Coupon Code	Coupon Code unblocked	Varchar	20
ReturnMessage	Return Message	Varchar	512

#### Error Codes:



Error Code	Description
0	Success
423	Invalid RequestID

**Sample:**

Type	JSON
Request	<pre>{   "RequestID": "001989",   "CouponCode": "YUOMN35VDB",   "UserName": "vaibhav@easyrewardz.com",   "SecurityToken":     "7FUyCvpq0+gEFD56ivjg44il07dZWNQQaFZ2c/IIWsE7hwd     hQz8XISdcDXniggq7iRTFRKA+OCs7D6lwgzRLsB2gTYWXOn     ZANijFZyE/eO6M24uoF0Wiyx13muxugJ61QNb6cAQITU/o     U5KfIR9w4YuXb3ongARO4THLRrf96lZa2Ugs7sVfhQfYZA27     /Jq" }</pre>
Response	<pre>{   "ReturnCode": 0,   "CouponCode": "YUOMN35VDB",   "ReturnMessage": "Success" }</pre>

**3.5.9. Update Coupon Status**

**Method Name and Signature:** wsUpdateCouponStatus (Varchar strRequest)

**API URL (REST):** [{{BaseUrl}}/UpdateCouponStatus](#)

**Method Description:** This method is required to update the status of the coupon.

**Approach:**

**Pre-Condition:** Coupon should already exist.

**Post-Condition:** Coupon status will be updated.

**Input Parameters:**

Parameter	M/O	Type	Length	
SecurityToken	M	Varchar	512	This is a unique token Id used to authenticate the request.
MemberID	M	Varchar	32	Mobile no. or membership no. of customer to which coupon is issued
StoreCode	M	Varchar	64	Store Code



CouponCode	M	Varchar	64	Coupon code to update
BillNo	M	Varchar	64	Bill to Which Coupon is applied
Discount	M	Decimal		Discount applied due to Coupon
TotalPaidAmount	M	Decimal		Total Bill Amount
UserName	M	Varchar	64	Partner's User Name, api username
ToStatus	M	Varchar	32	New Status to Changed
CountryCode	M	Varchar	64	It is the country code. To be used for support of international numbers

**Return Parameters:**

Parameter Name	Description	Data Type	Length
ReturnCode	Return Code indicating the status	Number	4
ReturnMessage	Return Message	Varchar	512
MemberID	Unique ID of the member	Varchar	16
CouponOldStatus	Coupon's Old status	Varchar	32
CouponNewStatus	Coupon's New status	Varchar	32
CouponOfferName	Coupon Offer Name	Varchar	64

**Sample:**

Type	JSON
Request	<pre> &lt;Request&gt; &lt;MemberID&gt;&lt;/MemberID&gt; &lt;StoreCode&gt;Demo&lt;/StoreCode&gt; &lt;CouponCode&gt;GR092OK7ZV&lt;/CouponCode&gt; &lt;BillNo&gt;11101&lt;/BillNo&gt; &lt;Discount&gt;100&lt;/Discount&gt; &lt;TotalPaidAmount&gt;1000&lt;/TotalPaidAmount&gt; &lt;UserName&gt;apparaod@revalsys.com&lt;/UserName&gt; &lt;SecurityToken&gt;iuuHSFP/08BZ57sRJC4/ulxBUN8HAaGROk3uiOhBVhPfsiKQ mFsrb8KYYaWIXcBbrYMIIdYWNLJUXMyBmPvKZzuRWIOtCFC3EnCykXwkWH ysRstcx3iK2M6qRE4qUAPVwGHxcip0TQ4YF4cCQbqOEABgmq5vu8B68Db2 XM4g4YrjG8uQWVthlydrRHD8FymYgfvdxHv/kremyf/ynH98yQ==&lt;/SecurityToken&gt; &lt;ToStatus&gt;Expired&lt;/ToStatus&gt; &lt;CountryCode&gt;91&lt;/CountryCode&gt; &lt;/Request&gt; </pre>
Response - Success	<pre> &lt;Response&gt;  &lt;ReturnCode&gt;0&lt;/ReturnCode&gt; </pre>



	<pre> &lt;ReturnMessage&gt;Success&lt;/ReturnMessage&gt;  &lt;MemberId&gt;7673977543&lt;/MemberId&gt;  &lt;CouponOldStatus&gt;Mapped&lt;/CouponOldStatus&gt;  &lt;CouponNewStatus&gt;Expired&lt;/CouponNewStatus&gt;  &lt;CouponOfferName&gt;GenOnReq0908&lt;/CouponOfferName&gt;  &lt;/Response&gt; </pre>
Response - Error	

### 3.6. Issuance and Redemption of Gift Voucher (GV)

GV is a tender which can be redeemed only at limited Payment Pages. Gift Vouchers are given out by retailers and marketers as part of a promotion strategy, to entice the recipient to come in or return to the store. Gift vouchers are generally redeemable only for purchases at the relevant retail premises and cannot be cashed out, and in some situations may be subject to an expiry date or fees.

This section describes the APIs used to issue and redeem a GV to/from a Member's account.

#### 3.6.1. Issue GV

**Method Name and Signature:** wslIssueGV (Varchar strRequest)

**API URL (REST):** [{{BaseUrl}}/IssueGV](#)

**Method Description:** This method is used to issue a GV. Required fields can be configured in LPaaS.

**Approach:** LPaaS supports multiple ways of issuing GV. It can be based sold, can be issued against redemption of points or can be issued as an offer reward. GV can be issued to the customer who is availing it or can be issued as a gift to someone by providing contact details or member ID.

**Best Implementation Practice:**

1. This API will be used to generate Credit Note of Amount which has to be returned
2. This API will be used to for Gift card Sale Use Case

**Pre-Condition:** GV Offer Code should exist.

**Post-Condition:** Customer/Member will be able to use the GV code issued

**Input Parameters:**

Field	Mandatory	Field Type	Field Length	Field Description
MemberID	M	Varchar	32	Member who has requested GV Issuance

<b>GVOfferCode</b>	M/O	Varchar	64	( This is a conditional parameter – if GV Code, member ID , store ID ,category are available then this is optional)
<b>GVCode</b>	M/O	Varchar	64	To issue specific GVcode to member
<b>PointsRedeemed</b>		decimal		Points to be redeemed to issue GV (for issue type redeem)
<b>IssueToMemberID</b>	M/O	Varchar	32	Member ID to which GV should be issued
<b>IssueToEmailId</b>	M/O	Varchar	32	Email id to which GV should be issued and alerted
<b>IssueToMobile</b>	O	Varchar	32	Mobile number to which GV should be issued
<b>StoreCode</b>	M	Varchar	64	Store Code
<b>GVAmount</b>		Decimal		Amount of GV to be issued ( mandatory for GV type sale and denomination is open or if BTA then this is transaction amount)
<b>SecurityToken</b>	M	Varchar	312	This is a unique token Id used to authenticate the request.
<b>UserName</b>	M	Varchar	64	Partner user name
<b>Date</b>		DateTime		#N/A
<b>TransactionId</b>		Varchar	64	Transaction Id
<b>OfferTransactionID</b>		Long		
<b>OfferCode</b>		Varchar	64	Offer Code
<b>Communicate</b>	O	Varchar	8	#N/A
<b>CouponOfferTypeCode</b>	O	Varchar	64	#N/A
<b>CountryCode</b>	M	Varchar	64	#N/A

**Return Parameters:**

Parameter Name	Description	Data Type	Length
----------------	-------------	-----------	--------

ReturnCode	ReturnCode indicating the status	Number	4
GVCode	GV code issued	Varchar	20
PIN	Pin issued for GV redemption	Varchar	10
GVExpiryDate	Expiry date for GV	DateTime	
GVValue	Value of GV	Number	
GVStatus	Status of GV issued	Varchar	20
GVBalance	Balance in GV	Number	
ReturnMessage	Return Message	Varchar	512

**Error Codes:**

Error Code	Description
0	Success
107	Input parameters are incorrect. Missing memberid or IssueToEmailId or IssueToMemberId or IssueToMobile
107	Input parameters are incorrect. missing security token
107	Input parameters are incorrect. Missing GV Offer Code
107	Input parameters are incorrect. Missing StoreCode
397	GVAmount Should be greater than Zero
107	Input parameters are incorrect. missing username
199	User does not exist.
394	Invalid Amount
198	Store code does not exist.
268	Invalid mobile number.
269	Invalid email id.
393	Invalid Issue to Member ID
395	GVOffer does not exist
277	Invalid points.
286	Invalid member id.



124	Security token verification failed.
-----	-------------------------------------

**Sample:**

Type	JSON
Request	<pre>{     "SecurityToken":     "7FUYCvpq0+gEFD56ivjg44il07dZWNQQaFZ2c/IIWsE7hwdhQz8XISdc     DXniggq7iRTFRKA+0Cs7D6lwgzRLsB2gTYWXOnZANiiFZyE/eO6M24u     oF0Wiyx13muxugJ61QNb6cAQITU/oU5KfIR9w4YuXb3ongARO4THLRr     f96lZa2Ugs7sVfhQfYZA27/Jq",     "MemberID": "9634098384",     "GVOfferCode": "DummyGV300",     "GVCode": "",     "PointsRedeemed": "",     "IssueToMemberID": "",     "IssueToEmailId": "",     "IssueToMobile": "",     "StoreCode": "DummyStore",     "Category": "",     "GVAmount": "",     "UserName": "vaibhav@easyrewardz.com",     "ThirdParty": "No",     "SegmentCode": "Test",     "TierCode": "",     "TransactionId": "",     "OfferTransactionID": "",     "OfferCode": "",     "Communicate": "Yes",     "CouponOfferTypeCode": "" }</pre>
Response – Success	<pre>{     "ReturnCode": 0,     "ReturnMessage": "Success",     "GVCode": "OPKJB9NF9U",     "PIN": "",     "GVExpiryDate": "2021-08-10T23:59:00",     "GVValue": "300.00",     "GVStatus": "Approved",     "GVBalance": "300.00" }</pre>
Response – Error	<pre>{     "ReturnCode": 124,     "ReturnMessage": "Security token verification failed." }</pre>

### 3.6.2. Get Available GVs

**Method Name and Signature:** wsGetAvailableGVs (Varchar strRequest)

**API URL (REST):** [{{BaseUrl}}/GetAvailableGVs](#)

**Method Description:** This method is used to rollback amount to GV.

**Approach:** This method is to be invoked if all the GVs issued to a customer are to be retrieved and shown before a customer wishes to apply a GV.

**Best Implementation Practice:**

1. This API will display Available GV Code Masked
2. Cashier should have flexibility to apply Coupon

**Pre-Condition:** Security token must be validated.

**Post-Condition:** GV details are returned and can be shown or applied by the customer.

**Input Parameters:**

Field	Mandatory	Field Type	Field Length	Field Description
MobileNumber	M/O	Varchar	32	Mobile number of member
SecurityToken	M	Varchar	512	This is a unique token Id used to authenticate the request.
DayInterval		Long		
UserName	M	Varchar	64	Partner user name
CountryCode	M	Varchar	64	Country Code
TransactionID	O			Bill number on which GV is Issued
StoreCode	O			Store code from which GV is issued
BillDate	O			Bill date on which GV is issued (DD MMM YYYY eg: 18 Jun 2019)
GV status	O	Varchar	32	GV Status

**Return Parameters:**

Parameter Name	Description	Data Type	Length
ReturnCode	ReturnCode indicating the status	Number	4
ReturnMessage	Return Message	Varchar	512





GVOfferName	GV Offer name	Varchar	64
GVCode	GVOffer Code	Varchar	32
GVNarration	Narration in GV Offer	Varchar	256
GVStatus	Status of GV (Generated, Expired, Blocked, Approved, Used, Partial, Merged)	Varchar	32
ExpiryDate	GV Expiry Date	Date	
AvailableBalance	Balance available in GV	Number	
Denomination	Denomination of GV	Number	

**Sample:**

Type	JSON
Request	<pre>{   "UserName": "apiuser@genisys.com",   "SecurityToken":     "xkk/IptYjbQo4z4RD2fB46uDsyV6t3eT0A6Haq/L5vho60DAwhkqISU1/gBxxvRmE39Hc9Xw     AJPqgUjD4wWDbL17qtrnmKi3/PVEW3ER3IS/DzgWDYIIAuBokXCDnTU/261kHO8XFRuKtmk     WhRxp6QbK17mawPhiqACmaRyjfi39AAossgg/8gUI7agU0xam",   "GVStatus": "",   "MobileNumber": "8447668782" }</pre>
Response {Success}	<pre>{   "ReturnCode": 0,   "ReturnMessage": "Success",   "GVOfferList": [     {       "GVOfferName": "MultiUseGV",       "GVCode": "MUCPHES9VNHX3",       "GVNarration": "MultiUseGV",       "GVStatus": "Approved",       "ExpiryDate": "2020-06-14T23:59:59",       "IstImageList": null,       "AvailableBalance": 1000.0000,       "Denomination": 1000.0000,       "TransactionID": "",       "MobileNumber": "8447668782",       "IssuedStoreCode": "ginesys",       "IssuedMobileNo": "",       "IssuedDate": "2020-04-30T00:00:00",       "RedeemBrandList": [         {           "BrandCode": "Ginesys"         }       ]     }   ],   {     "GVOfferName": "MultiUseGV",     "GVCode": "MU8XSP2NWGGV7",     "GVNarration": "MultiUseGV",     "GVStatus": "Approved",</pre>



```

"ExpiryDate": "2020-06-14T23:59:59",
"IstImageList": null,
"AvailableBalance": 1000.0000,
"Denomination": 1000.0000,
"TransactionID": "",
"MobileNumber": "8447668782",
"IssuedStoreCode": "ginesys",
"IssuedMobileNo": "",
"IssuedDate": "2020-04-30T00:00:00",
"RedeemBrandList": [
  {
    "BrandCode": "Ginesys"
  }
],
},
{
  "GVOfferName": "MultiUseGV",
  "GVCode": "MUU2IR379HM2P",
  "GVNarration": "MultiUseGV",
  "GVStatus": "Approved",
  "ExpiryDate": "2020-06-14T23:59:59",
  "IstImageList": null,
  "AvailableBalance": 1000.0000,
  "Denomination": 1000.0000,
  "TransactionID": "",
  "MobileNumber": "8447668782",
  "IssuedStoreCode": "ginesys",
  "IssuedMobileNo": "",
  "IssuedDate": "2020-04-30T00:00:00",
  "RedeemBrandList": [
    {
      "BrandCode": "Ginesys"
    }
  ]
}
]
}

```

### 3.6.3. Check Availability of GV

**Method Name and Signature:** wsCheckGVAvailability (Varchar strRequest)

**API URL (REST):** [{{BaseUrl}}/CheckGVAvailability](#)

**Method Description:** This method is used to verify GV code and its status before application.

**Approach:** Call this method to verify GV code, its status and expiry. LPaaS will verify the GV and return details for same, Same can also be chosen to be returned as alert message.

**Pre-Condition:** Security Token needs to be generated before calling this method. Each outlet (POS store code, online web channel, Mobile App etc.) needs to have a unique code assigned to them.

**Post-Condition:** GV details will be returned post verification.

**Input Parameters:**

Field	Mandatory	Field Type	Field Length	Field Description
<b>GVCode</b>	M	Varchar	64	GV Code
<b>SecurityToken</b>	M	Varchar	512	This is a unique token Id used to authenticate the request.
<b>MemberID</b>	M	Varchar	32	Unique ID of the member. This could be either Mobile Number or Client ID
<b>IsAlert</b>	M	Varchar	32	Set as true if alert is to be send in
<b>CountryCode</b>	M	Varchar	64	It is the country code. To be used for support of international numbers.
<b>GVOfferCode</b>			64	

**Return Parameters:**

Parameter Name	Description	Data Type	Length
ReturnCode	Return Code indicating the status	Number	4
ReturnMessage	Message description	Varchar	128
GVCode	GV code issued	Varchar	20
GVOfferCode	GV Offer Code	Varchar	
GVExpiryDate	Expiry date for GV	DateTime	
GVValue	Value of GV	Float	10
GVBalance	Available balance after redeeming GV	Float	
PartialAllowed	True/false (If multi use is selected in offer we are showing partial allowed as True else we are showing as False)	Varchar	5
GVStatus	Status of GV issued	Varchar	32

**Error Codes:**

Error Code	Description
0	Success
107	Input parameters are incorrect. missing security token
107	Input parameters are incorrect. missing MemberID



107	Input parameters are incorrect. missing GV Code
286	Invalid member id.
124	Security token verification failed.
392	Invalid GVOffer Code
105	Security token expired.

**Sample:**

Type	JSON
Request	<pre>{      "GVCode": "MU7FTDDWMH3B6",      "SecurityToken":     "xkk/lptYjbQo4z4RD2fB46uDsyV6t3eT0A6Haq/L5vho60     DAwhkqlSU1/gBxxvRmE39Hc9XwAJPqgUjD4wWDbL17q     trnmKi3/PVEW3ER3IS/DzgWDYIIAuBokXCDnTU/261kHO     8XFRuKtmkWhRxp6QbK17mawPhiqACmaRyjfi39AAossg     g/8gUI7agU0xam",      "MemberID": "8447668782",      "IsAlert": "",      "CountryCode": "91"  }</pre>
Response {Success}	<pre>{      "ReturnCode": 0,      "ReturnMessage": "",      "GVOfferCode": "MultiUseGV",      "GVCode": "MU7FTDDWMH3B6",      "GVStatus": "Approved",      "GVValue": "1000.00",      "GVBalance": "1000.00",      "GVExpiryDate": "14062020235959",      "PartialAllowed": true,      "CheCkGVAvailablityListResponse": null,</pre>

	<pre>"ArrayBillDetails": [] }</pre>
<b>Response</b> <b>{Error}</b>	<pre>&lt;Response&gt;     &lt;ReturnCode&gt;XXX&lt;/ReturnCode&gt;     &lt;ReturnMessage&gt;GVcode does not exists&lt;/ReturnMessage&gt; &lt;/Response&gt;</pre>

### 3.6.4. Redeem GV

**Method Name and Signature:** wsRedeemGV (Varchar strRequest)

**API URL (REST):** [{{BaseUrl}}/RedeemGV](#)

**Method Description:** This method is used to verify and block a GV for redemption against a transaction.

**Approach:** The method accepts GV code, verifies it and blocks it for redemption.

**Pre-Condition:** Security token should be generated and should be valid.

**Post-Condition:** GV will be blocked for redemption unless unblock GV is received. Any other request for GV use will be rejected.

**Input Parameters:**

Field	Mandatory	Field Type	Field Length	Field Description
<b>MemberID</b>	M	Varchar	32	Member ID
<b>GVCode</b>	M	Varchar	64	GV Code
<b>Date</b>	M	DateTime		Redemption date
<b>StoreCode</b>	M	Varchar	64	Store Code
<b>PIN</b>		Varchar	6	PIN if mandatory
<b>OTP</b>		Varchar	6	OTP if mandatory
<b>Amount</b>	M	Decimal		GV amount applied (in case multi use GV this is mandatory)
<b>UserName</b>	M	Varchar	64	Username
<b>SecurityToken</b>	M	Varchar	512	This is a unique token Id used to authenticate the request.



CountryCode	M	Varchar	64	It is the country code. To be used for support of international numbers.
-------------	---	---------	----	--------------------------------------------------------------------------

**Return Parameters:**

Parameter Name	Description	Data Type	Length
ReturnCode	Return Code indicating the status	Number	4
ReturnMessage	Message description	Varchar	128
Request ID	Request ID of GV code issued	Varchar	20
Value	Status of GV issued	Varchar	32
IsOTP	Is OTP needed for redemption or no ( 1 means OTP needed 0 means not needed	Number	
GVCode	GV code	Varchar	
GVExpiryDate	GV Expiry Date	Varchar	11
GVValue	GV Amount	Number	

**Error Codes:**

Error Code	Description
0	Success
253	Member Id does not exist.
124	Security token verification failed.
107	Input parameters are incorrect. missing GV Code
107	Input parameters are incorrect. missing security token
107	Input parameters are incorrect. missing date
255	Invalid date
198	Store code does not exist.
398	Member ID is not registered
435	Not Same Member
441	Not same store
404	Required PIN

402	Invalid PIN
104	Do not have permissions to access this method.

**Sample:**

Type	JSON
Request	<pre>{   "SecurityToken":     "7FUYCvpq0+gEFD56ivjg44il07dZWNQQaFZ2c/IIWsE7h     wdhQz8XISdcDXniggq7iRTFRKA+0Cs7D6lwgzRLsB2gTYW     XOnZANijFZyE/eO6M24uoF0Wiyx13muxugJ61QNb6cA     QITU/oU5KfIR9w4YuXb3ongARO4THLRrf96IZa2Ugs7sVf     hQfYZA27/Jq",   "MemberID": "9634098384",   "GVCode": "OPKJLANI8A",   "Date": "10 Aug 2020",   "StoreCode": "DummyStore",   "Amount": "300",   "UserName": "vaibhav@easyrewardz.com" }</pre>
Response – Success	<pre>{   "ReturnCode": 0,   "RequestID": "002555",   "Value": "0.00",   "IsOTP": true,   "ReturnMessage": "Success",   "GVCode": "OPKJLANI8A",   "GVExpiryDate": "2021-08-10T23:59:00",   "GVValue": "300.00",   "AppliedOn": "" }</pre>
Response – Error	<pre>&lt;Response&gt; &lt;ReturnCode&gt;253&lt;/ReturnCo de&gt; &lt;ReturnMessage&gt;Member ID does not exist&lt;/ReturnMessage&gt; &lt;/Response&gt;</pre>

### 3.6.5. GV OTP

**Method Name and Signature:** wsGVOTP (Varchar strRequest)

**API URL (REST):** [{{BaseUrl}}/GVOTP](#)

**Method Description:** There are some GV which require OTP. Such GV codes return IsOTPRequired field value as TRUE in the wsRedeemGV method call. During the execution of wsRedeemGV method for such GVs, an OTP is triggered to the redeeming member & requires validation for application of the GV in the bill.

**Approach:** If GV requires OTP validation, this method needs to be called by POS to confirm the validity of Member. If the Customer has EPIN, EPIN will be passed in the OTP field.

**Best Implementation Practice:**

1. Resend GV OTP has to be used along with the API

**Pre-Condition:** wsRedeemGV method needs to be called before calling this method. Redeem GV will return a flag indicating if the GV requires OTP or not. Basis this field value, the POS needs to show a field to accept the OTP. Then, this method is invoked.

**Post-Condition:** Post successful validation of the GV, proceed to applying the associated discount & proceed to saving the bill. Post bill save call the wsUseGV method to mark the use of the coupon in the bill.

**Input Parameters:**

Field	Mandatory	Field Type	Field Length	Field Description
GVCode	M	Varchar	64	GV Code
RequestID	M	Varchar	32	Request ID returned by Redeem GV
SecurityToken	M	Varchar	512	It is the country code. To be used for support of international numbers.
UserName	M	Varchar	64	Username
OTP	M	Varchar	6	OTP received by User
CountryCode	M	Varchar	64	It is the country code. To be used for support of international numbers.

**Return Parameters:**

Parameter Name	Description	Data Type	Length
ReturnCode	Return Code indicating the status	Number	4
ReturnMessage	Message description	Varchar	128

**Error Codes:**

Error Code	Description
0	Success
378	OTP Required
365	Invalid OTP



Sample:

Type	JSON
Request	<pre>{   "SecurityToken":     "7FUYCvpq0+gEFD56ivjg44il07dZWNQQaFZ2c/IIWsE7h     wdhQz8XISdcDXniggq7iRTFRKA+0Cs7D6lwgzRLsB2gTYW     XOnZANijFZyE/eO6M24uoF0Wiyx13muxugJ61QNb6cA     QITU/oU5KfIR9w4YuXb3ongARO4THLRrf96lZa2Ugs7sVf     hQfYZA27/Jq",   "GVCode": "OPKJ331KZP",   "RequestID": "002556",   "UserName": "vaibhav@easyrewardz.com",   "OTP": "133769" }</pre>
Response	<pre>{   "ReturnCode": 0,   "OTP": "",   "ReturnMessage": "Success" }</pre>

### 3.6.6. Use GV

**Method Name and Signature:** wsUseGV (Varchar strRequest)

**API URL (REST):** [{{BaseUrl}}/UseGV](#)

**Method Description:** This method is used to confirm redemption of GV amount as applied to bill.

**Approach:** This method is invoked once bill is paid via other modes to payment to confirm redemption of the used GV amount.

**Best Implementation Practice:**

1. API has to be called post Bill Save

**Pre-Condition:** Redeem GV request should have been successfully invoked.

**Post-Condition:** GV amount will be deducted as applied.

**Input Parameters:**

Field	Mandatory	Field Type	Field Length	Field Description
MemberID	M	Varchar	32	Member ID
GVCode	M	Varchar	64	GV Code
Date	M	DateTime		Date on which GV is being Redeemed

<b>StoreCode</b>	M	Varchar	64	Store Code
<b>RequestID</b>	M	Varchar	32	Request ID as in response of redeem request
<b>OTP</b>	M/O	Varchar	6	OTP
<b>BillNo</b>	M	Varchar	64	Bill number on which GV is applied
<b>UserName</b>	M	Varchar	64	Username
<b>SecurityToken</b>	M	Varchar	512	This is a unique token Id used to authenticate the request.
<b>AppliedAmount</b>	M	Decimal		GV amount applied
<b>PaidAmount</b>	M	Decimal		Total amount
<b>CountryCode</b>	M	Varchar	64	Country code – Required for multi-country program

#### Return Parameters:

Parameter Name	Description	Data Type	Length
ReturnCode	Return Code indicating the status	Number	4
ReturnMessage	Message description	Varchar	128
GVCode	GV code issued	Varchar	20
GVStatus	Status of GV issued	Varchar	32
Available balance	Available balance of GV	Float	10

#### Error Codes:

Error Code	Description
0	Success
124	Security token verification failed.
107	Input parameters are incorrect. missing security token
107	Input parameters are incorrect. missing membered
107	Input parameters are incorrect. missing GV Code
107	Input parameters are incorrect. missing RequestID
107	Input parameters are incorrect. missing storecode
107	Input parameters are incorrect. missing date

107	Input parameters are incorrect. missing bill no
410	Invalid Applied Amount
431	Invalid Paid Amount
199	User does not exist.
107	Input parameters are incorrect. Missing username
423	Invalid Request ID
125	Input parameters are not provided correctly.
262	User name does not exist.
263	Store does not exist.
253	Member ID does not exist.
124	Security token verification failed.
104	Do not have permissions to access this method.
105	Security token expired.

**Sample:**

Typ e	JSON
Req uest	<pre>{   "SecurityToken":     "7FUYCvpq0+gEFD56ivjg44il07dZWNQqAFZ2c/IIWsE7hwdhQz8XISdcDXniggq7iRTFRKA+0Cs7D6lwg     zRLsB2gTYWXOnZANijFZyE/eO6M24uoF0Wiyx13muxugJ61QNb6cAQITU/oU5KfIR9w4YuXb3ongA     RO4THLRrf96lZa2Ugs7sVfhQfYZA27/Jq",   "RequestID": "002556",   "MemberID": "9634098384",   "GVCode": "OPKJ331KZP",   "StoreCode": "DummyStore",   "Date": "10 Aug 2020",   "BillNo": "Bill1111",   "AppliedAmount": "300",   "PaidAmount": "1000",   "OTP": "133769",   "UserName": "vaibhav@easyrewardz.com" }</pre>
Res pon se – Suc cess	<pre>{   "ReturnCode": 0,   "ReturnMessage": "Success",   "GVCode": "OPKJ331KZP",   "GVStatus": "Used",   "AvailableBalance": "0.00" }</pre>

### 3.6.7. Unblock GV

**Method Name and Signature:** wsUnBlockGV (Varchar strRequest)

**API URL (REST):** [{{BaseUrl}}/UnBlockGV](#)

**Method Description:** This method unblocks a blocked GV in case customer aborts a transaction.

**Approach:** Once this method is called and it is successful, GV block is removed and GV can then be re-applied to another bill/transaction.

**Best Implementation Practice:**

This is an important API to call in all events where GV Code redemption could not be completed due to any reason for example Bill could not be saved. Undo of First Step of Redemption

**Pre-Condition:** Redeem GV request should be invoked and successful.

**Post-Condition:** GV will be unblocked and can be re – applied to another bill.

**Input Parameters:**

Field	Mandatory	Field Type	Field Length	Field Description
SecurityToken	M	Varchar	512	This is a unique token Id used to authenticate the request.
Date	M	DateTime		Redemption date
MemberID	M	Varchar	32	Member id
GVCode	M	Varchar	64	GV Code
StoreCode	M	Varchar	64	Store code
PIN		Varchar	6	PIN if mandatory
Amount	M	Decimal		GV amount applied
OTP		Varchar	6	OTP if mandatory
UserName	M	Varchar	64	Username

**Return Parameters:**

Parameter Name	Description	Data Type	Length
ReturnCode	Return Code indicating the status	Number	4
ReturnMessage	Message description	Varchar	128
GVCode	GV code	Varchar	20

**Error Codes:**

Error Code	Description
0	Success
124	Security token verification failed
107	Input parameters are incorrect. missing security token
107	Input parameters are incorrect. missing request id
107	Input parameters are incorrect. missing GV Code
403	GVCode does not Exists
409	Request ID does not exist
125	Input parameters are not provided correctly.
262	User name does not exist.
263	Store does not exist.
253	Member ID does not exist.
124	Security token verification failed.
104	Do not have permissions to access this method.
105	Security token expired.

**Sample:**

Type	JSON
Request	<pre>{   "SecurityToken":   "7FUyCvpq0+gEFD56ivjg44il07dZWNQqAFZ2c/IIWsE7hwdhQz8XISdcDXniggq7iRTFR   KA+0Cs7D6lwgzRLsB2gTYWXOnZANijFZYe/eO6M24uoF0Wiyx13muxugJ61QNb6cAQ   ITU/oU5KflR9w4YuXb3ongARO4THLRf96IZa2Ugs7sVfhQfYZA27/Jq",   "GVCode": "OPKJS26AZO",   "MemberID": "9634098384",   "Date": "11 Aug 2020",   "StoreCode": "DummyStore",   "RequestID": "002557",   "Amount": "300",   "UserName": "vaibhav@easyrewardz.com" }</pre>
Response – Success	<pre>{   "ReturnCode": 0,   "GVCode": "OPKJS26AZO",   "ReturnMessage": "Success" }</pre>

### 3.6.8. Rollback GV

**Method Name and Signature:** wsRollbackGV (Varchar strRequest)

**API URL (REST):** [{{BaseUrl}}/RollbackGV](#)

**Method Description:** This method is used to rollback amount to GV.

**Approach:** This method is to be invoked if the bill to which GV is applied is cancelled or rolled back or returned.

**Best Implementation Practice:**

This is an API used to Rollback Use GV

**Pre-Condition:** Security token must be validated and Use GV should have been invoked.

**Post-Condition:** Amount is rolled back to GV.

**Input Parameters:**

Field	Mandatory	Field Type	Field Length	Field Description
Amount	M	Decimal		GV amount applied
SecurityToken	M	Varchar	64	This is a unique token Id used to authenticate the request.
Date	M	DateTime		Bill number on which GV is applied
GVCode	M	Varchar	64	GV Code
MemberID	M	Varchar	32	Member id
SecurityToken	M	Varchar	512	
StoreCode	M	Varchar	64	Store code
TransactionId	M	Varchar	64	
UserName	M	Varchar	64	Username
CountryCode	M	Varchar	64	Country Code

**Return Parameters:**

Parameter Name	Description	Data Type	Length
ReturnCode	Return Code indicating the status	Number	4
ReturnMessage	Message description	Varchar	128
GVCode	GV code issued	Varchar	20
GVStatus	Status of GV issued	Varchar	32
GVValue	Amount of GV	Float	10
GVExpiryDate	Expiry date of GV	DateTime	22

**Error Codes:**

Error Code	Description
0	Success

107	Input parameters are incorrect. missing member id
107	Input parameters are incorrect. missing GV Code
107	Input parameters are incorrect. missing security token
107	Input parameters are incorrect. missing date
107	Input parameters are incorrect. missing amount (or) invalid Amount
107	Input parameters are incorrect. missing store code
107	Input parameters are incorrect. missing username
200	User does not have permission under this store
253	Member ID does not exist.
104	Do not have permissions to access this method.
105	Security token expired.

#### Sample:

Type	JSON
Request	<pre>{   "SecurityToken":     "7FUyCvpq0+gEFD56ivjg44il07dZWNQqAFZ2c/IIWsE7hwdhQz8XISdcDXniggq7iRTFRKA+0Cs7     D6lwgzRLsB2gTYWXOnZANiiFZyE/eO6M24uoF0Wiyx13muxugJ61QNb6cAQITU/oU5KflR9w4     YuXb3ongARO4THLRrf96lZa2Ugs7sVfhQfYZA27/Jq",   "Amount": "200",   "CountryCode": "91",   "Date": "11 Aug 2020",   "GVCode": "TWIML5MGUC",   "MemberID": "9634098384",   "StoreCode": "DummyStore",   "TransactionId": "Bill222",   "UserName": "vaibhav@easyrewardz.com" }</pre>
Response – Success	<pre>{   "ReturnCode": 0,   "ReturnMessage": "Success",   "GVCode": "TWIML5MGUC",   "GVStatus": "Approved",   "GVValue": "200.00",   "GVExpiryDate": "10082021235900" }</pre>
Response – Error	

### 3.6.9. Update GV Status

**Method Name and Signature:** wsUpdateGVStatus (Varchar strRequest)

**API URL (REST):** [{{BaseUrl}}/UpdateGVStatus](#)

**Method Description:** This method is used to update the GV status.

**Approach:**

**Pre-Condition:** Security token must be validated.

**Post-Condition:** GV status is updated.

**Input Parameters:**

Field	Mandatory	Field Type	Field Length	Field Description
MemberID	M	Varchar	32	Member ID
StoreCode	M	Varchar	64	Store Code
GVCode	M	Varchar	64	GV Code
ToStatus	M	Varchar	32	Status to be updated for GV
BillNo		Varchar	64	Bill Number
AmountUsed	M/O	Decimal		Amount used from the specific GV Code
TotalPaidAmount	M	Decimal		Total Net payable of invoice
UserName	M	Varchar	64	Partner user name
SecurityToken	M	Varchar	512	This is a unique token Id used to authenticate the request.
CountryCode	M	Varchar	64	Country Code

**Return Parameters:**

Parameter Name	Description	Data Type	Length
ReturnCode	ReturnCode indicating the status	Number	4
ReturnMessage	Return Message	Varchar	512
MemberID	Member Id	Varchar	
GVOldStatus			
GVNewStatus			
GVOfferName			

**Sample:**

Type	JSON
Request	<pre> &lt;Request&gt;   &lt;MemberID&gt;9989066604&lt;/MemberID&gt;   &lt;StoreCode&gt;Test45&lt;/StoreCode&gt;   &lt;GVCode&gt;NOV10QI2E4PB&lt;/GVCode&gt;   &lt;ToStatus&gt;Issued&lt;/ToStatus&gt;   &lt;BillNo&gt;15Feb2018&lt;/BillNo&gt;   &lt;AmountUsed&gt;10&lt;/AmountUsed&gt;   &lt;TotalPaidAmount&gt;500&lt;/TotalPaidAmount&gt;   &lt;UserName&gt;saieshwar&lt;/UserName&gt;   &lt;SecurityToken&gt;qxTPzKcdINzbfWuh1jZ2PdUNMKr9k+R4w+ </pre>





	r7M2EVjgbp3VVY1vHo1jhTqm7j3fIET3winTrxWP6EO2R6qnaUd9DAzr9Y woOmy8GyQ6a7zC/pPH/agd3QY5mUIHcPvXv0LV+iFdNs13UR3u+d7FP5I meP8LnhA8YkmN5+Q/7OVdid9Hv8GRStUJ1LjhKktY1c6/lfKqI7twioOGbc UnAYmq2NfjLnF7UgdyLfXiZkyy0=</SecurityToken> <CountryCode>91</CountryCode> </Request>
Response – Success	<Response JSONNs:xsi="http://www.w3.org/2001/JSONSchema- instance" JSONNs:xsd="http://www.w3.org/2001/JSONSchema"> <ReturnCode>0</ReturnCode> <ReturnMessage>Success</ReturnMessage> <MemberId /> <GVOldStatus>Approved</GVOldStatus> <GVNewStatus>Expired</GVNewStatus> <GVOfferName>GV22Nov02</GVOfferName> </Response>

### 3.7. Accruals based on SKUs

This section will detail the APIs required for accruals based on SKU/Item.

#### 3.7.1. GetSKUBillDetails

**Method Name and Signature:** wsGetSKUDetails (Varchar strRequest)

**API URL (REST):** [{{BaseUrl}}/GetSKUBillDetails](#)

**Method Description:** There are some business where we want to do accrual on SKU basis. This API retrieves the data from ER database.

**Approach:** Use SaveSKUBillDetails to save data then retrieve the same.

**Pre-Condition:** SaveSKUBillDetails method needs to be called before calling this method.

**Post-Condition:** GV details are returned and can be shown or applied by the customer.

#### Input Parameters:

Parameter Name	Description	Data Type	Length
SecurityToken (*)	This is a unique token Id used to authenticate the request.	Varchar	512
UserName	Partner user	Varchar	128
BillNo	Bill Number	Varchar	32
BillDate	Bill Date	DateTime	

StoreCode	POS store Code	Varchar	64
CountryCode	Country Code		64

**Return Parameters:**

Parameter Name	Description	Data Type	Length
ReturnCode	Return Code indicating the status	Number	4
ReturnMessage	Message description	Varchar	128
BillNo	Bill Number	Varchar	32
StoreCode	Store Code	Varchar	64
Bill Value	Net Payable Amount	Number	10
Channel	Channel Code	Varchar	50
Department	Department of the Item	Varchar	128
Category	Category of the Item	Varchar	128
Group	Brand/SubCategory of the Item	Varchar	256
ItemType	Sale or return Item Type	Varchar	1
ItemQty	Qty	Number	5
Unit	Unit Price	Number	10
ItemDiscount	Discount on all quantities of the item	Number	4
ItemTax	Tax on all quantities of the item	Number	4
TotalPrice	Qty*Unit+Tax	Number	10
BilledPrice	Total Price – ItemDiscount	Number	10
ItemId	SKU code or barcode or Item ID	Varchar	128
TenderCode	Tender Code	Varchar	64
TenderID	Reference ID for the invoice	Varchar	64
TenderValue	Value paid via specific Tender Code	Number	10

**Error Codes:**

Error Code	Description
0	Success

**Sample:**

Type	JSON
Request	<pre>{   "SecurityToken":     "7FUYCvpq0+gEFD56ivjg44il07dZWNQQaFZ2c/IIWsE7     hwdhQz8XISdcDXniggq7iRTFRKA+0Cs7D6lwgzRLsB2gT     YWXOnZANijFZyE/eO6M24uoF0Wiyx13muxugJ61QNb     6cAQITU/oU5KflR9w4YuXb3ongARO4THLRf96lZa2Ugs     7sVfhQfYZA27/Jq",   "BillNo": "NewBill898",   "BillDate": "12082020",   "StoreCode": "DummyStore",   "UserName": "vaibhav@easyrewardz.com",   "CountryCode": "91" }</pre>
Response	<pre>{   "ReturnCode": "0",   "ReturnMessage": "Success",   "BillNo": "NewBill898",   "StoreCode": "DummyStore",   "BillValue": "1500",   "Channel": "Offline",   "CustomerType": "Loyalty",   "SKUOfferCode": null,   "EasyId": "9634098384",   "TransactionDate": "8/12/2020 12:00:00 AM",   "Status": "Paid",   "PointsRedeemed": "0",   "PointsValueRedeemed": null,   "TotalPointsAccrued": "60",   "TransactionItems": {     "TransactionItem": [</pre>



```
{
  "ItemType": "s",
  "ItemQty": "1",
  "Unit": "1500.00",
  "ItemDiscount": "100.00",
  "ItemId": "VAG1111",
  "TotalPrice": "1600.00",
  "BilledPrice": "1500.00",
  "Department": "Clothes",
  "Category": null,
  "Group": "CASUAL",
  "ItemTax": "100.00",
  "PointsAccrued": "15"
},
{
  "ItemType": "s",
  "ItemQty": "1",
  "Unit": "1500.00",
  "ItemDiscount": "100.00",
  "ItemId": "VAG1111",
  "TotalPrice": "1600.00",
  "BilledPrice": "1500.00",
  "Department": "Clothes",
  "Category": null,
  "Group": "CASUAL",
  "ItemTax": "100.00",
  "PointsAccrued": "15"
},
{
  "ItemType": "s",
  "ItemQty": "1",
```



```
"Unit": "1500.00",  
"ItemDiscount": "100.00",  
"ItemId": "VAG2222",  
"TotalPrice": "1600.00",  
"BilledPrice": "1500.00",  
"Department": "Clothes",  
"Category": null,  
"Group": null,  
"ItemTax": "100.00",  
"PointsAccrued": "15"  
},  
{  
  "ItemType": "s",  
  "ItemQty": "1",  
  "Unit": "1500.00",  
  "ItemDiscount": "100.00",  
  "ItemId": "VAG2222",  
  "TotalPrice": "1600.00",  
  "BilledPrice": "1500.00",  
  "Department": "Clothes",  
  "Category": null,  
  "Group": null,  
  "ItemTax": "100.00",  
  "PointsAccrued": "15"  
}  
]  
},  
"PaymentMode": {  
  "TenderItem": {  
    "TenderCode": "CASH",  
    "TenderID": "4",
```

	<pre>         "TenderValue": "2000.00"       }     },     "PrimaryOrderNumber": null   } </pre>
--	-------------------------------------------------------------------------------------------------

### 3.7.2. SaveSKUBillDetails

**Method Name and Signature:** wsSaveSKUBillDetails (Varchar strRequest)

**API URL (REST):** [{{BaseUrl}}/SaveSKUBillDetails](#)

**Method Description:** There are some business where we want to do accrual on SKU basis or to do only analytics. This API pushes the SKU level transaction data into ER database.

**Approach:** Use SaveSKUBillDetails to save data the same.

**Best Implementation Practice:**

- All Bill Section ( Header Bill Value, Net Payable of Items,Tenders Net Payable) should match
- In case of Return Old Sale, Bill Reference Bill Number is must
- ER support up to 2 decimal places in all number fields
- Easy ID field will be mobile number in case valid mobile number is available else it will be unique identifier of the program
- Channel Code and Customer Type Should be Configurable
- Special Varcharacters are not supported.

**Pre-Condition:** Tender type setup should exist.

**Post-Condition:** Use GetSKUBillDetails API to pull the data from ER database.

**Input Parameters:**

Parameter	M/O	Type	Length	Description
SecurityToken	M	Varchar	512	This is a unique token Id used to authenticate the request
UserName	M	Varchar	64	Partner user
BillNo	M	Varchar	64	Bill Number
TransactionDate	M	DateTime		DateTime
StoreCode	M	Varchar	64	POS store Code
EasyId	M	Varchar	32	EasyId of the member
Channel	O	Varchar	50	Channel

CustomerType	O	Varchar	128	Customer Type
BillValue	M	Number	128	Bill Value
SKUOfferCode	O	Varchar	256	SKU code or barcode or Item ID
ItemType	M	Varchar	1	Type of Item (S- Sale, R- Return)
ItemQty	M	Number	5	Quantity of Item
Unit	M	Number	10	Unit Price of Item
ItemDiscount	O	Number	4	Discount on Item
ItemTax	O	Number	4	Tax on One Item
TotalPrice	M	Number	10	Qty*Unit Price +Tax
BilledPrice	M	Number	10	Total Price -ItemDiscount
Department	M	Varchar	128	Department
Category	M	Varchar	128	Category
Group	M	Varchar	256	Group
ItemId	M	Varchar	128	SKU ID
TenderCode	M	Varchar	64	Tender Code
TenderID	O	Varchar	64	Tender ID
TenderValue	M	Number	10	Value of Tender
CountryCode	M	Varchar	3	Country Code
PointsRedeemed	O	Number		No of Points Redeemed in the Bill
PointsValueRedeemed	O	Number		Value of Points Redeemed in the Bill
RefBillNo	M/O	Varchar	64	This is the Bill Number to which item Belongs if it is getting returned
PrimaryOrderNumber	O	Varchar		This is the bill number

**Return Parameters:**

Parameter Name	Description	Data Type	Length
ReturnCode	Return Code indicating the status	Number	4
ReturnMessage	Message description	Varchar	128
TransactionId	Transaction Id	Integer	8

**Error Codes:**

Error Code	Description
0	Success

**Sample:**

Type	JSON
Request	<pre>{   "SecurityToken":     "7FUyCvpq0+gEFD56ivjg44il07dZWNQQaFZ2c/IIWsE7hwd     hQz8XISdcDXniggq7iRTFRKA+0Cs7D6lwgzRLsB2gTYWXOn     ZANiiJFZyE/eO6M24uoFOWiyx13muxugJ61QNb6cAQITU/o     U5KfIR9w4YuXb3ongARO4THLRrf96lZa2Ugs7sVfhQfYZA27     /Jq",   "StoreCode": "DummyStore",   "TransactionDate": "11 Aug 2020",   "BillNo": "NewBill897",   "EasyId": "9634098384",   "UserName": "vaibhav@easyrewardz.com",   "Channel": "Offline",   "CustomerType": "Loyalty",   "BillValue": "3000",   "PointsRedeemed": "",   "PointsValueRedeemed": "",   "PrimaryOrderNumber": "",   "SKUOfferCode": "",   "CountryCode": "91",   "TransactionItems": {     "TransactionItem": [       {         "ItemType": "s",         "ItemQty": "1",         "ItemName": "",         "Size": "",         "Unit": "1500",         "ItemDiscount": "100",         "ItemTax": "100",         "TotalPrice": "1600",         "BilledPrice": "1500",         "Department": "Clothes",         "Category": "Shirts",         "Group": "Casual",         "ItemId": "VAG1111",         "RefBillNo": ""       },       {         "ItemType": "s",         "ItemQty": "1",</pre>



	<pre> "ItemName": "", "Size": "", "Unit": "1500", "ItemDiscount": "100", "ItemTax": "100", "TotalPrice": "1600", "BilledPrice": "1500", "Department": "Clothes", "Category": "Shirts", "Group": "Formal", "ItemId": "VAG2222", "RefBillNo": ""     }   ] }, "PaymentMode": {   "TenderItem": [     {       "TenderCode": "Cash",       "TenderID": "",       "TenderValue": "2000"     },     {       "TenderCode": "Point",       "TenderID": "",       "TenderValue": "1000"     }   ] } } </pre>
Response	<pre> {   "ReturnCode": 0,   "ReturnMessage": "Success.",   "TransactionId": 3515 } </pre>

### 3.8. Ecommerce Customer Journey

This section describes about the APIs used to be the part of online Loyalty program.

#### 3.8.1. Customer Signup

**Method Name and Signature:** Customer Sign Up

**API URL (REST):** [{{BaseUrl}}/customerSignUp](#)

**Method Description:** This method is required to sign up a customer into the Loyalty program.

**Approach:**

**Best Implementation Practice:**

**Pre-Condition:**

**Post-Condition:**

**Input Parameters:**

Fields	Mandatory	Field Type	Field Length	Field Description
SecurityToken	M	Varchar	512	This is a unique token Id used to authenticate the request.
FirstName	O	Varchar	32	First Name of the member
LastName	O	Varchar	32	Last name of the member
DateOfBirth	O	DateTime		The date of birth of the member. It should be in (DD MMM YYYY) format
PinCode	O	Varchar	6	The address pin code of the member.
EmailId	O	Varchar	128	Email id of the member. Should be unique
MobileNo	M	Varchar	32	Enrolling Member's Mobile number
EasyPin	O	Varchar	12	It should be a numeric value between 6 to 12 digits. If not provided, it is auto generated and sent to member through SMS or Email based on alert settings for the member.
UserName	M	Varchar	64	Partner's User Name
Gender	O	Varchar	32	Gender of the Member (Male, Female, M or F)
MemberShipCardNumber	M/O	Varchar	64	Membership Card Number (if the program supports MemberShipCard)
ServicePersonNo	O	Varchar	64	Service Person Number
EmailSubscribe	O	Boolean	1	Flat to capture if customer has opted for Email notifications
SMSSubscribe	O	Boolean	1	Flat to capture if customer has opted for SMS notifications
TierExpiryDate	O	DateTime		Tier Expiry Date
StoreCode	M	Varchar	64	Store Code used to enroll member. Store Code must already be configured in LPaaS before calling this method.
Address1	O	Varchar	64	Address of the Member – 1
Address2	O	Varchar	64	Address of the Member – 2

<b>ReferralCode</b>	O	Varchar	64	Referral Code of the existing Member if the enrolling Member is being referred by an existing member. Referral points may be given to both the enrolling and referring members (based on Referral Offer configured in LPaaS).
<b>EasyId</b>	M	Varchar	32	Unique Identifier of Member
<b>TierName</b>	O	Varchar	64	Tier Name for Tier Based Program
<b>Remarks</b>	O	Varchar	512	Any other Remarks
<b>Anniversary Date</b>	O	DateTime		Anniversary Date
<b>CompanyName</b>	O	Varchar	64	Company Name
<b>PANNo</b>	O	Varchar	32	PAN Number
<b>ClientCustomerID</b>	M/O	Varchar	128	Unique identifier of member apart from mobile/membershipnumber
<b>EnrollDate</b>	O	DateTime		Enrollment Date
<b>CustomerTypeCode</b>	O	Varchar	32	It is the Customer Type of Member. Customer Type must be configured in LPaaS before passing in this field. Sample values for Customer Type code are Loyalty, NonLoyalty etc.
<b>ChannelCode</b>	O	Varchar	64	It is the Channel used for enrolling a Member. The channel must be configured in LPaaS before passing in this field. Sample values for channel are Online, Offline etc.
<b>AgeGroup</b>	O	Varchar	128	Age Group
<b>CountryCode</b>	M	Varchar	64	Country Code
<b>NumberOfChildren</b>	O			It is the count of the children
<b>EcommerceCustomer</b>	M	Boolean		Ecommerce channel sign up is decided by this flag
<b>marketingacceptance</b>	M	Varchar	128	Marketing Acceptance Flag
<b>termsandconditionsacceptance</b>	M	Varchar	128	T&C acceptance Flag
<b>Occupation</b>	O	Varchar	128	Occupation of the member
<b>RelationshipStatus</b>	O	Varchar	128	Marital Status value like Married, Single
<b>AddressId</b>	M	Varchar	128	ID of Address
<b>Addresstype</b>	M	Varchar	128	Address Type like Home, Office
<b>CustomerName</b>	O	Varchar	64	Name of the member
<b>Mobile</b>	M			Mobile no. of the member
<b>Pincode</b>	O	Varchar	64	The address pin code of the member



State	O	Varchar	64	State
City	O	Varchar	64	City name of the member
Landmark	O	Varchar	64	Landmark of the member
AddressLine1	O	Varchar	64	Primary address of the member
AddressLine2	O	Varchar	64	Secondary address of the member
DefaultAddress	O	Varchar	64	Default address of the member

**Return Parameters:**

Parameter Name	Description	Data Type	Length
ReturnCode	Return Code indicating the status	Number	4
SmsStatus			
EmailStatus			
TPEnrollStatus	Enroll Status in Third Party (1→ success, 0 → failure)	Number	1
EREnrollStatus	Enroll Status in EasyRewardz (1→ success, 0 → failure)	Number	1
ProgramStatus	Program Creation Status of Member Dashboard in EasyRewardz (1→ success, 0 → failure)	Number	1
BonusPoints	Bonus Points Earned (in case of Enroll Bonus)	Number	6
CCIPoints	CCI Points Earned (in case of CCI Policy Claim)		
ReturnMessage	Return Message	Varchar	512
EnrollStatus	Enrolment Status		
ProfileStatus	Profile Status		
LoginStatus	Login Status		
UpdateProfileFlagSet true	UpdateProfile Status		

**Sample:**

Type	JSON
Request	<pre>{   "SecurityToken":     "9ACbrNGc5k8jlrImNOvcsUon4D4b+w5RVl1p/mE5l6uYCeT44dz     B7E0Q3krnbbcjnWma29e57uR7rKcST7TCPAAAtlgS+GABC/IPKljHm     VFJlNy53yS6MJ1C+IdJhTUxxfvDbnEdWAueGBEMVSAOWaIXVss     3iKuu8JrKpCtkq0L5ykl8DYqBRy02igmK16ROFLorrXniMIYL839i/RZ     mw==",   "UserName": "apiuser@bataweb.com",   "StoreCode": "Website",   "FirstName": "Vaibhav",   "LastName": "Gaur",   "EmailId": "gaurvaibhav@gmail.com",   "Password": "dc647eb65e6711e155375218212b3964",</pre>



	<pre>"MobileNo": "9897362211", "DateOfBirth": "02 May 1994", "AnniversaryDate": "11 Nov 2020", "EcommerceCustomer": "true", "Gender": "M", "CountryCode": "91", "marketingacceptance": "true", "termsandconditionsacceptance": "true", "Occupation": "Software", "RelationshipStatus": "Married", "ReferralCode": "", "AddressId": "", "Addresstype": "", "CustomerName": "vaibhav", "Mobile": "9897362211", "Pincode": "122003", "State": "Haryana", "City": "Gurgaon", "Landmark": "Behind Manav Rachna School", "AddressLine1": "H no 2123", "AddressLine2": "Sector 34", "DefaultAddress": "True" }</pre>
<b>Response</b>	<pre>{ "ReturnCode": 0, "SmsStatus": 0, "EmailStatus": 0, "TPEnrollStatus": 1, "EREnrollStatus": 0, "ProgramStatus": -1, "BonusPoints": 22.0, "CCIPoints": 0.0, "ReturnMessage": "Success", "EnrollStatus": 1, "ProfileStatus": -1, &gt;LoginStatus": 0, "UpdateProfileFlagSet": true }</pre>

### 3.8.2. Customer Login

**Method Name and Signature:** Customer Login

**API URL (REST):** [{{BaseUrl}}/CustomerLogin](#)

**Method Description:** This method is required to login a customer into a Loyalty program.

**Approach:**

**Best Implementation Practice:**

**Pre-Condition:**

**Post-Condition:**

**Input Parameters:**

Field	Mandatory	Field Type	Field Length	Field Description
SecurityToken	M	Varchar	512	This is a unique token Id used to authenticate the request.
UserName	M	Varchar	64	Partner User Name
MemberId	M	Varchar	32	MemberId (Mobile, Email ID)
StoreCode	M	Varchar	64	Store Code
Password	M	Varchar	32	Password
CountryCode	M	Varchar	64	Country Code

**Return Parameters:**

Parameter Name	Description	Data Type	Length
ReturnCode	ReturnCode indicating the status	Number	4
ReturnMessage	Return Message	Varchar	512
PasswordMatch	If password matches it returns true else false	Boolean	
TemporaryPassword	If login is happening via temporary password, it returns true else false	Boolean	
MobileNo	Mobile No	Varchar	32

**Sample:**

Type	JSON
Request	{



	<pre> "SecurityToken": "9ACbrNGc5k8jlrImNOvcsUon4D4b+w5RVI1p/mE5l6uYCeT44dzB7E0Q3krn bbcjnWma29e57uR7rKcST7TCPAAAtlgS+GABC/IPKljHmVFJNlny53yS6MJ1C+Id JhTUxxfvDbnEdWAueGBEMVSAOWalXVss3iKuu8JrKpCTkq0L5yKl8DYqBRy02i gmK16ROFLorrXniMIYL839i/RZmw==",  "UserName": "apiuser@bataweb.com",  "MemberId": "gaurvaibhav@gmail.com",  "StoreCode": "Website",  "Password": "dc647eb65e6711e155375218212b3964",  "Countrycode": "91"  } </pre>
<b>Response</b>	<pre> {  "ReturnCode": 0,  "ReturnMessage": "Success",  "PasswordMatch": "True",  "TemporaryPassword": false,  "MobileNo": "9897362211"  } </pre>

### 3.8.3. Customer Profile

**Method Name and Signature:** Customer Profile

**API URL (REST):** [{{BaseUrl}}/CustomerProfile](#)

**Method Description:** This method is used to fetch the customer profile details

**Approach:**

**Best Implementation Practice:**

**Pre-Condition:**

**Post-Condition:**

**Input Parameters:**

Parameter Name	Mandatory/Optional	Data Type	Length	Description
SecurityToken	M	Varchar	512	This is a unique token Id used to

				authenticate the request.
UserName	M	Varchar	64	This is access CRM User name
StoreCode	M	Varchar	64	Physical outlet ID
EasyId	M	Varchar	32	Unique Member Identifier
EmailId	O	Varchar	128	Email ID
MobileNo	M	Varchar	32	Mobile
CountryCode	M	Varchar	64	Country Code

#### Return Parameters:

Parameter Name	Description	Data Type	Length
ReturnCode	Return Code indicating the status	Number	4
ReturnMessage	Return Message	Varchar	512
FirstName	First Name of the member	Varchar	50
LastName	Last name of the member	Varchar	50
Email	Email of the member	Varchar	128
Mobile	Mobile number of the member	Number	10
ClientID	Client ID is the unique identifier of the Member in Partner system	Varchar	128
DateOfBirth	DateOfBirth of the member	Varchar	11
MembershipCardNumber	MembershipCardNumber of the member	Varchar	16
Address1	Address of the Member – 1	Varchar	64
Address2	Address of the Member – 2	Varchar	64
Gender	Gender	Varchar	10
CustomerType	Customer Type of the Member	Varchar	128
ReferralCode	Referral Code of the Member		
EmailSubscribe	Email Subscription Flag	Boolean	
SmsSubscribe	SMS Subscription Flag	Boolean	
AvailablePoints	Points available to a Member	Number	





## Sample:

Type	JSON
Request	<pre>{   "SecurityToken": "9ACbrNGc5k8jlrImNOvcsUon4D4b+w5RVI1p/mE5l6uYCeT44dzB7E 0Q3krnbbcjnWma29e57uR7rKcST7TCPAAtlgS+GABC/IPKljHmVFJNln y53yS6MJ1C+IdJhTUxxfvDbnEdWAueGBEMVSAOWalXVss3iKuu8JrK PcTkq0L5ykl8DYqBRy02igmK16ROFLorrXniMIYL839i/RZmw==",   "UserName": "apiuser@bataweb.com",   "StoreCode": "Website",   "EasyId": "9897362211",   "EmailId": "gaurvaibhav@gmail.com",   "MobileNo": "9897362211",   "CountryCode": "91" }</pre>
Response	<pre>{   "ReturnCode": 0,   "ReturnMessage": "Success",   "FirstName": "Ashu",   "LastName": "Sharma",   "Email": "gaurvaibhav@gmail.com",   "Mobile": "9897362211",   "ClientID": "",   "DateOfBirth": "11 Jun 1995",   "MembershipCardNumber": "",   "Address1": "",   "Address2": "",   "Gender": "Male",   "CustomerType": "Loyalty",   "ReferralCode": "SFRH308V3Y",   "ProfileStatus": "T", }</pre>

```

"EmailSubscribe": true,
"SmsSubscribe": true,
"AvailablePoints": 22.0000,
"NumberOfChildren": 2
}

```

### 3.8.4. Forgot Password

**Method Name and Signature:** Forgot Password

**API URL (REST):** [{{BaseUrl}}/forgotPassword](#)

**Method Description:** This method is required to reset the customer password.

**Approach:**

**Best Implementation Practice:**

**Pre-Condition:**

**Post-Condition:**

**Input Parameters:**

Parameter Name	Description	Data Type	Length
SecurityToken	M	Varchar	512
UserName	M	Varchar	64
StoreCode	M	Varchar	64
EmailId	O	Varchar	128
MobileNo	M	Varchar	32
CountryCode	M	Varchar	64

**Return Parameters:**

Parameter Name	Description	Data Type	Length
ReturnCode	ReturnCode indicating the status	Number	4
ReturnMessage	Return Message	Varchar	512

**Error Codes:**

**Sample:**



Type	JSON
Request	<pre>{    "SecurityToken":   "9ACbrNGc5k8jlrImNOvcsUon4D4b+w5RVI1p/mE5l6uYCeT44dz   B7E0Q3krnbbcjnWma29e57uR7rKcST7TCPAAAtlgS+GABC/IPKljHm   VFJNlny53yS6MJ1C+IdJhTUxxfvDbnEdWAueGBEMVSAOWalXVss   3iKuu8JrKPcTkq0L5ykl8DYqBRy02igmK16ROFLorrXniMIYL839i/RZ   mw==",    "UserName": "apisuer@bataweb.com",    "StoreCode": "Website",    "EmailId": "gaurvaibhav@gmail.com",    "MobileNo": "9897362211",    "Countrycode": "91" }</pre>
Response	<pre>{    "ReturnCode": 0,    "ReturnMessage": "Success" }</pre>

### 3.8.5. Update Password

**Method Name and Signature:** updatePassword

**API URL (REST):** [{{BaseUrl}}/UpdatePassword](#)

**Method Description:** This method is required to update the password.

**Approach:**

**Best Implementation Practice:**

**Pre-Condition:**

**Post-Condition:**

**Input Parameters:**

Parameter Name	Mandatory	Data Type	Length	Description
----------------	-----------	-----------	--------	-------------

SecurityToken	M	Varchar	512	This is a unique token Id used to authenticate the request.
UserName	M	Varchar	64	CRM User Name
StoreCode	M	Varchar	64	Physical Outlet ID
EmailId	M	Varchar	32	Email ID
MobileNo	M	Varchar	32	Mobile No
CurrentPassword	M	Varchar	32	Existing Password
NewPassword	M	Varchar	32	New Password
CountryCode	M	Varchar	64	Country Code

**Return Parameters:**

Parameter Name	Description	Data Type	Length
ReturnCode	ReturnCode indicating the status	Number	4
ReturnMessage	Return Message	Varchar	512

**Sample:**

Type	JSON
Request	<pre>{   "SecurityToken":     "9ACbrNGc5k8jlrImNOvcsUon4D4b+w5RV1p/mE5l6uYCeT44dzB     7E0Q3krnbbcjnWma29e57uR7rKcST7TCPAAAtlgS+GABC/IPKljHmVFJ     Nlny53yS6MJ1C+IdJhTUxxfvDbnEdWAueGBEMVSAOWalXVss3iKuu     8JrKPcTkq0L5ykl8DYqBRy02igmK16ROFLorrXniMIYL839i/RZmw=="   ,   "UserName": "apiuser@bataweb.com",   "StoreCode": "Website",   "EmailId": "gaurvaibhav@gmail.com",   "MobileNo": "9897362211",   "CurrentPassword": "dc647eb65e6711e155375218212b3964",   "NewPassword": "70c04e53f3216a44d9b4f3e517f9b45c",   "CountryCode": "91"</pre>



	}
Response	{  "ReturnCode": 0,  "ReturnMessage": "Success"  }

### 3.8.6. Member Address Detail

**Method Name and Signature:** Member Address Details

**API URL (REST):** [{{BaseUrl}}/memberAddressDetail](#)

**Method Description:** This method is required to store the member address details

**Approach:**

**Best Implementation Practice:**

**Pre-Condition:**

**Post-Condition:**

**Input Parameters:**

Parameter Name	Mandatory	Data Type	Length	Description
SecurityToken	M	Varchar	512	This is a unique token Id used to authenticate the request.
UserName	M	Varchar	64	CRM User Name
StoreCode	M	Varchar	64	Physical Outlet ID
MemberId	M	Varchar	64	Member Unique Identifier
AddressId	M	Varchar	64	Address ID
Addresstype	O	Varchar	64	Address Type
CustomerName	M	Varchar	64	Customer Name
CountryCode	M	Varchar	64	Country Code
Mobile	M	Varchar	64	Mobile
Pincode	O	Varchar	64	Pin Code



Country	O	Varchar	64	Country
State	O	Varchar	64	State
City	O	Varchar	64	City
Landmark	O	Varchar	64	Landmark
AddressLine1	O	Varchar	64	Address Line 1
AddressLine2	O	Varchar	64	Address Line 2
DefaultAddress	O	Varchar	64	Default Address flag

**Return Parameters:**

Parameter Name	Description	Data Type	Length
ReturnCode	ReturnCode indicating the status	Number	4
ReturnMessage	Return Message	Varchar	512

**Sample:**

Type	JSON
Request	<pre>{   "SecurityToken":     "9ACbrNGc5k8jlrImNOvcsUon4D4b+w5RVI1p/mE5l6uYCeT44dzB     7E0Q3krnbbcjnWma29e57uR7rKcST7TCPAAAtlgS+GABC/IPKljHmVFJ     Nlny53yS6MJ1C+IdJhTUxxfvDbnEdWAueGBEMVSAOWalXVss3iKuu     8JrKPcTkq0L5ykl8DYqBRy02igmK16ROFLorrXniMIYL839i/RZmw=="   ,   "UserName": "apiuser@bataweb.com",   "StoreCode": "Website",   "MemberId": "gaurvaibhav@gmail.com",   "CountryCode": "91",   "SavedAddresses": {     "AddressId": [],     "Addresstype": "Home 1",     "CustomerName": "Ashu",     "CountryCode": "91",     "Mobile": "9897362211",   } }</pre>



	<pre>"Pincode": "122004", "Country": "", "State": "Haryana", "City": "Gurgaon", "Landmark": "MG Road", "AddressLine1": "Lane 1, Round turn 2", "AddressLine2": "Lane 1, Round turn 1", "DefaultAddress": "True" }</pre>
<b>Response</b>	<pre>{ "ReturnCode": 0, "ReturnMessage": "Success" }</pre>

### 3.8.7. Get Address Detail

**Method Name and Signature:** Get Address Details

**API URL (REST):** [{{BaseUrl}}/getAddressDetail](#)

**Method Description:** This method is used to get all the address details of the customer.

**Approach:**

**Best Implementation Practice:**

**Pre-Condition:**

**Post-Condition:**

**Input Parameters:**

Parameter Name	Mandatory	Data Type	Length	Description
SecurityToken	M	Varchar	512	Token
StoreCode	M	Varchar	64	Store Code
EmailId	M	Varchar	64	Email ID
MobileNo	M	Varchar	64	Mobile No
CountryCode	M	Varchar	64	Country Code

**Return Parameters:**



Parameter Name	Description	Data Type	Length
AddressId	Address ID	Varchar	64
Addresstype	Address Type	Varchar	64
CustomerName	Customer Name	Varchar	64
CountryCode	Country Code	Varchar	64
Mobile	Mobile	Varchar	64
Pincode	Pin Code	Varchar	64
Country	Country	Varchar	64
State	State	Varchar	64
City	City	Varchar	64
Landmark	Landmark	Varchar	64
AddressLine1	Address Line 1	Varchar	64
AddressLine2	Address Line 2	Varchar	64
DefaultAddress	Default Address flag	Varchar	64

**Sample:**

Type	JSON
Request	<pre>{   "SecurityToken":   "9ACbrNGc5k8jlrImNOvcsUon4D4b+w5RVl1p/mE5l6uYCeT44dzB   7E0Q3krnbbcjnWma29e57uR7rKcST7TCPAAAtlgS+GABC/IPKljHmVFJ   Nlny53yS6MJ1C+IdJhTUxxfvDbnEdWAueGBEMVSAOWalXVss3iKuu   8JrKpCtkq0L5ykl8DYqBRy02igmK16ROFLorrXniMIYL839i/RZmw=="   ,   "UserName": "apiuser@bataweb.com",   "StoreCode": "Website",   "EmailId": "gaurvaibhav@gmail.com",   "MobileNo": "9897362211",   "CountryCode": "91" }</pre>
Response	<pre>{   "ReturnCode": "0",</pre>





```
"ReturnMessage": "Success",  
"SavedAddresses": [  
  {  
    "AddressId": 16724,  
    "Addresstype": "",  
    "CustomerName": "",  
    "CountryCode": "",  
    "Mobile": "9811159888",  
    "Pincode": "305801",  
    "Country": "",  
    "State": "Rajasthan",  
    "City": "Ajmer",  
    "Landmark": "MG Road",  
    "AddressLine1": "AL1",  
    "AddressLine2": "AL2",  
    "DefaultAddress": false  
  }  
]  
}
```

### 3.8.8. Save SKU Bill Details

**Method Name and Signature:** SaveSKUBillDetails

**API URL (REST):** [{{BaseUrl}}/saveSKUBilldetails](#)

**Method Description:** This method is used to push the SKU level details into OLTP System

**Approach:**

**Best Implementation Practice:**

**Pre-Condition:**

**Post-Condition:**

**Input Parameters:**



Parameter Name	Description	Data Type	Length	Description
SecurityToken	M	Varchar	512	This is a unique token Id used to authenticate the request.
StoreCode	M	Varchar	64	Virtual Store code for Ecommerce
TransactionDate	O	DateTime		Order Date
BillNo	M	Varchar	64	Order Number
EasyId	M	Varchar	32	Customer Unique Identifier
UserName	M	Varchar	64	CRM User Name
Channel	O	Varchar	50	Channel of the data
CustomerType	O	Varchar	128	Customer Type
BillValue	M	Number	128	Net Payable
PreDelivery	M	Varchar	128	Flag to define Ecommerce Order Value will be <b>true</b> in case of ecommerce
PointsRedeemed	O	Varchar	128	Points Redeemed
PointsValueRedeemed	O	Varchar	128	Cash Value of Points redeemed
PrimaryOrderNumber	O	Varchar	128	Order number
ShippingCharge	O	Number		Shipping Charges in Amount
SKUOfferCode	O	Varchar	256	SKU Offer
CountryCode	M	Varchar	3	Country code
ItemType	M	Varchar	1	Type of Item Sale , S for sale and R for return
ItemQty	M	Number	5	Quantity of Item
Unit	M	Number	10	Unit price of single quantity
ItemDiscount	O	Number	4	Discount of all quantities of Item
ItemTax	O	Number	4	Discount on all quantities of Item
TotalPrice	M	Number	10	Qty*Unit+ItemTax
BilledPrice	M	Number	10	TotalPrice-ItemDiscount
Department	M	Varchar	128	Department
Category	M	Varchar	128	Category
Group	M	Varchar	256	Sub Category/Brand of Item
ItemId	M	Varchar	128	Unique Product Code
RefBillNo	M/O	Varchar	64	If item is returning ,Reference of bill

				number where item was sold
TenderCode	M	Varchar	64	Tender Code
TenderID	O	Varchar	64	Tender ID
TenderValue	M	Number	10	Amount paid via specific Tender

#### Return Parameters:

Parameter Name	Description	Data Type	Length
ReturnCode	ReturnCode indicating the status	Number	
ReturnMessage	Return Message	Varchar	512
OrderLevelCredits	Provisional Points at Order level	Number	

#### Error Codes:

Error Code	Description
0	Success
125	Input parameters are not provided correctly.
247	Input parameter Membership card number missed.
186	Membership card number verification process failed.
187	Unable to generate membership card no. please try after some time.
229	Invalid Policy
200	User does not have permission under this store.
199	User does not exist.
222	Unauthenticated user.
223	Unauthenticated store.
224	Policy already claimed.
227	Your Policy has been deactivated.
225	Your Policy has expired.
228	Policy not started yet.
229	Invalid Policy

285	Offer does not exist.
198	Store code does not exist
315	Failed to update membership card details.
318	Failed to accrue points to member.
316	Failed to insert membership card details.
317	Membership card not exists.
261	Bill id already exists.
129	Cannot accrue as wallet does not have enough easyPoints.
264	Activity does not exist
262	User name does not exist.
263	Store does not exist.
299	Failed to register.
212	Member already exists.
124	Security token verification failed.
104	Do not have permissions to access this method.
105	Security token expired.
139	Cannot accrue, please try later
268	Invalid mobile number.
269	Invalid email id.
270	Invalid pincode.
271	Invalid date of birth.
273	Invalid lastname.
272	Invalid firstname.
274	Invalid membership card no.
275	Invalid issue card.



276	Invalid card value.
267	Easy pin should be between 6-12 letters and should include 1 number and 1 alpha Varcharacter.
207	Email id already exists.
139	Cannot accrue, please try later

**Sample:**

Type	JSON
Request	<pre>{   "SecurityToken":     "9ACbrNGc5k8jlrImNOvcsUon4D4b+w5RVI1p/mE5l6uYCeT44dzB     7E0Q3krnbbcjnWma29e57uR7rKcST7TCPAAAtlgS+GABC/IPKljHmVFJ     Nlny53yS6MJ1C+IdJhTUxxfvDbnEdWAueGBEMVSAOWalXVss3iKuu     8JrKPcTkq0L5ykl8DYqBRy02igmK16ROFLorrXniMIYL839i/RZmw=="   ,   "StoreCode": "Website",   "TransactionDate": "07 Jan 2021 10:27:00.0",   "BillNo": "O2",   "EasyId": "9811159888",   "UserName": "apiuser@bataweb.com",   "Channel": "",   "CustomerType": "",   "BillValue": "1700",   "PreDelivery": "true",   "PointsRedeemed": "",   "PointsValueRedeemed": "",   "PrimaryOrderNumber": "O2",   "SKUOfferCode": "",   "CountryCode": "91",   "TransactionItems": {     "TransactionItem": [</pre>



```
{  
  "ItemType": "s",  
  "ItemQty": "1",  
  "Unit": "900",  
  "ItemDiscount": "500",  
  "ItemTax": "500",  
  "TotalPrice": "1400",  
  "BilledPrice": "900",  
  "Department": "D1",  
  "Category": "C1",  
  "Group": "G1",  
  "ItemId": "Id2",  
  "RefBillNo": ""  
},  
{  
  "ItemType": "s",  
  "ItemQty": "1",  
  "Unit": "800",  
  "ItemDiscount": "",  
  "ItemTax": "",  
  "TotalPrice": "800",  
  "BilledPrice": "800",  
  "Department": "D1",  
  "Category": "C1",  
  "Group": "G1",  
  "ItemId": "Id1",  
  "RefBillNo": ""  
}  
]  
,
```



	<pre> "PaymentMode": {   "TenderItem": [     {       "TenderCode": "GV",       "TenderID": "",       "TenderValue": "1000"     },     {       "TenderCode": "Cash",       "TenderID": "",       "TenderValue": "700"     }   ] } </pre>
<b>Response</b>	<pre> {"ReturnCode":0,"ReturnMessage":"Success.,"OrderLevelCredits":12} </pre>

### 3.8.9. Lookup

**Method Name and Signature:** Lookup

**API URL (REST):** [{{BaseUrl}}/Lookup](#)

**Method Description:** The purpose of the method is to display provisional points at item level

**Approach:**

**Best Implementation Practice:**

**Pre-Condition:**

**Post-Condition:**

**Input Parameters:**

Parameter Name	Mandatory	Description	Data Type	Length
SecurityToken	M	Token	Varchar	512
UserName	M	CRM User Name	Varchar	64
StoreCode	M	Outlet ID	Varchar	64
MemberId	M	Member unique identifier	Varchar	64

TransactionDate	M	Order Date	DateTime	
PrimaryOrderNumber	M	Order Number	Varchar	64
Invoiceld	O	Invoice ID	Varchar	64
CountryCode	M	Country Code	Varchar	64

#### Return Parameters:

Parameter Name	Description	Data Type	Length
ReturnCode	ReturnCode indicating the status	Number	4
ReturnMessage	Return Message	Varchar	512
PrimaryOrderNumber	Order Number	Varchar	
OrderLevelCredits	Number of Points credited at order level	Number	
ItemId	Item ID	Varchar	128
LoyaltyPoints	Points accrued at Item level	Number	

#### Sample:

Type	JSON
Request	<pre>{   "SecurityToken":     "9ACbrNGc5k8jlrImNOvcsUon4D4b+w5RV1p/mE5l6uYCeT44dz     B7E0Q3krnbbcjnWma29e57uR7rKcST7TCPAAAtlgS+GABC/IPKljHm     VFJNlny53yS6MJ1C+IdJhTUxxfvDbnEdWAueGBEMVSAOWalXVss     3iKuu8JrKPcTkq0L5ykl8DYqBRy02igmK16ROFLorrXniMIYL839i/RZ     mw==",   "UserName": "apiuser@bataweb.com",   "StoreCode": "Website",   "MemberId": "9811159888",   "TransactionDate": "07 Jan 2021",   "PrimaryOrderNumber": "",   "Invoiceld": "",   "CountryCode": "91" }</pre>
Response	<pre>{"Response":{"ReturnCode":"0","ReturnMessage":"Success","Or ders":{"PrimaryOrderNumber":"O1","OrderLevelCredits":"12.00</pre>



```
00","OrderItemLevelPoints":{"ItemId":"Id2","LoyaltyPoints":"4.0000"},"{"ItemId":"Id1","LoyaltyPoints":"8.0000"}},{"PrimaryOrderNumber":"O2","OrderLevelCredits":"4.0000","OrderItemLevelPoints":{"ItemId":"Id2","LoyaltyPoints":"4.0000"}]]}]}
```

### 3.8.10. Order Status Update

**Method Name and Signature:** OrderStatusUpdate

**API URL (REST):** [{{BaseUrl}}/OrderStatusUpdate](#)

**Method Description:** This method is used to update Item delivery status (Cancelled, Delivered and Returned). The Returned status is passed post delivered status only

**Approach:**

**Best Implementation Practice:**

**Pre-Condition:**

**Post-Condition:**

**Input Parameters:**

Parameter Name	Mandatory	Description	Data Type	Length
SecurityToken	M	Token	Varchar	512
StoreCode	M	Outlet ID	Varchar	64
MemberId	M	Unique Member ID	Varchar	
UserName	M	CRM User Name	Varchar	64
TransactionDate	M	Order Date	DateTime	
PrimaryOrderNumber	M	Order Number	OrderNumber	64
OrderStatus	O	Order Status	Varchar	
CountryCode	M	Country Code	Varchar	64
InvoiceId	M	Invoice ID (Unique ID to identify item delivery journey)	Varchar	64
Date	M	Date of Item delivery status (Cancelled,Delivered and Returned) Change	DateTime	
ItemStatus	M	Item Status (Cancelled,Delivered and Returned)	Varchar	64
ItemQty	M	Quantity of Item	Number	
ItemId	M	Item SKU ID	Varchar	128

**Return Parameters:**



Parameter Name	Description	Data Type	Length
ReturnCode	ReturnCode indicating the status	Number	4
ReturnMessage	Return Message	Varchar	512

**Sample:**

Type	JSON
Request	<pre>{   "SecurityToken":     "9ACbrNGc5k8jlrImNOvcsUon4D4b+w5RVI1p/mE5l6uYCeT44dz     B7E0Q3krnbbcjnWma29e57uR7rKcST7TCPAAAtlgS+GABC/IPKljHm     VFJNlNy53yS6MJ1C+IdJhTUxxfvDbnEdWAueGBEMVSAOWalXVss     3iKuu8JrKPcTkq0L5ykl8DYqBRy02igmK16ROFLorrXniMIYL839i/RZ     mw==",   "StoreCode": "Website",   "MemberId": "9811159888",   "UserName": "apiuser@bataweb.com",   "TransactionDate": "07 Jan 2021",   "PrimaryOrderNumber": "O1",   "OrderStatus": "",   "CountryCode": "91",   "TransactionItems": {     "TransactionItem": {       "InvoiceId": "N1",       "Date": "07 Jan 2021",       "ItemStatus": "Delivered",       "ItemQty": "1.0",       "ItemId": "Id1"     }   } }</pre>
Response	<pre>{</pre>

```
"ReturnCode":0,
"ReturnMessage":"Success",
"AllTransactionItems":null
}
```

## 3.9. Miscellaneous

### 3.9.1. Easy Points Refund

**Method Name and Signature:** wsEasyPointRefund (Varchar strRequest)

**API URL (REST):** [{{BaseUrl}}/EasyPointRefund](#)

**Method Description:** This method refunds a transaction recorded in LPaaS. This transaction can be an accrual or redemption transaction.

**Approach:** The refund transaction will always be another entry in the member's account showing both the transactions in his/her statement; accrual transaction showing earning of program points & refund transaction showing deduction of program points from member's account.

**Best Implementation Practice:**

- POS has to read API response to find out discount amount in case of returned bill
- This API is used for complete bill refund

**Pre-Condition:** Transaction that needs to be refunded should exist in order for refund to take place.

**Post-Condition:** Points are debited from member's account for cash value returned or if member paid by point then points will be credited to member's account.

**Input Parameters:**

Field	Mandatory	Field Type	Field Length	Field Description
EasyId	M	Varchar	32	EasyId of the member.
SecurityToken	M	Varchar	512	This is a unique token Id used to authenticate the request.
StoreCode	M	Varchar	64	Store Code
TransactionCode	M	Varchar	64	Transaction Code (Bill Number) which is to be refunded
TransactionDate	M	DateTime		Transaction Date (DD MMM YYYY) of transaction which is to be refunded
RefundDate	M	DateTime		The date of refund transaction (DD MMM YYYY)

<b>RefundTransactionId</b>	M	Varchar	64	This will be a reference id for a refund transaction at vendor's end.
<b>RefundAmount</b>	M/O	Decimal		Amount related to transaction to refunded
<b>RefundEasyPoints</b>	M/O	decimal		EasyPoints collected or spent in the transaction to refunded
<b>TransactionDescription</b>	O	Varchar	128	This will be used to save the refund transaction details.
<b>UserName</b>	M	Varchar	64	Partner User Name
<b>CountryCode</b>	M	Varchar	64	Country Code

#### Return Parameters:

Parameter Name	Description	Data Type	Length
ReturnCode	ReturnCode indicating the status	Number	4
RefundId	RefundId for reference	Varchar	64
PointsDeducted	Points refunded	Number	
ReturnMessage	Return Message	Varchar	512

#### Error Codes:

Errors Code	Description
0	Success
253	Member ID does not exist.
261	Bill id already exists.
289	Customer does not have enough EasyPoints to redeem.
122	Invalid bill id.
125	Input parameters are not provided correctly.
293	Rollback is already done for this bill id.
294	Refund points are more than accrued/redeemed points.
295	Bill id does not exist.
296	Refund amount is not equal to billed amount.
297	No transaction has made to the customer.

189	Invalid store.
262	User name does not exist.
124	Security token verification failed.
104	Do not have permissions to access this method.
105	Security token expired.
301	Points are mandatory for activity-based accrual.
302	Cannot accrue EasyPoints as the amount is less.
170	Offer does not exist in this range.
292	Amount is required for this transaction.
129	Cannot accrue as wallet does not have enough easyPoints.
264	Activity does not exist.
285	Offer does not exist.
263	Store does not exist.
265	Invalid customer id.
255	Invalid date.
298	Refund date should be current date.
320	Invalid Price.
321	Invalid points.

**Sample:**

Type	JSON
Request	<pre>{   "SecurityToken":     "7FUYCvpq0+gEFD56ivjg44il07dZWNQQaFZ2c/IIWsE7hwdhQz8XISdcDXni     ggq7iRTFRKA+OCs7D6lwgzRLsB2gTYWXOnZANijFZyE/eO6M24uoF0Wiyx1     3muxugJ61QNb6cAQITU/oU5KflR9w4YuXb3ongARO4THLRrf96IZa2Ugs7s     VfhQfYZA27/Jq",   "EasyId": "9634098384",</pre>

	<pre> "StoreCode": "DummyStore", "UserName": "vaibhav@easyrewardz.com", "TransactionCode": "NewBill2266", "TransactionDate": "10 Aug 2020", "RefundDate": "10 Aug 2020", "RefundTransactionId": "Bill6512", "RefundAmount": "3000", "RefundEasyPoints": "1000", "TransactionDescription": "", "CountryCode": "91" } </pre>
Response	<pre> {   "ReturnCode": "0",   "RefundId": "0",   "PointsDeducted": "30" } </pre>

### 3.9.2. Get Tier Details

**Method Name and Signature:** wsGetTierDetails (Varchar strRequest)

**API URL (REST):** [{{BaseUrl}}/GetTierDetails](#)

**Method Description:** This method is used to get the Tier details of the customer.

**Approach:**

**Pre-Condition:**

**Post-Condition:**

**Input Parameters:**

Field	Mandatory	Field Type	Field Length	Field Description
SecurityToken	M	Varchar	512	This is a unique token Id used to authenticate the request.
EasyId	M	Varchar	32	EasyId of the member
UserName	M	Varchar	64	Partner user

**Return Parameters:**

Parameter Name	Description	Data Type	Length
ReturnCode	Return Code indicating the status	Number	4
ReturnMessage	Return Message	Varchar	512
TierName	Tier Name		
TierStartDate	Tier Start Date		
TierEndDate	Tier End Date		
TierSpend			
TierVisit			
UpgradeTierCriteria	This is the parent tag that will contain below tags		
->UpgradeTier			
->Condition			
->Points			
->Spendvalue			
->Visits			
RetainTierCriteria			
->DowngradeTier			
->Condition			
->Points			
->Spendvalue			
->Visits			

**Sample:**

Type	JSON
Request	<Request> <SecurityToken>KACcS4kg+VeZclxLunf+6xOFG+e774KMRZBZ6jtGQgw/ssiNLE0qQ5Hhkmi+GThS1wwQnA8MnqJ07QnM+qTIM4XAgG9f89PpaB0LgE4Z1



	g1/Ya+BLGY/VWq0zR1LMWLZg/SW151LtDyfWo1Q+6sHD2FxBgHf9qMfDF S9VbYhzlaYzcEor6de/dlboZGkQnxtHaZkpvHC8FIQlegs3bJoET6kqAsWly7X0 avg/mH3pF0=</SecurityToken> <EasyId>9811159888</EasyId> <UserName>tpposuser</UserName> </Request>
Response - Success	<?JSON version="1.0" ?> <ResponseJSONNs:xsi="http://www.w3.org/2001/XMLSchema- instance"JSONNs:xsd="http://www.w3.org/2001/XMLSchema"> <ReturnCode>0</ReturnCode> <TierName>Club tier</TierName> <TierStartDate>02 Jan 2017</TierStartDate> <TierEndDate>14 Mar 2018</TierEndDate> <TierSpend>72100.0000</TierSpend> <TierVisit>7</TierVisit> <UpgradeTierCriteria> <UpgradeTier/> <Condition/> <Points>0</Points> <Spendvalue>0</Spendvalue> <Visits>0</Visits> </UpgradeTierCriteria> <RetainTierCriteria> <DowngradeTier>Friend</DowngradeTier> <Condition>AND</Condition> <Points>0</Points> <Spendvalue>10000.0000</Spendvalue> <Visits>1</Visits> </RetainTierCriteria> </Response>
Response - Error	

### 3.9.3. Assign Tier

**Method Name and Signature:** wsAssignTier (Varchar strRequest)

**API URL (REST):** [{{BaseUrl}}/AssignTier](#)

**Method Description:** This method is used to assign tier to customer.

**Approach:** There are certain scenarios where customer is assigned tiers as per some business conditions (activities or transactions). The activity of assigning tiers can be done by AssignTier API. This method can also be used to change the tier of a customer who already belongs to a tier.

**Pre-Condition:** Customer may be enrolled before any tier can be assigned to it or Customer may be assigned to default Tier.

**Post-Condition:** Customer can be informed once tier is assigned or upgraded via SMS or email.



**Input Parameters:**

Field	Mandatory	Field Type	Field Length	Field Description
UserName	M	Varchar	64	Partner User Name
SecurityToken	M	Varchar	512	This is a unique token Id used to authenticate the request.
StoreCode	M	Varchar	64	Store Code
TierName	M	Varchar	64	Name of the Tier
EasyId	M	Varchar	32	EasyId of the member.
CountryCode	M	Varchar	64	It is the country code. To be used for support of international numbers.
TierExpiryDate		DateTime		Expiry date of the Tier
Remarks		Varchar	512	

**Return Parameters:**

Parameter Name	Description	Data Type	Length
ReturnCode	Return Code indicating the status	Number	4
ReturnMessage	Return Message	Varchar	512

**Sample:**

Type	JSON
Request	<pre> &lt;Request&gt; &lt;SecurityToken&gt;yHMOV43JOFr8pqEJQ6tZZ/P74Fc73yYjEEMMC4 infsPGVgDtI9VYkREBhIXzFVay+RDG+FcnVuEOqLKmLNFIE YD/ws7PtXjmnsfw1VIBIuvGyOtgQFExWxLit3Kf3un76w7H E9Lh1YQrMlzTOihKv+qChal7fzzHqJKx4GjluETATZZGqJ3re gMYUvmT3CttGAg4XiVB8r/kKM3XTmvGucP/2iaO4/68eao a8jUMxpc=&lt;/SecurityToken&gt; &lt;StoreCode&gt;Demo&lt;/StoreCode&gt; &lt;UserName&gt;apiuser&lt;/UserName&gt; &lt;TierName&gt;Club Tier&lt;/TierName&gt; &lt;EasyId&gt;9811100002&lt;/EasyId&gt; &lt;TierExpiryDate&gt;27112019&lt;/TierExpiryDate&gt; &lt;/Request&gt; </pre>
Response - Success	<pre> &lt;?JSON version="1.0"?&gt; &lt;Response JSONNs:xsi="http://www.w3.org/2001/XMLSchema- instance" </pre>



	<pre> JSONNs:xsd="http://www.w3.org/2001/XMLSchema"&gt; &lt;ReturnCode&gt;0&lt;/ReturnCode&gt; &lt;ReturnMessage&gt;Success&lt;/ReturnMessage&gt; &lt;/Response&gt; </pre>
Response - Error	<pre> &lt;Response&gt;   &lt;ReturnCode&gt;124&lt;/ReturnCode&gt;   &lt;ReturnMessage&gt;Security token verification failed. &lt;/ReturnMessage&gt; &lt;/Response&gt; </pre>

### 3.9.4. Get Customer Transaction Details

**Method Name and Signature:** wsGetCustomerTransactionDetails (Varchar strRequest)

**API URL (REST):** [{{BaseUrl}}/GetCustomerTransactionDetails](#)

**Method Description:** This method is used to get transaction details of a customer.

**Approach:**

**Pre-Condition:**

**Post-Condition:**

**Input Parameters:**

Field	Mandatory	Field Type	Field Length	Field Description
EasyId	M	Varchar	32	Member Unique ID (MobileNumber)
SecurityToken	M	Varchar	512	This is a unique token Id used to authenticate the request.
SearchWord	O	Varchar	128	
StoreCode	O	Varchar	64	Outlet ID/Virtual Store Code
UTCOffset	O	Integer		
TransactionDetailsCount	O	Integer		Number of Records to be fetched
PageSize	O	Integer		Number of records in a page to be displayed
PageNumber	O	Integer		Page Number
StartDate	O	DateTime		Start Date



EndDate	O	DateTime		End Date
BillNo	O	Varchar	64	Bill Number

**Return Parameters:**

The response will be an array of transaction details with following parameters.

Parameter Name	Description	Data Type	Length
ReturnCode	Return Code indicating the status	Number	4
BillNo	Bill Number	Varchar	32
BillDate	Bill Date		
TotalBilledAmount			
Mobile	Mobile Number of the Member	Number	10
TotalAccruedPoints			
TotalRedeemPoints			
EnrolledDate			
IsEnableReferralAccruals			
MemberShipCardNumber	Membership Card Number	Number	32
IsPointType			
Tier			
RecordCount			
StoreCode	Store Code	Varchar	50
EasyPointTypeId			

**Sample:**

Type	JSON
Request	<pre>&lt;Request&gt; &lt;EasyId&gt;8897334994&lt;/EasyId&gt;  &lt;SecurityToken&gt;xstKVyOqfehLSWSsolLJKScGeVMCvdPVg7JIP7OFpEpDVu</pre>



	<p>o6n4PIPukvDpxOYZ2YVKCtzJ1NAfN17pDaKM/zd0gfV6jB8LapczNsUM7ux3H7Dg7dJ90ciryshw+nr/ib+7OwHbBbxUXqwCf/wtoVnW3m0W2NQGuebMHRvSVeYSk0nJLllyEWBVfK41QsFUNmLXGK4vGujwhRblLaDSwwg==&lt;/SecurityToken&gt;</p> <p>&lt;TransactionTypeId&gt;0&lt;/TransactionTypeId&gt;</p> <p>&lt;TransactionDetailsCount&gt;30&lt;/TransactionDetailsCount&gt;</p> <p>&lt;PageSize&gt;10&lt;/PageSize&gt;</p> <p>&lt;PageNumber&gt;1&lt;/PageNumber&gt;</p> <p>&lt;StartDate&gt;05 Aug 2018&lt;/StartDate&gt;</p> <p>&lt;EndDate&gt;07 Aug 2018&lt;/EndDate&gt;</p> <p>&lt;BillNo&gt;070820181934534725&lt;/BillNo&gt;</p> <p>&lt;/Request&gt;</p>
Response - Success	<p>&lt;ArrayOfMemberTransactionResponseListDTO&gt;</p> <p>&lt;MemberTransactionResponseListDTO&gt;</p> <p>&lt;ReturnCode&gt;0&lt;/ReturnCode&gt;</p> <p>&lt;BillNo&gt;070820181934534725&lt;/BillNo&gt;</p> <p>&lt;BillDate&gt;2018-08-07T19:34:53.473&lt;/BillDate&gt;</p> <p>&lt;TotalBilledAmount&gt;0.0000&lt;/TotalBilledAmount&gt;</p> <p>&lt;Mobile&gt;8897334994&lt;/Mobile&gt;</p> <p>&lt;TotalAccruedPoints&gt;500.0000&lt;/TotalAccruedPoints&gt;</p> <p>&lt;TotalRedeemPoints&gt;0.0000&lt;/TotalRedeemPoints&gt;</p> <p>&lt;EnrolledDate&gt;0001-01-01T00:00:00&lt;/EnrolledDate&gt;</p> <p>&lt;IsEnableReferralAccruals&gt;&gt;false&lt;/IsEnableReferralAccruals&gt;</p> <p>&lt;UserName&gt;&lt;/UserName&gt;</p> <p>&lt;CardNumber&gt;&lt;/CardNumber&gt;</p> <p>&lt;IsVoucher&gt;&gt;false&lt;/IsVoucher&gt;</p> <p>&lt;IsRefunded&gt;&gt;false&lt;/IsRefunded&gt;</p> <p>&lt;Code&gt;&lt;/Code&gt;</p> <p>&lt;Brand&gt;itcinfotech&lt;/Brand&gt;</p> <p>&lt;Narration&gt;&lt;/Narration&gt;</p> <p>&lt;AccruedPoints&gt;4684&lt;/AccruedPoints&gt;</p> <p>&lt;MemberShipCardNumber&gt;&lt;/MemberShipCardNumber&gt;</p> <p>&lt;IsPointType&gt;&gt;true&lt;/IsPointType&gt;</p> <p>&lt;Tier&gt;&gt;false&lt;/Tier&gt;</p> <p>&lt;RecordCount&gt;1&lt;/RecordCount&gt;</p> <p>&lt;StoreCode&gt;Demostore&lt;/StoreCode&gt;</p>

	<pre> &lt;EasyPointTypeId&gt;5&lt;/EasyPointTypeId&gt;  &lt;IsRedeemed&gt;&gt;false&lt;/IsRedeemed&gt;  &lt;/MemberTransactionResponseListDTO&gt;  &lt;/ArrayOfMemberTransactionResponseListDTO&gt; </pre>
Response - Error	

### API Consumption Guidance Note

ER has exposed two APIs to get all orders of a customer/Point ledger entries

1. **GetCustomerTransactionDetails** : This is for all Orders which have crossed Return Period
2. **Lookup** : This is for all Orders which have not crossed Return Period (Ecommerce Orders)

### GetCustomerTransactionDetails

Get Customer Transaction Details API list down Transactions with multiple offers(Multiple SKU Offers, Redemption Offer,Activity Based Offer,Bonus Offers etc) executed on them.

Offers Type are shown in the API response with the field EasyPointTypeId. For combination of Mobile,BillNo,BillDate and StoreCode below Records has to be grouped to get all Unique Bills

EasyPointTypeId = 2 Record shows Point Redeemed Record Entry of Bill

EasyPointTypeId = 27 Record shows Point Accrual Record Entry of Bill

EasyPointTypeId = 15 Record shows Flat Redemption Record Entry of Bill(This is used in case Entry is inserted using Reconciliation instead of Live API)

Below fields needs to be summed up to get Point Accrued and Point redeemed

TotalAccruedPoints =Issued Points

TotalRedeemPoints =Redeemed Point

If there are more records on EasyPointTypeId =27 for same Mobile,BillNo,BillDate and StoreCode they have to be grouped also

Bill Amount will come from Omni System for that Bill

### Lookup

This API will show all Orders who are still not cross Return Period

PrimaryOrderNumber in the response shows Order Number which is passed in BillNo of SaveSKUBillDetails APIs

OrderLevelCredits in response shows Point Issued at Order Level

In case Item Level Points has to be shown **LoyaltyPoints** against each Item has to read

EasyPointTypeId descriptions

EasyPointTypeId	Description	Detail Understanding
1	Accrual	This is used for EasyPointAccrual/EOSS -- Ignore
2	Redemption	This is used for Redemptions - Debit
3	Refund to Member	This is used for refunding to member for accrual transactions – Credo
4	Refund to Merchant	This is used for refunding to Merchant
5	Flat Accrual	This is used for FlatAccrual Upload, Bonus points from Shopping Offer Service, Enrolment Bonus points, First Transaction bonus points
14	Lapse Points Redemption	This is used when points are lapsed
15	Flat Redemption	This is used for FlatRedemption
16	Referral Accrual	This is used for referral
27	SKU Accrual	This is used for SKU Transaction
30	Negative Accrual	This is for reversal of items post return depending upon point accrual vs point redeemed for the item

### 3.9.5. Get Referral Code

**Method Name and Signature:** wsGetReferralCode(Varchar strRequest)

**API URL (REST):** [{{BaseUrl}}/GetReferralCode](#)

**Method Description:** This method is required to fetch the referral code of a Member.

**Approach:**

**Pre-Condition:** Member should already be part of Loyalty program.

**Post-Condition:**

**Input Parameters:**

Field	Mandatory	Field Type	Field Length	Field Description
SecurityToken	M	Varchar	512	This is a unique token Id used to authenticate the request.
UserName	M	Varchar	64	Partner User Name
EasyId	M	Varchar	32	EasyId of the member.
CountryCode	M	Varchar	64	It is the country code. To be used for support of international numbers.

**Return Parameters:**

Parameter Name	Description	Data Type	Length
ReturnCode	Return Code indicating the status	Number	4
ReturnMessage	Return Message	Varchar	512
ReferralCode	Referral code	Varchar	128
CustomerName	Customer Name	Varchar	128

**Sample:**

Type	JSON
Request	<pre> &lt;Request&gt;  &lt;SecurityToken&gt;xstKVyOqfehLSWSsolLJKScGeVMCvdPVg7JIP7OFpEpDVuo6 n4PIPukvDpxOYZ2YVKCtzJ1NAfN17pDaKM/zd0gfV6jB8LapczNsUM7ux3H7D g7dJ90ciryshw+nr/ib+7OwHbBbxUXqwCf/wtoVnW3m0W2NQGUEbMHRvS VeYSk0nJLIlyEWBFvK41QsFUNmLXGK4vGujwhRbLLaDSwwg==&lt;/SecurityTo ken&gt;   &lt;UserName&gt;apparaod@revalsys.com&lt;/UserName&gt;   &lt;EasyId&gt;9416371613&lt;/EasyId&gt;   &lt;CountryCode&gt;91&lt;/CountryCode&gt; &lt;/Request&gt; </pre>
Response - Success	<pre> &lt;ReferralCodeResponse&gt;    &lt;ReturnCode&gt;0&lt;/ReturnCode&gt; </pre>



	<pre> &lt;ReferralCode&gt;REV9416371613&lt;/ReferralCode&gt;  &lt;ReturnMessage&gt;Success.&lt;/ReturnMessage&gt;  &lt;CustomerName&gt;Rakesh Kumar&lt;/CustomerName&gt;  &lt;/ReferralCodeResponse&gt; </pre>
Response - Error	

### 3.9.6. Update Bill Number

**Method Name and Signature:** wsUpdateBillNo (Varchar strRequest)

**API URL (REST):** [{{BaseUrl}}/UpdateBillNo](#)

**Method Description:** This method is used to update the Bill Number.

**Approach:**

**Pre-Condition:**

**Post-Condition:**

**Input Parameters:**

Field	Mandatory	Field Type	Field Length	Field Description
BillDate	M	DateTime		Billing date of the member
EasyId	M	Varchar	32	EasyId of the member.
OldBillNo	M	Varchar	64	Old Bill Number
NewBillNo	M	Varchar	64	New Bill Number
SecurityToken	M	Varchar	512	This is a unique token Id used to authenticate the request.
StoreCode	M	Varchar	64	Store Code
UserName	M	Varchar	64	Partner User Name
RewardType	O	Reward Type is		
Mode	O			
CountryCode	M	Varchar	64	Country code – to support international mobile number

**Return Parameters:**

Parameter Name	Description	Data Type	Length
ReturnCode	Return Code indicating the status	Number	4





ReturnMessage	Return Message	Varchar	512
---------------	----------------	---------	-----

**Sample:**

Type	JSON
Request	<pre> &lt;Request&gt;   &lt;BillDate&gt;17 Aug 2018&lt;/BillDate&gt;   &lt;CountryCode&gt;91&lt;/CountryCode&gt;   &lt;EasyId&gt;7673977543&lt;/EasyId&gt;   &lt;OldBillNo&gt;112333&lt;/OldBillNo&gt;   &lt;NewBillNo&gt;N234242&lt;/NewBillNo&gt;   &lt;SecurityToken&gt;iuuHSFP/08BZ57sRJC4/ulxBUN8HAaGROk3uiOhBVhPfsiKQ mFsrB8KYYaWIXcBbrYMIIdYWNlJUXMyBmPvKZzuRWIOtCFC3EnCyXwkwHy sRstcx3iK2M6qRE4qUAPVwGHxcip0TQ4YF4cCQbqOEABgmaq5vu8B68Db2X M4g4YrjG8uQWVthlydrRHD8FymYgfvdxchv/kremyf/ynH98yQ==&lt;/SecurityToken&gt;   &lt;RewardType&gt;&lt;/RewardType&gt;   &lt;Mode&gt;&lt;/Mode&gt;   &lt;StoreCode&gt;Demo&lt;/StoreCode&gt;   &lt;UserName&gt;apparaod@revalsys.com&lt;/UserName&gt; &lt;/Request&gt; </pre>
Response - Success	<pre> &lt;Response&gt;   &lt;ReturnCode&gt;0&lt;/ReturnCode&gt;   &lt;ReturnMessage&gt;Success.&lt;/ReturnMessage&gt; &lt;/Response&gt; </pre>
Response - Error	

**3.9.7. Generate OTP**

**Method Name and Signature:** wsGenerateOTP (Varchar strRequest)

**API URL (REST):** [{{BaseUrl}}/GenerateOTP](#)

**Method Name and Description:** This method is used to generate OTP for authenticating a customer. An OTP is generated on customer's mobile number and the same OTP is then validated using wsValidateOTP method.

**Best Implementation Practice:**

- This API will be used to generate OTP on Mobile Number during Registration of Member. This improves quality of Data

**Approach:****Pre-Condition:**

**Post-Condition:**
**Input Parameters:**

Field	Mandatory	Field Type	Field Length	Field Description
<b>MemberID</b>	M/O	Varchar	32	Unique Id of the member
<b>EmailID</b>	M/O	Varchar	128	Email Id of the member
<b>MobileNumber</b>	M/O	Varchar	32	Mobile Number of the member
<b>StoreCode</b>	M	Varchar	64	Store Code
<b>UserName</b>	M	Varchar	64	Partner User Name
<b>SecurityToken</b>	M	Varchar	512	This is a unique token Id used to authenticate the request.
<b>CountryCode</b>	M	Varchar	64	

**Return Parameters:**

Parameter Name	Description	Data Type	Length
ReturnCode	Return Code	Number	4
ReturnMessage	Return Message	Varchar	512
RequestID	ID of the request	Varchar	20

**Sample:**

Type	JSON
Request	<pre>{   "MemberID": "9811159888",   "EmailID": "manish@easyrewardz.com",   "MobileNumber": [],   "StoreCode": "webbot",   "UserName": "apiuser@webbot.com",   "SecurityToken":   "xkk/IptYjbQo4z4RD2fB46uDsyV6t3eT0A6Haq/L5vho60DAwhkqISU1/gBxxvR mE39Hc9XwAJPqgUjD4wWDbL17qtrnmKi3/PVEW3ER3IS/DzgWDYIIAuBokXCD nTU/261kHO8XFRuKtmkWhRxp6QbK17mawPhiqACmaRyjfi39AAossgg/8gUI7a gU0xam" }</pre>
Response – Success	<pre>{   "ReturnCode": 0,</pre>

	<pre>"RequestID": "IntegrationProd1752", "ReturnMessage": "Success" }</pre>
Response – Error	<pre>&lt;Response&gt;   &lt;ReturnCode&gt;365&lt;/ReturnCode&gt;   &lt;ReturnMessage&gt;Invalid OTP.&lt;/ReturnMessage&gt; &lt;/Response&gt;</pre>

### 3.9.8. Validate OTP

**Method Name and Signature:** wsValidateOTP (Varchar strRequest)

**API URL (REST):** [{{BaseUrl}}/ValidateOTP](#)

**Method Description:** This method is used to validate the OTP generated by wsGenerateOTP method.

**NOTE:**

1. All APIs should support Country Code
2. All Properties and Offer Codes should be made configurable and should be controlled from UI.

**Approach:**

**Pre-Condition:** The OTP should have been generated by the GenerateOTP method.

**Post-Condition:**

**Input Parameters:**

Field	Mandatory	Field Type	Field Length	Field Description
MemberID	M	Varchar	32	ID of the member/ Mobile Number
RequestID	M	Varchar	32	ID of the request
OTP	M	Varchar	6	OTP
UserName	M	Varchar	64	Partner Name
SecurityToken	M	Varchar	512	This is a unique token Id used to authenticate the request.
CountryCode	M	Varchar	64	Country Code

**Return Parameters:**

Parameter Name	Description	Data Type	Length
----------------	-------------	-----------	--------

ReturnCode	Return Code indicating the status	Number	4
ReturnMessage	Return Message	Varchar	512

#### Sample:

Type	JSON
Request	<pre>{   "MemberID": "9811159888",   "RequestID": "IntegrationProd1752",   "OTP": "971895",   "UserName": "apiuser@webbot.com",   "SecurityToken":     "xkk/IptYjbQo4z4RD2fB46uDsyV6t3eT0A6Haq/L5vho60DAwhkqISU1/gBx     xvRmE39Hc9XwAJPqgUjD4wWDbL17qtrnmKi3/PVEW3ER3IS/DzgWDYIIA     uBokXCDnTU/261kHO8XFRuKtmkWhRxp6QbK17mawPhiqACmaRyjfi39A     Aossgg/8gUI7agU0xam" }</pre>
Response – Success	<pre>{   "ReturnCode": 0,   "ReturnMessage": "Success" }</pre>
Response – Error	

## 4. List of Return Codes

Return Code	Description
0	Success.
104	Do not have permissions to access this method.
105	Security token expired.
109	Invalid billed price.
122	Invalid bill id.
124	Security token verification failed.
125	Input parameters are not provided correctly.
129	Cannot accrue as wallet does not have enough easy Points.
139	Cannot accrue, please try later
168	Invalid offer.
169	Amount should be greater than or equal to transaction unit.

170	Offer does not exists in this range.
171	Enter at least one field.
172	No results found.
173	Cannot redeem as customer does not have enough easy Points.
174	Member does not have enough points.
175	Requested points less than bill amount points
186	Membership card number verification process failed.
187	Unable to generate membership card no. please try after some time.
189	Invalid store.
198	Store code does not exists.
199	User does not exists.
200	User does not have permission under this store.
207	Email already exists.
212	Member already exist.
222	Unauthenticated user.
223	Unauthenticated store.
224	Policy already claimed.
225	Policy has expired on [#DATE#] and cannot be claimed.
227	Your Policy has been deactivated.
228	Policy cannot be claimed as it has not started and will start on [#DATE#].
229	Policy number is invalid please contact office.
246	Failed in generating security token.
247	Input parameter Membership card number missed.
248	Invalid user name or password.
249	Invalid appid.
250	Invalid devid.
251	Invalid vendor code supplied.
252	Invalid loyalty program.
253	Member ID does not exists.
254	Invalid easyId.
255	Invalid date.
256	Transaction date should be less than or equal to current date.
257	Points are mandatory for activity based redemption.
258	Amount should be greater than zero.
259	Cannot redeem easy Points as the amount is less.
260	Redemption is already done for this bill id.
261	Bill id already exists.

262	User name does not exists.
263	Store does not exists.
264	Activity does not exists.
265	Invalid customer id.
266	Redemption date should be current date.
267	Easy pin should be between 6-12 letters and should include 1 number and 1 alpha Var character.
268	Invalid mobile number.
269	Invalid email id.
270	Invalid pincode.
271	Invalid date of birth.
272	Invalid first name.
273	Invalid last name.
274	Invalid membership card no.
275	Invalid issue card.
276	Invalid card value.
285	Offer does not exists.
286	Invalid member id.
287	Invalid user name.
288	No transaction has made with this bill id.
289	Customer does not have enough easy Points to redeem.
290	Invalid security code.
291	Security code already used.
292	Amount is required for this transaction.
293	Rollback is already done for this bill id.
294	Refund points are more than accrued/redeemed points.
295	Bill id does not exists.
296	Refund amount is not equal to billed amount.
297	No transaction has made to the customer.
298	Refund date should be current date.
299	Failed to register.
300	Invalid member id.
301	Points are mandatory for activity based accrual.
302	Cannot accrue easy Points as the amount is less.
315	Failed to update membership card details.
316	Failed to insert membership card details.
317	Membership card not exists.
318	Failed to accrue points to member.



319	Loyalty program does not exists.
320	Invalid price.
321	Invalid points.
323	Invalid transaction time.
324	Invalid bill value.
325	Invalid coupon1.
326	Invalid offer1.
327	Invalid total discount.
328	Invalid SKU.
329	Invalid item type.
330	Invalid item no.
329	Invalid item type.
330	Invalid item no.
331	Invalid item quantity.
332	Invalid item price.
333	Invalid total price.
334	Invalid coupon1.
335	Invalid offer2.
336	Invalid item discount.
337	Item does not exists.
338	Item does not belong to store.
339	Item does not have points mapping.
340	Member needs to collect a minimum of [#POINTS#] points to do the first redemption.
341	Please contact call centre as policy is not found.
342	Offer is not applicable to this store.
343	Bill number does not exists.
344	New bill number already exists.
345	Invalid old bill number.
346	Invalid new bill number.
347	Invalid bill date.
348	Failed to update bill number.
349	New bill number should be different from old bill number.
350	Product not available.
351	Right now not Claim
352	Member cannot claim this shopping offer.
353	No shopping offer available.
354	Member is in inactive status.

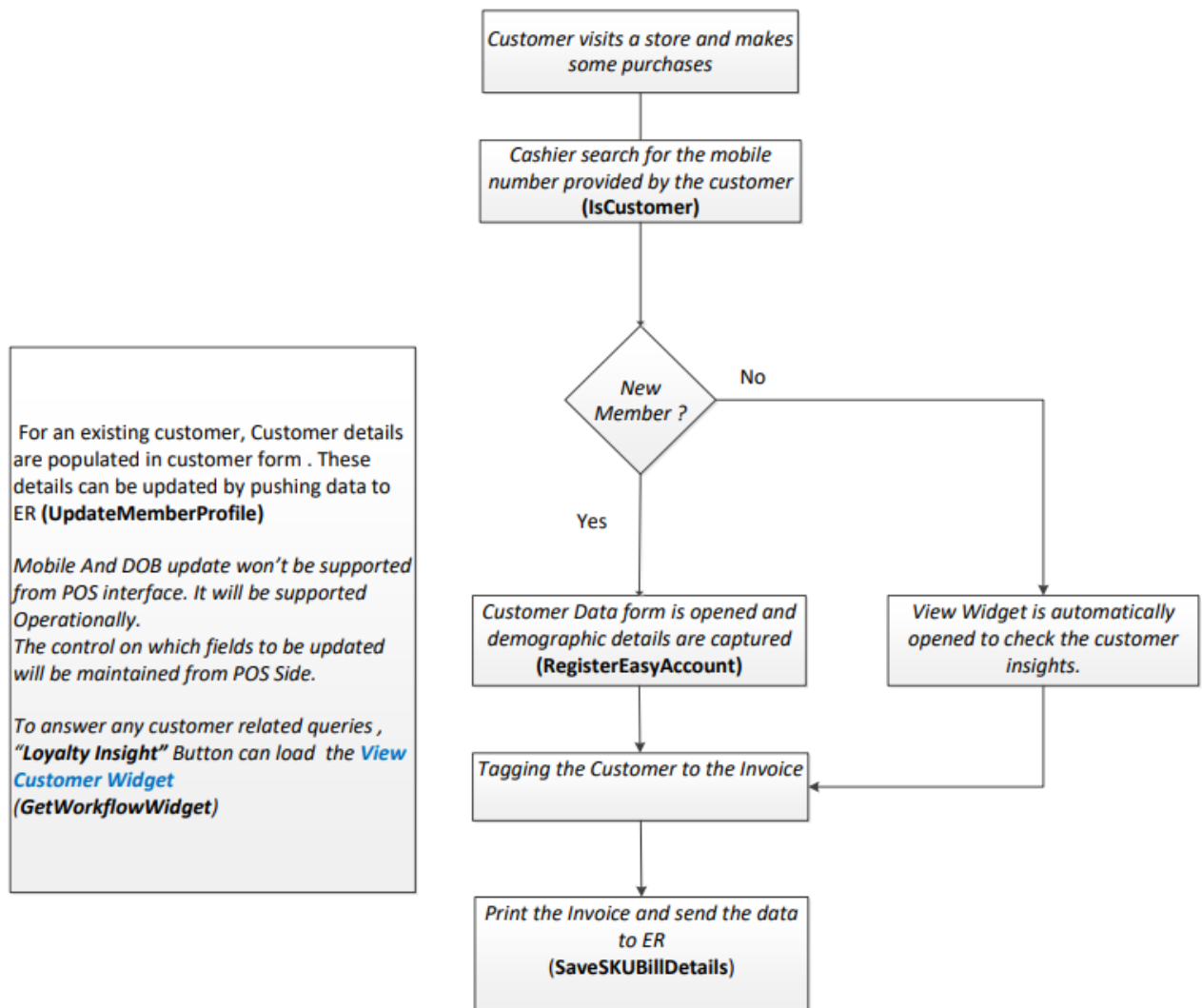
365	Invalid OTP.
366	Transaction doesn't belong to this member.
367	Pointed already Released.
369	Transaction not done in this store.
370	Transaction code does not exist.
371	Blocked points already released for this member.
365	Invalid OTP.
368	Invalid Coupon.
378	OTP Required
401	Coupon Offer already applied.
402	No Coupon codes are available.
423	Invalid RequestID
424	Invalid Request
434	No Coupon Offer Exists
435	Not Same Member
436	Not Eligible Store
437	Coupon already claimed
438	Coupon already blocked
439	Member is not registered
440	Coupon is expired
441	Not same store

## 5. Physical Store Customer Lifecycle

### 5.1. Member Enrolment Flow

The below diagram showcase customer journey of enrolling into loyalty program once he/she visits the physical store.





In case of Void Bill Option available, EasyPointRefund API needs to be integrated.

Figure 1: Customer Journey

## 5.2. View Customer Widget

There will be a button inside POS screen to check all CRM insights. It will load a URL to show all customer transaction across channel, Point ledger, Available rewards, Program construct details

Member Profile

**Rashmi Chatterjee**  
Member





**Birthday :** 04 Apr 2000      **Mobile :** \*\*\*\*\*

**Member Since :** 27 months

**Points -482**

**Coupons - 1**  
LVV6KZXXXX Expiry 19 Apr 2022 15% Off  
Maximum discount upto 500 (vld on fresh or non-discounted articles)

**Last Visit**  
64 days ago

Insights Message	Action
Cross Sell - Pitch member Women - Daily wear, Shoe Care, Women - Casual Flats, Comfit Women, School Shoe, Mens- Socks, Daily wear, Mens Sandal, Power - Mens Running, BUBBLEGUMMERS - Kids Casual and encourage trial	 
Wish member Happy Birthday & offer birthday discount displayed in "Coupons" Section	 

**Nearest points lapsing**

31 Jan 2023	482 Points
-------------	------------

**Bills** **Points Ledger** **Coupons**

☐ Recent Bills ☐ By Article Code ☐ Advance Filters

### 5.3. Point Redemption Flow

This flow defines Point redemption journey at store.

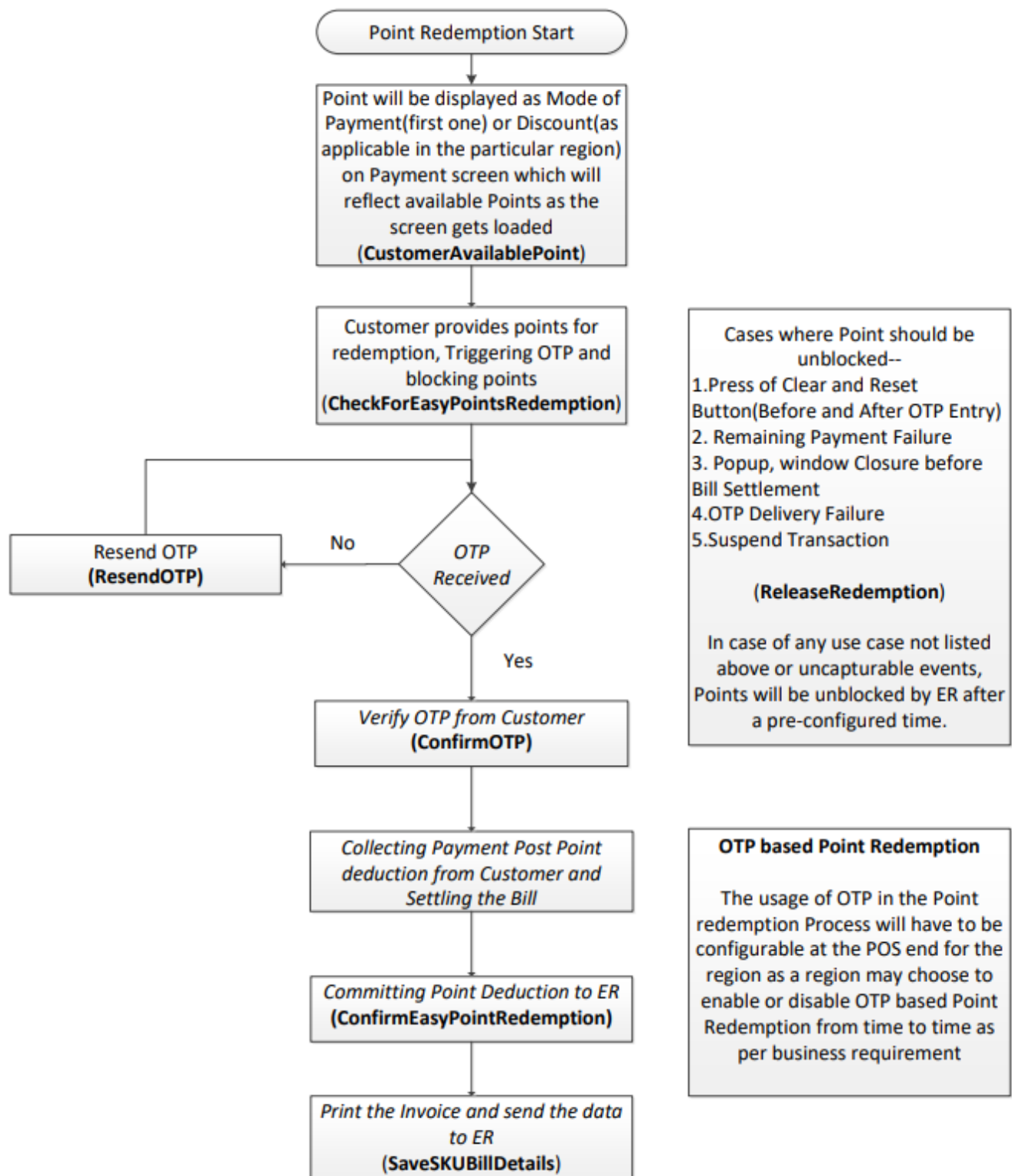


Figure 2: Points Redemption Flow

### 5.3.1.Point as a Discount - Impact/Concerns

ER suggestion - We recommend point redemption to be treated as a Tender/Mode of Payment. Below are few of the impact/concern areas if points are treated as discount.

We will explain each point with the following example

**Total MRP of the order - 1000**

**Point redeemed - 200**

1. If point is treated as a discount, then the bill value for the order in example will be 800 and revenue value will be lower for the brand.
2. When point is accrued, it is accrued as a liability, if redeemed as discount as per accounting principal the same can be adjusted as liability.
3. When an item is returned, on which points have been redeemed as discount, these points cannot be refunded as their revenue contribution was not accounted for.  
In the above example, since the item was recorded to be sold for Rs. 800 (instead of Rs. 1000), if it is returned, the POS accounting rule will allow refund of Rs. 800 only and not Rs. 1000. Hence points redeemed worth Rs. 200 cannot be rolled back.
4. In terms of Taxes, the process will depend prevailing tax rules on Mode of Payment and discount

### 5.3.2.OTP delivery Process:

**Channels available:** SMS, Email, WhatsApp

**Initial Go Live Channels:** SMS and Email

**Other solutions Possible:**

1. Pre generated E-Pin via missed or SMS Long Code

- This will pre generate an OTP for the customer with X hours validity, so that customer can get the OTP from any location where he has network before coming to the store. This will work for a customer with clear intent of redemption before visit)

2. We can enable an OTP fetching Keyword on WABA for customers to request for the generated OTP via brand WABA (This will require an OTP to have been generated by the POS first)
3. On call Password delivery (Dependant on country specific Vendor's feature availability)

MPIN from App cannot be used for OTP.

## 5.4. Coupon Redemption Flow

Customer provides the coupon code to cashier. Cashier applies the coupon code in the text box provided. On pressing redeem button If Coupon Code is valid otp is triggered to customer. Customer provides the OTP and Coupon is blocked . Once invoice is created successfully Coupon redemption process is completed.

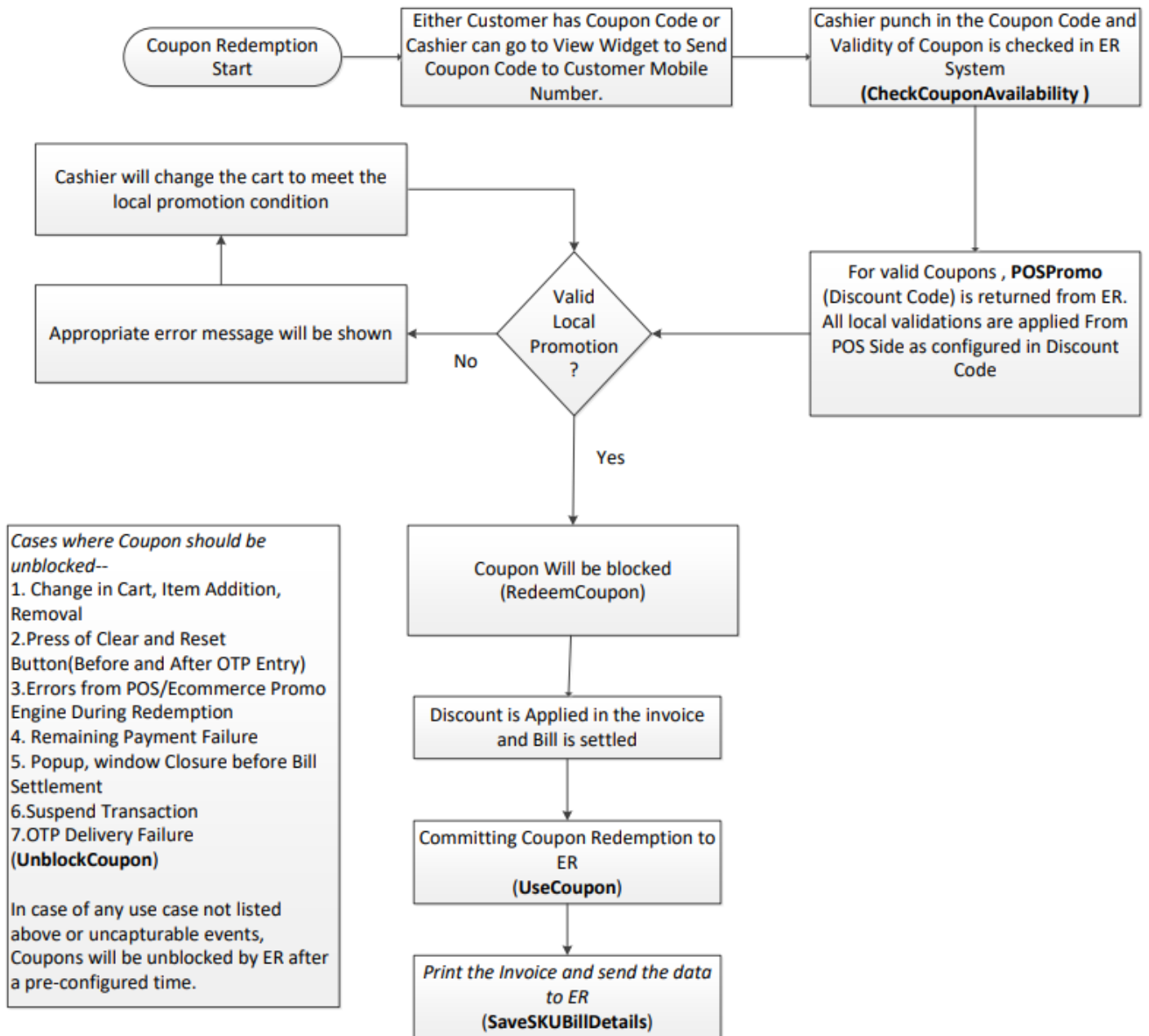


Figure 3: Customer Journey-Coupon Redemption

**POS Promo** - POSPromo is the binding ID between CRM and POS Engine. POSPromo will be unique rule ID of the promotion created in the POS HO System ( i.e ORCE in case of Xstore). For any promotion execution below steps will be followed:

1. Creation of promotion in POS
2. Sharing of POSPromo(DealID) with CRM

3. CRM creating couponcodes against DealID
4. During Redemption DealID will be used by POS Local Engine to apply all conditions and benefits.

## 5.5. Sales Return

When customer visits the store for returning an item, below scenarios are applicable

### 5.5.1. Customer is carrying the invoice

Item will be returned with the reference of the invoice and Points (Earned and Burned in that invoice) will be rolled back (Pro rata basis). This is also applicable if same invoice can be searched using Customer Mobile Number

### 5.5.2. Invoice is not available

Item return will depend upon brand business policy, Points won't be rolled back.

### 5.5.3. Non loyalty Customer

Item Return will happen as per Business policy. Data won't be pushed to ER

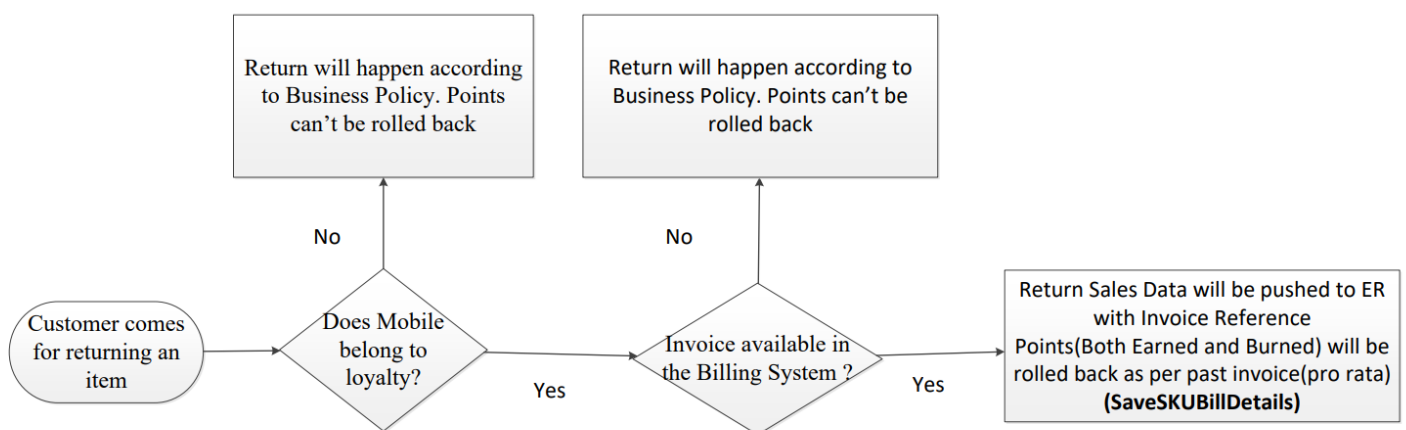


Figure 4 Sales Return Scenarios

## 5.6. Offline Stores/Internet not working

There are two scenarios where CRM APIs may not respond

1. Internet unavailability
2. CRM not online

In case internet is not available in Stores Rewards redemption won't be possible. The registration and Sale/Return Invoices will be stored locally with the flag kept that these data are not synced with CRM.

Once internet is up, both data will be synced with CRM.

In case any time out or nonresponse from CRM, POS will retry for 3 times till the time CRM System provides acknowledgement.

In case of any failure even with above fail-safe handling approach, EOD file generated by HO will be used to Reconcile POS Sales with CRM.

## 5.7. GV Issuance Flow

GV Issuance is a process done for the Gift Voucher (Single Use/Multi Use) when GV are getting sold from physical store /Ecommerce. The process will depend on whether GV is physical or EGV

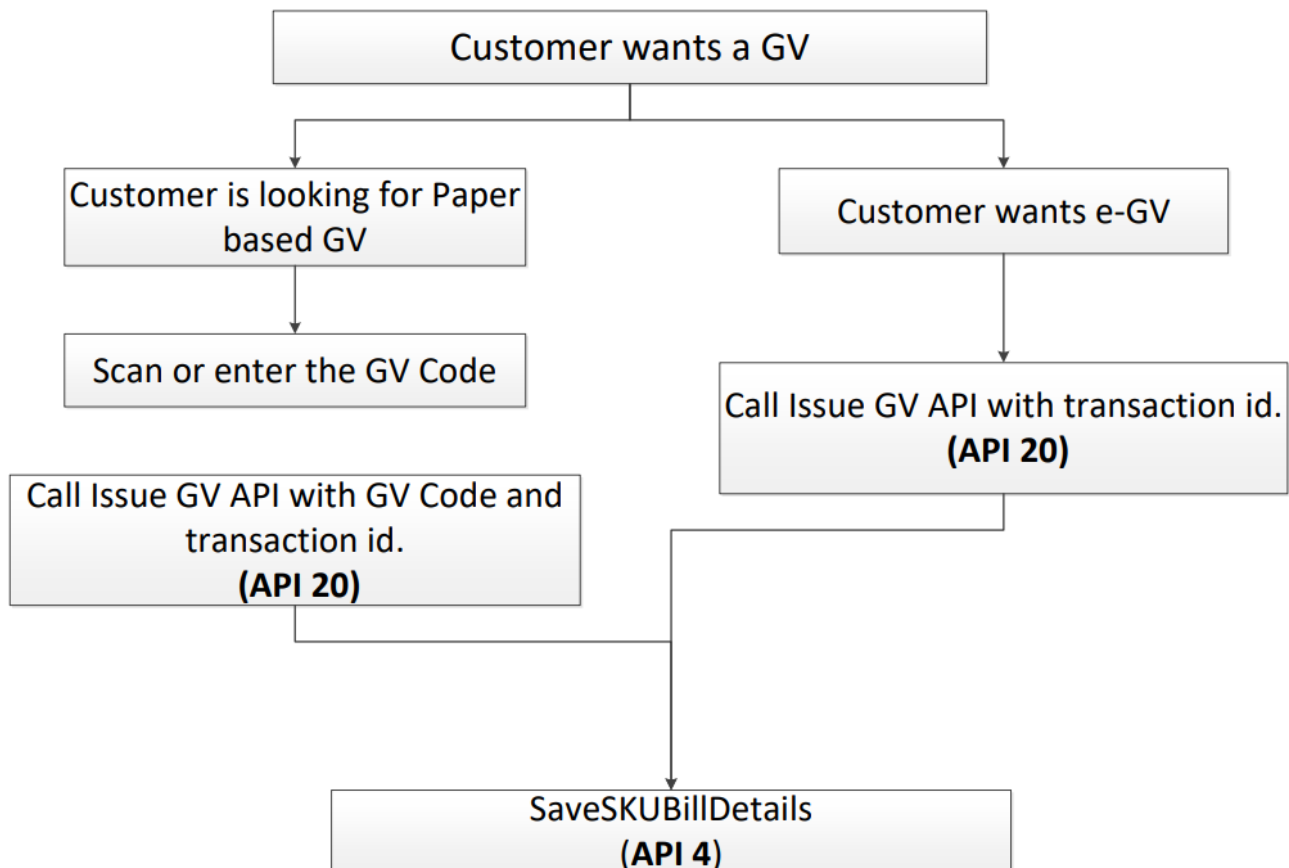


Figure 2:Customer Journey-Issue GV

## 5.8. GV Redemption Flow

GV Redemption process is also two step process. On the payment Page GV Code is provided by customer to Cashier. Cashier Starts the redemption process and blocks the GV. Post OTP Validation GV is blocked and remaining payment is completed. Once invoice is created the GV is redeemed.



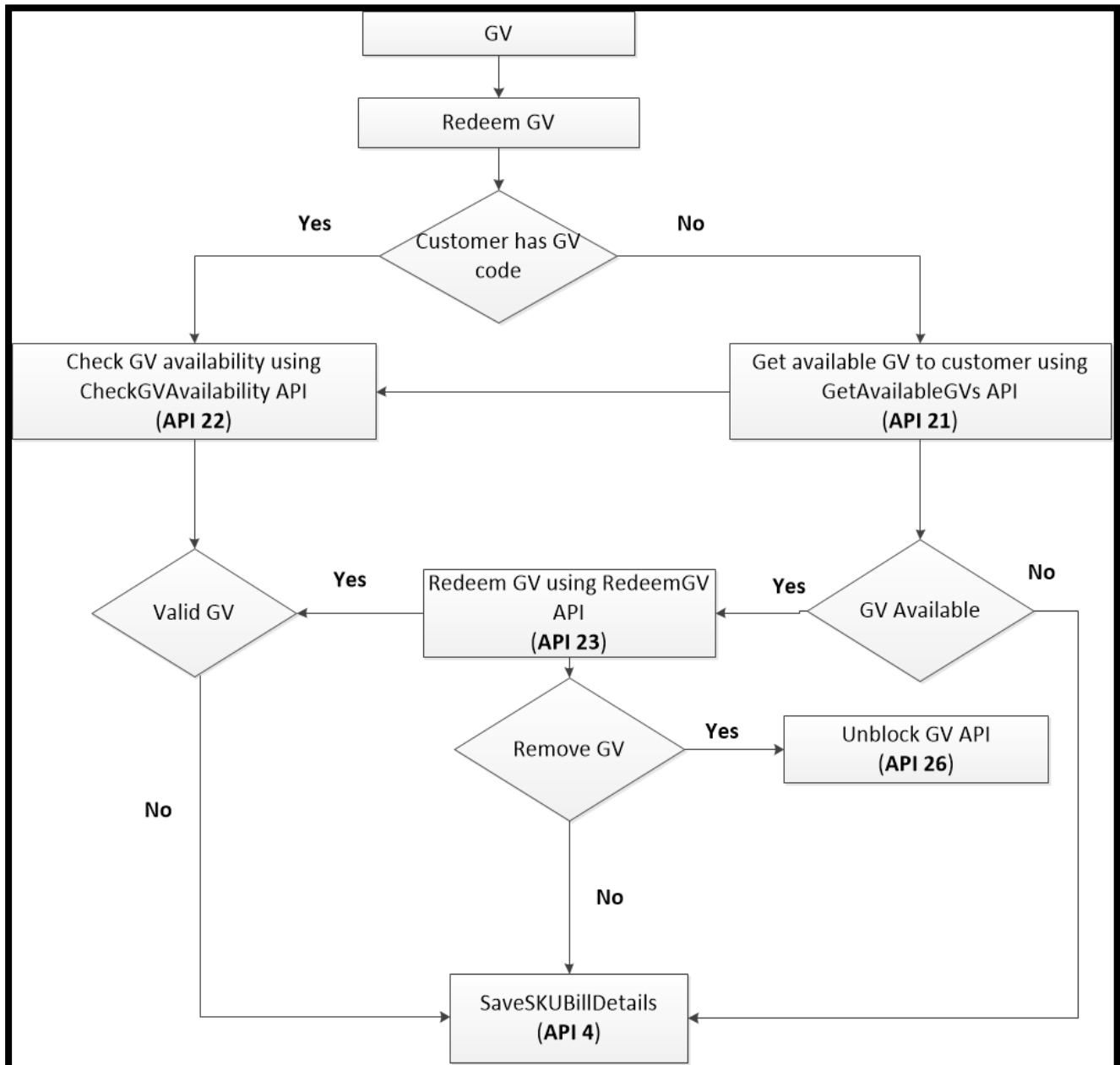


Figure 3: Customer Journey-Redeem GV

## 6. Widgets

Widgets URL will be sent as response from ER to POS. These widget URL will open the corresponding widget screen at the POS to perform the corresponding action like redemption of point or coupon.

There are four types of widgets supported by ER:

- Redeem Points
- Redeem Coupon
- Add/Edit Customer
- View Customer

## 6.1. Redeem Points

**Widget Name:** Redeem Points Widget Code: RP

**Description:** This widget will open at POS using the redeem point widget URL returned from ER to POS. It will enable the member to redeem loyalty points on the bill at the POS.

**Pre-Condition:** Irrespective of whether loyalty points are integrated in the POS/transaction system as a discount or a payment mode, this method is always called prior to bill save. The OTP generated (if required) during this call is triggered to the member's mobile no. registered during enrollment.

**Post-Condition:** If Member decides not to redeem points, then Release redemption points should be called. Process:  
The POS will need to integrate ER's widget invocation API.

The name of the method to invoke the workflow widget is GetWorkflowWidget & the rest URL to call this API is {{BaseUrl}}/GetWorkflowWidget

When the request parameters for the redeem point widget is passed, the Redeem Point widget URL is returned in response of the API.

This URL redirects to the web page of the Redeem Point widget.

In the web page of the redeem point widget below parameters are displayed:

**Bill Amount-** This gets pre-populated from the bill amount passed in the request for widget & is non-editable.

**Available Points-** This is pre-populated & non-editable. This is fetched from the response of the Customer Available Points API called from the Redeem Point widget level itself.

**Point Rate-** This is pre-populated & non-editable. This is fetched from the response of the Customer Available Points API called from the Redeem Point widget level itself.

**Point to Redeem-** This is pre-populated & editable. This is fetched from the response of the Customer Available Points API called from the Redeem Point widget level itself. The maximum no of points that can be redeemed on a bill is auto populated here.

**Point Value-** This is pre-populated based on the maximum points to redeem auto populated & is non-editable. Its value changes based on the input given by the user in the Points to Redeem screen. The value displayed is calculated as

Point Value= Points to Redeem \* Point Rate.

Implicit calls are made to generate OTP, Confirm OTP & resend OTP from Check for easy point redemption API through the Redeem Point widget level.

ER API's invoked implicitly from the Redeem Point widget:

Get Customer Available Points API

Check for Easy Points Redemption API-Invoked on the click of Redeem CTA

Confirm OTP

Resend OTP- Invoked on the click of Resend OTP CTA

Release Redemption Points- Invoked on the click of Release CTA

#### 6.1.1. Input Parameters for Redeem Point Widget URL

Parameter Name	Description	Data Type	Length
SecurityToken (*)	This is a unique token Id used to authenticatethe request.	Varchar	512
StoreCode (*)	Store Code	Varchar	50
UserName (*)	Partner User Name	Varchar	64
MemberID (*)	Mobile no of the member	Number	10
WidgetCode (*)	This corresponds to the pre-defined code of the Widget in LPaaS. Every widget type will have a unique widget code. The widget codesused are:  AEC-Add/Edit Customer  VC- View Customer  RC- Redeem Coupon  RP- Redeem Points	Varchar	10
OfferCode	This corresponds to the code of the redemption offer to be applied on Bill	Varchar	64
BillAmount	This corresponds to the amount of the transaction/Bill	float	10
BillIID	This corresponds to the unique identifier ofthe bill.	Varchar	50

### 6.1.2. Sample Request/Response for the Redeem Point widget in GetWorkflowWidget API

Type	XML
Request for RPwidget Invocation	<pre> &lt;Request&gt;   &lt;SecurityToken&gt;PhUtWS4aUzb0LuOo9iuZwCDrtSYMs/Y+6DYZyzomORlbgST   30   MrHBRVImO/bQyPqxZ34kRJl4WXjwEhAh9xW4pbBGyLQoqkCMRhhUA3cOZR   G4adfWld   t+vDmwDcNdVJCx8K7QGqZS5X6L55r7nsmIWE4dpVMafMrb4qltLztPw4= &lt;/SecurityToken&gt;    &lt;MemberId&gt;9640830846&lt;/MemberId&gt;    &lt;UserName&gt;<a href="mailto:apparaod@revalsys.com">apparaod@revalsys.com</a>&lt;/UserName&gt;    &lt;StoreCode&gt;demo01&lt;/StoreCode&gt;    &lt;WidgetCode&gt;RP&lt;/WidgetCode&gt;    &lt;OfferCode&gt;Redemption01&lt;/OfferCode&gt;    &lt;BillId&gt;dfdfhdfhh&lt;/BillId&gt;    &lt;BillAmount&gt;100&lt;/BillAmount&gt;  &lt;/Request&gt; </pre>
Response for RPwidget	<pre> &lt;?xml version="1.0"?&gt;  &lt;Response    xmlns:xsd="http://www.w3.org/2001/XMLSchema"   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"&gt;    &lt;ReturnCode&gt;0&lt;/ReturnCode&gt;    &lt;ReturnMessage&gt;Success&lt;/ReturnMessage&gt;    &lt;WidgetURL&gt;http://SOCHPWW.erstaging.com/PWW/RedeemPoints.   aspx?Requ   estCode=SOCH_523445_16082018125513709&lt;/WidgetURL&gt;  &lt;/Response&gt; </pre>

From the URL returned in the tag <widgetURL> in response is opened in the browser, below shown web page of redeem pointwidget appears.

### 6.1.3. Redeem Point Widget Screen

SD

siv deepu  
Mobile : 9640830846

Bill Amount \*  
100.00

Points to Redeem  
0

Available Points : 102671  
Point Rate : 1.00

Point Value  
0

Redeem

OTP sent Successfully

Payment Authorization

Enter OTP  
  
(Enter the code sent on mobile)

Pay

Amount	100.0
Points redeeming	100
Points Value	100.00
Balance to Pay	0

Release

Resend OTP

Populated from Customer AvailablePoints

Click on this populated below details

### 6.2. Redeem Coupons Widget

**Widget Name:** Redeem Coupon Widget Code: RC

**Description:** The url for redeem coupon widget returned from ER to POS will open the Redeem Coupon widget at the POS. Member can then redeem his available coupon on the bill/transaction to avail the benefit.

**Approach:** Once this method is called the coupon is first validated & if the same is successful, then is blocked by LPaaS for use by this member in this transaction. The same also returns to the transaction system/POS all details including promotion code, that is needed to trigger the associated promotion in the transaction.

**Pre-Condition:** This method is called before prepayment, to apply the promotion & corresponding discount in the transaction.

**Post-Condition:** The promotion code returned by the response of this method is used to trigger the discount in in the transaction. If the promotion code is not to be triggered, then the Transaction Type can also use the Discount Type & Discount Value to trigger the discount.

**Process:**

The POS will need to integrate ER's widget invocation API.

The name of the method to invoke the workflow widget is GetWorkflowWidget & the rest URL to call this API is

{{BaseUrl}}/GetWorkflowWidget

When the request parameters for the redeem coupon widget is passed, the Redeem coupon widget URL is returned in response of the API.

This URL redirects to the web page of the Redeem Coupon widget.

In the web page of the redeem Coupon widget below parameters are displayed:

**Coupon Codes-** This column in the grid contains the list of coupon codes of the available coupons issued to the member.

**Validity-** This column in the grid corresponds to the Validity end date of the corresponding Coupon code.

**Narration-** This column in the grid corresponds to Coupon description.

**Available Uses-** This column in the grid corresponds to the no of times the corresponding coupon can still be used by the member.

**Action- Redeem-** This CTA populates the coupon code selected to redeem with two CTA Redeem & Clear

**Redeem CTA-** blocks the coupon & calls Redeem Coupon API.

**Clear CTA-** This CTA clears the coupon code selected to redeem & another coupon code can be selected to redeem.

Implicit calls are made to Coupon OTP & Resend OTP from Redeem Coupon API through the Redeem Coupon widget level.

ER API's that will be implicitly invoked from the Redeem Coupon widget level will be:

Get Available Coupons API

Redeem Coupon API- Invoked on the click of Redeem CTA

Coupon OTP

Resend OTP- Invoked on the click of Resend CTA

Unblock Coupon API-- Invoked on the click of Cancel CTA

### 6.2.1. Input Parameters for Redeem Coupon Widget URL

Parameter Name	Description	Data Type	Length
SecurityToken (*)	This is a unique token Id used to authenticatethe request.	Varchar	512
StoreCode (*)	Store Code	Varchar	50
UserName (*)	Partner User Name	Varchar	64
MemberID (*)	Mobile no of the member	Number	10
WidgetCode (*)	This corresponds to the pre-defined code of the Widget in LPaaS. Every widget type will have a unique widget code. The widget codesused are: <ul style="list-style-type: none"> <li>➤ AEC-Add/Edit Customer</li> <li>➤ VC- View Customer</li> <li>➤ RC- Redeem Coupon</li> <li>➤ RP- Redeem Points</li> </ul>	Varchar	10
BillGUID	This corresponds to the unique identifier ofthe bill. This will be unique identifier generated from the POS.	Varchar	64

### 6.2.2. Sample Request/Response for the Redeem Coupon widget


Type	XML
Request for RC widget Invocation passed in wsGetWorkflow Widget	<pre> &lt;Request&gt;   &lt;SecurityToken&gt;PhUtWS4aUzb0LuOo9iuZwCDrtSYMs/Y+6DYZyzomO   RIbgST3OMrHBRV   ImO/bQyPqxZ34kRJI4WXjwEhAh9xW4pbBGyLQoqkCMRhhUA3cOZRG4adfWI   dt+vDmwDcNdVJC   x8K7QGqZS5X6L55r7nsmIWE4dpVMafMrb4qltLztPw4=&lt;/SecurityToken&gt;   &lt;MemberId&gt;9640830846&lt;/MemberId&gt;   &lt;UserName&gt;apparaod@revalsys.com&lt;/UserName&gt;   &lt;StoreCode&gt;demo01&lt;/StoreCode&gt;   &lt;WidgetCode&gt;RC&lt;/WidgetCode&gt;   &lt;BillGUID&gt;sample&lt;/BillGUID&gt; &lt;/Request&gt; </pre>
Response	<pre> &lt;?xml version="1.0"?&gt; &lt;Response   xmlns:xsd="http://www.w3.org/2001/XMLSchema"   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"&gt;   &lt;ReturnCode&gt;0&lt;/ReturnCode&gt;   &lt;ReturnMessage&gt;Success&lt;/ReturnMessage&gt;   &lt;WidgetURL&gt;http://SOCHPWW.erstaging.com/PWW/RedeemCoupon.   aspx?RequestCo de=SOCH_952776_17082018113315479&lt;/WidgetURL&gt; </pre>

</Response>

P.S: The BillGUID used in the request for the Redeem Coupon widget API is the system generated unique identifier for a bill (generated at POS end) which is passed & used at ER as the unique identifier of the bill

### 6.2.3. Redeem Coupon Widget Screen

Redeem Coupon



siv deepu

Mobile : 9640830846

Have a coupon Code?

Coupon Code	Validity	Narration	Available Uses	
YTXXXXXQ	31 Aug 2018		1	Redeem
GRXXXXUB	24 Oct 2018		1	Redeem
GSXXXX1L	17 Oct 2018		2	Redeem
TEXXXXNU	31 Dec 2018		5	Redeem
TXXXXE	30 Aug 2018		10	Redeem
TEXXXXWZ	25 Sep 2018		5	Redeem

Redeem Coupon

Coupon Code **TEXXXXNU**

Redeem

Clear



## 6.3. Add/Edit Customer Widget

**Widget Name:** Add/Edit Customer

**Widget Code:** AEC

**Description:** The url for Add/Edit customer widget returned from ER to POS will open the Add/Edit customer widget at the POS. Two functionalities can be performed using this widget:

1. **Add Customer Functionality:** To register a member in loyalty program
2. **Edit Customer Functionality:** To update the profile of a registered member

### 1. Add Customer:

**Approach:** LPaaS supports multiple ways of enrollment process. It can be based on Mobile number, Membership card no. or both. Membership no. can be from pre-mapped series or dynamically allocated. To enroll a member, Register Easy Account API is called from the AEC widget level.

**Pre-Condition:** Security Token needs to be generated before calling this method. Each outlet (POS store code, online web channel, Mobile App etc.) needs to have a unique code assigned to them. All the configuration needs to be present in LPaaS before Register Easy Account method is called through widget.

**Post-Condition:** After registration Member is ready to accrue and redeem in the system

### 2. Edit Customer

**Approach:** There are scenarios where customer data needs to be modified or more details to be entered at later visits or via URLs. There we need to call the Update Member Profile API which is also invoked from the AEC widget level. It will update all data accepted via RegisterEasyAccount API except mobile number.

**Pre-Condition:** Customer should be enrolled before by RegisterEasyAccount API.

**Post-Condition:** Customer can be assigned coupons via Sms or email on updating their profile data.

#### **Process:**

The POS will need to integrate ER's widget invocation API.

The name of the method to invoke the workflow widget is GetWorkflowWidget & the rest URL to call this API is {{BaseUrl}}/GetWorkflowWidget

When the request parameters for the Add/Edit Customer widget is passed, the Add/Edit customer widget URL is returned in response of the API.

This URL redirects to the web page of the Add/Edit Customer widget.

#### Input Parameters for AEC Widget Request

Parameter Name	Description	Data Type	Length
SecurityToken (*)	This is a unique token Id used to authenticate the request.	Varchar	512
StoreCode (*)	Store Code	Varchar	50
UserName (*)	Partner User Name	Varchar	64
MemberID (*)	Mobile no of the member	Number	10
WidgetCode (*)	This corresponds to the pre-defined code of the Widget in LPaaS. Every widget type will have a unique widget code. The widget codes used are:  AEC-Add/Edit Customer  VC- View Customer  RC- Redeem Coupon  RP- Redeem Points	Varchar	10

#### Output Parameters

In response to the request parameters passed for the AEC widget invocation, we get below Output parameters:

**Return Code:** It can be "0" for success & specific failure code for failure

**Return Message:** It can be "Success" or predefined error message corresponding to the error code returned. Widget

**URL:** It will have the Program Code, PWW (stand for Program Workflow widget), environment (staging or production), Widget Name and Request Code.

**Request Code:** will have a mobile token generated specific for the request. This will be stored in cache and will have validity of 30 seconds.

#### Rules & Validations

- AEC widget URL generated opens the web page with the input parameters for the member profile details like name, DOB, Gender etc.
- If the member ID passed in the request for AEC widget is not enrolled with the program, then the AEC widget URL generated will open the web page with the member details parameter with all the input parameters blank except the mobile no (same as passed in the request).

- If the member ID passed in the request for AEC widget is already enrolled with the program, then the AEC widget URL generated will open the web page with the member details parameter with those input parameters pre-populated for which the details are already stored in the system.
- In both the above cases, all the fields except mobile no are editable.

### 6.3.1. Sample Request & Response for AEC customer widget

Type	XML
Request for AEC widget Invocation	<pre> &lt;Request&gt;     &lt;SecurityToken&gt;PhUtWS4aUzb0LuOo9iuZwCDrtSY     Ms/Y+6DYZyzo     mORlbgST3OMrHBRVImO/bQyPqxZ34kRJI4WXjwEhAh9     xW4pbBGyLQoqk     CMRhhUA3cOZRG4adfWldt+vDmwDcNdVJCx8K7QGqZS5X     6L55r7nsmIWE     4dpVMafMrb4qItLztPw4=&lt;/SecurityToken&gt;     &lt;MemberId&gt;9640830846&lt;/MemberId&gt;     &lt;UserName&gt;apparaod@revalsys.com&lt;/UserName&gt;     &lt;StoreCode&gt;demo01&lt;/StoreCode&gt;     &lt;WidgetCode&gt;AEC&lt;/WidgetCode&gt; &lt;/Request&gt; </pre>
Response for AEC widget	<pre> &lt;?xml version="1.0"?&gt; &lt;Response     xmlns:xsd="http://www.w3.org/2001/XMLSchema"     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"&gt;     &lt;ReturnCode&gt;0&lt;/ReturnCode&gt;     &lt;ReturnMessage&gt;Success&lt;/ReturnMessage&gt;     &lt;WidgetURL&gt;http://SOCHPWW.erstaging.com/P     WW/AddMem     ber.aspx?RequestCode=SOCH_181197_1608201813460     4695&lt;/Widget URL&gt; &lt;/Response&gt; </pre>

### Sample Widget Screens

➤ Edit Member screen

If the member already exists then the AEC widget opens with the update profile option as shown below:

**Update Profile**

First Name

siv

Last Name

deepu


Email\*


MANRETTT@GMAIL.COM

Mobile No\*


9640830846

Gender






D.O.B




11/10/1988

Child DOB



20/07/2012

Expected Date Of Delivery



Expected Date Of Delivery

Tier

Gold ▼

Are you from a corporate firm ?\*

☐ Yes

☒ No

Do you have referral code

☒ Yes

☐ No

Referral Code\*

RE26UQ3Z

Customer Type

Loyalty ▼

Cancel

Update

➤ Add Member Screen

If the MemberID passed in the request does not exists, then through AEC widget we can enroll the member

**Enroll New Member**

First Name

First Name

Last Name

Last Name


Email\*


Email Id

Mobile No\*


8435050666

Gender






D.O.B




DOB

Child DOB



Child DOB

Expected Date Of Delivery



Expected Date Of Delivery

Tier

-- Select Tier--

Are you from a corporate firm ?\*

☐ Yes

☒ No

Do you have referral code

☐ Yes

☒ No

Customer Type

-Select Customer Type-

Cancel

Add Member

## 6.4. View Customer Widget

**Widget Name:** View Customer

**Widget Code:** VC

**Description:** This widget provides complete information about the member's profile which includes member details added through Add/Edit customer widget, member transaction details, member reward details. It displays member available points, list of available coupons/GV's.

### 6.4.1. Input Parameters for View Customer widget

Parameter Name	Description	Data Type	Length
SecurityToken (*)	This is a unique token Id used to authenticate the request.	Varchar	512
StoreCode (*)	Store Code	Varchar	50
UserName (*)	Partner User Name	Varchar	64
MemberID (*)	Mobile no of the member	Number	10
WidgetCode (*)	<p>This corresponds to the pre-defined code of the Widget in LPaaS. Every widget type will have a unique widget code. The widget codes used are:</p> <p>AEC-Add/Edit Customer</p> <p>VC- View Customer</p> <p>RC- Redeem Coupon</p> <p>RP- Redeem Points</p>	Varchar	10

#### 6.4.2. Sample Request & Response for the View Customer Widget

Type	XML
Request forView Customer Widget Invocation	<pre> &lt;Request&gt;   &lt;SecurityToken&gt;PhUtWS4aUzb0LuOo9iuZwCDrtSYM5/Y+ 6DYZyzomORlbgST3OMrHBRVImO/bQyPqxZ34kRJI4WXjw EhAh9xW4pbBGyLQoqkCMRhhUA3cOZRG4adfWldt+vDm wDcNdVJCx8K7QGqZS5X6L55r7nsmIWE4dpVMafMrb4qIt LztPw4=&lt;/SecurityToken&gt;   &lt;MemberId&gt;9640830846&lt;/MemberId&gt;   &lt;UserName&gt;apparaod@revalsys.com&lt;/UserName&gt;   &lt;StoreCode&gt;demo01&lt;/StoreCode&gt;   &lt;WidgetCode&gt;VC&lt;/WidgetCode&gt; &lt;/Request&gt; </pre>
Response ofView Customer Widget API	<pre> &lt;?xml version="1.0"?&gt; &lt;Response xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema- instance"&gt;   &lt;ReturnCode&gt;0&lt;/ReturnCode&gt;   &lt;ReturnMessage&gt;Success&lt;/ReturnMessage&gt;   &lt;WidgetURL&gt;http://SOCHPWW.erstaging.com/PWW/Cus tomerProfile.aspx?RequestCode=SOCH_842383_170820 18115838255&lt;/WidgetURL&gt; &lt;/Response&gt; </pre>

### 6.4.3. Customer Widget Screen

#### Member Profile




siv deepu Gold

Email : MANRETTT@GMAIL.COM

Member Since :06 Jun 2018

Favourite Store : demo

Mobile : 9640830846

Referral Code : RE26UQ3Z 

Date of Birth :11 Oct 1988

Member  
For  
**2**  
months

Available  
Points  
**1,02,671**

Last  
Visit  
**30 Jul 2018**  
**01:03:50 PM**

Nearest points lapsing	
30 Jul 2018	400 Points
30 Jun 2019	100000 Points
31 Jul 2019	60 Points

Period	Spends	Visits	ATV
Lifetime Sales	Rs. 6,900.00	9	767
YTD Sales	Rs. 21,605.00	60	360
Last 90 days	Rs. 21,605.00	60	360

Referral Rewards		
Referral Count	Referral Points	Remaining Referrals
0	0	0

When clicked on Show detailed profile, the screen expands with below details.

Transactions Coupons Gift Vouchers

Coupon Code	Expiry Date	Benefit
YTXXXXQ	31 Aug 2018	500.0000
GRXXXXUB	24 Oct 2018	100.0000
GSXXXX1L	17 Oct 2018	100.0000
TEXXXXNU	31 Dec 2018	100.0000
TXXXXE	30 Aug 2018	100.0000
TEXXXXWZ	25 Sep 2018	100.0000
J1XXXXKY	31 Dec 2018	500.0000

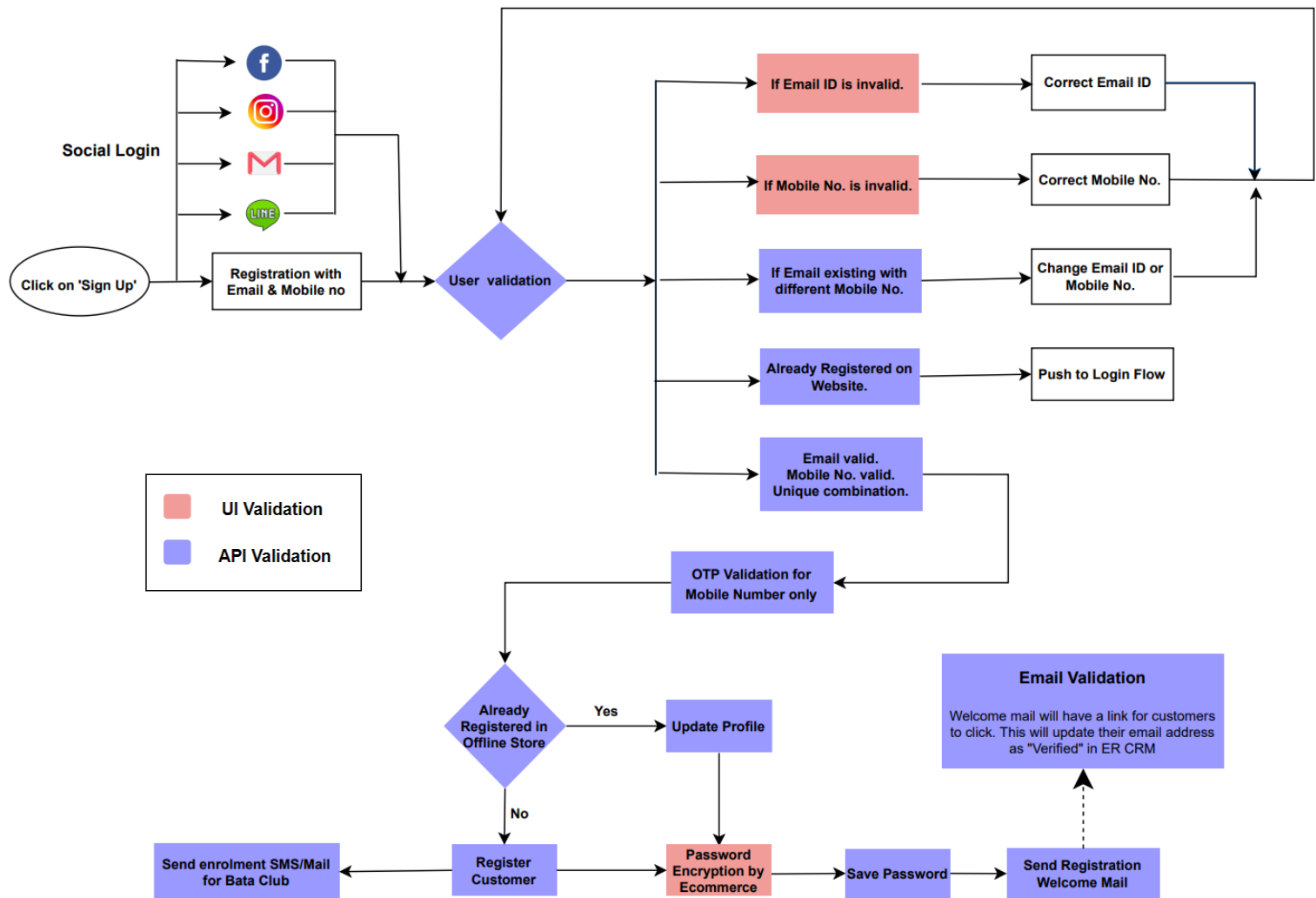


Hide Details

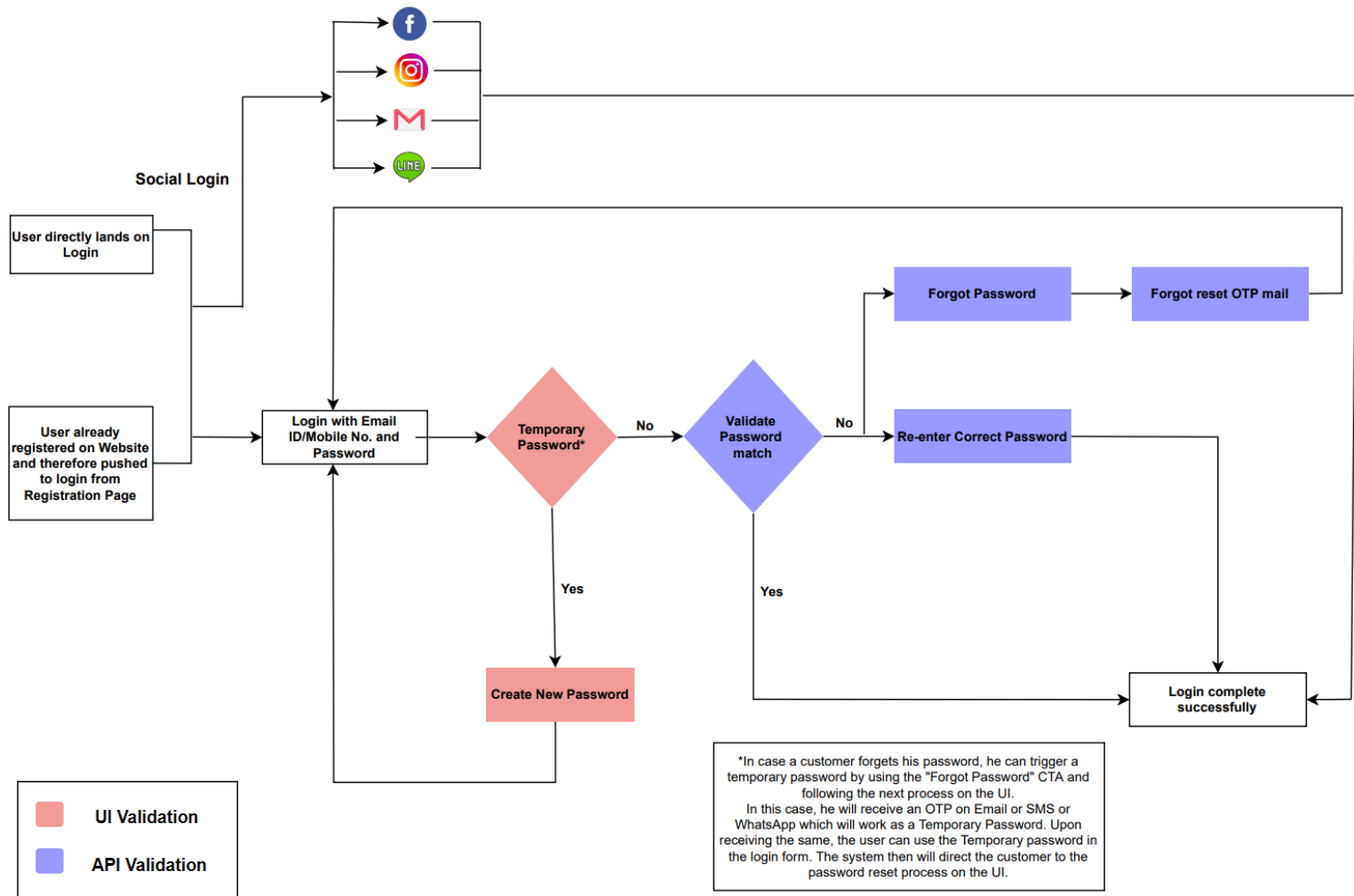


## 7. Ecommerce Workflows

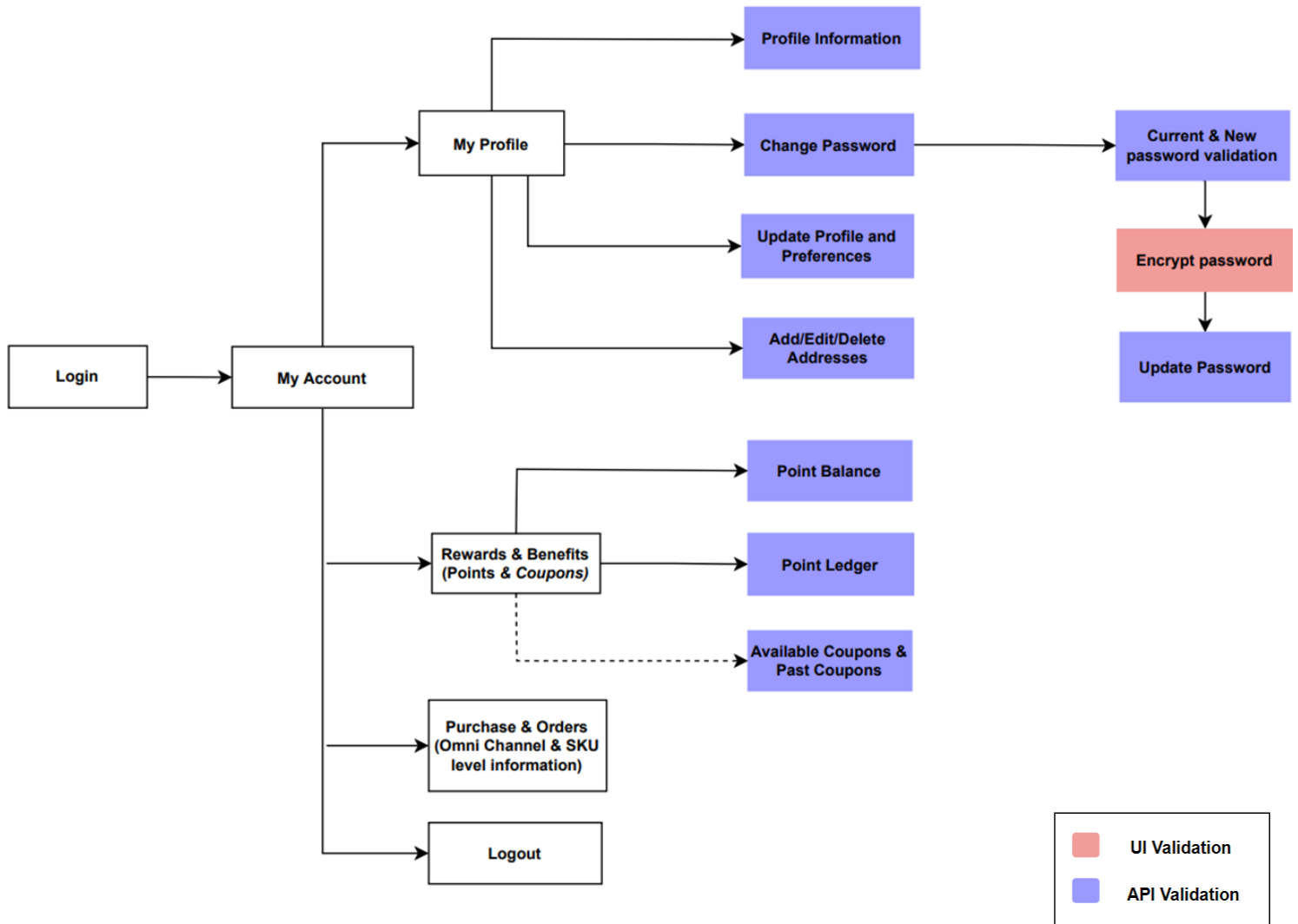
### 7.1. Registration Flow



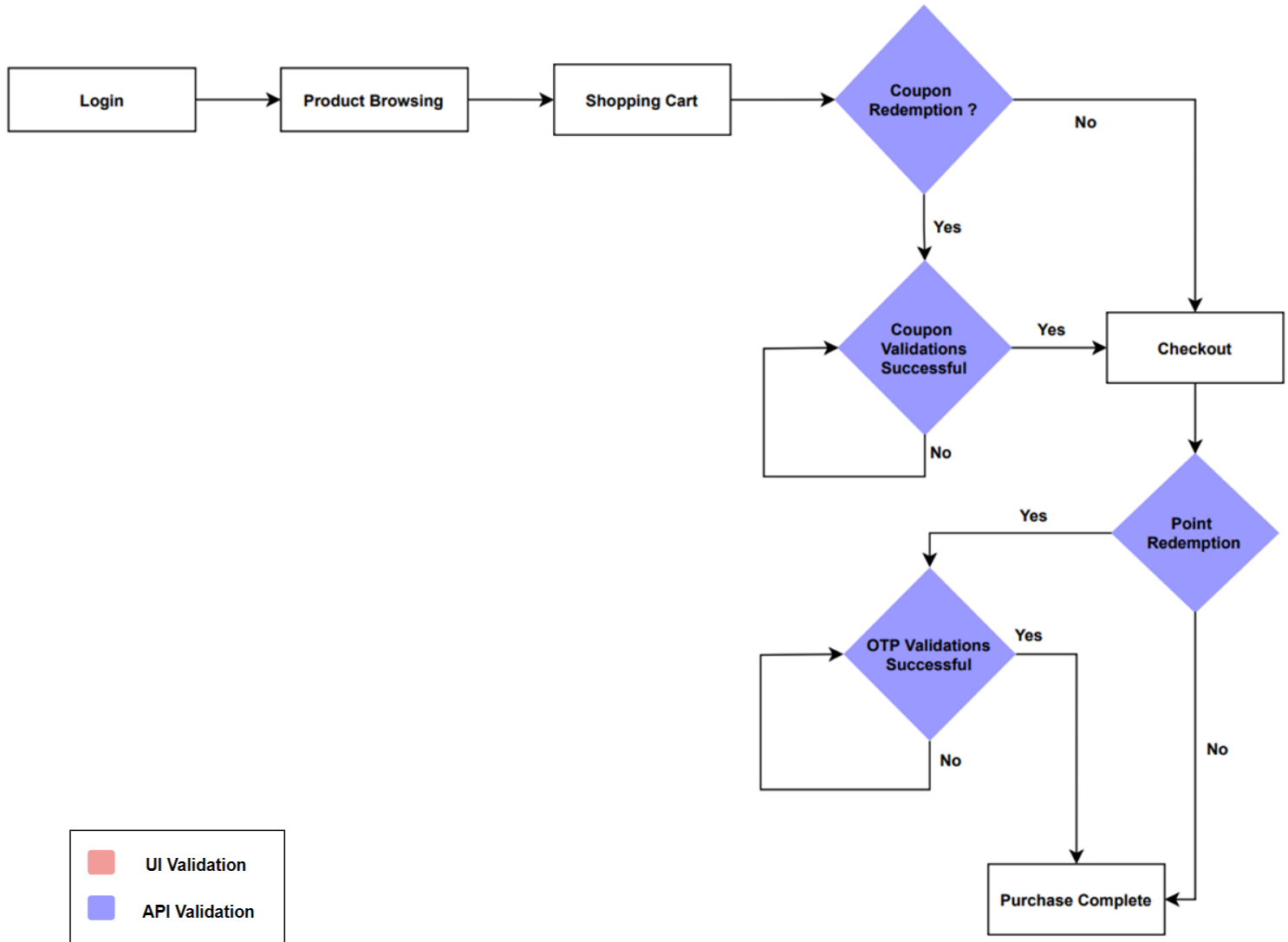
## 7.2. Login Flow



### 7.3. My Account Flow



## 7.4. Redemption Flow



## 7.5. Accrual Flow

