



TIBCO[®] Reward

API Reference Guide

Version 22.3.0
August 2022



Contents

Contents	2
Introduction to TIBCO® Reward APIs	5
Integrate TIBCO Reward with Other Applications	5
Supported Operations	5
Objects and the API	6
Standards Compliance	6
Development Platforms	6
API Support Policy	6
Related Resources	7
General Data Protection Regulation	7
Loyalty API Batch Processing and Integration	12
Record Level Business Process Application	12
Dates and Times	12
Shopper Matching	12
Bulk Data Processing	13
Null Data Processing	13
TIBCO Reward Entities	14
Entity List	14
Related Actions or Operations	16
API Entity Attribute Description Tables	17
Entity Processing Information and Usage Notes	17
API Entity Attributes and Properties	19
Shopper Entity Attributes and Properties	19
Offer Status and Offer Attribute Properties	27
Registered Card Attribute Properties	38

Event Instance Properties	42
Reward Product Properties	44
Overview of APIs	47
Characteristics of API Calls	47
Typical API Call Sequence	48
Start with Authentication	48
List of APIs	48
Overview of Objects or Entities	65
Security and the API	66
User Authentication	66
Implicit Restrictions for Objects and Fields	67
Error Handling	67
Error Handling for Session Expiry	67
Error Handling for Unauthorized Data	67
Error Handling for API	68
Generic Error Codes	69
API Related Error Codes	69
TravelSuite API Error Codes	72
List of APIs and Related Error Codes	73
Implementation Considerations	75
Preparing to Use the API	77
Ensure Your Company is Configured for APIs	77
Obtain a TIBCO Reward API User Account	77
Generate or Obtain the Web Service WSDL File	78
Import the WSDL File into Your Development Platform	78
Walk Through the Sample Code	80
Sample Code Walkthrough	82
C# Sample Code	82
TIBCO Documentation and Support Services	85

Legal and Third-Party Notices **87**

Introduction to TIBCO® Reward APIs

This chapter introduces TIBCO Reward application programming interfaces (APIs). TIBCO Reward's APIs provide simple and secure access to a client's data.

Integrate TIBCO Reward with Other Applications

TIBCO Reward APIs enable data exchanges between your internal systems or another third party provider and TIBCO Reward.

See [Overview of APIs](#) for an introduction to API concepts.

Supported Operations

Using your existing development environment, you can construct web service client applications that use standard web service protocols to:

- Log in and receive authentication information to be used for subsequent calls
- Retrieve customers via various identifiers
- Create, retrieve, and edit a wide range of customer data
- Incorporate capabilities of TIBCO Reward's customer-facing loyalty program website into your own web environment, including offers and rewards.

For each operation, client applications submit a synchronous request to the API, await the response, and process the results.

The API commits any changed data automatically.

See [Loyalty API Batch Processing and Integration](#) for a summary of how API processing compares to TIBCO Reward batch processing.

Objects and the API

In database modeling, an entity represents a discrete object. In this document, the two terms, entity and object, are nearly equivalent. You can use the API to perform query, insert, update, and delete operations on select Entities. Entities are representations of your company's data. Entity attributes represent fields in those entities, and client applications set or retrieve data values via these operations. For example, customers are represented by a Shopper Entity, and a Shopper Entity has attributes (fields) representing First Name, Last Name, Phone Number, Email Address, Customer Tier, Point Balance, etc.

Standards Compliance

The API is implemented to comply with the following specifications:

- **Simple Object Access Protocol (SOAP) 1.1**
<http://www.w3.org/TR/2000/NOTE-SOAP-20000508>
- **Web Services Description Language (WSDL) 1.1**
<http://www.w3.org/TR/2001/NOTE-wsdl-20010315>

Development Platforms

The API works with current Simple Object Access Protocol (SOAP) development environments, including, but not limited to, Visual Studio.NET and Apache Axis. In this document, we provide examples in Java and C# (.NET). The Java examples are based on Apache Axis 1.3 and JDK 5.0 onwards (Java 2 Platform Standard Edition Development Kit 5.0). For more information about Apache Axis 1.3, go to <http://axis.apache.org/axis/>.

API Support Policy

TIBCO Reward recommends that your client applications use the most recent version of the TIBCO Reward Web Services Description Language (WSDL) file. When you first implement an API or when a new version is released, use the following steps to update your WSDL:

- Generate the WSDL file (see [Generate or Obtain the Web Service WSDL File](#)), then

import it into your environment (see [Import the WSDL File into Your Development Platform](#)).

TIBCO Reward maintains all prior versions of its APIs to ensure backward-compatibility. Currently, supported solutions written with earlier versions of the API will continue to work unmodified.

Under certain circumstances, TIBCO Reward may “retire” older API calls. As a result, updated WSDL files may not include older APIs. Prior solutions written with older versions of the WSDL file generally will work with a newer version, other than cases of a retired API call.

Advanced notice of API retirement will be given well in advance of the retirement, usually at least six months, although unforeseen circumstances may force updates at any time. All changes to the API Set will be communicated by TIBCO Reward’s Client Services team, posted to our extranet, posted to our RSS announcement feed, and emailed to your named account representative.

Related Resources

TIBCO Reward's entire collection of documents and technical references about the API is accessible at the following location: <https://docs.tibco.com/products/tibco-reward>.

[The LoyaltyAPI Class definitions site](#) is the fundamental TIBCO Reward API information resource.

This document, *TIBCO Reward API Reference*, is an adjunct to the *TIBCO Reward Integration Guide*, which provides guidance and information regarding the transfer of data through batch files.

General Data Protection Regulation

As per the General Data Protection Regulation (GDPR), TIBCO Reward has implemented AnonymizeShopper API so that customers can remove all their personal identifiable information from Reward.

AnonymizeShopper API when invoked for a shopper using one of the retailer supported unique identifiers, anonymizes the Personal Identifiable Information(PII) and Payment Card Industry(PCI) data.

PII data includes **Firstname, LastName, Password, Address, PhoneNumber, UniqueIdentifiers (RetailerShopperId, EmailAddress, Username, LoyaltyCard, CreditCard and so on).**

PCI data includes **RegisteredCards.**

After a shopper is anonymized, the system anonymizes the identified PII and PCI data and then unenrolls the shopper from the loyalty program and from that point the shopper is not identifiable in the system.

Authentication Request is a prerequisite for calling AnonymizeShopper API.

API Authentication Request

```
<?xml version="1.0"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <AuthenticateUserResponse
xmlns="http://www.loyaltylab.com/loyaltyapi/">
      <AuthenticateUserResult>
        <RetailerGuid>878ijgbh-dsfd-vfvv-hg78-
vsdvrgvvhgfg</RetailerGuid>
        <Authenticated>>true</Authenticated>

<Token>565857byftvif7st6d5d7ho8sdg6s7d65sg7d4sd6bst6dts5dgshd67sbdst</To
ken>
        <ICSUserID>5645</ICSUserID>
      </AuthenticateUserResult>
    </AuthenticateUserResponse>
  </soap:Body>
</soap:Envelope>
```

API Authentication Response

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <AuthenticateUserResponse
xmlns="http://www.loyaltylab.com/loyaltyapi/">
      <AuthenticateUserResult>
        <RetailerGuid>878ijgbh-dsfd-vfvv-hg78-
vsdvrgvvhgfg</RetailerGuid>
        <Authenticated>>true</Authenticated>
```

```

    <Token>ubysdsbdsudtsnidsbudssdsdsferewr-
ZlYw6L2+ererergfvdvdv/fdffefgeffcee.</Token>
    <ICSUserID>5645</ICSUserID>
  </AuthenticateUserResult>
</AuthenticateUserResponse>
</soap:Body>
</soap:Envelope>

```

API AnonymizeShopper Request using the authenticated token

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <AuthenticationResult xmlns="http://www.loyaltylab.com/loyaltyapi/">
      <RetailerGuid>878ijgbh-dsfd-vfvv-hg78-vsdvrgvvhgfg</RetailerGuid>
      <Authenticated>true</Authenticated>
      <Token>ubysdsbdsudtsnidsbudssdsdsferewr-
ZlYw6L2+ererergfvdvdv/fdffefgeffcee.</Token>
      <ICSUserID>5645</ICSUserID>
    </AuthenticationResult>
  </soap:Header>
  <soap:Body>
    <AnonymizeShopper xmlns="http://www.loyaltylab.com/loyaltyapi/">
      <shopperIdentifier>
        <IdentifierType>RetailerShopperId</IdentifierType>
        <IdentifierValue>22732885</IdentifierValue>
      </shopperIdentifier>
    </AnonymizeShopper>
  </soap:Body>
</soap:Envelope>

```

API AnonymizeShopper Response on Success

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <AnonymizeShopperResponse
xmlns="http://www.loyaltylab.com/loyaltyapi/">
      <AnonymizeShopperResult>Anonymize shopper request is processed
successfully!</AnonymizeShopperResult>
    </AnonymizeShopperResponse>
  </soap:Body>
</soap:Envelope>

```

```

</soap:Body>
</soap:Envelope>

```

API AnonymizeShopper Response on Failure scenario I

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <soap:Fault>
      <faultcode>C204</faultcode>
      <faultstring>LoyaltyAPILibrary.SoapCustomAPIException: Unable to
find shopper from the specified RetailerShopperId
      at
LoyaltyAPIEntities.LoyaltyAPIBaseAuthenticatedService.LogAndThrowCustomE
xception(Exception ex, NameValueCollection additionalExceptionInfo,
Int32 errorCode, String message, Boolean includeFaultCode) in
g:\p4Úy.Kanagala_UBHANUPR-Z600VM_
1737\LoyaltySystem\Development\InStorecard\Web\Services\LoyaltyAPI\CodeB
ehind\LoyaltyAPIUtility.cs:line 1242
      at LoyaltyAPI.LoyaltyAPI.AnonymizeShopper(ShopperIdentifier
shopperIdentifier) in g:\p4Úy.Kanagala_UBHANUPR-Z600VM_
1737\LoyaltySystem\Development\InStorecard\Web\Services\LoyaltyAPI\CodeB
ehind\LoyaltyAPI.cs:line 3469</faultstring>
      <detail>
        <code>204</code>
        <description>Unable to find shopper from the specified
RetailerShopperId</description>
      </detail>
    </soap:Fault>
  </soap:Body>
</soap:Envelope>

```

API AnonymizeShopper Response on Failure scenario II

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <soap:Fault>
      <faultcode>C416</faultcode>
      <faultstring>LoyaltyAPILibrary.SoapCustomAPIException: Shopper is
already Anonymized

```

```
at
LoyaltyAPIEntities.LoyaltyAPIBaseAuthenticatedService.LogAndThrowCustomE
xception(Exception ex, NameValueCollection additionalExceptionInfo,
Int32 errorCode, String message, Boolean includeFaultCode) in
g:\p4Úy.Kanagala_UBHANUPR-Z600VM_
1737\LoyaltySystem\Development\InStorecard\Web\Services\LoyaltyAPI\CodeB
ehind\LoyaltyAPIUtility.cs:line 1242
    at LoyaltyAPI.LoyaltyAPI.AnonymizeShopper(ShopperIdentifier
shopperIdentifier) in g:\p4Úy.Kanagala_UBHANUPR-Z600VM_
1737\LoyaltySystem\Development\InStorecard\Web\Services\LoyaltyAPI\CodeB
ehind\LoyaltyAPI.cs:line 3477</faultstring>
  <detail>
    <code>416</code>
    <description>Shopper is already Anonymized</description>
  </detail>
</soap:Fault>
</soap:Body>
</soap:Envelope>
```

Loyalty API Batch Processing and Integration

The Loyalty API and batch file processing are the two key technologies for data transfer between TIBCO Reward and its clients. The two processes are designed to serve different needs, and as such behave differently.

Record Level Business Process Application

Batch Process: The batch process enforces certain data requirements and business logic based on the set up decisions you made during the TIBCO Reward onboarding process.

Loyalty API: The Loyalty API will not apply any business logic to your incoming data regardless of application settings. API use assumes that business logic is implemented and applied in the application that calls the API. The application set up completed during the onboarding process does not apply to API processing.

Dates and Times

Batch Process: Batch processing will apply default date and time stamps to various data.

Loyalty API: The Loyalty API will not automatically populate all relevant date-time fields. However, you can populate them yourself using external business logic.

Shopper Matching

Batch Process: Using the batch process you can configure multiple matching criteria (unique shopper identifiers). This ensures that shoppers are matched correctly. For example, this ensures that shoppers are correctly matched to transactions when you are using a transaction batch file.

Loyalty API: The Loyalty API will only enforce email address uniqueness when adding a record. If you use other customer identifiers, TIBCO Reward suggests that you use available Loyalty API methods to check for uniqueness before adding the record.

Bulk Data Processing

Batch Process: The batch process is designed for inserting multiple records at once, thus much of the processing overhead is streamlined for this process.

Loyalty API: The Loyalty API is meant for singleton manipulation of data. Although processing may occur almost instantaneously, only one record is processed at a time.

Null Data Processing

Batch Process: The batch process will not update a non-null field to a null. If null data is passed through by means of a batch file and there is existing data, the existing information in the database will be retained.

Loyalty API: The Loyalty API will allow you to null out any field that is “nullable” (refer the "Nulls Allowed" column in the [API Entity Attributes and Properties](#) topic).

TIBCO Reward Entities

TIBCO Reward CRMS is built around common entities like Shoppers and Offers. TIBCO Reward APIs perform actions on these objects, primarily the base CRUD functions (Create, Read, Update, and Delete).

A typical use of TIBCO Reward APIs includes the following:

- an authentication step
- requests for information from TIBCO Reward for use by the source application
- additions or updates to the TIBCO Reward entities

Entity List

The following entities are exposed via the APIs: (See [API Entity Attributes and Properties](#) for more specific information about the entities you will use).

Shopper

Description: A representation of a registered member of your loyalty program. Includes a wide range of attributes available for use in your applications.

Available Actions:

- Get
- Create
- Score
- Update
- Deactivate
- Get Point Balance for Shopper

Many retrieval keys are honored for this entity.

Programs

Description: Programs have benefits, creative assets, and other attributes. Currently, there are three programs: Main, Tier, and Club. Your enrolled customers can be members of either a tier (T) program or a main (M) program, but not both. If you have a main (M) program, each enrolled customer is enrolled in the main program. If you use a tier (T) program, each enrolled customer must belong to just one of the tiers. Whether you use the main or the tier program, you can also use club programs to further segment your customers and provide additional benefits. Customers can be enrolled in zero-to-many clubs.

Available Actions: Get, Join Shopper to Program.

UnregisteredShopper

Description: A representation of a customer who is not a member of your loyalty program.

Available Actions: Create, Update, Get.

RegisteredCard

Description: A credit or loyalty card (or other identifier) that is associated to a specific shopper.

Available Actions: Create, Get.

CustomAttributes

Description: A set of client-defined attributes describing a shopper. Maintained separate from the Shopper entity itself.

Available Actions: Get, Update Value.

Rewards

Description: A representation of the rewards available to a shopper.

Available Actions: Get

Authentication Token

Description: A session-specific identifier needed for every API interaction.

Available Actions: Get.

Offers

Description: A representation of the offers available to a shopper. Includes a qualification rule and an award.

Available Actions: Get, Update Status for Shopper.

Reminder

Description: A representation of the reminders set by a shopper.

Available Actions: Get, Update for Shopper.

Event Instance

Description: An instance of a generic event (i.e., an action taken by a customer).

Available Actions: Create event instance.

Related Actions or Operations

For a list of the actual API calls that perform the actions listed for the entities above, see [List of APIs](#). See also the listed methods at the API site: <http://api.loyaltylab.com/loyaltyapi/loyaltyapi.asmx>.

API Entity Attribute Description Tables

This topic provides information describing API entity attributes.

Use the API to perform operations on selected entities. Entities are representations of your company's data, and entity attributes represent fields in those entities. Your applications configure or retrieve data values representing these attributes via these API operations. For example, customers are represented by a Shopper Entity, and a Shopper Entity has attributes (fields) representing First Name, Last Name, Phone Number, Email Address, Customer Tier, Point Balance, etc.

The tables below reveal details about the standard Entities upon which TIBCO Reward's APIs perform operations, especially at the attribute level. These tables include metadata and explanations about individual entity attributes that do not appear elsewhere in the API documentation.

The information in these tables is meant to be used in conjunction with the formal API description available at this site: <http://api.loyaltylab.com/loyaltyapi/loyaltyapi.asmx>.

Entity Processing Information and Usage Notes

- API processing is not a substitute for high-volume batch file processing. To process large quantities of data, use standard TIBCO Reward batch processing methods. Batch file processing also performs many data validation functions. For details, see the TIBCO Reward Integration Guide.
- Unless stated otherwise, attributes have the same meaning as fields in batch files with the same names, as these fields are defined in the TIBCO Reward Integration Guide.
- API processing includes no data validation.
- If clients want to validate data, they must implement local data checking edits and processing.
- Email address validation is not enforced through APIs, so the API can create a shopper with an invalid email address. If not caught and corrected, this would be an

unserviceable customer record.

- API processing physically allows you to insert null values, even where these can cause problems. The tables provided in this chapter indicate which fields must not be nulled.
- TIBCO Reward's API uses generic error-handling messages to pass back exceptions for web service calls.
- If the TIBCO Reward system is unavailable because of network problems (at the client, at TIBCO Reward, or at any third party server/service) or because the TIBCO Reward system is down, no data transmitted via API can be processed or queued by TIBCO Reward. You, the client, should implement a loosely coupled queuing system in case such problems occur. If you are not familiar with the errors you might receive, see this article: <http://ws.apache.org/axis/java/client-side-axis.html>. It includes a section on troubleshooting network problems and what errors you might receive.
- Clients must handle SOAP exceptions. For examples, see the samples provided in [Preparing to Use the API](#).
- API methods for adding new shoppers must be used only to add new shoppers and not to update existing customers. Before you use any of the APIs to create shoppers, you must verify that a record does not already exist for this shopper.
- The Create and Score Shopper method illustrates some of the implications of these very basic rules:
 - The Create and Score shopper API will only work for new shoppers.
 - Clients must check whether the customer exists before calling this API.
 - If the customer does not exist, this API creates a new shopper and scores the shopper as eligible for offers.
 - If the customer does exist, this API does not fail. It returns the shopper data, and updates the create date, based on the date passed. However, it does not score the shopper, nor does it enroll the shopper as a loyalty program member.
 - If the customer exists, and you want to score the customer, then you must call the Add Shopper to Program API to enroll the customer. Overnight, the newly enrolled shopper will be scored.
- Email addresses must be unique, and can be used as unique customer identifiers. If a client uses the email address as its unique customer identifier, then it cannot create

a new customer record that has the same email address as an existing record in the database. An attempt to create such a new record will fail. However, if another customer ID (other than email address) is used, the API method cannot assume that the other ID is unique within the system, and the API will create a new record.

API Entity Attributes and Properties

This topic has five tables of API entity attributes and their properties:

- [Shopper Attributes and Properties](#)
- [Offer Attribute Properties](#)
- [Registered Card Properties](#)
- [Event Instance Properties](#)
- [Reward Product Properties](#)

Because the TIBCO Reward API is both flexible and powerful, it is crucial that you understand how the different properties interact with the loyalty systems. The APIs essentially provide you with the same options that a TIBCO Reward systems developer has when developing new features. However, with this opportunity comes risk. After implementing API components, you must rigorously test them.

Shopper Entity Attributes and Properties

Also known as the Customer Entity Attributes. The term "shopper" is equivalent to "customer" throughout.

Shopper Entity Attribute Properties - XML Table

Property Name	Description	Type	Length	Nulls Allowed	Sample Data	Default Data	Notes
ShopperID	TIBCO Reward	Int	4 bytes	No	123459876	None	Used in many API calls

Property Name	Description	Type	Length	Nulls Allowed	Sample Data	Default Data	Notes
	's generated unique identifier for customer						
RetailerGUID	Unique identifier for the Client/Retailer		16 bytes	No		None	On submit, set to microsite's retailer guid (globally unique identifier). Note: This should NOT be exposed in the API.
EmailAddress	Customer's email address	nvarchar	200	No	mstone@example.com	None	Valid email addresses follow this format: <alphanumeric>@<alphanumeric>.<alpha> Note: The API does not perform any data validation.
EmailFrequency	Not Used	int		No	7	7	Required: Pass in 7 for standard frequency
EmailFrequency	Not	char	1	No	D	D	Required: Pass in D

Property Name	Description	Type	Length	Nulls Allowed	Sample Data	Default Data	Notes
Unit	Used						for standard frequency
EmailFormat	The type of email to send out: TEXT or HTML.	nvarchar	200	No	HTML	HTML	Pass in HTML for standard emails
Password	Password	nvarchar	200	No	123456789a sdfghj	Empty String	This data is only updated if shopper information updates are enabled and AddressLine1 is provided. Stored as a one-way hashed value.
Status	Customer Status On submit, set to 'A'	char	1	No	A	no default	Should be correct for use with address. A indicates Active and enrolled, meaning the customer is a loyalty member and active.
LastName	Required	nvarchar	200	No	Doe	Empty String	

Property Name	Description	Type	Length	Nulls Allowed	Sample Data	Default Data	Notes
MiddleInitial	Not Required	nvarchar	200	Yes	B		
FirstName	Required	nvarchar	200	No	John	Empty String	
Address1	Not Required	nvarchar	200	No	412 Sixty-first Street	Empty String	
Address2	Not required. Second line of address detail.	nvarchar	200	No	Apt 666	Empty String	
City	Shopper's City	nvarchar	200	No	Oakland	Empty String	Should be correct for use with address.
State	Shopper's State	String	2		CA	Empty String	Should be correct for use with address.

Property Name	Description	Type	Length	Nulls Allowed	Sample Data	Default Data	Notes
Zip	Shopper's ZIP code	nvarchar	200	No	94609	Empty String	Required for location-based targeting.
PhoneNumber	Shopper's given phone number, including area code	nvarchar	200	Yes	510-555-1233	Empty String	Shopper phone number. Not used for shopper ID.
MobilePhoneEmail	Shopper's given phone number, including area code	nvarchar	200	Yes	510-555-2142	Empty String	Not currently used. API can set value.
ProfileCreateTime	Date when profile was originally created for custom	datetime		Yes	2012-06-26T00:00:00.0000000+02:00		On submit, set to NULL.

Property Name	Description	Type	Length	Nulls Allowed	Sample Data	Default Data	Notes
	er.						
ProfileDateUpdateTime	Date when profile was last updated for the customer	date time		Yes	2012-06-26T00:00:00.0000000+02:00		On submit, set to current date time.
CreateDateTime		date time		No	2012-06-26T00:00:00.0000000+02:00	Null	On submit, set to current date/time.
LoyaltyCreatedataTime		date time		Yes	2012-06-26T00:00:00.0000000+02:00	None	On submit, set to current date/time.
StatusUpdateDateTime		date time		Yes	2012-06-26T00:00:00.0000000+02:00	None	On submit, set to current date/time.
PasswordLastChanged		date time		No	null	None	On submit, set to NULL.
Origin	On	char	1	No		None	Can be used to

Property Name	Description	Type	Length	Nulls Allowed	Sample Data	Default Data	Notes
	submit, set to 'W' Not used					e	indicate where customer came from. Historically has been used inconsistently.
RetailerShoppeId	ID assigned to customer by client	nvarchar	200	Yes	123456789012	None	Use the unique identifier you use within your own system.
FileImportId		int		No	0	0 (zero)	Used for bulk imports.
BulkEmail	This is the commercial email opt-in flag. 1 = opted in (true/yes), 0 = NOT opted in	int		No	1	0 (zero)	In CRMS and CSR, this is the Email opt in. Note: export email functionality filters on 1 (opted-in = true).
BulkEmailSource	Not used	char	1	Yes	0	null	

Property Name	Description	Type	Length	Nulls Allowed	Sample Data	Default Data	Notes
BulkEmailSourceDescription	Not used	nvarchar	510	Yes		null	
LoyaltyMember	Flag indicating if this customer is a loyalty program member	bit		No	1	0	<p>This field determines LoyaltyMembership. Set to 1 when using CreateShopper;; Set to 0 when using CreateUnregisteredShopper.</p> <p>To enroll an existing unregistered shopper, set this to 1 using the UpdateShopper API, then use the AddShopperToProgram API.</p> <div style="border: 1px solid gray; padding: 5px; background-color: #f0f0f0; margin-top: 10px;"> <p>Note: Do not update this value.</p> </div>
RetailerRegistered	Not used	bit		No		0	
MailOptIn	Not used	bit		No		0	Not used.

Property Name	Description	Type	Length	Nulls Allowed	Sample Data	Default Data	Notes
PhoneOptIn	Not used	bit		No		0	
SourceReference	Not used	nvarchar	60	Yes		0	
RetailerShopperCreationDate	Creation date from Client	datetime		No	2012-05-31T00:00:00.0000000+02:00	Current Time	If no retailer data, set to current time.
PersonStatus	For TIBCO Reward internal use	char	1	No	P	Empty String	P for person.

Offer Status and Offer Attribute Properties

i Note: None of the Offer Entity attribute data are calculated data. This data is read-only. When updating a shopper's offer status, you should first request the OfferStatus values and then pass them back updated.

Offer Status and Attribute Properties - XML Data

Property Name	Description	Type	Length	Nulls Allowed	Sample Data	Default Data	Notes
IsClipped	Flag indicating whether the offer is clipped or not.	Bit	1	No	1	0	Is the offer clipped for the shopper? If the offer is a "clipped offer", and if the shopper qualifies for the offer, then the benefits of this offer are applied to the shopper.
LastUpdate	Date and time of last offer update.	Date Time	8	No	2012-05-31T00:00:00.000000+02:00	Last date time of offer score.	When was the offerstatus last updated for this shopper.
OfferScore	Relevance score applied to shopper-LL internal use	int	4	No	0	0	For use by TIBCO Reward only.

Property Name	Description	Type	Length	Nulls Allowed	Sample Data	Default Data	Notes
	only.						
Offer	Meta-description of the offer	Offer (see offer ID)	NA	No	NA	NA	See next table for composite object.
OfferId	Offer ID number assigned in CRMS.	int	4 bytes	No	3703	0	Unique offer ID.
RetailerGUID	Client-retailer's unique identifier.	GUID	16 bytes	No	<GUID>	None	Globally unique identifier used by a TIBCO Reward client.
OfferName	Name of offer as assigned in CRMS.	nvarchar	200	No	Sample offer	Empty string	
RecordType	Letter code	char	1	No	F	Empty string	The type of Offer. For example, O

Property Name	Description	Type	Length	Nulls Allowed	Sample Data	Default Data	Notes
	referring to the type of the offer.						refers to Offer.
QualifyingTypeId	Numeric code that refers to the type of qualification for the offer.	int	4	No	7	0 (zero)	The type of qualification for the Offer, where each type is represented by a number code.
AwardTypeId	Numeric code that refers to the award type for the offer, e.g., Points, Bonus.	int	4	No	5	1	The type of award associated with the Offer, where each type is represented by a number code.
Status	The	char	1	No	L	Empty string	L (Live)

Property Name	Description	Type	Length	Nulls Allowed	Sample Data	Default Data	Notes
	current status of the offer.						
StartDateTime	Start date for the offer.	datetime	8	Yes	2012-05-31T00:00:00.000000+02:00	Null	Start date for the offer
EndDateTime	End date for the offer.	datetime	8	Yes	2012-05-31T00:00:00.000000+02:00	Null	End date for the offer
QualifyStartDateTime	The starting date from when one can qualify the Offer.	datetime	8	Yes	2012-05-31T00:00:00.000000+02:00	Null	The date when a shopper can start qualifying for this offer.
QualifyEndTime	The date, after which a	datetime	8	Yes	2012-05-31T00:00:00.000000+02:00	Null	The date, after which a shopper can no longer

Property Name	Description	Type	Length	Nulls Allowed	Sample Data	Default Data	Notes
	shopper can no longer qualify for this offer.						qualify for this offer.
ActualStartDateTime	The date when the Offer actually started.	datetime	8	Yes	2012-05-31T00:00:00.000000+02:00	Current date time when offer is first awarded.	Actual date offer was made live.
ActualEndDateTime	The date when the Offer actually ended.	datetime	8	Yes	2012-05-31T00:00:00.000000+02:00	Null	Actual date the offer was ended.
ClipLimit	Not used.	int	4	No	1	0	Not used.
Headline	The headline for the	nvarchar	200	Yes	Introducing a New Offer	Null	The headline for the offer: pulled from data entered

Property Name	Description	Type	Length	Nulls Allowed	Sample Data	Default Data	Notes
	offer.						in CRMS.
OfferText	The actual text of the offer.	nvarchar	1000	Yes	Double Points on first purchase of Product X	Null	The actual text of the offer: pulled from Offer Description text entered in CRMS.
Terms	The terms of the offer.	nvarchar	4000	Yes	Offer valid while supplies last	Null	The terms of the offer: pulled from Offer Terms and Conditions text entered in CRMS.
ImageUrl	The URL for the image for the offer.	varchar	500	Yes	<URL>	Null	The URL for the image for the offer: pulled from data entered in CRMS.
MediaUrl	Reserved for future use. The media URL for the	varchar	500	Yes	<URL>	Null	not used.

Property Name	Description	Type	Length	Nulls Allowed	Sample Data	Default Data	Notes
	offer.						
BuyUrl	The buy URL for the offer.	varchar	500	Yes	<URL>	Null	The buy URL for the offer: pulled from Offer data entered in CRMS, specifically, the Offer description field labeled, 'Online Product Page URL to "Buy Now":'
InfoUrl	Reserved for future use. The information URL for the offer.	varchar	500	Yes	<URL>	Null	Reserved for future use.
IsAutoClip	Used to indicate if the offer is	int	4	No	0	Null	Used to indicate if the offer is auto clip, viewed or clipped:

Property Name	Description	Type	Length	Nulls Allowed	Sample Data	Default Data	Notes
	auto clip, viewed or clipped. Also specifies user acceptance behavior.						<ul style="list-style-type: none"> • AcceptOffer = 0 //Shopper must accept the offer * • ViewOffer = 1 //Shopper must view the offer* • AutoQualify Offer = 2 // no action required by Shopper <p>Shopper must accept or view the offer in the</p>

Property Name	Description	Type	Length	Nulls Allowed	Sample Data	Default Data	Notes
							offer showcase.
ShopperRedemptionLimit	The redemption limit per shopper for the offer.	int	4	No	1	1	The redemption limit per shopper for the offer. This is set in the CRMS.
TargetRedemption	Limit Not supported.	int	4	No	NA	NA	Not supported.
EmailNotificationThreshold	Not supported.	int	4	No	NA	NA	Not supported.
FlashTemplateStyleId	Code referring to a graphic template.	int	4	No	1	Empty string	Code referring to a graphic template.
UpdateDateTime	The last update time	datetime	8	No	2012-05-31T00:00:00.000000+	UpdateDateTime	The last update time for the offer.

Property Name	Description	Type	Length	Nulls Allowed	Sample Data	Default Data	Notes
	for the offer.				02:00		
CreatedByUserId	The user id of the user who created the offer.	int	4	No	NA	(assigned by system)	User Id - used internally for TIBCO Reward auditing.
LastModifiedByUserId	The user id of the user who last updated the offer. For use by TIBCO Reward.	int	4	No	NA	(assigned by system)	User Id - used internally for TIBCO Reward auditing.
IsAlwaysAwarded	Reserved for future use.	bit		No	1	0	

Registered Card Attribute Properties

i Note: Registered Card Attribute values are calculated data.

Registered Card Attribute Properties

Property Name	Description	Type	Length	Nulls Allowed	Sample Data	Default Data	Notes
RegisteredCardID	TIBCO Reward unique identifier for a customer card	Int	4	No	123459876	0	TIBCO Reward unique identifier for a customer card.
CardNumber	Card number, e.g., VISA card no. (Will be encrypted.)	nvarchar	800	Yes	1111222233334444	Null	This will come back as an encrypted field after being saved to TIBCO Reward the first time.
ShopperId	TIBCO Reward shopper Id assigned to this card	int	4 bytes	Yes	123459876	Null	Precision,scale = 10,0; e.g., 1234567890
CommonName	This can be used to	nvarchar	200	No	Acme Co Loyalty Card	Empty String	This can be used to store extra

Property Name	Description	Type	Length	Nulls Allowed	Sample Data	Default Data	Notes
	store extra information about the registered card					g	information about the registered card.
AlternateCard Identifier	If you need an unencrypted card number, use this field	nvarchar	200	Yes	1234567	Null	This field is not encrypted and should not be used for the storage of any monetary instrument data.
CardType	Type identifier for credit card	char	1	No	V	Empty String	The Credit Card type. supported values are: Visa = V MasterCard = M AmericanExpress = A Discover = D DinersClub = C JapaneseCommerceBank = J NonCreditCardL

Property Name	Description	Type	Length	Nulls Allowed	Sample Data	Default Data	Notes
							oyaltyCard = L PrivateLabelCreditCard = P
ExpirationMonth	Standard	int	4	No	12	None	Precision,scale = 10,0; e.g., 1234567890
ExpirationYear	Standard	int	4	No	2010	None	Precision,scale = 10,0; e.g., 1234567890
LastFour	The last four digits of the card number	nvarchar	8	No	8162	Empty String	The last four digits of the card number, this can be card number or alternate card number. The value used to search for members in the CSR tool.
SecurityCode	Do not use	nvarchar	8	Yes	858	Null	Not used.
CardHolderName	The name as printed on the card	nvarchar	200	No	John B Doe	Empty String	The cardholder name as printed on the card.

Property Name	Description	Type	Length	Nulls Allowed	Sample Data	Default Data	Notes
Status	Status of the card	char	1	No	A	Empty String	A/I (active or inactive). Only active cards are used by the TIBCO Reward system in transaction matching etc.
CreateDateTime	Date time the card is created	datetime	8	No	2012-04-23T00:00:00.000000+02:00	None	Date time the card is created.
IsPreferred	Reserved for future use	char	1	No	N	Empty String	<<Blank>>
FileImportId	Id of the import file that created the card	int	4	No	4321	0	Zero indicates that this was not created by file import.
EncryptedCardNumber	Encrypted version of card number	varbinary	128	Yes	NA	Null	Not for external use.
FirstSix	For	nvarchar	12	Yes	123456	Null	For future use.

Property Name	Description	Type	Length	Nulls Allowed	Sample Data	Default Data	Notes
	future use. Some clients allow the first six digits of a card to be used, to help differentiate between cards.						

Event Instance Properties

Event Instance Properties - XML Data

Property Name	Description	Type	Length	Nulls Allowed	Sample Data	Default Data	Notes
EventDate	The date/time when the event occurred	DateTime	20	Yes	02/12/2012 10:11:12	Current date/time If time is omitted, defaults to	Format: MM/DD/YYYY HH:Mi:SS, if time is omitted, defaults to midnight.

Property Name	Description	Type	Length	Nulls Allowed	Sample Data	Default Data	Notes
						midnight.	
RetailerEventId	Client's identifier for the event instance	nvarchar	100	Yes	12345	Null	
EventReferenceTag	Reference tag of the event definition of which the record is part	nvarchar	100	No	Addwishlist	Empty String	Must match the reference tag specified when the event definition was created.
Subject	Client's identifier of the event's subject	nvarchar	100	No	test@foo.com	Empty String	
Object	Client's identifier of the event's object	nvarchar	100	Yes	SKU100	Null	
MasterValue	Master value of	Int	4	Yes	10	Null	Applicable

Property Name	Description	Type	Length	Nulls Allowed	Sample Data	Default Data	Notes
	the event						e only if Master Value is enabled for the event

Reward Product Properties

Reward Product Properties - XML Data Table

Property Name	Description	Type	Length	Nulls Allowed	Sample Data	Default Data	Notes
ProductId	The ID of the Reward Product	int	4 bytes	No	1234	0	ProductId
RetailerGUID	Unique identifier for the Client/Retailer	nvarchar	16 bytes	No	GUID	None	RetailerGUID
SKU	The SKU of the Reward Product	string	200 chars	No	ABC	Empty string	SKU
ProductName	The name assigned to the Reward Product	string	510 chars	No	ABC	Empty string	ProductName

Property Name	Description	Type	Length	Nulls Allowed	Sample Data	Default Data	Notes
ProductDescription	Text description of the Reward Product	string	16 bytes	Yes	ABC	null	ProductDescription
RedeemValue	The Points needed to redeem this reward	int	4 bytes	No	100	0	RedeemValue
ImageURL	The URL specified by the marketer to display the image of the Reward Product	string	200 chars	Yes	url for image	null	ImageURL
BrandName	Brand Name of the reward item (optional)	string	200 chars	Yes	ABC	Null	
Status	The status of the Point Redemption Item	string	200 chars	Yes	A	Null	A = Active I = Inactive
LastUpdate	Date and Time when the reward item data was most recently	DateTime	8 bytes	No	Any Datetime	Default datetime.	

Property Name	Description	Type	Length	Nulls Allowed	Sample Data	Default Data	Notes
	updated						
UPCCode	The UniversalProduct Code assigned to the reward item	string	26 chars	Yes	Any string	Null	
InternalNote	Notes by and for the client regarding the Reward Product. - not for display to the Customer	string	510 chars	Yes	Any string	Null	
ImageDownloaded	Flag indicating whether the image was downloaded	boolean	1 char	No	1	0	

Overview of APIs

API calls represent specific operations that your client applications can invoke at runtime to perform tasks, for example:

- Query data maintained in the TIBCO Reward customer database.
- Create, update, and delete data.
- Obtain metadata.

All API calls are synchronous or asynchronous requests made to the API. Any changes to your data are committed automatically.

Characteristics of API Calls

All API calls are:

- Service requests and responses – Your client application prepares and submits a service request to the API, the API processes the request and returns a response, and the client application handles the response.
- Synchronous – Once the API call is invoked, your client application waits until it receives a response from the service. Asynchronous calls are not supported.
- Committed automatically – Every operation that writes to a TIBCO Reward object is committed automatically. This is analogous to the AUTOCOMMMIT setting in SQL.

For create, update, and delete calls that attempt to write to multiple records in an object, the write operation for each record is treated as a separate transaction. For example, if a client application attempts to create two new accounts, they're created using mutually exclusive insert operations that succeed or fail individually, not as a group.

Typical API Call Sequence

For each call, your client application typically:

- prepares the request by defining request parameters, if applicable.
- invokes the call, which passes the request with its parameters to the API for processing.
- receives the response from the API.
- handles the response, either by processing the returned data (for a successful invocation) or by handling the error (for a failed invocation).

Start with Authentication

Before invoking any other calls, a client application must first authenticate its credentials to establish a session and set the returned AuthenticationResult SOAP header in all subsequent API calls. For more information, see [Sample Code Walkthrough](#).

List of APIs

The following table lists the API calls as of the 13.1 release and provides the standard API descriptions.

 **Important:** Refer to <http://api.loyaltylab.com/loyaltyapi/> for a list of these same API calls. This online list includes links to a formal description of each API. This description includes the following:

- basic definition of the operation
- links to definitions of the key operation parameters and exceptions
- a link to the XML schema

See [API Entity Attributes and Properties](#) for additional information about the data used in the APIs.

Method Name	Description
AddRewardProductToCatalog	Adds a reward product to a reward catalog (successful if the product is already associated with the reward catalog).
AddShopperToProgram	Adds a single customer to a given program. Use this API to add an existing customer to your Loyalty program. In addition, this API can be used to add a customer to a club.
AddShopperToTier	Adds a shopper to a tier. If tier downgrading is not allowed and the passed program would result in a downgrade, the new tier is not assigned.
AddTransaction	Deprecated. Use ImportTransactions instead.
AdjustShopperPoints	Changes the point balance for a Customer. Use this API to increase or decrease the current point balance by a specified quantity of points.
AdjustShopperPointsCustomAttributes	Adjusts the point balance for a customer. Use this API to increase or decrease the current point balance by a specified quantity of points. You can also optionally specify the point type, and a textual description along with this call. In addition to adjusting points, you can use this API to update Custom

Method Name	Description
	Attributes for the specific shopper.
AdjustShopperPointsWithExpirationDate	Adjusts the point balance for a customer. Use this API to increase or decrease the current point balance by a specified quantity of points. You can also optionally specify the point type, and a textual description along with this call. Note: PointLifecycle must be enabled to use this method.
AdjustShopperPointsWithExpirationDateCustomAttributes	Adjusts the point balance for a customer. Use this API to increase or decrease the current point balance by a specified quantity of points. You can also optionally specify the point type, and a textual description along with this call. Note: PointLifecycle must be enabled to use this method. In addition to adjusting points, you can use this API to update Custom Attributes for the specific shopper.
AdjustShopperPointsWithRedemptionCustomAttributeCheck	Adjusts the point balance for a customer. Use this API to increase or decrease the current point balance by a specified quantity of points. You can also optionally specify the point type, and a textual description along with this call. In addition to

Method Name	Description
	adjusting points and setting Custom Attributes for the specified shopper, you can use this API to update Custom Attributes for Reward Redemptions as well.
AnonymizeShopper	Removes shopper personal identifiable information from Reward's system.
AssociatePurchase	Associates an external point purchase identifier with a point purchase in Reward's system.
AuthenticateUser	Creates and returns an authentication token for the API user. This token is good for a rolling 20 minutes.
ChangePointStateByExternalId	Changes the state of a set of points in a given point bucket matching the point purchase transaction on the given purchase id generated by the partner system.
ChangePointStateByLoyaltyLabId	Changes the state of a set of points in a given point bucket matching the point purchase transaction on the given purchase id generated by the Reward system.
CreateAndScoreShopper	Creates a new enrolled customer and immediately score the customer for offers and clubs.

Method Name	Description
CreateEventInstance	Creates a new instance of a generic event. Use with a specified generic event.
CreateRewardProduct	Creates a new reward product and returns the newly created entity with updated Reward ID and image URL. If the RewardProductID is specified it will be ignored and overwritten in the response. The image will be queued for download in our system from the supplied URL and overwritten with our URL in the response. The last update will be set automatically.
CreateRewardRedemption	Creates a Reward Redemption. For clients using the reward catalog, this API creates a redemption for an item for a customer.
CreateRewardRedemptionWithEvent	Create Reward Redemption With Event. For clients using the reward catalog, this API creates a redemption for an item for a customer and creates an event to record the redemption as a generic event.
CreateRegisteredCard	Creates a Registered Card for the specified Customer by its TIBCO Reward internal ID.
CreateShopper	Creates a new enrolled Customer.

Method Name	Description
CreateShopperWithCard	Creates a Customer entity that is a member of the Main Program. In addition, it will create the included RegisteredCard and associate it with the Shopper.
CreateUnregisteredShopper	Creates a new customer that is not a member of the Loyalty Program.
EnrollShopper	Enrolls shopper. This request can be processed for unregistered customers (that is, customers not enrolled in a loyalty program). Response must include a response code according to standards.
GetCSRAuthenticationToken	Gets Single Sign On authentication token for a CSR user by email address. Please refer to the Single Sign On Integration Guide for usage information.
GetCodesForShopper	An obsolete method. Please call GetCodesWithSkuForShopper.
GetCodesWithSkuForShopper	Allows client to retrieve all reward code redemption information for a single shopper by shopper id.
GetCurrentPointPrice	Returns a point price for the current base price, based on the product pricing criteria currently in effect.

Method Name	Description
GetCurrentPointPrices	Returns point prices for multiple products using their current base prices, based on the product pricing criteria currently in effect.
GetCustomAttribute	Gets the value of a custom attribute for a customer.
GetCustomEntitiesByReferenceTag	Gets custom entities and the referenced object from a custom entity reference tag, reference type, and reference ID.
GetCustomQuestionsForShopper	Gets custom questions and answers entity for a given customer. If the customer had previously answered the questions, the responses will be included.
GetEventDefinitions	Get available Event Definitions.
GetPlcAvailablePointsByDate	Gets Point Life Cycle available point balance for shopper from RedemptionPointState by transaction date range.
GetPlcAvailablePointsByDateByPointGroup	Gets Point Life Cycle available point balance for shopper from RedemptionPointState by transaction date range and by point group.
GetPointPrices	API that returns point prices for multiple products using their current base prices, based on

Method Name	Description
	the product pricing criteria currently in effect.
GetProductBySKU	Gets the product entity associated to the SKU.
GetProfilesByShopperId	Gets an array of profiles that the specified shopper is associated with.
GetPurchaseAwardForReturns	Gets award amount for returns.
GetReferredFriends	Returns the list of referred friends.
GetReferenceObjectFromUniqueCustomEntity	Retrieves the reference object (the shopper, redemption, etc.) given a custom entity reference tag, reference type, and field data of a custom entity field or fields marked unique.
GetReferenceObjectsFromSearchableCustomEntity	Retrieves an array of reference objects (shoppers, redemptions, etc.) from a custom entity reference tag, reference type, and field data of a custom entity field or fields marked as searchable.
GetRegisteredCard	Gets registered card information for a customer by the TIBCO Reward internal ID.
GetRewardCatalogById	Gets a reward catalog entity from the Reward ID.

Method Name	Description
GetRewardCatalogByRetailerCatalogId	Gets a reward catalog entity by the retailer reward catalog ID.
GetRewardCatalogsByRewardProductId	Gets an array of reward catalog entities to which the passed reward product ID belongs.
GetRewardProductById	Gets a reward product entity from a Reward reward product ID.
GetRewardProductByUniqueSKU	Gets a reward product entity for a SKU. (Assumes the SKU is unique in the Loyalty system).
GetRewards	Gets all active reward Items in the reward catalog. Note this is all rewards, not rewards by customer.
GetShopper	Gets customer information by TIBCO Reward shopper ID. Gets an extended customer entity, which includes the point balance.
GetShopperAuthenticationToken	Gets Shopper authentication token.
GetShopperByEmail	Gets the details of the customer whose email address is provided in the request.
GetShopperById	Gets the details of the customer whose Reward shopper ID is provided in the request.

Method Name	Description
GetShopperByMergedVictimID	Gets a customer by the client's unique customer key. It is expected that this is your unique internal identifier for the customer. Please contact client services before using this API, since the initial integration with your company may have repurposed the RetailerShopperID parameter for another use.
GetShopperByRegisteredCard	Gets customer information by registered card. This only applies if the customer is a member of the loyalty program and your company is collecting registered card information.
GetShopperByRetailerID	Gets a customer by the client's unique customer key. It is expected that this is your unique internal identifier for the customer. Please contact your Account Manager before using this API, since the initial integration with your company may have repurposed the RetailerShopperID parameter for another use.
GetShopperByUserName	Gets a customer by the customer's alias or username. It is expected that this is your unique internal identifier for the Customer. Please contact your Account Manager before using

Method Name	Description
	this API.
GetShopperOffers	Gets all the offers that are both live and scored for a customer. Includes the offer clip status of all of the offers for the specified customer.
GetShopperOffersExtended	Gets all the offers that are both live and scored for a customer. Includes the clip status of all of the offers for the specified customer.
GetShopperPointBalance	Gets a customer's current point balance.
GetShopperPointBalanceByBalanceType	Gets a customer's current point balance of a certain point balance.
GetShopperPointBalanceByDate	Gets total point balance for the shopper by points post date range.
GetShopperPointBalanceByPointGroup	Gets a shopper's point balance for all point types within the specified point group.
GetShopperPointBalanceByPointState	Returns the current number of points for the specified customer in the specified state.
GetShopperPointBalanceByRetailerID	Gets a customer's current point balance. The customer identifier passed is the identifier in the client's system.

Method Name	Description
GetShopperPointsExpiring	Finds the number of points of a given type expiring within a given time window for a shopper.
GetShopperPrograms	Gets all live clubs available for a customer to join, current club memberships, and tier or main program membership information. Note: Program membership information is very dependent on loyalty program setup.
GetShopperRedemptionsByDate	Returns the set of awards earned by a shopper in the specified date range.
GetShopperRedemptionsByLoyaltyProgramId	Gets information regarding the redemptions of the given shopper as defined by Loyalty program ID.
GetShopperRedemptionsByShopperId	Gets information regarding the redemptions of the given shopper as defined by their shopper ID.
GetShopperRedemptionsOrderItemsByDate	Gets redemption order items for a shopper during a date range.
GetShopperRegisteredCards	Gets all registered cards for a customer.
GetStores	Returns the stores. Data is split

Method Name	Description
	up by pageIndex (starting record) and pageSize, so calls should be done in smaller page sizes for faster response and transmission.
GetTierPrograms	Gets a detailed list of tiers.
GetTotalPointsEarnedByEmailAddress	Returns the total number of points earned by a shopper in a given time frame, the shopper being identified by the email address.
GetTotalPointsEarnedByLoyaltyCardNumber	Returns the total number of points earned by a shopper in a given time frame, the shopper being identified by the loyalty program's card number.
GetTotalPointsEarnedByRetailerShopperId	Returns the total number of points earned by a shopper in a given time frame, the shopper being identified by the client's shopper identifier.
GetTotalPointsEarnedByShopperId	Returns the total number of points earned by a shopper in a given time frame, the shopper being identified by Reward's internal shopper identifier.
GetTransactions	Gets the complete transaction history (all Transactions) for a customer.
GetTransactionsByDate	Gets all transactions for the

Method Name	Description
	customer within the specified date range.
GetTransactionsWithTenders	Gets transaction history with tenders for a customer.
ImportTransaction	Provides the ability for transaction data (purchases and returns) to be processed in real time against any live purchase based offers that award points, setup and configured via the CRMS.
ImportTransactions	Provides the ability for transaction data (purchases and returns) to be processed in real time against any live purchase-based offers that award points, setup and configured via the CRMS.
MergeAccounts	Merges one customer account into another, retaining attributes for the second account.
MergeShoppers	Deprecated. Use MergeAccounts instead.
PointsPurchase	Computes a point price for the current base price, based on the product pricing criteria currently in effect and applies those points towards purchasing the specified sku. If point groups are enabled, do not use this API but instead use

Method Name	Description
	PointsPurchaseWithPointGroup.
PointsPurchaseWithPointGroup	Computes a point price for the current base price, based on the product pricing criteria currently in effect and applies those points towards purchasing the specified SKU.
RedeemOffer	Returns the ShopperRewardItemRedemption based on the offer that is setup for qualifying on generic events and awarding reward codes.
RedeemReward	For Clients using the reward catalog, this converts points into the specified reward for a customer.
RedeemShopperRewardCerificate	Redeems a reward certificate number which was generated for the shopper.
ReferFriend	Use this API to refer a friend for the loyalty program.
RemoveRewardProductFromCatalog	Removes a reward product from a reward catalog (successful if the reward product is not currently a member of the reward catalog).
ReversePurchaseAwardForReturns	Redeems amount on Stored Value Card for Returns. (This API is only supported for some clients. Please check with your

Method Name	Description
	Account Manager).
ScoreShopper	Sends a message to the scoring queue.
SendCommunicationToEmailAddress	<p>Accepts one email address and one email communication ID and uses this data to send one email to the one customer identified by the email address.</p> <p>Validations: the customer must be in active state; communication must be in live status. Override will bypass shopper opt-in flag.</p>
SendCommunicationToEmailAddresses	<p>Accepts a string array of email addresses and one email communication ID and uses this data to send the email to the multiple customers identified by these email addresses.</p> <p>Validations: shoppers must be in active state; communication must be in live status. Override will bypass shopper opt-in flag.</p>
SendCommunicationToShopper	<p>Accepts one retailer shopper ID and one email communication ID and uses this data to send the email to the one customer.</p> <p>Validations: shopper must be in active state; communication must be in live status. Override will bypass shopper opt-in flag.</p>
SendCommunicationToShoppers	Accepts multiple retailer

Method Name	Description
	shopperIDs and one email communication ID and uses these to send the email to the multiple customers at the same time. Validations: shoppers must be in active state; communication must be in live status. Override will bypass shopper opt-in flag.
ShopperSignIn	Returns the shopper object data based on the correct shopper username/password.
UnEnrollShopper	Unenrolls the loyalty member and deactivates the customer. As a result, the customer is no longer active in the TIBCO Reward system and will no longer accrue any benefits from the program.
UpdateCustomAttribute	Sets the value of a custom attribute for an associated entity. In most cases, this will be for a customer unless directed otherwise by Account Management.
UpdateCustomAttributes	Sets the value of multiple custom attributes for multiple entities. This will be for customers or products unless directed otherwise by Account Management.
UpdateCustomEntity	Updates custom entity field

Method Name	Description
	data.
UpdateCustomQuestionForShopper	Updates the answers for a given set of custom questions for a given customer.
UpdateRewardProduct	Updates a reward product and returns the newly updated product.
UpdateShopper	Updates the customer's information. You need to pass the entire entity with the information you want to update.
UpdateShopperOfferStatus	Accept Offer for a customer. Use this API to clip an offer for a given customer. Pass the "true" value in the status parameter, and the offer will be clipped.
UpdateShopperPointsBalanceWithCustomAttributes	Update Shopper's Point balance with Custom Attributes.

Overview of Objects or Entities

In the API, objects are representations of sets of data, analogous to database tables, defined for your organization. Object properties represent fields in those data objects, and client applications set or retrieve data values via these properties.

i Note: In this document, "object properties," "fields," and "entity attributes" are analogous, and in most contexts are virtually interchangeable. Depending on the development environment and products you work with, you may see these variations in nomenclature in the documentation you use.

See [TIBCO Reward](#) and [API Entity Attributes and Properties](#) for specific information about the entities used by the TIBCO Reward API.

Security and the API

Client applications that access your organization's TIBCO Reward data are subject to the same security protections that are used in the TIBCO Reward user interface:

- User Authentication
- Profile Configuration
- Sharing
- Implicit Restrictions for Objects and Fields

User Authentication

Client applications must log in using valid credentials for an organization. The server authenticates these credentials and, if valid, provides the client application with:

- SOAP authentication header that must be set into the session header so that all subsequent calls to the API are authenticated
- a URL for the client application's API requests

TIBCO Reward supports only the Secure Sockets Layer (SSL) protocol SSLv3 and the Transport Layer Security (TLS) protocol.

Ciphers must have a key length of at least 128 bits.

i Note: Client applications try to connect with the SFTP default secure ciphers first and again with legacy insecure ciphers if initial secure connection fails with an ssh exception. A warning is logged if a connection is opened using insecure legacy ciphers. Insecure reconnecting might be disabled in an environment setting. It is enabled on production environments to avoid service disruptions.

Implicit Restrictions for Objects and Fields

Certain objects can be created or deleted only in the TIBCO Reward user interface. Other objects are read-only: client applications cannot create, update, or delete such objects. Similarly, certain fields within some objects can be specified on create but not on update. Other fields are read-only: client applications cannot specify field values in create or update calls. See [API Entity Attributes and Properties](#) for information related to the respective Entity Attribute descriptions.

Error Handling

The API calls return error data that your client application can use to identify and resolve runtime errors. If an error occurs during the invocation of most API calls, then the API provides the following types of error handling:

- For errors resulting from badly formed messages, failed authentication, or similar problems, the API returns a SOAP fault message.
- TIBCO Reward's API uses generic error handling messages to pass back exceptions for web service calls.
- For most calls, if the error occurs because of a problem specific to the query, the API returns an Error. For example, if a create request contains more than 200 objects.

Error Handling for Session Expiry

When you make the initial authentication call, a new client session begins and a corresponding unique authentication token is generated. Authentication tokens automatically expire after 20 minutes of inactivity. When your token expires, a SOAP exception of type `SoapAuthenticationException` is returned. If this happens, you must invoke the authentication call again.

Error Handling for Unauthorized Data

When you make any API call by passing input data that results in an authorization violation, either `SoapInputAuthorizationException` or `SoapOutputAuthorizationException` is returned. The usual case is passing an identifier that results in finding an API entity that

does not belong to you. If this happens, you must correct the input data and call the API again.

Error Handling for API

When you make any API call that results in an error (SOAP fault) message, specific information about the nature of error is provided. Various error cases are handled and related error information is returned for most of the APIs that we consider helpful. To extract the error information, the SOAP fault message must be parsed. APIs return error information in three parts within SOAP fault message as explained below:

- `faultString` (string)– generic message with standard or specific message
- `detail.code` (integer) – numeric code
- `detail.description` (string) – short description about the error

The following is the sample SOAP fault message:

```
<soap:Fault>
  <faultcode>soap:Server</faultcode>
  <faultstring>An error occurred in processing your request, please try
  again
  </faultstring>
  <detail>
    <code>201</code>
    <description>Unable to find shopper from the specified email address
    </description>
  </detail>
</soap:Fault>
```

Generic Error Codes

Error Code	Description
100	Unauthenticated request
101	Unauthorized input
102	Unauthorized output
103	Unknown system error
104	Database error

API Related Error Codes

Error Code	Description
200	Unable to update shopper likely due to email address collision
201	Unable to find shopper from the specified email address
202	Unable to create shopper due to system error
203	Registered card already exists for the shopper
204	Unable to find shopper from the specified RetailerShopperId

Error Code	Description
205	Shopper email address is not valid
206	Shopper has been updated but could not be enrolled into program
207	Unable to find event for the specified event instance
208	Unable to find matching subject for the subject text
209	Unable to find shopper from the specified ShopperId
210	Unable to find program from the specified ProgramId
211	Unable to find offer from the specified OfferId
212	Unable to find pointredemptioncatalog from the specified PointRedemptionCatalogId
213	Unable to find loyalty card from the specified Id
214	Unable to find matching object for the subject text
215	Unable to create/update shopper due to a Retailer Shopper ID collision

Error Code	Description
216	Unable to create/update shopper due to a user name collision
217	Unable to create/update shopper due to a phone number collision
218	Unable to create/update shopper due to a loyalty card collision
219	Unable to create/update shopper due to a credit card collision
220	Unable to create/update shopper because at least one of the configured unique fields must be specified
254	Shopper with shopperId is already merged or inactive
415	Unable to find the matching original Purchase with the TransactionLink information provided in the return list. Unable to process the Return
416	Shopper is already Anonymized

TravelSuite API Error Codes

Error Code	Description
300	The point purchase transaction with the specified TIBCO Reward purchase Id could not be located.
301	The point purchase transaction with the specified externalId could not be located.
302	The customer does not have enough points to make this point purchase.
303	The specified point state is invalid for the requested transaction.
304	PointLifecycle is not enabled, please contact TIBCO Reward to use this functionality.

Error Code	Description
305	The external transaction identifier already exists in the database. It must be a unique value.
306	The product with the specified SKU could not be located.

List of APIs and Related Error Codes

API	Error Code or Error Codes
AnonymizeShopper	204, 416
UpdateShopper	200, 215, 216, 217, 218, 219, 220
SendCommunicationToEmailAddress	201
UpdateCustomAttribute	103, 104
CreateShopper	103, 200,

API	Error Code or Error Codes
	202, 206, 215, 216, 217, 218, 219, 220
CreateAndScoreShopper	103, 200, 202, 206, 215, 216, 217, 218, 219, 220
CreateRegisteredCard	203
CreateEventInstance	207, 208, 214
AssociatePurchase	304
ChangePointStateByLoyaltyLabId	304
GetShopperPointBalanceByPointState	304, 204
GetShopperRedemptionsByLoyaltyProgramId	304, 204
ChangePointStateByExternalId	304, 303
PointsPurchase	304, 303, 302, 306, 204
ImportTransactions	103, 415
GetCurrentPointPrice	304, 306, 204

API	Error Code or Error Codes
GetCurrentPointPrices	304, 306, 204
GetPointPrices	304
GetShopperRedemptionsByShopperId	304
All APIs updating entities could receive entity validation error	209, 210, 211, 212, 213
All above and remaining APIs could receive generic error conditions	103,104

Implementation Considerations

Review the following topics to make sure you understand how the API manages data and communication before building your client applications.

XML Compliance

The API is based on XML, which requires all documents to be well-formed. Part of that requirement is that certain Unicode characters are not allowed, even in an escaped form, and that others must be encoded according to the API location. Normally, this is handled by any standard SOAP or XML client. Clients must be able to parse any normal XML escape sequence and must not pass up invalid XML characters. Some characters, as mentioned, are illegal even if they are escaped. The illegal characters include the Unicode surrogate blocks and a few other Unicode characters. All are seldom-used control characters that are usually not important in any data and tend to cause problems with many programs. Although they are not allowed in XML documents, illegal characters are allowed in HTML documents and may be present in TIBCO Reward data.

The illegal characters will be stripped from any API response. The following characters are illegal:

- 0xFFFE
- 0xFFFF
- Control characters 0x0 - 0x19, except the following characters, which are legal: 0x9, 0xA, 0xD, tab, newline, and carriage return)
- 0xD800 - 0xDFFF

Content Type Requirement

In the API, all requests must contain a correct content type HTTP header, for example: Content-Type:text/xml;. Earlier versions of the API do not enforce this requirement.

HTML Persistent Connections

Most clients achieve better performance if they use HTTP 1.1 persistent connection to reuse the socket connection for multiple requests. Persistent connections are normally handled by your SOAP/WSDL client automatically. For more details, see the HTTP 1.1 specification at: <http://www.w3.org/Protocols/rfc2616/rfc2616-sec8.html#sec8.1>.

HTTP Chunking

Clients that use HTTP 1.1 may receive chunked responses. Chunking is normally handled by your SOAP/WSDL client automatically.

Preparing to Use the API

If you have a solid background in implementing API services, and you want to get off to a fast start using the TIBCO Reward API in your development environment, perform the following steps:

Procedure

1. [Ensure Your Company is Configured for APIs](#)
2. [TIBCO Reward](#)
3. [Generate or Obtain the Web Service WSDL File](#)
4. [Import the WSDL File into Your Development Platform](#)
5. [Walk Through the Sample Code](#)

Ensure Your Company is Configured for APIs

Contact your Client Services Manager and request API access to a development environment.

This should take approximately two days, as your development environment must be built based on the unique configuration of your current TIBCO Reward implementation. TIBCO Reward will also assess any impact to your loyalty program.

Obtain a TIBCO Reward API User Account

Generally, there should be only one API User Account for your company. Log in to the CRMS and select **Admin > CRMS Users**.

See if an API account already exists. (User Type will be API User.) If not, you can create one by selecting API User from the select user pull-down menu and then clicking **Add**. This opens the create new API user screen. Enter fields marked with asterisks and then click **Done**.

Generate or Obtain the Web Service WSDL File

To access the TIBCO Reward API service, you need a WSDL file, sometimes called the service description.

The WSDL file defines the Web service that is available to you. Your development platform uses this WSDL file to generate an API to access the web service it defines. For more information about WSDLs, see <http://www.w3.org/TR/wsdl>.

Generating the WSDL File for Your Organization

If you access the service URL link provided below (with "?wsdl" appended), TIBCO Reward will automatically generate a WSDL (service description) and return it as XML in your browser. The generated WSDL defines all of the API calls, objects, and fields that are available for API access for your company.

To generate the WSDL file for your organization, use <http://api.loyaltylab.com/loyaltyapi/loyaltyapi.asmx?wsdl>.

Import the WSDL File into Your Development Platform

Once you have the WSDL file, you need to import it into your development platform so that your development environment can generate the necessary objects for use in building client Web service applications in that environment.

The following two sets of instructions in this document provide sample instructions. One set is for Apache Axis Java; the other is for Microsoft Visual Studio. For instructions about other development platforms, see that platform's product documentation.

Instructions for Java Environments (Apache Axis)

Java environments access the API through Java objects that serve as proxies for their server-side counterparts. Before using the API, you must first generate these objects from your organization's WSDL file. Each SOAP client has its own tool for this process. For Apache Axis, use the WSDL2Java utility. (See <http://axis.apache.org/axis/java/user-guide.html#UsingWSDLWithAxis> for additional background and guidance).

i Note: Before you run WSDL2Java, you must have Axis installed on your system and all of its component JAR files must be referenced in your classpath. For Axis installation information, see <http://axis.apache.org/axis/java/install.html>.

The basic syntax for WSDL2Java is:

```
java -classpath pathToJAR/Filename org.apache.axis.wsdl.WSDL2Java -a
pathToWsdL/WsdLFilename
```

The `-a` switch generates code for all elements, referenced or not, which may be necessary depending on your WSDL. For more information on this, see the [WSDL2Java online reference guide](#).

If you have JAR files in more than one location, list them with a semicolon separating the files. For example, if the Axis JAR files are installed in `C:\axis-1_3`, and the WSDL is named `loyalty.wsdl` and is stored in `C:\myswds`: `java -classpath`.

```
c:\axis-1_3\lib\axis.jar;c:\axis-1_3\lib\axi-ant.jar;c:\axis-1_
3\lib\axis-schema.jar; c:\axis-1_3\lib\commons-discovery-
0.2.jar;c:\axis-1_3\lib\commons-logging-1.0.4.jar; c:\axis-1_
3\lib\jaxrpc.jar;c:\axis-1_3\lib\log4j-1.2.8.jar;c:\axis-1_
3\lib\saaj.jar; c:\axis-1_3\lib\wsdl4j-1.5.2.jar;
org.apache.axis.wsdl.WSDL2Java -a C:\myswds\loyalty.wsdl
```

This command will generate a set of folders and Java source code files in the same directory in which it was run. After these files are compiled, they can be included in your Java programs for use in creating client applications. For most Java development environments, you can use wizard-based tools for this process instead of the command line. For a WSDL2Java Reference, see <http://ws.apache.org/axis/java/reference.html>.

Instructions for Microsoft Visual Studio

Visual Studio languages access the API through objects that serve as proxies for their server-side counterparts. Before using the API, you must first generate these objects from your organization's WSDL file. Visual Studio provides two approaches for importing your WSDL file and generating an XML Web service client: an IDE-based approach and a command line approach.

i Note: Before you begin, create a new application or open an existing application in Visual Studio.

In addition, you need to have generated the WSDL file, as described in [Generate or Obtain the Web Service WSDL File](#).

An XML web service client is any component or application that references and uses an XML web service. This does not necessarily need to be a client-based application. In fact, in many cases, your XML web service clients might be other web applications, such as web forms or even other XML web services. When accessing XML web services in managed code, a proxy class and the .NET framework handle all of the infrastructure coding.

To access an XML web service from managed code:

1. Add a web reference to your project for the XML web service that you want to access. The web reference creates a proxy class with methods that serve as proxies for each exposed method of the XML web service.
2. Add the namespace for the web reference.
3. Create an instance of the proxy class and then access the methods of that class as you would access the methods of any other class.

To add a web reference:

1. On the **Project** menu, choose **Add Web Reference**.
2. In the URL box of the **Add Web Reference** dialog box, type the URL to obtain the service description of the XML web service you want to access, such as:
`C:\WSDLFiles\loyalty.wsdl`.
3. Click **Go** to retrieve information about the XML web service.
4. In the web reference name box, rename the web reference to `LoyaltyAPI`, which is the namespace you will use for this web reference.
5. Click **Add Reference** to add a web reference for the target XML web service. For more information, see the topic “Adding and Removing Web References” in the Visual Studio documentation.
6. Visual Studio retrieves the service description and generates a proxy class to interface between your application and the XML web service.

Walk Through the Sample Code

Once you have imported your WSDL file, you can begin building client applications that use the API.

The fastest way is to learn by example: start by walking through the code example provided in [Sample Code Walkthrough](#).

Sample Code Walkthrough

This topic provides sample code snippets that use the API.

C# Sample Code

This section walks through a sample C# client application. The purpose of this sample application is to show the required steps for logging in and to demonstrate the invocation and subsequent handling of several API calls.

This sample application performs the following tasks:

1. Sets an authentication token with the returned authentication token, which is required for authentication on subsequent API calls.
2. Creates a new Shopper object, which is a customer in the TIBCO Reward system.
3. Sets the customer's attribute values.
4. Adds 50 points to the customer's point balance.
5. Gets and displays offers, as follows:
 - a. Gets the offers for the new customer
 - b. Checks to see if any of the retrieved offers are live
 - c. Displays the offer if it is live
 - d. Clips all of the live offers
 - e. Saves the updated offer status.
6. Gets all of the available rewards in the Rewards Catalog.
7. Redeems the first reward in the Rewards Catalog.
8. Gets the updated customer point balance.

```

////////////////////////////////////
////////////////////////////////////
//At this point we have added a web reference to
https://api.loyaltylab.com/loyaltyapi/ //
//This web reference is named loyaltyapi. //
//An API user has been created in CRMS with the email of
apiuser@yourdomain.com //

////////////////////////////////////
////////////////////////////////////

loyaltyapi.LoyaltyLabAPI api = new ApiTest.loyaltyapi.LoyaltyLabAPI();
api.AuthenticationResultValue = api.AuthenticateUser
("apiuser@yourdomain.com", "yourpassword");
//Create a new customer object (which is named Shopper)

loyaltyapi.Shopper newCustomer = new loyaltyapi.Shopper();

//Set the user's context to be yours (alternatly you can pass in an
empty guid and this

//will be set for you)
newCustomer.RetailerGUID = api.AuthenticationResultValue.RetailerGuid;
//Sets the new customer's attributes newCustomer.FirstName = "Joe";
newCustomer.LastName = "User"; newCustomer.ProfileCreateDateTime =
DateTime.Now; newCustomer.CreateDateTime = DateTime.Now;
newCustomer.LoyaltyLabCreateDateTime = DateTime.Now;
newCustomer.EmailAddress = "apites3t@apitest2.com";
newCustomer.LoyaltyMember = true;
newCustomer.Status = "A"; newCustomer.Password = "dfds";
newCustomer.PersonStatus = "P";

//Creates the new customer in the TIBCO Reward system and scores the new
customer for
//offers

newCustomer = api.CreateAndScoreShopper(newCustomer);
//Add 50 points to the customer's point balance

int bal = api.AdjustShopperPoints(newCustomer.ShopperId, 50);
//Get offers for the new customer
loyaltyapi.OfferStatus[] oStatus = api.GetShopperOffers
(newCustomer.ShopperId);
for(int i =0;i<offers.Length;i++){
//See if the offer is live now

```

```
if(oStatus[i].Offer.StartDateTime>DateTime.Now && oStatus
[i].Offer.EndDateTime
<DateTime.Now){
//You could display offer description using oStatus[i].Offer.OfferText

//Clip the offers if not clipped (you generally would not
//want to clip all offers, this is just to show how you would clip one)
if(!oStatus[i].IsClipped){
oStatus[i].IsClipped = true;
//Save the status change api.UpdateShopperOfferStatus(oStatus[i]);

//Get all of the available rewards loyaltyapi.RewardItem[] rewards =
api.GetRewards();
//Redeem the first reward - will fail if the customer does not have
enough points for the

//reward if(rewards.Length >0){
api.RedeemReward(rewards[0].PointRedemptionCatalogId,
newCustomer.ShopperId);

}

//Get the customer's new point balance
int bal = api.GetShopperPointBalance(newCustomer.ShopperId);
```

TIBCO Documentation and Support Services

For information about this product, you can read the documentation, contact TIBCO Support, and join TIBCO Community.

How to Access TIBCO Documentation

Documentation for TIBCO products is available on the [TIBCO Product Documentation](#) website, mainly in HTML and PDF formats.

The [TIBCO Product Documentation](#) website is updated frequently and is more current than any other documentation included with the product.

Product-Specific Documentation

Documentation for TIBCO® Reward is available on the [TIBCO® Reward Product Documentation](#) page.

To directly access documentation for this product, double-click the following file:

`TIBCO_HOME/release_notes/TIB_loyalty_22.3.0_docinfo.html` where `TIBCO_HOME` is the top-level directory in which TIBCO products are installed. On Windows, the default `TIBCO_HOME` is `C:\tibco`. On UNIX systems, the default `TIBCO_HOME` is `/opt/tibco`.

The following documents for this product can be found in the TIBCO Documentation site:

- *TIBCO® Reward Release Notes*
- *TIBCO® Reward User Guide*
- *TIBCO® Reward User Guide V3*
- *TIBCO® Reward CSR User Guide*
- *TIBCO® Reward Integration Guide*
- *TIBCO® Reward API Reference Guide*
- *TIBCO® Reward Analytics User's Guide*

How to Contact TIBCO Support

Get an overview of [TIBCO Support](#). You can contact TIBCO Support in the following ways:

- For accessing the Support Knowledge Base and getting personalized content about products you are interested in, visit the [TIBCO Support](#) website.
- For creating a Support case, you must have a valid maintenance or support contract with TIBCO. You also need a user name and password to log in to [TIBCO Support](#) website. If you do not have a user name, you can request one by clicking **Register** on the website.

How to Join TIBCO Community

TIBCO Community is the official channel for TIBCO customers, partners, and employee subject matter experts to share and access their collective experience. TIBCO Community offers access to Q&A forums, product wikis, and best practices. It also offers access to extensions, adapters, solution accelerators, and tools that extend and enable customers to gain full value from TIBCO products. In addition, users can submit and vote on feature requests from within the [TIBCO Ideas Portal](#). For a free registration, go to [TIBCO Community](#).

Legal and Third-Party Notices

SOME TIBCO SOFTWARE EMBEDS OR BUNDLES OTHER TIBCO SOFTWARE. USE OF SUCH EMBEDDED OR BUNDLED TIBCO SOFTWARE IS SOLELY TO ENABLE THE FUNCTIONALITY (OR PROVIDE LIMITED ADD-ON FUNCTIONALITY) OF THE LICENSED TIBCO SOFTWARE. THE EMBEDDED OR BUNDLED SOFTWARE IS NOT LICENSED TO BE USED OR ACCESSED BY ANY OTHER TIBCO SOFTWARE OR FOR ANY OTHER PURPOSE.

USE OF TIBCO SOFTWARE AND THIS DOCUMENT IS SUBJECT TO THE TERMS AND CONDITIONS OF A LICENSE AGREEMENT FOUND IN EITHER A SEPARATELY EXECUTED SOFTWARE LICENSE AGREEMENT, OR, IF THERE IS NO SUCH SEPARATE AGREEMENT, THE CLICKWRAP END USER LICENSE AGREEMENT WHICH IS DISPLAYED DURING DOWNLOAD OR INSTALLATION OF THE SOFTWARE (AND WHICH IS DUPLICATED IN THE LICENSE FILE) OR IF THERE IS NO SUCH SOFTWARE LICENSE AGREEMENT OR CLICKWRAP END USER LICENSE AGREEMENT, THE LICENSE(S) LOCATED IN THE “LICENSE” FILE(S) OF THE SOFTWARE. USE OF THIS DOCUMENT IS SUBJECT TO THOSE TERMS AND CONDITIONS, AND YOUR USE HEREOF SHALL CONSTITUTE ACCEPTANCE OF AND AN AGREEMENT TO BE BOUND BY THE SAME.

This document is subject to U.S. and international copyright laws and treaties. No part of this document may be reproduced in any form without the written authorization of TIBCO Software Inc.

TIBCO, the TIBCO logo, the TIBCO O logo, and are either registered trademarks or trademarks of TIBCO Software Inc. in the United States and/or other countries.

Java and all Java based trademarks and logos are trademarks or registered trademarks of Oracle Corporation and/or its affiliates.

This document includes fonts that are licensed under the SIL Open Font License, Version 1.1, which is available at: <https://scripts.sil.org/OFL>

Copyright (c) Paul D. Hunt, with Reserved Font Name Source Sans Pro and Source Code Pro.

All other product and company names and marks mentioned in this document are the property of their respective owners and are mentioned for identification purposes only.

This software may be available on multiple operating systems. However, not all operating system platforms for a specific software version are released at the same time. See the readme file for the availability of this software version on a specific operating system platform.

THIS DOCUMENT IS PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS DOCUMENT COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THIS DOCUMENT. TIBCO SOFTWARE INC. MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS DOCUMENT AT ANY TIME.

THE CONTENTS OF THIS DOCUMENT MAY BE MODIFIED AND/OR QUALIFIED, DIRECTLY OR INDIRECTLY, BY OTHER DOCUMENTATION WHICH ACCOMPANIES THIS SOFTWARE, INCLUDING BUT NOT LIMITED TO ANY RELEASE NOTES AND "READ ME" FILES.

This and other products of TIBCO Software Inc. may be covered by registered patents. Please refer to TIBCO's Virtual Patent Marking document (<https://www.tibco.com/patents>) for details.

Copyright © 2004-2022. TIBCO Software Inc. All Rights Reserved.