**Triumph Business Capital Web API Detailed Reference Guide**

**Version 1.2**

**Spring 2016**

Table of Contents

[Invoice Management Web API Information 4](#_Toc451947784)

[Use of Web API Methods 4](#_Toc451947785)

[REST Services 4](#_Toc451947786)

[Authentication 4](#_Toc451947787)

[OAuth2 4](#_Toc451947788)

[Dates 5](#_Toc451947789)

[Errors 6](#_Toc451947790)

[Required Fields 6](#_Toc451947791)

[Authentication Service Methods 6](#_Toc451947792)

[POST /v1/Login 6](#_Toc451947793)

[POST /v1/Logout 6](#_Toc451947794)

[GET /v1/OAuthUrl 7](#_Toc451947795)

[GET /v1/OAuthToken 7](#_Toc451947796)

[GET /v1/OAuthRevoke 8](#_Toc451947797)

[POST /v1/SingleSignOnUrl 8](#_Toc451947798)

[POST /v1/EmailPasswordReset 8](#_Toc451947799)

[POST /v1/UserInfo 9](#_Toc451947800)

[Listing Service Methods 10](#_Toc451947801)

[GET /v1List/BatchStatusTypes 10](#_Toc451947802)

[GET /v1List/CarrierPaymentTerms 10](#_Toc451947803)

[GET /v1List/CreditResultTypes 10](#_Toc451947804)

[GET /v1List/Customers 11](#_Toc451947805)

[GET /v1List/DocTypes 11](#_Toc451947806)

[GET /v1List/FundingOptions 11](#_Toc451947807)

[GET /v1List/InputChargeTypes 12](#_Toc451947808)

[GET /v1List/InvoiceCutoffTime 12](#_Toc451947809)

[GET /v1List/InvoiceStatusTypes 12](#_Toc451947810)

[Credit Service Methods 13](#_Toc451947811)

[POST /v1Credit/SearchCredit 13](#_Toc451947812)

[Submission Service Methods 13](#_Toc451947813)

[POST /v1Submit/CreateInput 13](#_Toc451947814)

[POST /v1Submit/CreateInputsFromArray 14](#_Toc451947815)

[POST /v1Submit/UpdateBrokerCarrierDetails 15](#_Toc451947816)

[POST /v1Submit/UpdateBrokerCarrierDetailsFromArray 16](#_Toc451947817)

[POST /v1Submit/UpdateTransportationInputDetails 16](#_Toc451947818)

[GET /v1Submit/PendingInput/*inputId* 17](#_Toc451947819)

[GET /v1Submit/AllPendingInputs 19](#_Toc451947820)

[DELETE /v1Submit/DeletePendingInput/*inputId* 19](#_Toc451947821)

[DELETE /v1Submit/DeleteAllPendingInputs 19](#_Toc451947822)

[POST /v1Submit/CreateDocument 19](#_Toc451947823)

[GET /v1Submit/DocumentsForInputId/*inputId* 20](#_Toc451947824)

[GET /v1Submit/Document/*documentId* 20](#_Toc451947825)

[DELETE /v1Submit/DeleteDocument/*documentId* 20](#_Toc451947826)

[POST /v1Submit/AppendDocumentFromReferenceKey/ 21](#_Toc451947827)

[POST /v1Submit/CreateInputCharge 21](#_Toc451947828)

[GET /v1Submit/InputChargesForInputId/*inputId* 21](#_Toc451947829)

[GET /v1Submit/InputCharge/*inputChargeId* 22](#_Toc451947830)

[DELETE /v1Submit/DeleteInputCharge/*inputChargeId* 22](#_Toc451947831)

[POST /v1Submit/FinalizePendingInput 22](#_Toc451947832)

[POST /v1Submit/FinalizePendingInputArray 22](#_Toc451947833)

[Reporting Service Methods 23](#_Toc451947834)

[GET /v1Report/AccountBalance 23](#_Toc451947835)

[GET /v1Report/GetInputIdByReferenceKey/*referenceKey* 23](#_Toc451947836)

[GET /v1Report/GetInputStatusByReferenceKey/*referenceKey* 23](#_Toc451947837)

[GET /v1Report/GetInputStatusArrayByReferenceKeys 24](#_Toc451947838)

[GET /v1Report/ProblemInvoiceSummary 24](#_Toc451947839)

[GET /v1Report/GetRecentBatchSummary 24](#_Toc451947840)

[GET /v1Report/GetRecentBatches 25](#_Toc451947841)

# Invoice Management Web API Information

Customers and vendors may use the Triumph Business Capital Web Services to integrate real-time with the Invoice transaction system. The API allows customers to create individual inputs, add additional details, upload images, and then, once they are ready, they can finalize all pending inputs

# Use of Web API Methods

## REST Services

The API is implemented as REST services. The methods accept standard HTTP parameters and will returns output encoded as a JSON object.

The root for the documented test API can be found at: [*https://testapi.mytriumph.com/*](https://testapi.mytriumph.com/)

The root for the documented production API can be found at: [*https://api.mytriumph.com/*](https://api.mytriumph.com/)

These sites will allow you to connect using a browser to explore the API.

## Authentication

All API methods require authentication. The only exceptions are the 4 methods used in generating authentication tokens: **/v1/Login, /v1/OAuthUrl, /v1/OAuthToken,** and **/v1/OAuthRevoke.**

The API offers two ways to get an access token. **/v1/Login** can be used to generate a token if you have a customer’s username and password credentials. Alternatively, the API supports OAuth2 authentication. OAuth2 is the industry standard way of linking 2 separate systems together without sharing their username or password. Further details about the **/v1/OAuthUrl, /v1/OAuthToken,** and **/v1/OAuthRevoke** methods can be found below.

To gain access to all the other API methods, pass the access token’s value as a HTTP cookie named “SessionToken” or set the “Authentication” HTTP header to the token’s value.

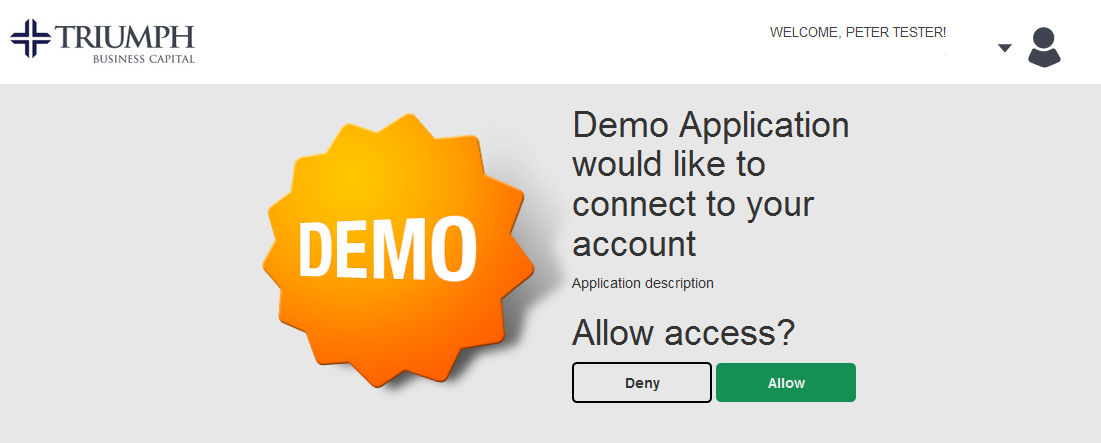
When the user has completed the call or group of calls they should call **/v1/Logout** or **/v1/OAuthRevoke** to make the access token invalid. After 20 minutes of inactivity the token will automatically be marked invalid.

OAuth2

OAuth 2.0 defines a server handshake process to securely link a user in your application with an account in the MyTriumph system. There are several tutorials of how OAuth 2 works online, but a brief summary of the process is described in this section.

Before an OAuth handshake can begin, your application and the Triumph API need to be setup. We will provide an API key (client\_id) and secret password (client\_secret) that will be used for your application to identify itself. You will need to provide your application’s name and redirect URL along with optionally providing a logo and description.

The OAuth handshake is initiated by the customer using their browser. You will provide a specially crafted link to the MyTriumph site. The API can build this link for you if you call the **/v1/OAuthUrl** method. When the customer’s browser displays the MyTriumph page, they will be presented with the following screen where the name, logo, and description will be shown to the customer.



If the customer presses the allow button, the Triumph API will generate an authorization code and append the code to the redirect\_url as a parameter named “code”; the customer will be redirected back a page in your application specified by the redirect url, and your page should use the code parameter for the next step of the OAuth handshake. If they press the deny button, they will be redirected, using the redirect\_url) back to your application with a parameter named “error” and the value of “access\_denied”.

At this point your application will have an authorization code. Since this authorization code has been passed from the Triumph API to your application using the client’s browser, it will need to be exchanged for an access token which is communicated directly from the Triumph API to your application. Your application will need take the authorization code and send it to **/v1/OAuthToken** which will respond with an access token that will allow access to the rest of the API. The access token will expire after 20 minutes of inactivity.

If you wish to access the Triumph API without requiring the customer to re-grant access every 20 minutes, then at the time you call the **/v1/OAuthUrl** method, be sure to set the access\_type parameter to offline. When offline access is requested, then **/v1/OAuthToken** will respond with a refresh token in addition to an access token. A refresh token is valid indefinitely, and can be re-passed to **/v1/OAuthToken** to obtain a new access token. Please note that each customer of your application can only have one refresh token tied to their MyTriumph account at a time. If you redirect the customer’s browser to the **/v1/OAuthUrl** requesting offline access a 2nd time, when they press the allow button, a new refresh token will be created and the existing refresh token will be expired.

## Dates

Dates should be passed using the yyyy-mm-dd format. When a DateTime is requested, the format should be yyyy-mm-dd hh24:mm:ss. All time should be passed in 24 hour format. Developers should be careful to note that all of Triumph Business Capital’s servers are central time.

## Errors

All methods will return a valid response or an error status with a message. Some methods don’t require a value to be returned but will still return a valid response object to say that the call was successful. Some methods do not specify any faults returned. This implies that any faults will be exceptional and should be logged and discussed with Triumph Business Capital to determine how to resolve them.

## Required Fields

Parameters listed in the Service Methods section that are required are indicated by an \*.

# Authentication Service Methods

Most service methods under the /v1/ URL are accessible without an access token. The exceptions will be noted. They focus on methods involving authentication.

## POST /v1/Login

* Provides an access token. See the Authentication section for more information.
* Input
  + *username\** – string: the account username. Accounts setup on the Triumph Business Capital client portal (<https://mytriumph.com> ) can be used. We recommend setting up a separate account specifically for the API to avoid breaking any integration if the customer resets their account password. Please contact Triumph Business Capital if you need help accessing the client portal to setup an account.
  + *password\** – int: the account password
  + *apiKey\** - string: the API key. Provided by Triumph Business Capital. Used to identify your application.
* Output
  + *SessionToken* - string (20): The valid session token
  + *Error* – bool: Is set to true when an error occurs.
  + *ErrorMessage* – string: will contain a message when *Error* is set to true

## POST /v1/Logout

* Must have a valid access token to use this method. After calling this method, the token will no longer be active.
* Input
  + *None*
* Output
  + *Error* – bool: Is set to true when an error occurs.
  + *ErrorMessage* – string: will contain a message when *Error* is set to true

GET /v1/OAuthUrl

* The first step in the OAuth2 handshake is to direct the user’s browser to the API to obtain an authorization code. The **/v1/OAuthUrl** can be used to build the url to direct a user to. This URL will not change and can be cached in your application.
* Input
  + *response\_type – string:* must be set to “code”. Indicates the type of response to be returned from the handshake. Included to conform to the OAuth2 spec.
  + *client\_id\* – string:* the API key. Provided by Triumph Business Capital. Used to identify your application.
  + *redirect\_uri\* – string:* determines to where the user’s browser should be redirected with the authorization code. The value of this parameter must be predetermined and given to Triumph Business Capital. The value passed must match exactly the value provided including case, the https://, and any trailing /.
  + *scope – string:* the level of access to be obtained from the handshake. If this is passed, it must be set to “all”. Included to conform to the OAuth2 spec.
  + *state – string:* provides any state that might be useful to your application when the user’s browser is redirected back to the redirect\_uri along with the authentication code. The API roundtrips this parameter, so your application receives the same value it sent. To mitigate against cross-site request forgery (CSRF), it is strongly recommended to include an anti-forgery token in the state, and confirm it in the response.
  + *access\_type* – string: the type of access that is being requested. Available options are “online” and “offline”. Indicates whether your application needs to access the API when the user is not present at the browser. This parameter defaults to “online”. If your application needs to refresh access tokens when the user is not present at the browser, then use “offline”. See the OAuth section for more information.
* Output
  + *Url* - string: The url to redirect any user to obtain an authorization code. This url will not change and can be cached.
  + *Error* – bool: Is set to true when an error occurs.
  + *ErrorMessage* – string: will contain a message when *Error* is set to true

GET /v1/OAuthToken

* Provides an access token. This method will provide an access token if you provide a valid refresh token or a valid authorization code. See the Authentication and OAuth sections for more information.
* Input (when using an authorization code).
  + *code\* – string:* the authorization code. Your application will get this code from the customer’s browser when they are returned to the redirect\_url.
  + *client\_id\* – string:* the API key. Please contact Triumph Business Capital to get one. The same API key can be used for multiple customers; it is used to uniquely identify the company that is integrating with our API.
  + *client\_secret\* – string:* Please contact Triumph Business Capital to get one. This is required to prove that your application is authorized to use the client\_id parameter.
  + *redirect\_uri – string:* the url that the customer’s browser was directed to by Triumph API when the code was created.
  + *grant\_type\* – string:* must be set to “authorization\_code”
* Input (when using a refresh token).
  + *refresh\_token\* – string:* the refresh token.
  + *client\_id\* – string:* the API key. Please contact Triumph Business Capital to get one. The same API key can be used for multiple customers; it is used to uniquely identify the company that is integrating with our API.
  + *client\_secret\* – string:* Please contact Triumph Business Capital to get one. This is required to prove that your application is authorized to use the client\_id parameter.
  + *grant\_type\* – string:* must be set to “refresh\_token”
* Output (when input was valid)
  + *access\_token* - string: The access token. Used to access the Triumph API. Expires after 20 minutes of inactivity.
  + *refresh\_token* – string: The refresh token. Used obtain new access tokens. Never expires. Will only be present if access\_type was “offline” during the authorization\_code request.
  + *token\_type –* string: will always be Bearer. Included to conform to the Oauth2 spec.
* Output (when input was invalid)
  + *error* - string: Describes why the token was not provided.

GET /v1/OAuthRevoke

* Used to expire an active access token or refresh token.
* Input
  + *token – string:* the active token that should be revoked.
* Output (when input was invalid)
  + *success:* string- set to “Token revoked”
* Output (when input was invalid)
  + *error* - string: Describes why the token was not revoked. Typically, this will only happen if the token was previously expired.

## POST /v1/SingleSignOnUrl

* Must have a valid access token to use this method. This method will return a one-time url to automatically log the user into the MyTriumph.com website without having to enter a username and password.
* Input
  + *None*
* Output
  + *Url –* string: the Url to redirect the customer to the MyTriumph dashboard.
  + *Token –* string: a token that can be appended to any MyTriumph url to allow access.
  + *Error* – bool: Is set to true when an error occurs.
  + *ErrorMessage* – string: will contain a message when *Error* is set to true

## POST /v1/EmailPasswordReset

* Emails a new password to the user.
* Input
  + *username\** – string: the account username. Accounts setup on the Triumph Business Capital client portal (<https://mytriumph.com> ) can be used. We recommend setting up a separate account specifically for the API to avoid breaking any integration if the customer resets their account password. Please contact Triumph Business Capital if you need help accessing the client portal to setup an account.
* Output
  + *Error* – bool: Is set to true when an error occurs. When set to false, the email was successfully sent.
  + *ErrorMessage* – string: will contain a message when *Error* is set to true

## GET /v1/UserInfo

* Returns information about the user
* Input
  + *None*
* Output
  + email – string: Email address of username
  + name – string: Full name of username
  + phone\_number – string: Phone number of username
  + customer\_name – string: Name of the company that the username it tied to
  + customer\_number – string: The number of the company used by Triumph’s marketing department
  + is\_freight\_broker – bool: Returns true or false when the client that the user is currently associated with is a freight broker
  + *Error* – bool: Is set to true when an error occurs. When set to false, the email was successfully sent.
  + *ErrorMessage* – string: will contain a message when *Error* is set to true

# Listing Service Methods

Service methods under the /v1List/ URL require a valid access token and provide enumerable listings of ids that are required for other methods.

## GET /v1List/BatchStatusTypes

* A method that is used to retrieve all available Batch Status Types. This list does not change and can be cached. A batch’s status provides insight on the lifetime of the invoice from the point it has been submitted by the customer to the point where the customer is paid. The batchStatusTypeId integer is returned by **/v1Report/GetInputStatus** and  **/v1Report/GetInputStatusByReferenceKey**
* Input
  + *None*
* Output (array)
  + Error – bool: Is set to true when an error occurs.
  + ErrorMessage – string: will contain a message when Error is set to true
  + batchStatusTypes (array)
    - *batchStatusTypeId* – int: The ID of the batch status type.
    - *name* – string: the name of the batch status type.

GET /v1List/CarrierPaymentTerms

* A method that is used to retrieve Carrier Payment Terms. This list does not change for a customer and can be cached. carrierPmtTerms is a field in the **/v1Submit/UpdateProkerCarrierDetails** and **/v1Submit/UpdateProkerCarrierDetailsFromArray**
* Input
  + *None*
* Output (array)
  + Error – bool: Is set to true when an error occurs.
  + ErrorMessage – string: will contain a message when Error is set to true
  + carrierPmtTerms (array)
    - *name* – string: the name of the carrier payment term.

## GET /v1List/CreditResultTypes

* A method that is used to retrieve all details about possible credit results. This list does not and can be cached. *creditResultTypeId*  is given by **/v1Credit/SearchCredit** and **/v1List/Customers**
* Input
  + *None*
* Output (array)
  + Error – bool: Is set to true when an error occurs.
  + ErrorMessage – string: will contain a message when Error is set to true
  + CreditResultTypes (array)
    - *creditResultTypeId*  – int: The ID of the credit result type.
    - *name* – string: the plain text name of the credit result type.
    - *creditLimit* – decimal: the approved credit limit for the debtor.

## GET /v1List/Customers

* A method that is used to retrieve all debtors from the Triumph API. To obtain a list of all currently active debtors for an account, do not include a searchText parameter and set existingCustomersOnly to true. The *DebtorKey* field is called *triumphCustomerId*  when it is used by **/v1Submit/CreateInput**
* Input
  + *searchText -* string: the result list will be filtered to only include debtors whose name is similar to the passed text.
  + *existingCustomersOnly -* bool: the result list will be filtered to only include debtors who currently have an existing relationship with the logged in user. Defaults to true.
  + topNRows – int: the result list will be limited to the given number of rows. Defaults to 100 if not passed.
* Output (array)
  + Error – bool: Is set to true when an error occurs.
  + ErrorMessage – string: will contain a message when Error is set to true
  + Customers (array)
    - *DebtorKey* – int: Triumph’s unique identifier for the customer.
    - *Name\** – string(100): The name of the bill-to customer.
    - *MotorCarrNo* – string(30): The MC or DOT # of the customer.
    - *Addr1* – string(100): The address of the customer.
    - *Addr2* – string(100): The address of the customer.
    - *City* – string(50): The city of the customer.
    - *State* – string(2): The state of the customer.
    - *ZipCode* – string(10): The zip code of the customer.
    - *Phone1* – string(20): The phone number of the customer.
    - *Fax* – string(20): The fax number of the customer.
    - *CreditLimit* – decimal: The credit limit of the customer.
    - *IsOriginalRequired* – bool: True if this customer requires the original paperwork be sent.
    - HasRelationship – bool: True if this customer has had previous interaction with the currently logged in account
    - CreditResultTypeId – int: The Credit rating of this debtor. All available types are found by calling the **/v1List/CreditResultTypes** method.

## GET /v1List/DocTypes

* A method that is used to retrieve all available Doc Types. This list does not change for a customer and can be cached. documentTypeId is used by **/v1Submit/CreateDocument**
* Input
  + *None*
* Output (array)
  + Error – bool: Is set to true when an error occurs.
  + ErrorMessage – string: will contain a message when Error is set to true
  + documentTypes (array)
    - *documentTypeId* – int: The ID of the doc type.
    - *name* – string: the name of the doc type.

GET /v1List/FundingOptions

* A method that is used to retrieve all built in funding options. Funding options can be used to populate the fundingInstructions field in **/v1Submit/FinalizePendingInput**  and **/v1Submit/FinalizePendingInputArray**
* Input
  + *None*
* Output (array)
  + Error – bool: Is set to true when an error occurs.
  + ErrorMessage – string: will contain a message when Error is set to true
  + fundingOptions (array)
    - cliBankId – int: The unique identifier for this bank funding type. Required by **/v1Submit/FinalizeFuelAdvanceInputsFromArray**
    - *isDefault* – bool: True for the customer’s preferred funding method.
    - *name* – string: the string value for the fundingInstructions

GET /v1List/InputChargeTypes

* A method that is used to retrieve all available Input Charge Types. This list does not change for a customer and can be cached. inputChargeTypeId is required by **/v1Submit/CreateInputCharge**
* Input
  + *None*
* Output (array)
  + Error – bool: Is set to true when an error occurs.
  + ErrorMessage – string: will contain a message when Error is set to true
  + inputChargeTypes (array)
    - *inputChargeTypeId* – int: The ID of the input charge type.
    - *name* – string: the name of the input charge type.

GET /v1List/InvoiceCutoffTime

* A method that is used to retrieve the daily cutoff time for an customer. Invoices that are submitted after this time will not be processed until the next business day. This value does not change for a customer and can be cached.
* Input
  + *None*
* Output (array)
  + Error – bool: Is set to true when an error occurs.
  + ErrorMessage – string: will contain a message when Error is set to true
  + CutoffTime – string: the time that can be displayed to a client

## GET /v1List/InvoiceStatusTypes

* A method that is used to retrieve all available Invoice Status Types. This list does not change and can be cached. An invoice’s status provides insight on the lifetime of the invoice from the point it has been submitted by the customer, past the point where the customer is paid, and onto the point that the debtor has paid Triumph. The invoiceStatusTypeId integer is returned by **/v1Report/GetInputStatus, /v1Report/GetInputStatusByReferenceKey** and **/v1Report/ProblemInvoiceSummary**
* Input
  + *None*
* Output (array)
  + Error – bool: Is set to true when an error occurs.
  + ErrorMessage – string: will contain a message when Error is set to true
  + invoiceStatusTypes (array)
    - *invoiceStatusTypeId* – int: The ID of the invoice status type.
    - *name* – string: the name of the invoice status type.

# Credit Service Methods

Service methods under the /v1Credit/ URL require a valid access token and are provide methods to retrieve credit information about debtors from the Triumph API. If the logged in user does not have “Credit” rights on their MyTriumph account, then all the methods in this section will return an “Access Denied” error.

## POST /v1Credit/SearchCredit

* A method to retrieve the credit status of a debtor.
* Input
  + *docket\_number\** - string(10): The MC# or FF# for the debtor to get credit details on.
  + *docket\_type\** – string(2): Either “”MC” or “FF”. Indicates the type of docket\_number to search on.
* Output
  + *Error* – bool: Is set to true when an error occurs.
  + *ErrorMessage* – string: will contain a message when *Error* is set to true
  + *creditResultTypeId* – int: The credit. Details on the *searchCreditResultTypeId* can be found using the **/v1List/CreditResultTypes** method.
  + *companyName* – string: The name of the company that matches the submitted docket\_number.
  + *city* – string: The company’s city that matches the submitted docket\_number.
  + *state* – string: The company’s state that matches the submitted docket\_number.
  + *phone* – string: The phone number of the company that matches the submitted docket\_number.

# Submission Service Methods

Service methods under the /v1Submit/ URL require a valid access token and are provide methods to create and submit inputs for funding. If the logged in user does not have “Funding” rights on their MyTriumph account, then all the methods in this section will return an “Access Denied” error.

Inputs are complex objects with a lot of configurable fields. So the API enables the generation of these items using a “shopping cart” approach. The first step in the process will always be call the **/v1Submit/CreateInput** method, which returns an inputId. Additional /v1Submit/ methods can be called with that inputId to configure Broker specific details, assign Documents, and add InputCharges. Once the input has been fully configured, it can be submitted to Triumph for processing by “checking out” using either **/v1Submit/FinalizePendingInput**  or **/v1Submit/FinalizePendingInputArray**. Once Finalize has been called, Triumph will begin to process the input; no further modifications can be made.

## POST /v1Submit/CreateInput

* A method used to create an input to be purchased by Triumph Business Capital. This is the first method that should be called when submitting an Input to be purchased by Triumph Business Capital.
* Input
  + *referenceKey* – string (20): The sender’s unique key to reference this input.
  + invoiceNumber*\** – string (30): The invoice number for the input.
  + *invoiceDate* – Date (yyyy-mm-dd): The Date of the Invoice for the input.
  + *referenceNumber\** string (50): The Reference Number of the Input.
  + *grossAmount\** – decimal: The Gross Amount of the input.
  + *miscCharges* - decimal: The sum of other charges that affect the net invoice amount.
  + *isMiscInvoice\** – bool: Indicates whether the invoice is a miscellaneous invoice.
  + *triumphCustomerId* – int: Triumph’s unique identifier for the customer. Populating this field speeds up the payment processing for the input. Values for this field come from the *DebtorKey* response in the **/v1List/Customers** method.
  + *customerName\** – string(100): The name of the bill-to customer.
  + *customerId* – string(30): The MC or DOT # of the customer.
  + *customerAddr1* – string(100): The address of the customer.
  + *customerAddr2* – string(100): The address of the customer.
  + *customerCity* – string(50): The city of the customer.
  + *customerState* – string(2): The state of the customer.
  + *customerZipCode* – string(10): The zip code of the customer.
  + *customerPhone* – string(20): The phone number of the customer.
  + *customerFax* – string(20): The fax number of the customer.
  + *originCity* – string (50): The city of the first pickup location.
  + *originState* – string (2): The state of the first pickup location.
  + *originZip* – string (10): The zip code of the first pickup location.
  + *originPickupDate* – Date (yyyy-mm-dd): The date of the first pickup.
  + *destinationCity* – string (50): The city of the last delivery.
  + *destinationState* – string (2): The state of the last delivery.
  + *destinationZip* – string (10): The zip code of the last delivery.
  + *deliveryDate* – Date (yyyy-mm-dd): The date of the last delivery.
  + *equipmentType* – string (50): The type of equipment used in the haul. Possible values are “Van”, “Flatbed”, “Refrigerated”, “Intermodal”, “LTL”, and “Specialized”.
* Output
  + *InputId* – int: The Id of the created Input.
  + *Error* – bool: Is set to true when an error occurs.
  + *ErrorMessage* – string: will contain a message when *Error* is set to true

## POST /v1Submit/CreateInputsFromArray

* A method used to create multiple inputs to be purchased by Triumph Business Capital. This is the first method that should be called when submitting multiple Inputs to be purchased by Triumph Business Capital.
* Input
  + Array of CreateInput items. See **/v1Submit/CreateInput** for the full listing of parameters. Each item will need to be prefixed by the index number of the item (0 based). For example: the referenceKey parameter for the first input will be sent with the name “[0].referenceKey”. The invoiceDate parameter for the 2nd input will be sent with the name “[1].invoiceDate”.
* Output
  + *InputId* – int[]: An array of InputIds created.
  + *Error* – bool: Is set to true when an error occurs.
  + *ErrorMessage* – string: will contain a message when *Error* is set to true

## POST /v1Submit/UpdateBrokerCarrierDetails

* A method used to add or update the carrier details about an Input that has been created. This method requires the InputId returned by the **/v1Submit/CreateInput** method.
* Input
  + *inputId\** – int: The InputId returned by the **/v1Submit/CreateInput** method.
  + *carrierInvoiceNumber\** – string (30): The Invoice number of the carrier invoice.
  + *carrierGrossAmount* – decimal: The Gross amount due to the carrier.
  + *carrierName\** – string (50): The name of the carrier.
  + *carrierMc* – string (10): The Motor Carrier Code of the carrier.
  + *carrierDot* – string (10): The DOT code of the carrier.
  + *carrierAddr1* – string (100): First line of carrier payment address.
  + *carrierAddr2* – string (100): Second line of carrier payment address.
  + *carrierCity* – string (50): The city for carrier payments.
  + *carrierState* – string (2): The state for carrier payments.
  + *carrierZip* – string (10): The zip code for carrier payments.
  + *carrierPhone1* – string (15): Phone number of carrier.
  + *carrierPhone2* – string (15): Phone number of carrier.
  + *carrierFax* – string (15): Fax number of carrier.
  + *payableToFactor\** - bool: Flag indicating whether or not the payment instructions belong to a factor.
  + *factorName* – string(50): Name of the Factor that the carrier payment will be sent to.
  + *carrierPaymentNotificationEmail* – string (50): Email address for Carrier payment notifications.
  + *carrierPmtType\** – string (50): The type of payment that the carrier wants
  + carrierPmtTerms\* – string (50): The terms which the carrier will be paid in.
  + carrierRemitBankName – string (50): Name of the carrier’s bank.
  + *carrierRemitAddr1* – string (50): First line of carrier payment address.
  + *carrierRemitAddr2* – string (100): Second line of carrier payment address.
  + *carrierRemitCity* – string (50): The city for carrier payments.
  + *carrierRemitState* – string(2): The state for carrier payments.
  + *carrierRemitZip* – string (5): The zip code for carrier payments.
  + *carrierRemitAcctNumber\** – string (50): Account number for carrier account.
  + *carrierRemitABANumber* – string (50): ABA Number for carrier account.
  + *carrierRemitInterBankName –* string (50): Bank name for Intermediary Bank (Only Applicable for Wire Payments).
  + *carrierRemitInterAccountNumber –* string (50): Account number at intermediary bank (Only Applicable for Wire Payments).
  + *carrierRemitInterAbaNumber –* string (50): ABA number at intermediary bank (Only Applicable for Wire Payments).
* Output
  + *InputId* – int: The Ids of the affected Input.
  + *Error* – bool: Is set to true when an error occurs.
  + *ErrorMessage* – string: will contain a message when *Error* is set to true

## POST /v1Submit/UpdateBrokerCarrierDetailsFromArray

* A method used to add or update multiple carrier details about an Input that has been created. This method requires the InputIds returned by the **/v1Submit/CreateInput** method.
* Input
  + Array of AddBrokerCarrierDetails items. See **/v1Submit/AddBrokerCarrierDetail** for the full listing of parameters. Each item will need to be prefixed by the index number of the item (0 based). For example: the inputId parameter for the first input will be sent with the name “[0].inputId”. The carrierRemitZip parameter for the 2nd input will be sent with the name “[1].carrierRemitZip ”.
* Output
  + *InputId* – int[]: An array of Ids affected.
  + *Error* – bool: Is set to true when an error occurs.
  + *ErrorMessage* – string: will contain a message when *Error* is set to true

## POST /v1Submit/UpdateTransportationInputDetails

* A method that is used to add or update Transportation Details for an Input. This method requires the InputId returned by the CreateInput method.
* Input
  + *inputId\** – int: The InputId returned by the **/v1Submit/CreateInput** method.
  + *closedDateTime* – DateTime (yyyy-mm-dd hh24:mm:ss): The date and time that the load was closed in the TMS.
  + *originGpsLatitude* – string(20): The latitude of the truck at the first pickup.
  + *originGpsLongitude –* string(20): The longitude of the truck at the first pickup.
  + *destinationGpsLatitude –* string(20): The latitude of the truck at the last destination.
  + *destinationGpsLongitude –* string(20): The longitude of the truck at the last destination.
  + *totalAmountFuelPurchased* – decimal: Total dollar amount spent on fuel for the haul.
  + *totalGallonsFuelPurchased* – int: Total number of gallons of fuel used for the haul.
  + *truckOrderNotUsed* – bool: When true the Input is considered a “Truck Order Not Used”.
  + *shipperName* – string (50): The shipper name.
  + *pickupAddress* – string (50): The address of the pickup location.
  + *totalPickUpCount* – int: The total number of pickups for the input.
  + *pickupApptRequired* – bool: Set the value to True when a pickup appointment is required.
  + *firstPickupApptStartTime* – string (5): The first pickup appointment start time. Use 24 hr format.
  + *firstPickupApptEndTime* – string(5): The first pickup appointment end time. Use 24 hr format.
  + *consigneeName* – string (50): The name of the consignee.
  + *consigneeAddress* – string (50): The address of the consignee.
  + *totalDrops* – int: The total number of drops.
  + *deliveryApptRequired* – bool: Set the value to True when a delivery appointment is required.
  + *firstDeliveryApptTime* – string (5): The time of the first delivery appointment. Use 24 hr format.
  + *hazmat* – bool: Set to true if load is a hazmat haul.
  + *piecePalletCount* – int: Number of pieces/pallets on haul.
  + *weight* – int: The weight of the haul.
  + *freightValue* – decimal: The estimated value of the haul.
  + *practicalMileage* – int: The estimated number of miles of the haul.
  + *brokeredFreight* – bool: Set to true if haul was awarded from a broker.
  + *orderBookDateTime* – DateTime (yyy-mm-dd hh24:mm:ss): The date and time the order was booked.
* Output
  + *InputId* – int: The Id of the affected Input.
  + *Error* – bool: Is set to true when an error occurs.
  + *ErrorMessage* – string: will contain a message when *Error* is set to true

GET /v1Submit/PendingInput/*inputId*

* A method that is used to retrieve information about a pending input. This method requires the InputId returned by the **/v1Submit/CreateInput** method.
* Input
  + *inputId\** – int: The InputId returned by the **/v1Submit/CreateInput** method.
* Output
  + Input – Pending Input object
    - *inputId* – int: The InputId returned by the **/v1Submit/CreateInput** method.
    - referenceKey – string (20): The sender’s unique key to reference this input.
    - invoiceNumber\* – string (30): The invoice number for the input.
    - invoiceDate\* – Date (yyyy-mm-dd): The Date of the Invoice for the input.
    - referenceNumber\* string (50): The Reference Number of the Input.
    - grossAmount\* – decimal: The Gross Amount of the input.
    - miscCharges - decimal: The sum of other charges that affect the net invoice amount.
    - isMiscInvoice\* – bool: Indicates whether the invoice is a miscellaneous invoice.
    - customerName\* – string(100): The name of the bill-to customer.
    - customerId – string(30): The MC or DOT # of the customer.
    - customerAddr1 – string(100): The address of the customer.
    - customerAddr2 – string(100): The address of the customer.
    - customerCity – string(50): The city of the customer.
    - customerState – string(2): The state of the customer.
    - customerZipCode – string(10): The zip code of the customer.
    - originCity – string (50): The city of the first pickup location.
    - originState – string (2): The state of the first pickup location.
    - originZip – string (10): The zip code of the first pickup location.
    - originPickupDate – Date (yyyy-mm-dd): The date of the first pickup.
    - destinationCity – string (50): The city of the last delivery.
    - destinationState – string (2): The state of the last delivery.
    - destinationZip – string (10): The zip code of the last delivery.
    - deliveryDate – Date (yyyy-mm-dd): The date of the last delivery.
    - equipmentType – string (50): The type of equipment used in the haul. Possible values are “Van”, “Flatbed”, “Refrigerated”, “Intermodal”, “LTL”, and “Specialized”.
    - *carrierInvoiceNumber* – string (30): The Invoice number of the carrier invoice.
    - *carrierGrossAmount* – decimal: The Gross amount due to the carrier.
    - *carrierName* – string (50): The name of the carrier.
    - *carrierMc* – string (10): The Motor Carrier Code of the carrier.
    - *carrierDot* – string (10): The DOT code of the carrier.
    - *carrierAddr1* – string (100): First line of carrier payment address.
    - *carrierAddr2* – string (100): Second line of carrier payment address.
    - *carrierCity* – string (50): The city for carrier payments.
    - *carrierState* – string (2): The state for carrier payments.
    - *carrierZip* – string (10): The zip code for carrier payments.
    - *carrierPhone1* – string (15): Phone number of carrier.
    - *carrierPhone2* – string (15): Phone number of carrier.
    - *carrierFax* – string (15): Fax number of carrier.
    - *payableToFactor* - bool: Flag indicating whether or not the payment instructions belong to a factor.
    - *factorName* – string(50): Name of the Factor that the carrier payment will be sent to.
    - *carrierPaymentNotificationEmail* – string (50): Email address for Carrier payment notifications.
    - *carrierPmtType* – string (50): The type of payment that the carrier wants
    - carrierPmtTerms – string (50): The terms which the carrier will be paid in.
    - carrierRemitBankName – string (50): Name of the carrier’s bank.
    - *carrierRemitAddr1* – string (50): First line of carrier payment address.
    - *carrierRemitAddr2* – string (100): Second line of carrier payment address.
    - *carrierRemitCity* – string (50): The city for carrier payments.
    - *carrierRemitState* – string(2): The state for carrier payments.
    - *carrierRemitZip* – string (5): The zip code for carrier payments.
    - *carrierRemitAcctNumber\** – string (50): Account number for carrier account.
    - *carrierRemitABANumber* – string (50): ABA Number for carrier account.
    - *carrierRemitInterBankName –* string (50): Bank name for Intermediary Bank (Only Applicable for Wire Payments).
    - *carrierRemitInterAccountNumber –* string (50): Account number at intermediary bank (Only Applicable for Wire Payments).
    - *carrierRemitInterAbaNumber –* string (50): ABA number at intermediary bank (Only Applicable for Wire Payments).
    - closedDateTime – DateTime (yyyy-mm-dd hh24:mm:ss): The date and time that the load was closed in the TMS.
    - originGpsLatitude – string(20): The latitude of the truck at the first pickup.
    - originGpsLongitude – string(20): The longitude of the truck at the first pickup.
    - destinationGpsLatitude – string(20): The latitude of the truck at the last destination.
    - destinationGpsLongitude – string(20): The longitude of the truck at the last destination.
    - totalAmountFuelPurchased – decimal: Total dollar amount spent on fuel for the haul.
    - totalGallonsFuelPurchased – int: Total number of gallons of fuel used for the haul.
    - truckOrderNotUsed – bool: When true the Input is considered a “Truck Order Not Used”.
    - shipperName – string (50): The shipper name.
    - pickupAddress – string (50): The address of the pickup location.
    - totalPickUpCount – int: The total number of pickups for the input.
    - pickupApptRequired – bool: Set the value to True when a pickup appointment is required.
    - firstPickupApptStartTime – string (5): The first pickup appointment start time. Use 24 hr format.
    - firstPickupApptEndTime – string(5): The first pickup appointment end time. Use 24 hr format.
    - consigneeName – string (50): The name of the consignee.
    - consigneeAddress – string (50): The address of the consignee.
    - totalDrops – int: The total number of drops.
    - deliveryApptRequired – bool: Set the value to True when a delivery appointment is required.
    - firstDeliveryApptTime – string (5): The time of the first delivery appointment. Use 24 hr format.
    - hazmat – bool: Set to true if load is a hazmat haul.
    - piecePalletCount – int: Number of pieces/pallets on haul.
    - weight – int: The weight of the haul.
    - freightValue – decimal: The estimated value of the haul.
    - practicalMileage – int: The estimated number of miles of the haul.
    - brokeredFreight – bool: Set to true if haul was awarded from a broker.
    - orderBookDateTime – DateTime (yyy-mm-dd hh24:mm:ss): The date and time the order was booked.

GET /v1Submit/AllPendingInputs

* A method that is used to view all pending inputs that have not yet been submitted to Triumph for processing.
* Input
  + *None*
* Output
  + *Error* – bool: Is set to true when an error occurs.
  + *ErrorMessage* – string: will contain a message when *Error* is set to true
  + *Inputs* – Array of PendingInput items. See **/v1Submit/PendingInput** for the object detail list.

## DELETE /v1Submit/DeletePendingInput/*inputId*

* A method that is used to delete a pending input. This method requires the InputId returned by the CreateInput method.
* Input
  + *inputId\** – int: The InputId returned by the **/v1Submit/CreateInput** method.
* Output
  + *Error* – bool: Is set to true when an error occurs.
  + *ErrorMessage* – string: will contain a message when *Error* is set to true

## DELETE /v1Submit/DeleteAllPendingInputs

* A method that is used to delete all pending inputs.
* Input
  + none
* Output
  + *Error* – bool: Is set to true when an error occurs.
  + *ErrorMessage* – string: will contain a message when *Error* is set to true

POST /v1Submit/CreateDocument

* A method used to add an image file to an input. This method requires the InputId returned by the CreateInput method. This method supports Tif, JPG, and PDF file formats.
* Input
  + *inputId\** – int: The InputId returned by the **/v1Submit/CreateInput** method.
  + filename\* – string (50): The name of the file.
  + *fileData\** – byte[]: A byte array of the file.
  + *docType* - int[]: An integer array of doc types for each page of the file. See **/v1Report/GetDocTypes** for available docType values. Parameters will be
* Output
  + *documentId* – int[]: An integer array of the added document Ids, one for each page of the submitted file
  + *Error* – bool: Is set to true when an error occurs.
  + *ErrorMessage* – string: will contain a message when *Error* is set to true

GET /v1Submit/DocumentsForInputId/*inputId*

* A method that is used to retrieve a list of documents for a pending input. This method requires the inputId returned by the **/v1Submit/CreateInput** method.
* Input
  + *documentId\** – int: The DocumentId returned by the **/v1Submit/AddDocument** method.
* Output
  + *Error* – bool: Is set to true when an error occurs.
  + *ErrorMessage* – string: will contain a message when *Error* is set to true
  + *Documents[]*: an array of document objects with the following fields
    - documentId – int: An integer array of the added document Ids.
    - inputId – int: The InputId returned by the CreateInput method.
    - filename – string (50): The name of the file.
    - filesize – int: the number of bytes in the file
    - docType - int[]: An integer array of doc types for each page of the file. See **/v1Report/GetDocTypes** for available docType values.

GET /v1Submit/Document/*documentId*

* A method that is used to retreive a file that has been previously uploaded for a pending input. This method requires the documentId returned by the **/v1Submit/AddDocument** method.
* Input
  + *documentId\** – int: The DocumentId returned by the **/v1Submit/AddDocument** method.
* Output
  + *Error* – bool: Is set to true when an error occurs.
  + *ErrorMessage* – string: will contain a message when *Error* is set to true
  + *Document*: document object
    - documentId – int: An integer array of the added document Ids.
    - inputId – int: The InputId returned by the CreateInput method.
    - filename – string (50): The name of the file.
    - fileData – byte[]: A byte array of the file.
    - docType - int[]: An integer array of doc types for each page of the file. See GetDocTypes for available docType values.

DELETE /v1Submit/DeleteDocument/*documentId*

* A method that is used to delete a file from a pending input. This method requires the documentId returned by the **/v1Submit/AddDocument** method.
* Input
  + *documentId\** – int: The DocumentId returned by the **/v1Submit/AddDocument** method.
* Output
  + *Error* – bool: Is set to true when an error occurs.
  + *ErrorMessage* – string: will contain a message when *Error* is set to true

POST /v1Submit/AppendDocumentFromReferenceKey/

* Method used to add an image file to an existing Input based on the Reference Key passed into the **/v1Submit/CreateInput** method. This method requires the Reference Key that is managed by the user to find the Input. This method supports Tif, JPG, and PDF file formats. If an input is not found to add the documents to, an error message will be populated in the Output.
* Input
  + *referenceKey\** – int: The Reference Key passed into the **/v1Submit/CreateInput** method.
  + filename\* – string (50): The name of the file.
  + *fileData\** – byte[]: A byte array of the file.
  + *docType* - int[]: An integer array of doc types for each page of the file. See **/v1Report/GetDocTypes** for available docType values. Parameters will be
* Output
  + *Error* – bool: Is set to true when an error occurs.
  + *ErrorMessage* – string: will contain a message when *Error* is set to true

POST /v1Submit/CreateInputCharge

* A method used to record a charge for an input. Charges reflect expenses that will be have been deducted from an input’s grossAmount. This method requires the InputId returned by the **/v1Submit/CreateInput** method.
* Input
  + *inputId\** – int: The InputId returned by the **/v1Submit/CreateInput** method.
  + inputChargeType\* – int: An integer describing the type of input charge. See **/v1Report/GetInputChargeType** for available inputChargeType values.
  + *amount\** –decimal: The amount of the charge.
* Output
  + *inputChargeId* – int: The id of the InputCharge
  + *Error* – bool: Is set to true when an error occurs.
  + *ErrorMessage* – string: will contain a message when *Error* is set to true

GET /v1Submit/InputChargesForInputId/*inputId*

* A method that is used to list all details about input charges that have been previously submitted for a pending input. This method requires the inputId returned by the **/v1Submit/CreateInput** method.
* Input
  + *inputId\** – int: The InputId returned by the **/v1Submit/CreateInput** method.
* Output
  + *Error* – bool: Is set to true when an error occurs.
  + *ErrorMessage* – string: will contain a message when *Error* is set to true
  + *InputCharges[]*: an array of inputCharge objects
    - inputChargeId – int: An integer array of the added inputCharge Ids.
    - inputId – int: The InputId returned by the CreateInput method.
    - inputChargeType – int: An integer describing the type of input charge. See **/v1Report/GetInputChargeType** for available inputChargeType values.
    - amount –decimal: The amount of the charge.

GET /v1Submit/InputCharge/*inputChargeId*

* A method that is used to retreive details on an input charge that has been previously submitted for a pending input. This method requires the inputChargeId returned by the AddInputCharge method.
* Input
  + *inputChargeId\** – int: The InputChargeId returned by the **/v1Report/CreateInputCharge** method.
* Output
  + *Error* – bool: Is set to true when an error occurs.
  + *ErrorMessage* – string: will contain a message when *Error* is set to true
  + *InputCharge*: inputCharge object
    - inputChargeId – int: An integer array of the added inputCharge Ids.
    - inputId – int: The InputId returned by the CreateInput method.
    - inputChargeType – int: An integer describing the type of input charge. See **/v1Report/GetInputChargeType** for available inputChargeType values.
    - amount –decimal: The amount of the charge.

DELETE /v1Submit/DeleteInputCharge/*inputChargeId*

* A method that is used to delete a charge from a pending input. This method requires the inputChargeId returned by the **/v1Report/CreateInputCharge** method.
* Input
  + *inputChargeId\** – int: The InputChargeId returned by the **/v1Report/CreateInputCharge** method.
* Output
  + *Error* – bool: Is set to true when an error occurs.
  + *ErrorMessage* – string: will contain a message when *Error* is set to true

POST /v1Submit/FinalizePendingInput

* A method that is used to finalize a created Input and begin processing. An input cannot be modified once it is finalized. This is the last method that should be called when submitting Inputs to be purchased by Triumph Business Capital.
* Input
  + *inputId\* – int:* The InputId returned by the **/v1Submit/CreateInput** method.
  + fundingInstructions – string (200): Text describing how the funds should be disbursed. Predefined values for this can be listed with **/v1List/FundingOptions**
* Output
  + *confirmationCode* – string: A unique confirmation code of a successful finalization.
  + *Error* – bool: Is set to true when an error occurs.
  + *ErrorMessage* – string: will contain a message when *Error* is set to true

POST /v1Submit/FinalizePendingInputArray

* A method that is used to finalize a set of created Inputs and begin processing. An input cannot be modified once it is finalized. This is the last method that should be called when submitting Inputs to be purchased by Triumph Business Capital.
* Input
  + *inputIds[]\* – int:* The list of InputIds returned by the **/v1Submit/CreateInput** method.
  + fundingInstructions – string (200): Text describing how the funds should be disbursed. Predefined values for this can be listed with **/v1List/FundingOptions**
* Output
  + *confirmationCode* – string: A unique confirmation code of a successful finalization.
  + *Error* – bool: Is set to true when an error occurs.
  + *ErrorMessage* – string: will contain a message when *Error* is set to true

## POST /v1Submit/CreateFuelAdvanceInputsFromArray

* A method used to create multiple Fuel Advance Inputs to be purchased by Triumph Business Capital.
* Input
  + CreateFuelAdvanceInput items.
    - referenceKey – int: The sender’s unique key to reference this input.
    - referenceNumber – string: The Reference Number of the Fuel Advance.
    - grossAmount – decimal: Total load amount. Must be greater than 0.
    - advanceAmount – decimal: total requested advance amount. Must be less than 0.
    - debtorId – int: Triumph’s unique identifier for the customer. Populating this field speeds up the payment processing for the input. Values for this field come from the *DebtorKey* response in the **/v1List/Customers** method.
    - debtorName – string: The name of the debtor.
    - debtorBusinessId –string: The MC# or DOT# of the debtor.
    - debtorAddr1 –string: The address of the debtor.
    - debtorAddr2 –string: The address of the debtor.
    - debtorCity –string: The city of the debtor.
    - debtorState –string: The state of the debtor.
    - debtorZipCode –string: The zip code of the debtor.
    - debtorPhone –string: The phone number of the debtor.
    - debtorFax –string: The fax number of the debtor.
* Output
  + *InputId* – int[]: An array of InputIds created.
  + *Error* – bool: Is set to true when an error occurs.
  + *ErrorMessage* – string: will contain a message when *Error* is set to true

POST /v1Submit/FinalizeFuelAdvanceInputsFromArray

* A method that is used to finalize a set of created Fuel Advance Inputs and begin processing. A Fuel Advance input cannot be modified once it is finalized. This is the last method that should be called when submitting Fuel Advance Inputs to be purchased by Triumph Business Capital.
* Input
  + *inputIds[]\* – int:* The list of InputIds returned by the **/v1Submit/CreateInput** method.
  + additionalFundingDetails – string: Text describing how the funds should be disbursed.
  + cliBankId - int: The Triumph bank account key that the advance is related to. Value can be left null for the client to set the default value. The list of available cliBankIds is returned by the **/v1List/FundingOptions** method.
* Output
  + *confirmationCode* – string: A unique confirmation code of a successful finalization.
  + *Error* – bool: Is set to true when an error occurs.
  + *ErrorMessage* – string: will contain a message when *Error* is set to true

# Reporting Service Methods

Service methods under the /v1Report/ URL require a valid access token and are provide methods to retrieve information from the Triumph API. If the logged in user does not have “Reporting” rights on their MyTriumph account, then all the methods in this section will return an “Access Denied” error.

GET /v1Report/AccountBalance

* A method to retrieve the balances on a logged in customer’s account
* Input
  + None
* Output
  + *Error* – bool: Is set to true when an error occurs.
  + *ErrorMessage* – string: will contain a message when *Error* is set to true
  + *balance* – decimal: the dollar amount in account receivables.
  + *cashReserveBalance* – decimal: the dollar amount in cash reserves.
  + *escrowReserveBalance* – decimal: the dollar amount held in escrow.

## GET /v1Report/GetInputIdByReferenceKey/*referenceKey*

* A method to retrieve an InputId for a sender’s unique reference key.
* Input
  + *referenceKey\** – string: The sender’s reference key.
* Output
  + *Error* – bool: Is set to true when an error occurs.
  + *ErrorMessage* – string: will contain a message when *Error* is set to true
  + *inputId* – int: The InputId returned by the **/v1Submit/CreateInput** method.

GET /v1Report/GetInputStatusByReferenceKey/*referenceKey*

* A method to retrieve the status of an input within Triumph Business Capital using the sender’s unique reference key
* Input
  + *referenceKey\** – string: The sender’s reference key.
* Output
  + *Error* – bool: Is set to true when an error occurs.
  + *ErrorMessage* – string: will contain a message when *Error* is set to true
  + *invoiceStatusId* – int: the status of the input once is has become an invoice. See **/v1List/InvoiceStatusTypes** for a full list of types that can be returned. Will return null if the passed referenceKey is not found.
  + *batchStatusId* – int: the status of the purchase batch that this invoice is included in. See **/v1List/BatchStatusTypes** for a full list of types that can be returned. Will return null if the passed referenceKey is not found.

GET /v1Report/GetInputStatusArrayByReferenceKeys

* A method to retrieve the status of an input within Triumph Business Capital using the sender’s unique reference key
* Input
  + *referenceKeys[]\** – string: The sender’s reference keys.
* Output
  + *Error* – bool: Is set to true when an error occurs.
  + *ErrorMessage* – string: will contain a message when *Error* is set to true
  + *InputStatuses[] – an array of InputStatus objects*
    - *invoiceStatusId* – int?: the status of the input once is has become an invoice. See **/v1List/InvoiceStatusTypes** for a full list of types that can be returned. Will return null if the passed referenceKey is not found.
    - *batchStatusId* – int?: the status of the purchase batch that this invoice is included in. See **/v1List/BatchStatusTypes** for a full list of types that can be returned. Will return null if the passed referenceKey is not found.

## GET /v1Report/ProblemInvoiceSummary

* A method to display the invoice counts that are Disputed, Held or Over Age.
* Input
  + *none*
* Output
  + *Error* – bool: Is set to true when an error occurs.
  + *ErrorMessage* – string: will contain a message when *Error* is set to true
  + *Rows[]* – Array of InvoiceSummary object
    - *Measure* – string: The name for the invoice type, will be Disputed, Held or Over Age
    - *Count* – int: the count of invoices that are of this type
    - *InvoiceSummaryType* – int: the invoiceTypeId that matches the measure. The **/v1List/InvoiceStatusTypes** method provides the full list of invoice types

GET /v1Report/GetRecentBatchSummary

* A method to retrieve the status of batches that have been submitted today. This returns the same information as the “Where’s my Money” widget on the MyTriumph.com website.
* Input
  + *none*
* Output
  + *Error* – bool: Is set to true when an error occurs.
  + *ErrorMessage* – string: will contain a message when *Error* is set to true
  + *Rows[]* – Array of BatchSummary object
    - *BatchNo* – int: The Batch Number assigned by Triumph. Can be null if the Batch has not been fully processed yet.
    - *BatchStatusId* – int: the status of the batch. See **/v1List/BatchStatusTypes** for a full list of types that can be returned.
    - *Amount* – decimal: The total amount of invoices in the batch
    - *CarrierAmount* – decimal: The total amount of the carrierGrossAmount in the invoices in the batch. See *carrierGrossAmount* in **/v1Submit/UpdateBrokerCarrierDetails**
    - *TotalInvoiceCount – int:* The total number of invoices in the batch
    - *TotalInvoiceDeniedCount – int:* The total number of invoices in the batch that have been denied.
    - *TotalInvoiceHeldCount – int:* The total number of invoices in the batch that have been held.

GET /v1Report/GetRecentBatches

* A method to retrieve the most recent batches.
* Input
  + *topRows – int: the top number of results to return.*
* Output
  + *Error* – bool: Is set to true when an error occurs.
  + *ErrorMessage* – string: will contain a message when *Error* is set to true
  + *rows[]* – Array of RecentBatchResults object
    - *Amount* – decimal: The total dollar amount in the batch.
    - *BatchNo* – int: The user friendly Batch Number assigned by Triumph.
    - *Date* – date: The date of the Batch.
    - *Invoices* – int: The total number of invoices in the batch
    - *Payout – string:* The method of funding that the batch amount is sent to.
    - *TransKey – int:* The key assigned by Triumph related to the Batch.
    - *Type – string:* The type description of the Batch.