apexportal®

Technical Guide

apexportal API and OAuth Documentation

Contents

i. Overview

Feature Name	Portal API and OAuth
Description	The Portal API, integrated with OAuth authentication, provides a secure and efficient way to interact with apexportal and access its various features. This technical guide offers detailed information on authentication options, the OAuth token workflow, and the usage of different integration APIs such as Vendor Registration, Invoice and Payment, Employee, and Cash Management.
Value Proposition	The Portal API with OAuth simplifies the integration process and enhances security by allowing clients to securely access apexportal's features and retrieve essential data. It provides a standardized and scalable solution for seamless communication between client systems and the portal, enabling efficient data exchange and streamlined workflows.
Pre-requisites	Before utilizing the Portal API with OAuth, clients need to ensure the following: **apexportal Access:** You need to have access to apexportal and API credentials to interact with the Portal API. • **Integration Credentials:** For OAuth integration, an Implementation Specialist will provide you with a Client ID, Client Secret, Username, and Password, along with the Identity Provider REST endpoint. • **Secure Communication:** You should have a secure communication channel to receive the OAuth credentials.
Configuration Overview	To acquire an access token and utilize the Portal API with OAuth, follow these steps: 1. **One-time Manual Process:** The Implementation Specialist creates the Client Id, Client Secret, Username, and Password for OAuth access on behalf of the client. 2. **Store Credentials Securely:** Safely store the credentials generated in step 1 for future use. 3. **Request Access Token:** Send an HTTP POST request to the Identity provider token endpoint, including the Client Id and Client Secret in the "Authorization" header. The request body should contain the scope, grant_type, username, and password. 4. **Receive Access Token:** If the provided credentials are valid, the Identity provider will respond with an "access_token," which is a base 64 encoded JSON string. This token expires within an hour and can be cached for subsequent API requests. 5. **Use Access Token for API Calls:** Include the access token in the "Authorization" header of API requests using the format "Bearer {access_token}". Avoid sending credentials in the request body when using OAuth authorization.

	6. **Receive Data:** If the token is valid, you will receive the requested data from the API. If the token validation fails, you will receive an HTTP Status code 401 - Unauthorized.

ii. Client-Specific Details

Field	Description	Example
Client Swagger URL	The URL where the client can download the Swagger definition file.	www.example.com/swagger/docs/v1
Client Id	The unique identifier assigned to the client for OAuth access.	{client_id}
Client Secret	The secret key associated with the client Id for OAuth access.	{client_secret}
Username	The username provided for OAuth access.	{username}
Password	The password associated with the username for OAuth access.	{password}
Identity Provider REST Endpoint	The endpoint URL for the Identity provider to acquire the access token.	{https://identity.apexportal.net/connect/token}

iii. In-depth Feature Rundown

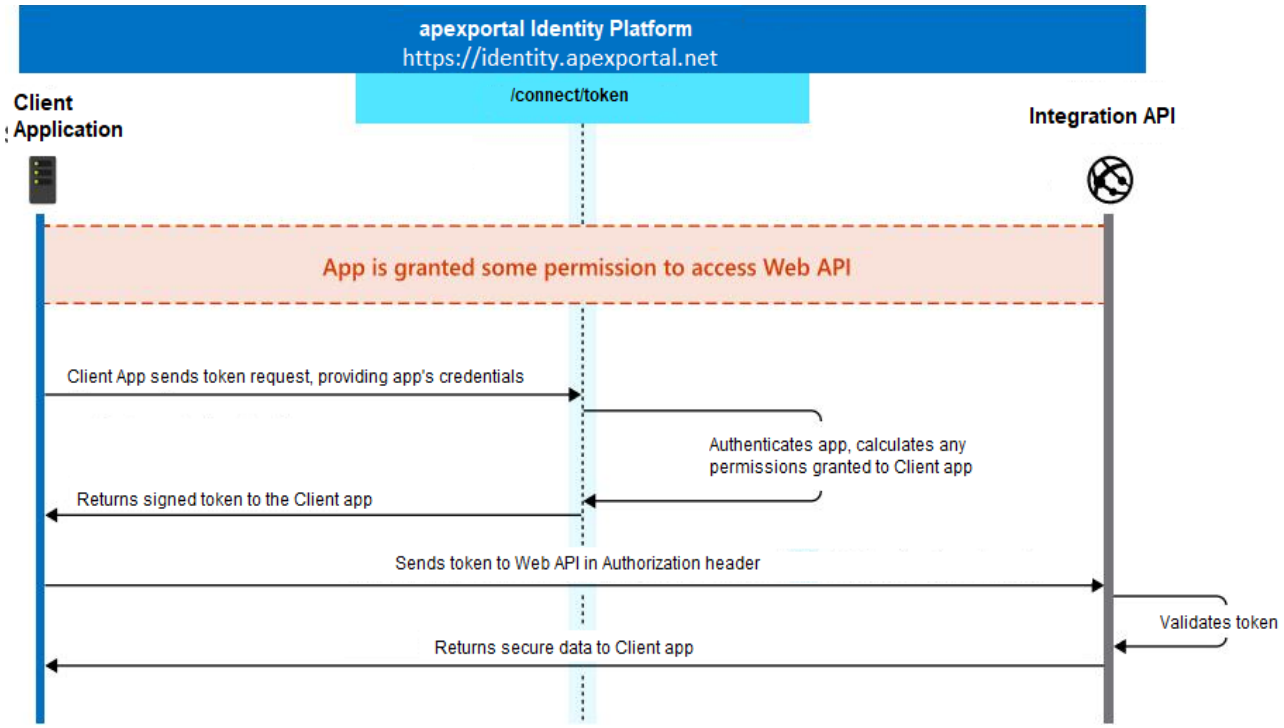
Authentication

apexportal and its APIs are enabled with HTTPS along with Transport Layer Security protocol (TLS) 1.2 or greater. Following are the different authentication options for the APIs:

- **Credentials (Username and Password)**
    - System generated unique Username/Password will be shared with client.
    - Clients must send their credentials for every request to Portal API within the request parameters for either REST or SOAP. These will be validated and a failure response will be sent back if invalid.
- **API Key Authentication (roadmap)**
    - A unique key is assigned to each client API user. When the user attempts to invoke the API, their unique key is used to authenticate to perform the transaction. This feature is in roadmap and will be available soon.
- **IP Whitelisting**
    - IP Whitelisting can be enabled to limit and control access only to trusted systems and networks.
- **Integration APIs with OAuth**
    - This section provides end users with the steps to call apexportal Integration APIs with OAuth authorization.

Swagger Definition File Download Link: <<Client URL>>/FirstStrike.VendorPortal.Integration/swagger/docs/v1

OAuth Token Workflow



Steps:

One-time manual process:

1. An implementation specialist will create the Client Id, Client Secret, Username, and Password for OAuth access on behalf of the client in the Apex system.
2. The implementation specialist will mark the client as Enabled for OAuth access in the Apex system.
3. The implementation specialist will securely communicate the credentials created in step 1 and the Identity provider REST endpoint (https://identity.apexportal.net/connect/token) to the client through a secure communication channel.

Steps to acquire an access token:

a) Send an HTTP POST request to the Identity provider token endpoint from step 1 with the following details:

1. Include the Client Id and Client Secret shared in step 1 in the "Authorization" header of the request as "Basic Authorization."
2. The request body should contain the following data:
3. scope=api_access&
4. grant_type=password&
5. username={Username from step 1}&
6. password={Password from step 1}

b) If the credentials passed in the request are valid, the Identity provider will respond with an "access_token."

c) The "access_token" is a base64-encoded JSON string that expires within an hour. The token can be cached in the client application until it expires.

d) The token can be used to access the following integration APIs:
/api/VendorRegistration
/api/InvoiceAndPayment
/api/Employee
/api/CashManagement

e) To call the apexportal integration API, include the "access_token" in the "Authorization" header of the API request. The format will be:
Authorization: Bearer {access_token}

f) Note that when an "access_token" is present, pass an empty object in the Credentials within the request body. The system behaves as follows: If the Credentials object is present in the request body, the system will use the Username and password from it, even if the access token is sent. Otherwise, the system uses the access_token to validate the user. Therefore, it's important not to include the credentials in the request body if you are using OAuth authorization.

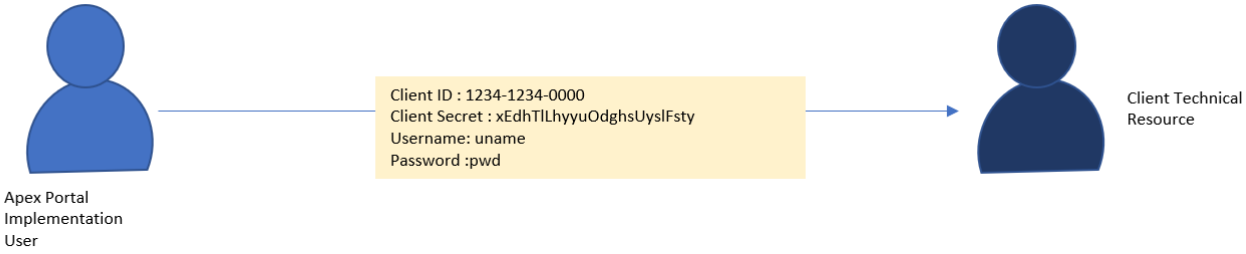
g) If the token validates, you will receive the requested data from the API.

h) If the token validation fails, you will receive an HTTP Status code 401 - Unauthorized. The response body may contain an error message indicating the cause of failure, such as an expired access token or insufficient user roles.

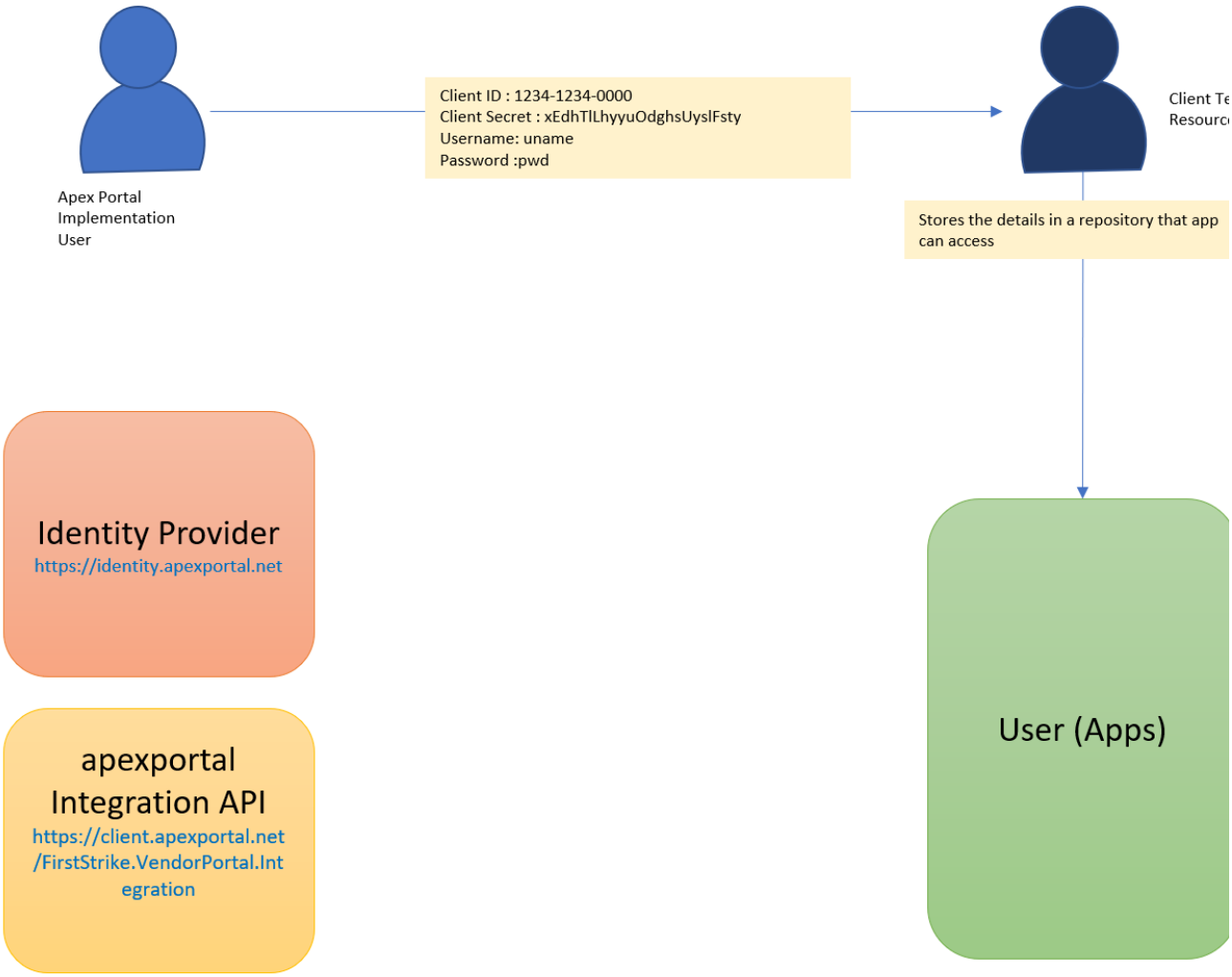
Sample error responses:
The access token has expired.
User doesn't have the required roles to access this resource.

Step by Step Flow (Diagrammatic Representation)

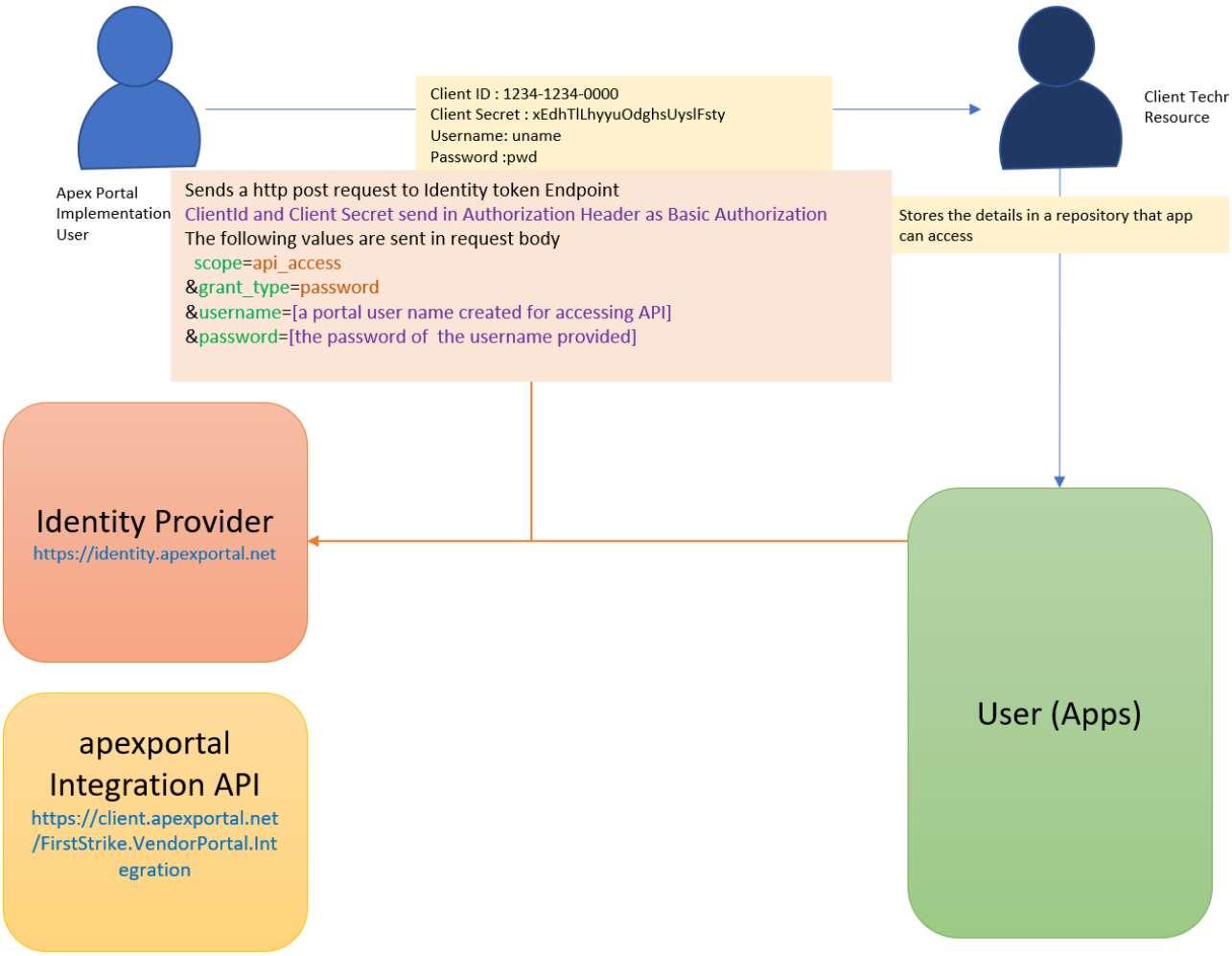
Step 1. Manual One-time process of sharing credentials



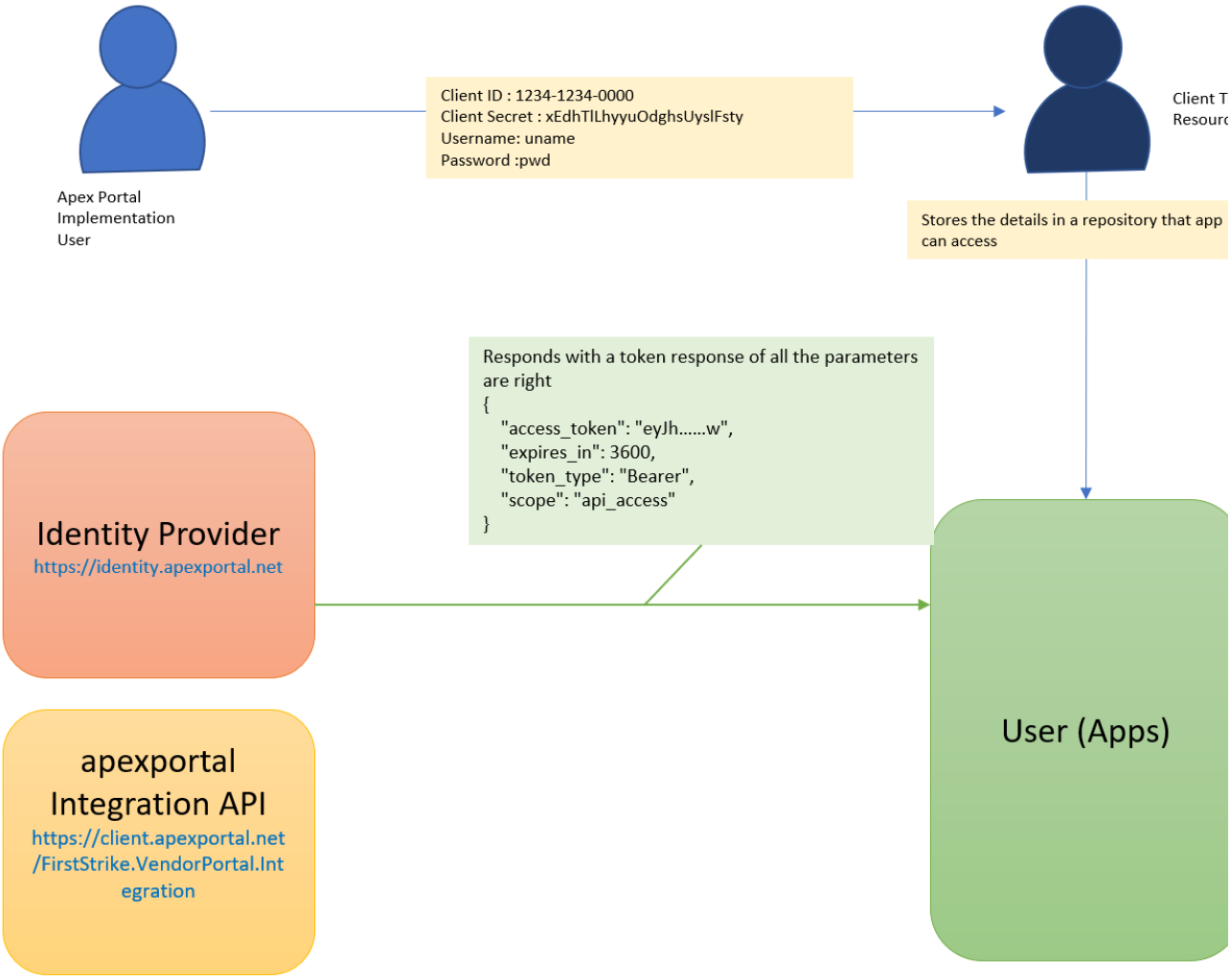
Step 2. Store the credentials securely



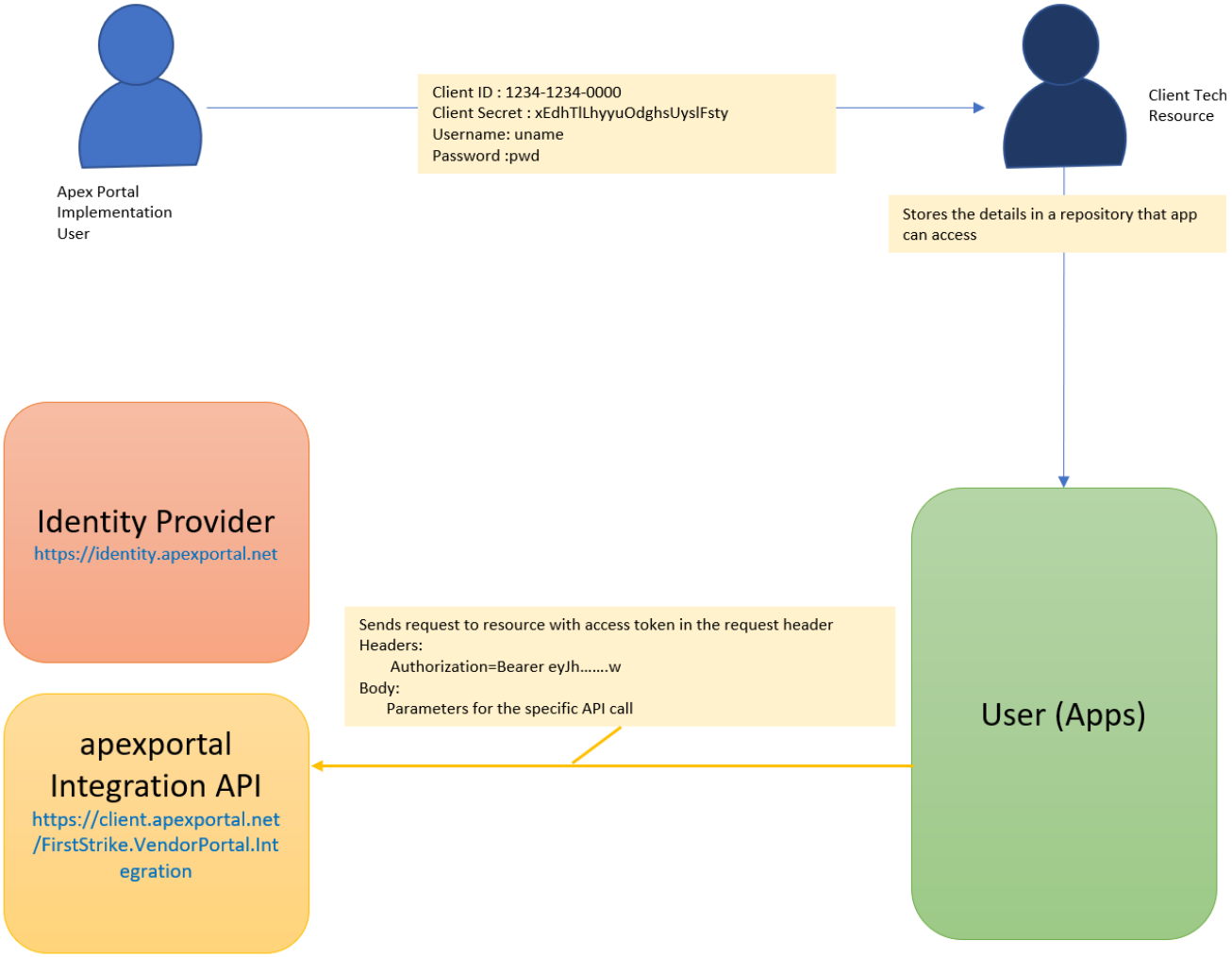
Step 3. Send a request for the token



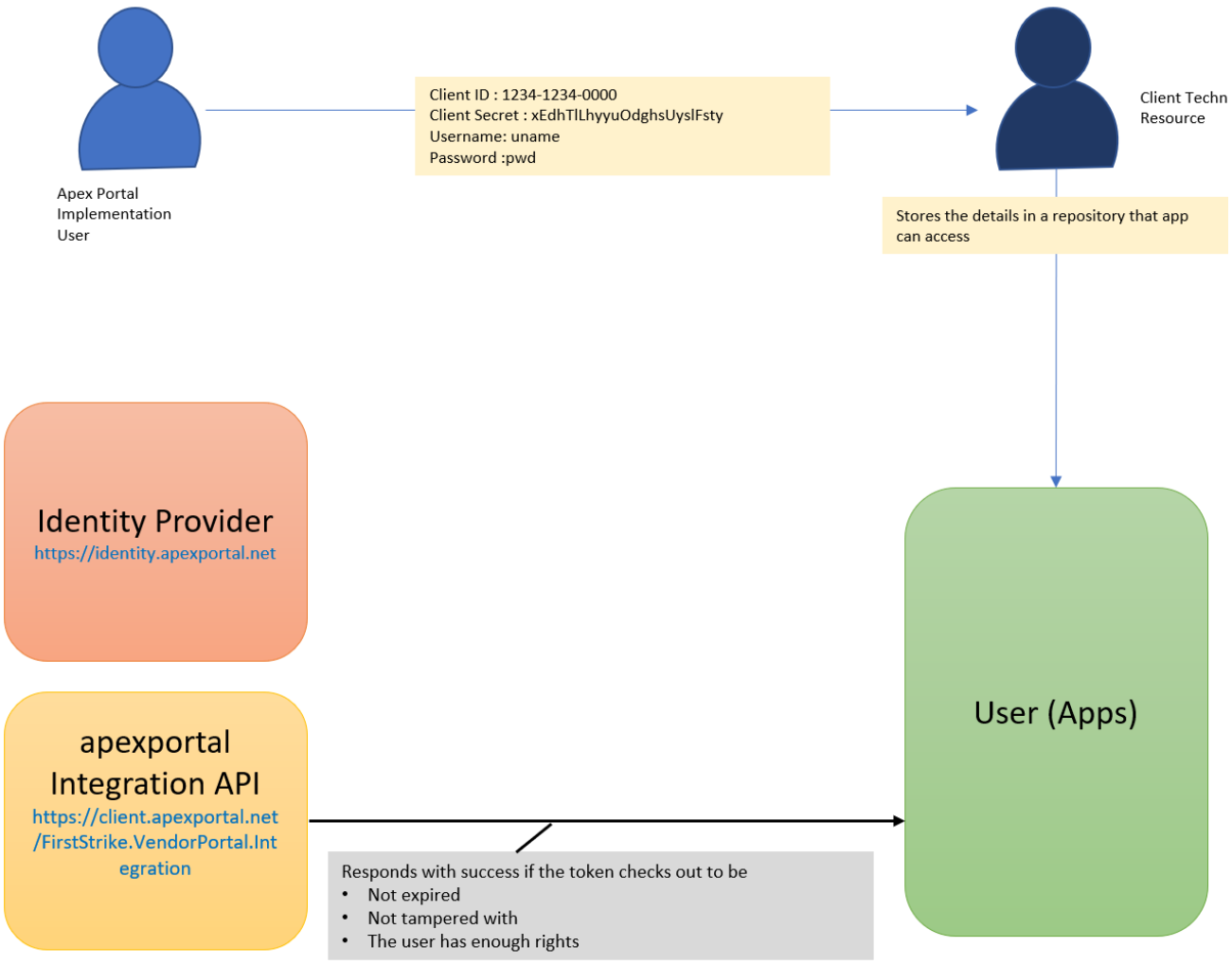
Step 4. Receive the token



Step 5. Use the token to access protected data



Step 6. Receive the protected data



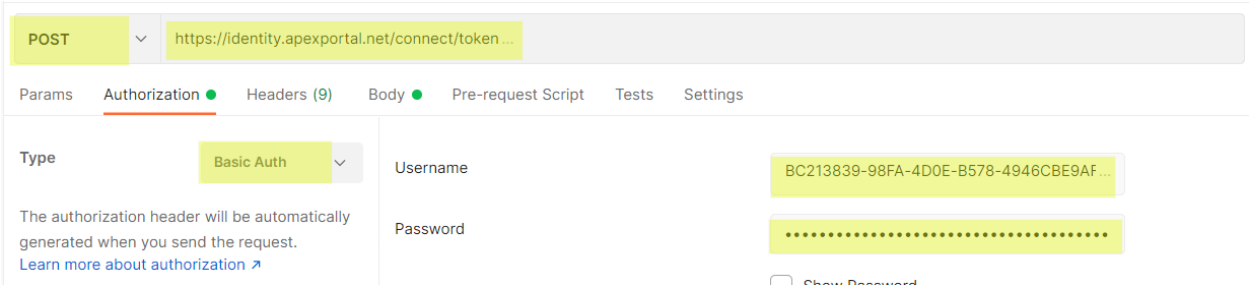
ERP to Portal

To fetch the bearer token, the client should send a POST request to the identity server using Basic Auth. The client ID should be provided as the username and the client secret key as the password. The authorization type must be set to "Basic Auth". The integration API of apexportal can be accessed by including the obtained "access_token" in the "Authorization" header.

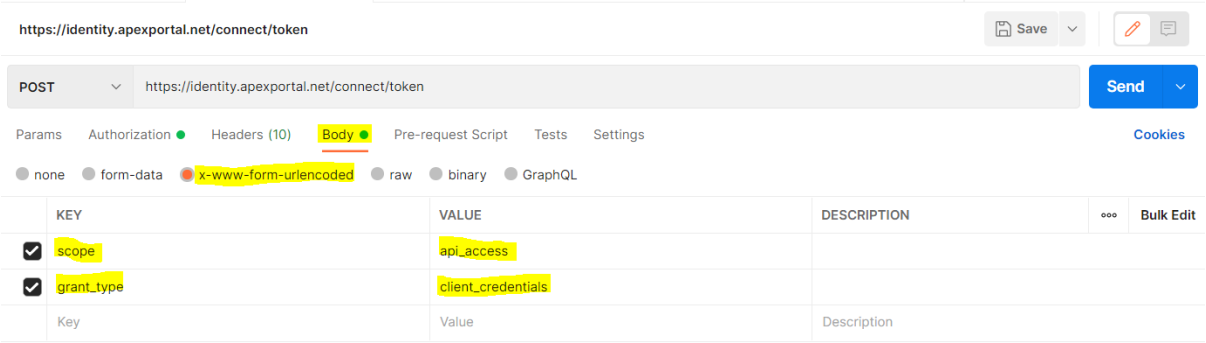
Step to acquire access token:

1. To successfully fetch token from our identity server, client needs to send a post request to our identity server. Regardless of the environment, whether it is a QA or UAT or Prod, client needs to send the request to "https://identity.apexportal.net/connect/token".

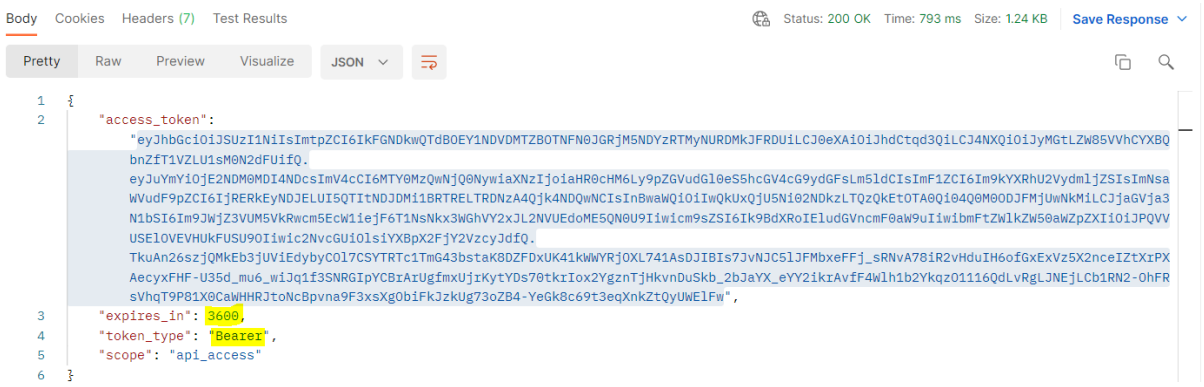
   Client needs to send the client id as username and client secret key as password and set auth-type as. Basic Auth. Please refer to the screenshot below.



2. In the request body, the client needs to include two parameters: "scope" and "grant_type". The value of the "scope" parameter should be set to "api_access", and the value of the "grant_type" parameter should be set to "client_credentials". These values are static and should not be changed. The client should send these parameters in the "application/x-www-form-urlencoded" format. Please refer to the screenshot below.



3. Upon receiving a valid POST request, the identity server will respond with a token that is valid for 3600 seconds. Please refer to the screenshot below for an example. When the token expires, the client needs to obtain a new one by following the above process.



Sample HTTP Requests

Sample Token Request:

```
POST /connect/token HTTP/1.1
Host: identity.apexportal.net
Authorization: Basic
QkMyMTM4MzktOThGQS00RDBFLUI1NzgtNDk0NkNCRTlBRjEzOkNCN0ZCRDc2MEEyOUU
1RDc5RkE5QzYwMzc5NDk0MDlDMERGM0I0QzIyMThFMDQ=
Content-Type: application/x-www-form-urlencoded
Content-Length: 46

scope=api_access&grant_type=client_credentials
```

Sample Token Response:

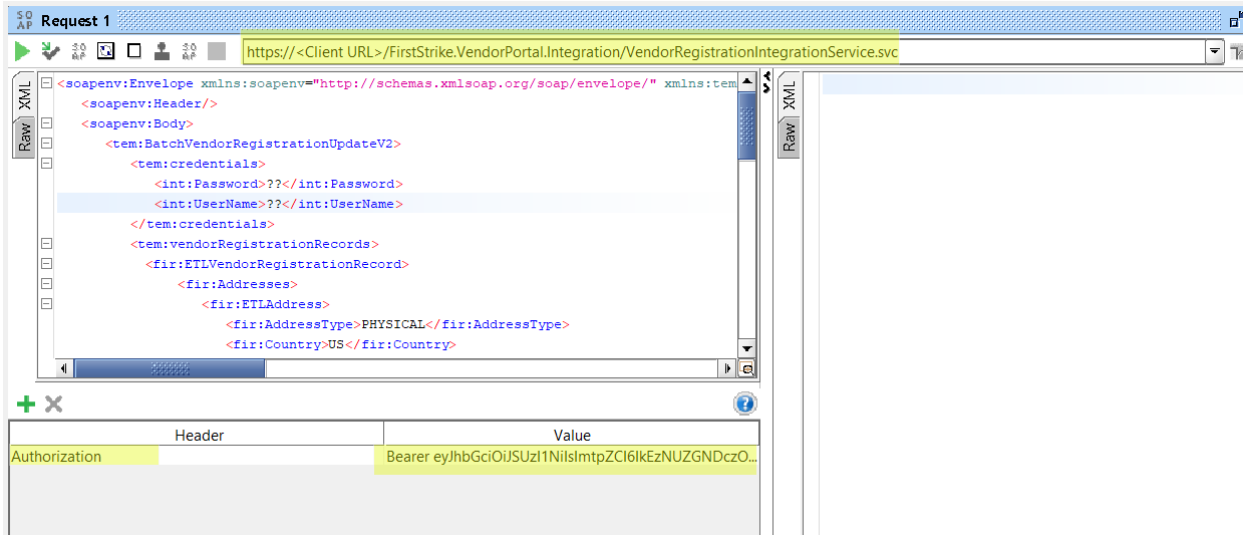

```
{
  "access_token": "eyJhbGciOiJSUzI1NiIsImtpZCI6IjVGOUM5RTUwRTEzMDNENzE3NkY4NUE5QkMwRjdFM0NDRUYzRjc4
OTMiLCJ0eXAiOiJhdCtqd3QiLCJ4NXQiOiJYNXllVU9Fd1BYRjItRnFid1BmanpOF9lSk0ifQ.eyJuYmYiOjE1ODUxODU4MTUsI
mV4cCI6MTU4NTE4OTQxNSwiaXNzIjoiaHR0cHM6Ly9sb2NhbGhvc3Q6NDQzNDkiLCJhdWQiOiJvZGF0YVNlcnZpY2UiLCJjb
GllbnRfaWQiOiI1MUM3MTlCNS1FN0ZDLTRFNTEtQTVDQS1FRTE2QTY0QjREMjEiLCJzdWIiOiJyYWphaml0IiwiYXV0aF90a
W1lIjoxNTg1MTg1ODEyLCJpZHAiOiJsb2NhbCIsInBwaWQiOiI0M2ViYmJhNS1lNmZjLTRkMWMtYjBkMy02YWI2YjE5OTE3Z
jUiLCJjaGVja3N1bSI6IlRtWWtHdmd3Qnh4ZlhyM29tYVVRSHAyOG5YdWtxUnBVRzJST0tBZWRTRUU9Iiwicm9sZSI6Ik9BdXR
oIEludGVncmF0aW9uIiwic2NvcGUiOlsiYXBpX2FjY2VzcyJdLCJhbXIiOlsiY3VzdG9tII19.Cmsfjxqk5CTFDCPW5EKArsSwcwRA3
A8CUa-2RQbN1PjnZ67D4PTd_qngTnsPw-
mPS2SUst47Vk_MmJXmSd0YXrEJlZjsbrd0YOBx9BEIS14QQQMH0NLanH97WjSFJtbE2z0cqAKOAxF8UsFBfkXExVFUxr1T4E
fJF-Bwdj1Q7CDh8UMkfGZRLFPi2P1pgSe-oRRzSNYdHe3t70MlO16FMbTiM_-
kSXK6cm1U2TzHcQJqLitcI5tZhQbt0V0A758HJ3zrR3gYndGrKxBVDWZk1ROEB9ThtPk1-TVrkaTEa1_qD7K-
ydYoMHf_bwUuBO1CgEzYSE3cA28vfABNLlkcPQ",
  "expires_in": 3600,
  "token_type": "Bearer",
  "scope": "api_access"
}
```

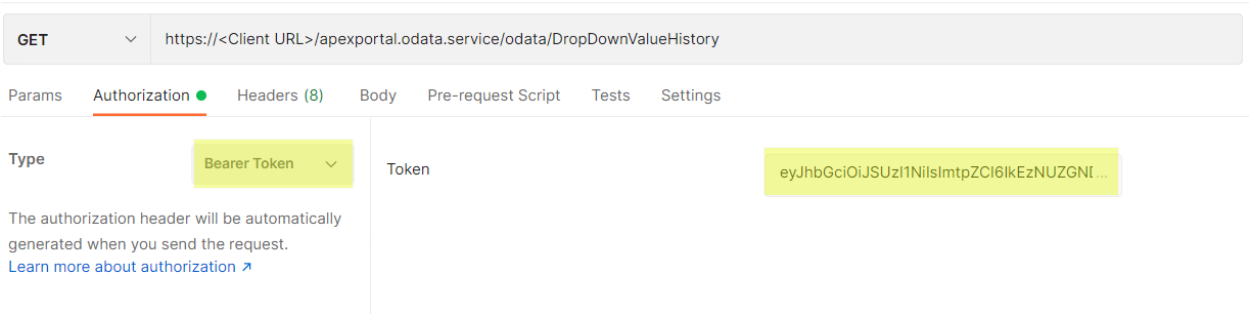

Use token to access API/Odata endpoint:

1. Once the client successfully fetches the token, they can utilize it to authenticate our APIs/OData endpoints. The client needs to include the token in the Authorization header of the

API request. The header value should be a string in the following format: "Bearer {fetchedToken}". The token should be preceded by the word "Bearer" followed by a space.

Please refer to the screenshots below.





Portal to ERP

To fetch the bearer token from the Token URL, Apex sends the username and password in the body of a POST request, rather than using authentication headers. When fetching the bearer token, the Apexportal configuration must be set to Authorization Type = "No Auth".

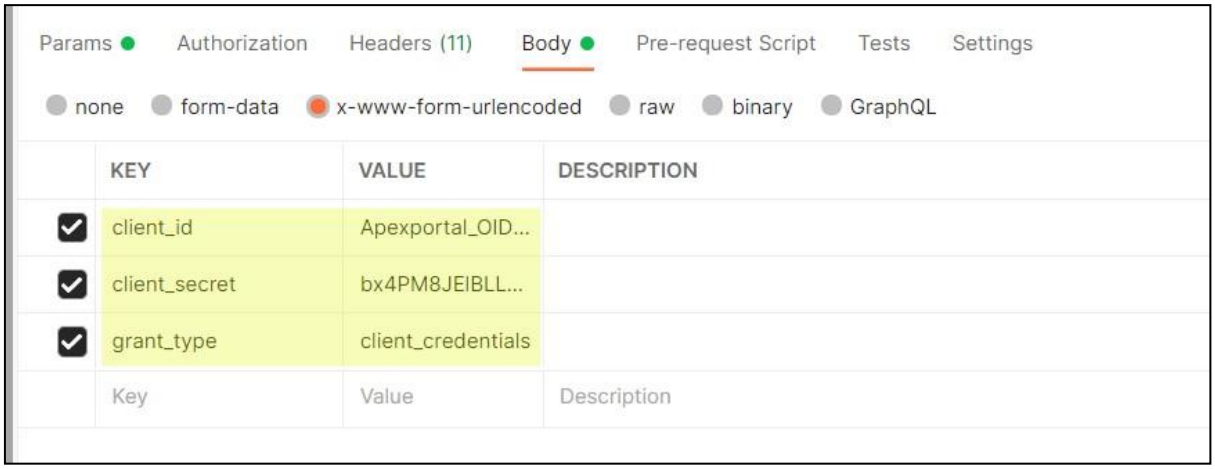
Steps:

Fetch token from provided endpoint via Authorization Type = No Auth, username and password. Below are the steps generated from Postman to fetch the bearer token.

1. Authorization Type = No Auth Whitelist below IP addresses
   216.250.228.0 - 216.250.228.255

216.250.229.0 - 216.250.229.255

2. Pass grant_type and Credentials as client_id and client_secret in body



3. Response Received – Status: 200 OK and Access_Token received in response body.

Sample HTTP Requests:

Sample Token Request:

Url:/connect/token
Method:POST
Headers
Authorization=Basic
RDk0NjFFOTAtRTk2Ni00NzQxLUE1OUUtQkU2MTA0REMxNzZEOjBFNERCMkEwJTI2QyMx
RjdCOTA0RDMwRTRGOCo0MjcwKjk1NTQqMTI3NDNGNkM=
Content-Length=67
Content-Type=application/x-www-form-urlencoded
Body
scope=api_access&grant_type=password&username=rajajit&password=password@1234

Sample Token Response:

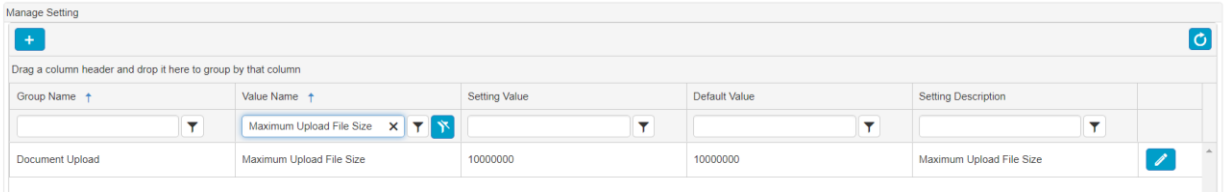
{
"access_token": "eyJhbGciOiJSUzI1NiIsImtpZCI6IjVGOUM5RTUwRTEzMDNENzE3NkY4NUE5QkMwRjdFM0NDRUYzRjc4
OTMiLCJ0eXAiOiJhdCtqd3QiLCJ4NXQiOiJYNXllVU9Fd1BYRjItRnFid1BmanpPOF9ISk0ifQ.eyJuYmYiOjE1ODUxODU4MTUsI
mV4cCI6MTU4NTE4OTQxNSwiaXNzIjoiaHR0cHM6Ly9sb2NhbGhvc3Q6NDQzNDkiLCJhdWQiOiJvZGF0YVNlcnZpY2UiLCJjb

```
GllbnRfaWQiOiI1MUM3MTlCNS1FN0ZDLTRFNTEtQTVDQS1FRTE2QTY0QjREMjEiLCJzdWIiOiJyYWphaml0IiwiYXV0aF90a
W1lIjoxNTg1MTg1ODEyLCJpZHAiOiJsb2NhbCIsInBwaWQiOiI0M2ViYmJhNS1lNmZjLTRkMWMtYjBkMy02YWI2YjE5OTE3Z
jUiLCJjaGVja3N1bSI6IlRtWWtHdmd3Qnh4ZlhyM29tYVVRSHAyOG5YdWtxUnBVRzJST0tBZWRTRUU9Iiwicm9sZSI6Ik9BdXR
oIEludGVncmF0aW9uIiwic2NvcGUiOlsiYXBpX2FjY2VzcyJdLCJhbXliOlsiY3VzdG9tII19.Cmsfjxqk5CTFDCPW5EKArsSwcwRA3
A8CUa-2RQbN1PjnZ67D4PTd_qngTnsPw-
mPS2SUst47Vk_MmJXmSd0YXrEJlZjsbrd0YOBx9BEIS14QQQMH0NLanH97WjSFJtbE2z0cqAKOAxF8UsFBfkXExVFUxr1T4E
fJF-Bwdj1Q7CDh8UMkfGZRLFPi2P1pgSe-oRRzSNYdHe3t70MIO16FMbTiM_-
kSXK6cm1U2TzHcQJqLitcI5tZhQbt0V0A758HJ3zrR3gYndGrKxBVDWZk1ROEB9ThtPk1-TVrkaTEa1_qD7K-
ydYoMHf_bwUuBO1CgEzYSE3cA28vfABNLlkcPQ",
  "expires_in": 3600,
  "token_type": "Bearer",
  "scope": "api_access"
}
```

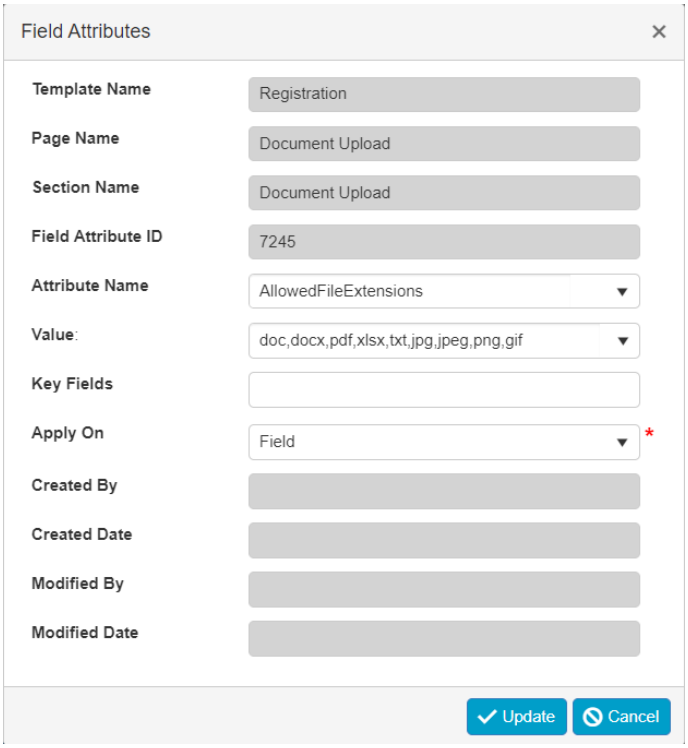
Documents API

APEX Size Restrictions and Formats

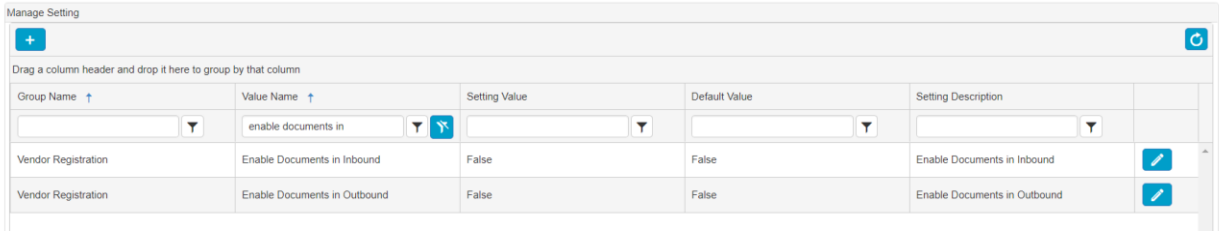
The size and type restrictions are fully customizable. We provide support for documents up to 2GB in size. However, by default, APEX has a size limit of 10MB, which can be found in the following sys_setting:



You can control the file types that users are allowed to upload by linking an attribute, specifically the AllowedFileExtensions, to the File Upload Control. By specifying the desired file extensions, only documents with those extensions will be permitted for users to upload. This restriction ensures that only the specified file types are accepted.



In order to have these documents sent to / received from the client, turn on the following sys_settings:



Once these settings have been enabled, the API responsible for retrieving Vendor Records from APEX to the client (GetVendorRegistration) will include the document content. The API structure remains unchanged; there are no new modifications required.

However, it is important to note that the document content will be encrypted using APEX's Private Key. APEX will decrypt the document content before sending it. The content will be in the form of a byte[] array, ensuring secure transmission of the document data.

Method Structure for API Documentation

Below is the sample API structure for a document using GetVendorRegistration API.


```
"Documents": [
  {
    "VendorId": "778899",
    "Description": "",
    "DocumentId": 93003,
    "DocumentName": "147.docx",
    "eSignDate": null,
    "eSigningUser": "",
    "ExpirationDate": null,
    "FileContent": "UEsDBBQABgAIAAAAIQCj77sdZQEAAFIFAAATAAgCW0NvbnRlbnRfVHlwZ
    "FileName": "147C",
    "FileSize": 280067,
    "FileType": "147C",
    "IsAutoGenerated": false,
    "Sent": null,
    "SentDate": null,
    "UploadedBy": "TEST USER",
    "UploadedDate": "2020-02-10T17:46:23.77",
    "UploadedUser": "fc55c121-2bf8-44a8-a8d8-2802d9efd91d",
    "UserIdentity": "1",
    "VR_DocumentId": null,
    "ETLVendorId": 27026,
    "VR_Id": 2821211,
    "ETLGuid": null,
    "DataSource": null,
    "ERPDocumentUniqueID": null,
    "ETLDocumentId": 10851,
    "File_Id": null,
    "IssueDate": null
  },
```


Vendor Registration REST APIs

Available REST Actions:

1. **BatchVendorRegistrationUpdate**
2. **GetVendorRegistrationRecords**
3. **GetVendorTransactions**
4. **APIConnectivityTest**
5. **LogEvent**
6. **VendorRegistrationUpdate**

Available REST Actions Description:

1. **BatchVendorRegistrationUpdate**

This method will post a batch of vendor records to portal. Max allowed number is 100 by default which can be increased if needed.
Method Type: POST

Note: Invite data to be pushed with both VendorId and VR_ID empty or null and ERPLegacyId have populated.

Vendor invitation record - ERPLegacyId, Vendor Country, Company Name, First Name, Last Name, Email Address, Company Code details and Address information as defined in wireframe. Mapping for required fields will be done in Vendor Functional SOA document.

Endpoint URL: <<*client URL*>>/FirstStrike.VendorPortal.Integration/api/VendorRegistration/BatchVendorRegistration Update

Sample Request (the complete sample can be provided upon request from the client):


```
{
   "Credentials": {
      "UserName": "string",
      "Password": "string",
      "APIKey": "string"
   },
   "vendorRegistrationRecords": [{
         "ErrorMessage": "string",
         "ETLVendorId": 0,
         "ETLTransmissionReference": 0,
         "VendorID": "string",
         "AdditionalVAT_Numbers": "string",
         "AgreementAccepted": true,
         "AgreementAcceptedDate": "2020-01-27T20:49:26.975Z",
```


Sample Response (the complete sample can be provided upon request from the client):


```
{
   "IsSuccess": true,
   "ResponseCode": "string",
   "ServiceResponses": [{
         "UniqueIdentifier": 0,
         "VR_Id": 0,
         "IsSuccess": true,
         "ExceptionDetails": "string",
         "ETLVendorId": 0,
         "DataSource": "string",
         "ERPUniqueTransmissionId": "string",
         "Status": "string",
         "Description": "string",
         "VendorId": "string"
```

```
      }
    ],
    "Message": "string"
}
```


2. GetVendorRegistrationRecords

This method returns an array of vendorRegistration records from portal. It will return ETL vendor records that are not yet processed/sent to client. Clients can use this method to fetch latest/recent approved record in portal.

Method Type: POST
Endpoint URL: <<*client URL*>>/FirstStrike.VendorPortal.Integration/api/VendorRegistration/GetVendorRegistrationRecords

Sample Request:


```
{
  "UserName": "string",
  "Password": "string",
  "APIKey": "string"
}
```


Sample Response (the complete sample can be provided upon request from the client):


```
{
  "Status": "string",
  "Message": "string",
  "Error": "string",
  "VendorCount": 0,
  "VendorRecords": [
    {
      "ErrorMessage": "string",
      "ETLVendorId": 0,
      "ETLTransmissionReference":
```


3. GetVendorTransactions

This method provides recon API for vendor data transmission between the portal and the Client ERP. This method will return the import/export records in Portal Staging table for any TransmissionId or DateRange.

Method Type: POST
Endpoint URL: <<*client URL*>>/FirstStrike.VendorPortal.Integration/api/VendorRegistration/GetVendorTransactions

Sample Request (the complete sample can be provided upon request from the client):


```
{
  "Credentials": {
    "UserName": "string",
    "Password": "string",
    "APIKey": "string"
  },
  "VendorTransactionArgs": {
    "ERPUniqueTransmissionId": "string",
    "IsInbound": true,
    "FromDate": "2020-01-27T20:49:27.791Z",
    "ToDate": "2020-01-27T20:49:27.791Z",
    "PageSize": 0,
```

```
    "PageNumber": 0
  }
}
```


Sample Response (the complete sample can be provided upon request from the client):


```
{
  "IsSuccess": true,
  "Message": "string",
  "VendorRecords": [
    {
      "ErrorMessage": "string",
      "ETLVendorId": 0,
      "ETLTransmissionReference": 0,
      "VendorID": "string",
            "AdditionalVAT_Numbers": "string",
```


4. APIConnectivityTest

This method tests the connectivity to Portal Integration API.

Method Type: GET
Endpoint URL: <<*client URL*>>/FirstStrike.VendorPortal.Integration/api/VendorRegistration/APIConnectivityTest

5. LogEvent

The method logs the event to Portal and post any failure log to portal. Clients can use it to log any failure on ERP system to Portal.

Method Type: POST
Endpoint URL: <<*client URL*>>/FirstStrike.VendorPortal.Integration/api/VendorRegistration/LogEvent

Sample Request:


```
{
  "LogEventArgs": {
    "VR_Id": 0,
    "VendorId": "string",
    "Message": "string",
    "ErrorMessage": "string",
    "DetailMesssage": "string"
  },
  "Credentials": {
    "UserName": "string",
    "Password": "string"
  }
}
```


Sample Response:


```
{
  "IsSuccess": true,
  "Message": "string"
}
```


6. VendorRegistrationUpdate

This method updates and posts a single vendor record to portal.

Method Type: POST
Endpoint URL: <<*client URL*>>/FirstStrike.VendorPortal.Integration/api/VendorRegistration/VendorRegistrationUpdate

Sample Request (the complete sample can be provided upon request from the client):


```
{
  "Credentials": {
    "UserName": "string",
    "Password": "string",
    "APIKey": "string"
  },
  "VendorRegistrationRecord": {
    "ErrorMessage": "string",
    "ETLVendorId": 0,
    "ETLTransmissionReference": 0,
    "VendorID": "string",
    "AdditionalVAT_Numbers": "string",
    "AgreementAccepted": true,
```


Sample Response (the complete sample can be provided upon request from the client):


```
{
  "UniqueIdentifier": 0,
  "VR_Id": 0,
  "IsSuccess": true,
  "ExceptionDetails": "string",
  "ETLVendorId": 0,
  "DataSource": "string",
  "ERPUniqueTransmissionId": "string",
  "Status": "string",
  "Description": "string",
  "VendorId": "string"
}
```


Inquiry REST APIs

Swagger: <<Client URl>>/FirstStrike.VendorPortal.Integration/swagger/ui/index#/

Available REST Actions:

1. **ImportInvoicesAndPayments**
2. **GetInvoiceAndPaymentTransactions**
3. **UpdateInvoiceEarlyPayments**

Available REST Actions Description:

1. ImportInvoicesAndPayments

This method can be used to post invoice/Payments records to portal. It can be used to import invoice,payments,inoviceLineItema and/or Xref record to portal. Max allowed of invoice is 100 by default which can be increased if needed.

Method Type: POST
Endpoint URL: <<*client*/api/InvoiceAndPayment/ImportInvoicesAndPayments

Sample Request (the complete sample can be provided upon request from the client):


```
{
  "Account": {
    "Username": "string",
    "Password": "string",
     "APIKey": "string"
  },
  "InvoicesAndPayments": {
    "Invoices": [
      {
        "Invoice_ID": 0,
        "Date_Added": "2019-02-01T15:22:01.925Z",
        "Division_Code": "string",
        "Vendor_ID": "string",
        "Company_Name": "string",
```


Sample Response (the complete sample can be provided upon request from the client):


```
{
  "Invoices": [
    {
      "Invoice_ID": 0,
      "Invoice_Number": "string",
      "Vendor_ID": "string",
      "Location_Code": "string",
      "Fiscal_Year": 0,
      "Voucher": "string",
      "Data_Source": "string",
      "Error": "string",
      "ETLInvoiceId": 0,
      "Invoice_Doc_ID": "string",
      "ERPUniqueTransmissionId": "string",
      "Status": "string",
      "Description": "string"
    }
  ],
```


2. GetInvoiceAndPaymentTransactions

This method provides recon API for Invoice/payment Transaction between client ERP and Portal. It also will return invoice/payment records which are transmitted between the client ERP and Portal for given transmissionId or DateRange.

Sample Request:


```
{
  "Credentials": {
    "UserName": "string",
    "Password": "string",
     "APIKey": "string"

  },
  "InvoiceAndPaymentTransactionArgs": {
    "ERPUniqueTransmissionId": "string",
    "FromDate": "2019-02-01T15:22:02.027Z",
    "ToDate": "2019-02-01T15:22:02.027Z",
    "PageSize": 0,
    "PageNumber": 0,
    "RecordType": 0
  }
}
```


Sample Response (the complete sample can be provided upon request from the client):

{ "IsSuccess": true, "Message": "string", "InvoiceAndPaymentRecords": { "ETLInvoices": [{ "ETLInvoiceId": 0, "ETLDirection": true, "ETLProcessed": true, "ETLProcessedDate": "2019-02-01T15:22:01.973Z", "ETLFileName": "string", "ETLFileLineNumber": 0, "Invoice_ID": 0, "File_Log_ID": 0, "Date_Added": "2019-02-01T15:22:01.973Z", "Division_Code": "string", "Vendor_ID": "string", "Company_Name": "string", "Invoice_Number": "string", "Invoice_Reference": "string", "Voucher": "string", "Purchase_Order": "string", "Invoice_Date": "2019-02-01T15:22:01.973Z",

4. UpdateInvoiceEarlyPayments

This method can be used to patch invoice Earlypayment information to portal. It allows for multiple invoices to be sent in the request.
Method Type: PATCH
Endpoint URL:
'<<client/FirstStrike.VendorPortal.Integration/api/InvoiceAndPayment/UpdateInvoiceEarlyPayments

Sample Request:

```
{
  "Credentials": {
    "UserName": "string",
    "Password": "string",
    "APIKey": "string"
  },
  "InvoicesAndPayments":
[
{
    "Invoice_ID": int,
    "Earlypayment_ID": int,
    "EarlyPaymentNetInvoiceAmount": decimal,
    "EarlyPaymentDiscountAmount": decimal
  }
]
}
```

Sample Response (the complete sample can be provided upon request from the client):

```
{
  "Status": "string",
  "Message": "string",
  "Error": null,
  "FailedInvoices": [],
  "SuccessfulInvoices": [
    int
  ]
}
```

Employee REST APIs

Available REST Actions:

1. ProcessEmployeeRecords

Available REST Actions Description:

1. ProcessEmployeeRecords
This action can be used to post the employee records from client system to portal.
Method Type: POST
Endpoint URL: <<*client URL*>>/FirstStrike.VendorPortal.Integration/api/Employee/ProcessEmployeeRecordsAsync

Sample Request:


```
{
  "Credentials": {
    "UserName": "string",
    "Password": "string",
    "APIKey": "string"

  },
  "EmployeeRecords": [
    {
      "EmployeeId": 0,
      "ClientERPEmployeeId": "string",
      "FirstName": "string",
      "LastName": "string",
      "EmailAddress": "string",
      "ManagerEmployeeId": 0,
      "ClientERPManagerEmployeeId": "string",
      "UserName": "string",
      "StartDate": "2019-02-01T15:22:01.906Z",
      "EndDate": "2019-02-01T15:22:01.906Z",
      "CreatedBy": "string",
      "ModifiedBy": "string",
      "Status": "string",
      "DataSource": "string",
      "Roles": "string",
      "CompanyCodes": "string",
      "Regions": "string",
      "Country": "string"
    }
  ]
}
```


Sample Response:
None

Cash Management REST APIs

Swagger: <<*Client URI*>>/FirstStrike.VendorPortal.Integration/swagger/ui/index#/CashManagement

Available REST Actions:

1. **GetAccelerateCashStatus**
2. **AccelerateCashStatusUpdate**

Available REST Actions Description:

1. GetAccelerateCashStatus
This method fetch all Approved AccelerateCash Record from portal. This method will return an array of AccelerateCashStatus record with details which are recently approved in portal.

Method Type: POST
Endpoint URL: <<*client URL*>>/FirstStrike.VendorPortal.Integration/api/CashManagement/GetAccelerateCashStatus

Sample Request:


```
{
  "Credentials": {
    "UserName": "string",
    "Password": "string",
    "APIKey": "string"
  }
}
```


Sample Response (the complete sample can be provided upon request from the client):


```
{
  "IsSuccess": true,
  "Message": "string",
  "AccelerateCashStatusRecords": [
    {
      "AccelerateCashID": 0,
      "AccelerateCashStatus": "string",
      "CompanyName": "string",
      "VendorId": "string",
      "AccelerateCashPaymentAmount": 0,
      "AccelerateCashPaymentDate": "2019-02-01T15:22:01.834Z",
      "WeightedDiscountPercentage": 0,
      "NetWeightedDiscountPercentage": 0,
      "Count": 0,
      "AccelerateCashDiscountAmount": 0,
```

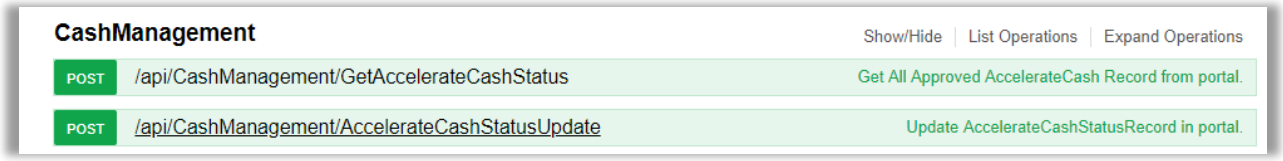

2. AccelerateCashStatusUpdate

This method can be used to update AccelerateCashStatusRecord in portal and to update/active the AccelerateCashStatus. It can also be used to log failure any failure from ERP system to portal for any AccelerateCashStatusID.
Method Type: POST
Endpoint URL: <<*client URL*>>/FirstStrike.VendorPortal.Integration/api/CashManagement/AccelerateCashStatusUpdate
Sample Request:


```
{
  "Credentials": {
    "UserName": "string",
    "Password": "string",
    "APIKey": "string"
  },
  "AccelerateCashStatusUpdateRequestDetails": [
    {
      "IsERPSuccess": true,
      "Message": "string",
      "AccelerateCashStatusID": 0,
      "VendorId": "string"
    }
  ]
}
```


Sample Response:

```
{
 "IsSuccess": true,
 "Message": "string",
 "AccelerateCashStatusUpdateResponseDetails": [
  {
   "IsSuccess": true,
   "Message": "string",
   "AccelerateCashStatusID": 0,
   "VendorId": "string"
  }
 ]
}
```

DropDown REST APIs

Swagger: <<*Client URl*>/FirstStrike.VendorPortal.Integration/swagger/ui/index#/ Dropdown

Available REST Actions:

1. **Get**
2. **Update**
3. **Delete**

Available REST Actions Description:

1. Get
This method fetch all dropdown and its values. if one group id/id's or name(s) passed in the request it will get the records matching.

Method Type: POST
Endpoint URL: <<*client URL*>>/FirstStrike.VendorPortal.Integration/api/DropDown/Get

Sample Request (the complete sample can be provided upon request from the client):


```
{
 "Credentials": {
  "UserName": "string",
  "Password": "string",
  "APIKey": "string"
 },
 "DropDownGroup": [
  {
   "DropDownId": 0,
   "DropDownName": "string",
```


Sample Response (the complete sample can be provided upon request from the client):


```
{
 "Status": "string",
 "IsSuccess": true,
 "Message": "string",
 "DropDowns": [
```

```
  {
    "DropDownId": 0,
    "DropDownName": "string",
    "Description": "string",
    "GroupType": "string",
    "ClientId": 0,
    "DropDownValues": [
     {
       "ID": 0,
       "DropdownGroupId": 0,
       "Value": "string",
       "Name": "string",
```


2. Update

This method can be used to Add/update dropdown/values. If dropdown id is passed it will update the dropdown and its values. If value id is not passed it will add new value. If the existing all values to delete and re-add with new values then "DeleteAndRecreateValues" flag needs to be send as true.

Method Type: POST
Endpoint URL: <<*client URL*>>/FirstStrike.VendorPortal.Integration/api/DropDown/Update

Sample Request (the complete sample can be provided upon request from the client):


```
{
  "Credentials": {
    "UserName": "string",
    "Password": "string",
    "APIKey": "string"
  },
  "DropDownArgs": [
    {
      "DropDownGroup": {
        "DropDownId": 0,
        "DropDownName": "string",
        "Description": "string",
        "GroupType": "string",
        "ClientId": 0,
        "DropDownValues": [
         {
           "ID": 0,
           "DropdownGroupId": 0,
```


Sample Response:


```
{
  "Status": "string",
  "IsSuccess": true,
  "Message": [
    "string"
  ]
}
```


3. Delete

This method can be used to delete dropdown.

Method Type: DELETE

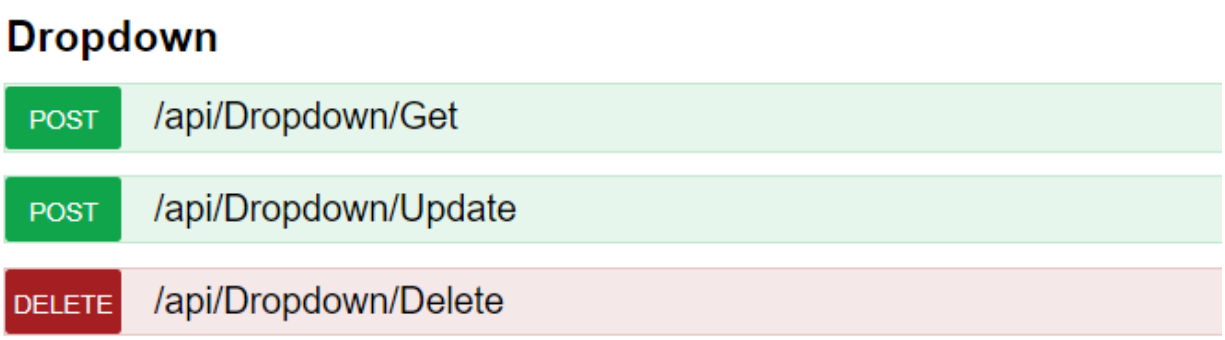
Endpoint URL: <<*client URL*>>/FirstStrike.VendorPortal.Integration/api/DropDown/Delete

Sample Request (the complete sample can be provided upon request from the client):


```
{
  "Credentials": {
    "UserName": "string",
    "Password": "string",
    "APIKey": "string"
  },
  "DropDownGroups": [
    {
      "DropDownId": 0,
      "DropDownName": "string",
      "Description": "string",
      "GroupType": "string",
      "ClientId": 0,
      "DropDownValues": [
        {
          "ID": 0,
          "DropdownGroupId": 0,
          "Value": "string",
```


Sample Response:


```
{
  "Status": "string",
  "IsSuccess": true,
  "Message": [
    "string"
  ]
}
```

Vendor Registration SOAP APIs

WSDL: <<*Client URL*>>
/FirstStrike.VendorPortal.Integration/VendorRegistrationIntegrationService.svc?wsdl

Endpoint:
<<*clientURl*>>/FirstStrike.VendorPortal.Integration/VendorRegistrationIntegrationService.svc

Available SOAP Actions:

1. **BatchVendorRegistrationUpdateV2**
2. **BatchVendorRegistrationUpdate**
3. **GetVendorRegistrationRecord**

4. **GetVendorRegistrationRecords**
5. **GetVendorTransactions**
6. **APIConnectivityTest**
7. **LogEvent**
8. **VendorRegistrationUpdateV2**
9. **VendorRegistrationUpdate**

Available SOAP Actions Description:

1. **BatchVendorRegistrationUpdateV2**
   This SOAP method can be used to post/push the Batch of VendorRegistration records from client system to portal. Multiple vendors can be posted in a single request.

   Sample Request (the complete sample can be provided upon request from the client):

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:tem="http://tempuri.org/"
xmlns:int="http://schemas.datacontract.org/2004/07/Integration.Data_Contracts"
xmlns:fir="http://schemas.datacontract.org/2004/07/FirstStrike.VendorPortal.SharedDataContracts">
  <soapenv:Header/>
  <soapenv:Body>
    <tem:BatchVendorRegistrationUpdateV2>
      <!--Optional:-->
      <tem:credentials>
        <!--Optional:-->
        <int:Password></int:Password>
        <!--Optional:-->
        <int:UserName></int:UserName>
      </tem:credentials>
      <!--Optional:-->
      <tem:vendorRegistrationRecords>
```

   Sample Response (the complete sample can be provided upon request from the client):

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:tem="http://tempuri.org/"
xmlns:int="http://schemas.datacontract.org/2004/07/Integration.Data_Contracts">
  <soapenv:Header/>
  <soapenv:Body>
    <tem:BatchVendorRegistrationUpdateV2Response>
      <!--Optional:-->
      <tem:BatchVendorRegistrationUpdateV2Result>
        <!--Optional:-->
        <int:IsSuccess>?</int:IsSuccess>
        <!--Optional:-->
        <int:Message>?</int:Message>
        <!--Optional:-->
        <int:ResponseCode>?</int:ResponseCode>
```

2. **BatchVendorRegistrationUpdate**
   This SOAP method can be used to post/push the Batch of VendorRegistration records from client system to portal. Multiple vendors can be posted at a single time. This is a legacy SOAP action and not recommended for use by new clients. **BatchVendorRegistrationUpdateV2** should be used instead.

   Sample Request (the complete sample can be provided upon request from the client):

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:tem="http://tempuri.org/"
xmlns:int="http://schemas.datacontract.org/2004/07/Integration.Data_Contracts"
xmlns:fir="http://schemas.datacontract.org/2004/07/FirstStrike.VendorPortal.SharedDataContracts"
xmlns:arr="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
```

```
<soapenv:Header/>
<soapenv:Body>
  <tem:BatchVendorRegistrationUpdate>
    <!--Optional:-->
    <tem:credentials>
      <!--Optional:-->
      <int:Password></int:Password>
      <!--Optional:-->
      <int:UserName></int:UserName>
    </tem:credentials>
    <!--Optional:-->
    <tem:vendorRegistrationRecords>
```

Sample Response (the complete sample can be provided upon request from the client):

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:tem="http://tempuri.org/"
xmlns:int="http://schemas.datacontract.org/2004/07/Integration.Data_Contracts">
  <soapenv:Header/>
  <soapenv:Body>
    <tem:BatchVendorRegistrationUpdateResponse>
      <!--Optional:-->
      <tem:BatchVendorRegistrationUpdateResult>
        <!--Optional:-->
        <int:IsSuccess></int:IsSuccess>
        <!--Optional:-->
        <int:Message></int:Message>
        <!--Optional:-->
        <int:ResponseCode></int:ResponseCode>
        <!--Optional:-->
        <int:ServiceResponses>
```

3. **GetVendorRegistrationRecord**
   This SOAP Action can be used to fetch vendor records for requested args (vrid or vendorid or ETLVendorId). It will return those records which were transmitted earlier to Client ERP. This is helpful when clients need to debug or re-fetch the record on failure on their system. It can return one or more records at time.

Sample Request (the complete sample can be provided upon request from the client):

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:tem="http://tempuri.org/"
xmlns:int="http://schemas.datacontract.org/2004/07/Integration.Data_Contracts">
  <soapenv:Header/>
  <soapenv:Body>
    <tem:GetVendorRegistrationRecord>
      <!--Optional:-->
      <tem:credentials>
        <!--Optional:-->
        <int:Password></int:Password>
        <!--Optional:-->
        <int:UserName></int:UserName>
      </tem:credentials>
      <!--Optional:-->
      <tem:getVendorRecordRequestArgs>
```

Sample Response (the complete sample can be provided upon request from the client):

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:tem="http://tempuri.org/"
xmlns:int="http://schemas.datacontract.org/2004/07/Integration.Data_Contracts"
xmlns:fir="http://schemas.datacontract.org/2004/07/FirstStrike.VendorPortal.SharedDataContracts">
```

```
<soapenv:Header/>
<soapenv:Body>
  <tem:GetVendorRegistrationRecordResponse>
    <!--Optional:-->
    <tem:GetVendorRegistrationRecordResult>
      <!--Optional:-->
      <int:Error></int:Error>
      <!--Optional:-->
      <int:Message></int:Message>
      <!--Optional:-->
      <int:Status></int:Status>
      <!--Optional:-->
      <int:VendorCount></int:VendorCount>
      <!--Optional:-->
      <int:VendorRecords>
```

4. **GetVendorRegistrationRecords**

   This SOAP Action can be used to fetch Vendor records that are ready to be exported from portal staging tables and which have not been sent already to client ERP.

   Sample Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:tem="http://tempuri.org/"
xmlns:int="http://schemas.datacontract.org/2004/07/Integration.Data_Contracts">
  <soapenv:Header/>
  <soapenv:Body>
    <tem:GetVendorRegistrationRecords>
      <!--Optional:-->
      <tem:credentials>
        <!--Optional:-->
        <int:Password></int:Password>
        <!--Optional:-->
        <int:UserName></int:UserName>
      </tem:credentials>
    </tem:GetVendorRegistrationRecords>
  </soapenv:Body>
</soapenv:Envelope>
```

   Sample Response (the complete sample can be provided upon request from the client):

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:tem="http://tempuri.org/"
xmlns:int="http://schemas.datacontract.org/2004/07/Integration.Data_Contracts"
xmlns:fir="http://schemas.datacontract.org/2004/07/FirstStrike.VendorPortal.SharedDataContracts">
  <soapenv:Header/>
  <soapenv:Body>
    <tem:GetVendorRegistrationRecordsResponse>
      <!--Optional:-->
      <tem:GetVendorRegistrationRecordsResult>
        <!--Optional:-->
        <int:Error></int:Error>
        <!--Optional:-->
        <int:Message></int:Message>
        <!--Optional:-->
        <int:Status></int:Status>
        <!--Optional:-->
        <int:VendorCount></int:VendorCount>
        <!--Optional:-->
        <int:VendorRecords>
```

5. **GetVendorTransactions**

   This SOAP action can be used for reconciliation of the vendor data transactions between the client ERP and portal. It will return all the records from the staging table based on the provided requestArgs(uniqueTransmissionId or Date range).

   Sample Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:tem="http://tempuri.org/"
xmlns:int="http://schemas.datacontract.org/2004/07/Integration.Data_Contracts">
  <soapenv:Header/>
  <soapenv:Body>
    <tem:GetVendorTransactions>
      <!--Optional:-->
      <tem:getVendorTransactionRequestArgs>
        <!--Optional:-->
        <int:Credentials>
          <!--Optional:-->
          <int:Password></int:Password>
          <!--Optional:-->
          <int:UserName></int:UserName>
        </int:Credentials>
        <!--Optional:-->
        <int:VendorTransactionArgs>
          <!--Optional:-->
          <int:ERPUniqueTransmissionId></int:ERPUniqueTransmissionId>
          <!--Optional:-->
          <int:FromDate></int:FromDate>
          <!--Optional:-->
          <int:IsInbound></int:IsInbound>
          <!--Optional:-->
          <int:PageNumber></int:PageNumber>
          <!--Optional:-->
          <int:PageSize></int:PageSize>
          <!--Optional:-->
          <int:ToDate></int:ToDate>
        </int:VendorTransactionArgs>
      </tem:getVendorTransactionRequestArgs>
    </tem:GetVendorTransactions>
  </soapenv:Body>
</soapenv:Envelope>
```

Sample Response (the complete sample can be provided upon request from the client):

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:tem="http://tempuri.org/"
xmlns:int="http://schemas.datacontract.org/2004/07/Integration.Data_Contracts"
xmlns:fir="http://schemas.datacontract.org/2004/07/FirstStrike.VendorPortal.SharedDataContracts">
  <soapenv:Header/>
  <soapenv:Body>
    <tem:GetVendorTransactionsResponse>
      <!--Optional:-->
      <tem:GetVendorTransactionsResult>
        <!--Optional:-->
        <int:CurrentCount></int:CurrentCount>
        <!--Optional:-->
        <int:IsSuccess></int:IsSuccess>
        <!--Optional:-->
        <int:Message></int:Message>
        <!--Optional:-->
        <int:PageNumber></int:PageNumber>
        <!--Optional:-->
        <int:PageSize></int:PageSize>
        <!--Optional:-->
        <int:TotalCount></int:TotalCount>
        <!--Optional:-->
```

6. **APIConnectivityTest**

   This SOAP action can be used to test the connectivity between client system and portal. If response is return from portal, then there is no connection issue.

   Sample Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:tem="http://tempuri.org/">
  <soapenv:Header/>
  <soapenv:Body>
    <tem:IntegrationConnectivityTest>
```

```
      <!--Optional:-->
      <tem:InputString></tem:InputString>
    </tem:IntegrationConnectivityTest>
  </soapenv:Body>
</soapenv:Envelope>
```

Sample Response:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:tem="http://tempuri.org/">
  <soapenv:Header/>
  <soapenv:Body>
    <tem:IntegrationConnectivityTestResponse>
      <!--Optional:-->
      <tem:IntegrationConnectivityTestResult></tem:IntegrationConnectivityTestResult>
    </tem:IntegrationConnectivityTestResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

7. LogEvent

This SOAP action can be used to log a failure event to portal if ERP failure occurs during vendor update in client system. Based on this event, portal can be configured to take some action.

Sample Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:tem="http://tempuri.org/"
xmlns:int="http://schemas.datacontract.org/2004/07/Integration.Data_Contracts">
  <soapenv:Header/>
  <soapenv:Body>
    <tem:LogEvent>
      <!--Optional:-->
      <tem:eventArgs>
        <!--Optional:-->
        <int:Credentials>
          <!--Optional:-->
          <int:Password></int:Password>
          <!--Optional:-->
          <int:UserName></int:UserName>
        </int:Credentials>
        <!--Optional:-->
        <int:LogEventArgs>
          <!--Optional:-->
          <int:DetailMesssage></int:DetailMesssage>
          <!--Optional:-->
          <int:ErrorMessage></int:ErrorMessage>
          <!--Optional:-->
          <int:Message></int:Message>
          <!--Optional:-->
          <int:VR_Id></int:VR_Id>
          <!--Optional:-->
          <int:VendorId></int:VendorId>
        </int:LogEventArgs>
      </tem:eventArgs>
    </tem:LogEvent>
  </soapenv:Body>
</soapenv:Envelope>
```

Sample Response:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:tem="http://tempuri.org/"
xmlns:int="http://schemas.datacontract.org/2004/07/Integration.Data_Contracts">
  <soapenv:Header/>
  <soapenv:Body>
    <tem:LogEventResponse>
      <!--Optional:-->
      <tem:LogEventResult>
        <!--Optional:-->
        <int:IsSuccess></int:IsSuccess>
```

```
        <!--Optional:-->
        <int:Message></int:Message>
      </tem:LogEventResult>
    </tem:LogEventResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

8. **VendorRegistrationUpdateV2**

   This SOAP Action can be used to post/push single vendor from client ERP system to portal.

   Sample Request (the complete sample can be provided upon request from the client):

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:tem="http://tempuri.org/"
xmlns:int="http://schemas.datacontract.org/2004/07/Integration.Data_Contracts"
xmlns:fir="http://schemas.datacontract.org/2004/07/FirstStrike.VendorPortal.SharedDataContracts">
  <soapenv:Header/>
  <soapenv:Body>
    <tem:VendorRegistrationUpdateV2>
      <!--Optional:-->
      <tem:credentials>
        <!--Optional:-->
        <int:Password></int:Password>
        <!--Optional:-->
        <int:UserName></int:UserName>
      </tem:credentials>
```

   Sample Response:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:tem="http://tempuri.org/"
xmlns:int="http://schemas.datacontract.org/2004/07/Integration.Data_Contracts">
  <soapenv:Header/>
  <soapenv:Body>
    <tem:VendorRegistrationUpdateV2Response>
      <!--Optional:-->
      <tem:VendorRegistrationUpdateV2Result>
        <!--Optional:-->
        <int:DataSource></int:DataSource>
        <!--Optional:-->
        <int:Description></int:Description>
        <!--Optional:-->
        <int:ERPUniqueTransmissionId></int:ERPUniqueTransmissionId>
        <!--Optional:-->
        <int:ETLVendorId></int:ETLVendorId>
        <!--Optional:-->
        <int:ExceptionDetails></int:ExceptionDetails>
        <!--Optional:-->
        <int:IsSuccess></int:IsSuccess>
        <!--Optional:-->
        <int:Status></int:Status>
        <!--Optional:-->
        <int:UniqueIdentifier></int:UniqueIdentifier>
        <!--Optional:-->
        <int:VR_Id></int:VR_Id>
        <!--Optional:-->
        <int:VendorId></int:VendorId>
      </tem:VendorRegistrationUpdateV2Result>
    </tem:VendorRegistrationUpdateV2Response>
  </soapenv:Body>
</soapenv:Envelope>
```

9. **VendorRegistrationUpdate**

   This SOAP Action can be used to post/push single vendor from the client ERP system to portal. This is a legacy SOAP action and not recommended for new clients.
   VendorRegistrationUpdateV2

should be used instead.

Sample Request (the complete sample can be provided upon request from the client):

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:tem="http://tempuri.org/"
xmlns:int="http://schemas.datacontract.org/2004/07/Integration.Data_Contracts"
xmlns:fir="http://schemas.datacontract.org/2004/07/FirstStrike.VendorPortal.SharedDataContracts"
xmlns:arr="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
  <soapenv:Header/>
  <soapenv:Body>
    <tem:VendorRegistrationUpdate>
      <!--Optional:-->
      <tem:credentials>
        <!--Optional:-->
        <int:Password></int:Password>
        <!--Optional:-->
        <int:UserName></int:UserName>
      </tem:credentials>
```

Sample Response:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:tem="http://tempuri.org/"
xmlns:int="http://schemas.datacontract.org/2004/07/Integration.Data_Contracts">
  <soapenv:Header/>
  <soapenv:Body>
    <tem:VendorRegistrationUpdateResponse>
      <!--Optional:-->
      <tem:VendorRegistrationUpdateResult>
        <!--Optional:-->
        <int:DataSource></int:DataSource>
        <!--Optional:-->
        <int:Description></int:Description>
        <!--Optional:-->
        <int:ERPUniqueTransmissionId></int:ERPUniqueTransmissionId>
        <!--Optional:-->
        <int:ETLVendorId></int:ETLVendorId>
        <!--Optional:-->
        <int:ExceptionDetails></int:ExceptionDetails>
        <!--Optional:-->
        <int:IsSuccess></int:IsSuccess>
        <!--Optional:-->
        <int:Status></int:Status>
        <!--Optional:-->
        <int:UniqueIdentifier></int:UniqueIdentifier>
        <!--Optional:-->
        <int:VR_Id></int:VR_Id>
        <!--Optional:-->
        <int:VendorId></int:VendorId>
      </tem:VendorRegistrationUpdateResult>
    </tem:VendorRegistrationUpdateResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

Inquiry SOAP APIs

WSDL: <<*Client URL*>>/FirstStrike.VendorPortal.Integration/InvoicePaymentService.svc?wsdl

Endpoint: <<*clientURL*>>/FirstStrike.VendorPortal.Integration/InvoicePaymentService.svc

Available SOAP Actions:

1. ImportInvoicesAndPayments

2. GetInvoiceAndPaymentTransactions

Available SOAP Action Description:

1. **ImportInvoicesAndPayments**
   This SOAP method can be used to post/push the batch of invoices/payements/invoicelineItems and invoicePaymentxrefs records to portal. The client can push each type (invoice, payment, lineItems and xref) separately or in a single batch. Max allowed invoices in a batch is 100 by default which can be increased if needed.

Sample Request (the complete sample can be provided upon request from the client):

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:tem="http://tempuri.org/"
xmlns:int="http://schemas.datacontract.org/2004/07/Integration.Shared.Dtos">
  <soapenv:Header/>
  <soapenv:Body>
    <tem:ImportInvoicesAndPayments>
      <!--Optional:-->
      <tem:account>
        <!--Optional:-->
        <int:Password></int:Password>
        <!--Optional:-->
        <int:Username></int:Username>
      </tem:account>
      <!--Optional:-->
<tem:invoicesAndPayments>
```

2. **GetInvoiceAndPaymentTransactions**
   This SOAP action can be used for reconciliation of the inquiry data transaction between the client ERP and portal. It will return all the records from staging table based on the provided requestArgs(uniqueTransmissionId or Date range and record type).

Sample Request (the complete sample can be provided upon request from the client):

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:tem="http://tempuri.org/"
xmlns:int="http://schemas.datacontract.org/2004/07/Integration.Data_Contracts">
  <soapenv:Header/>
  <soapenv:Body>
    <tem:GetInvoiceAndPaymentTransactions>
      <!--Optional:-->
      <tem:getInvoiceAndPaymentTransactionRequestArgs>
        <!--Optional:-->
        <int:Credentials>
          <!--Optional:-->
          <int:Password></int:Password>
          <!--Optional:-->
<int:UserName></int:UserName>
```

Sample Response (the complete sample can be provided upon request from the client):

```
 <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:tem="http://tempuri.org/"
xmlns:int="http://schemas.datacontract.org/2004/07/Integration.Data_Contracts"
xmlns:fir="http://schemas.datacontract.org/2004/07/FirstStrike.VendorPortal.SharedDataContracts"
xmlns:fir1="http://schemas.datacontract.org/2004/07/FirstStrike.VendorPortal.SharedDataContracts.ETL">
  <soapenv:Header/>
  <soapenv:Body>
    <tem:GetInvoiceAndPaymentTransactionsResponse>
      <!--Optional:-->
      <tem:GetInvoiceAndPaymentTransactionsResult>
        <!--Optional:-->
```

```
<int:CurrentCount></int:CurrentCount>
  <!--Optional:-->
<int:InvoiceAndPaymentRecords>
```

Employee SOAP APIs

WSDL: <<*Client URL*>>/FirstStrike.VendorPortal.Integration/EmployeeIntegrationService.svc?wsdl

Endpoint: <<*clientURL*>>/FirstStrike.VendorPortal.Integration/ EmployeeIntegrationService.svc

Available SOAP Action:

1. **ProcessEmployeeRecords**

Available SOAP Action Description:

1. ProcessEmployeeRecords:
This SOAP action can be used to post/push the batch of Employee Records from the client system to portal. Multiple employees can be posted in a single request.

Sample Request (the complete sample can be provided upon request from the client):

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:tem="http://tempuri.org/"
xmlns:int="http://schemas.datacontract.org/2004/07/Integration.Data_Contracts"
xmlns:fir="http://schemas.datacontract.org/2004/07/FirstStrike.VendorPortal.SharedDataContracts">
  <soapenv:Header/>
  <soapenv:Body>
    <tem:ProcessEmployeeRecords>
      <!--Optional:-->
      <tem:credentials>
```

Sample Response:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:tem="http://tempuri.org/">
  <soapenv:Header/>
  <soapenv:Body>
    <tem:ProcessEmployeeRecordsResponse>
      <!--Optional:-->
      <tem:ProcessEmployeeRecordsResult></tem:ProcessEmployeeRecordsResult>
    </tem:ProcessEmployeeRecordsResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

Sample Response (the complete sample can be provided upon request from the client):

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:tem="http://tempuri.org/"
xmlns:int="http://schemas.datacontract.org/2004/07/Integration.Shared.Dtos">
  <soapenv:Header/>
  <soapenv:Body>
    <tem:ImportInvoicesAndPaymentsResponse>
      <!--Optional:-->
      <tem:ImportInvoicesAndPaymentsResult>
        <!--Optional:-->
        <int:InvoiceLineItem>
          <!--Zero or more repetitions:-->
          <int:InvoiceLineItemResponseDto>
            <!--Optional:-->
<int:Description></int:Description>
```

Cash Management SOAP APIs

WSDL: <<*Client URL*>>
/FirstStrike.VendorPortal.Integration/CashManagementIntegrationService.svc?wsdl

Endpoint:
<<*clientURL*>>/FirstStrike.VendorPortal.Integration/CashManagementIntegrationService.svc

Available SOAP Actions:

1. **GetAccelerateCashStatus**
2. **AccelerateCashStatusUpdate**

Available SOAP Actions Description:

1. **GetAccelerateCashStatus**

This SOAP action can be used to fetch the acceleratecashstatus records from portal which are approved in status. This action will return an array of acceleratecashstatus records. Each record will consist of details and related invoice information.

Sample Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:tem="http://tempuri.org/"
xmlns:int="http://schemas.datacontract.org/2004/07/Integration.Data_Contracts">
  <soapenv:Header/>
  <soapenv:Body>
    <tem:GetAccelerateCashStatus>
      <!--Optional:-->
      <tem:getAccelerateCashStatusRequestArgs>
        <!--Optional:-->
        <int:Credentials>
          <!--Optional:-->
          <int:Password></int:Password>
          <!--Optional:-->
          <int:UserName></int:UserName>
        </int:Credentials>
      </tem:getAccelerateCashStatusRequestArgs>
    </tem:GetAccelerateCashStatus>
  </soapenv:Body>
</soapenv:Envelope>
```

Sample Response (the complete sample can be provided upon request from the client):

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:tem="http://tempuri.org/"
xmlns:int="http://schemas.datacontract.org/2004/07/Integration.Data_Contracts"
xmlns:fir="http://schemas.datacontract.org/2004/07/FirstStrike.VendorPortal.SharedDataContracts.CashManagement"
xmlns:fir1="http://schemas.datacontract.org/2004/07/FirstStrike.VendorPortal.SharedDataContracts.ETL">
  <soapenv:Header/>
  <soapenv:Body>
    <tem:GetAccelerateCashStatusResponse>
      <!--Optional:-->
      <tem:GetAccelerateCashStatusResult>
        <!--Optional:-->
        <int:AccelerateCashStatusRecords>
          <!--Zero or more repetitions:-->
          <fir:AccelerateCashStatusRecord>
```

2. **AccelerateCashStatusUpdate**

This SOAP action can be used to acknowledge/confirm that the accelerateCashstatus records were received in the client's ERP system. With this action, client can log

whether the received accelerate cash records were successfully processed or failed in their system. Portal can be configured to take some action based on this log.

Sample Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:tem="http://tempuri.org/"
xmlns:int="http://schemas.datacontract.org/2004/07/Integration.Data_Contracts"
xmlns:fir="http://schemas.datacontract.org/2004/07/FirstStrike.VendorPortal.SharedDataContracts.CashManagement">
  <soapenv:Header/>
  <soapenv:Body>
    <tem:AccelerateCashStatusUpdate>
      <!--Optional:-->
      <tem:accelerateCashStatusUpdateRequestArgs>
        <!--Optional:-->
        <int:AccelerateCashStatusUpdateRequestDetails>
          <!--Zero or more repetitions:-->
          <fir:AccelerateCashStatusUpdateRequestDetails>
            <!--Optional:-->
            <fir:AccelerateCashStatusID></fir:AccelerateCashStatusID>
            <!--Optional:-->
            <fir:IsERPSuccess></fir:IsERPSuccess>
            <!--Optional:-->
            <fir:Message></fir:Message>
            <!--Optional:-->
            <fir:VendorId></fir:VendorId>
          </fir:AccelerateCashStatusUpdateRequestDetails>
        </int:AccelerateCashStatusUpdateRequestDetails>
        <!--Optional:-->
        <int:Credentials>
          <!--Optional:-->
          <int:Password></int:Password>
          <!--Optional:-->
          <int:UserName></int:UserName>
        </int:Credentials>
      </tem:accelerateCashStatusUpdateRequestArgs>
    </tem:AccelerateCashStatusUpdate>
  </soapenv:Body>
</soapenv:Envelope>
```

Sample Response:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:tem="http://tempuri.org/"
xmlns:int="http://schemas.datacontract.org/2004/07/Integration.Data_Contracts"
xmlns:fir="http://schemas.datacontract.org/2004/07/FirstStrike.VendorPortal.SharedDataContracts.CashManagement">
  <soapenv:Header/>
  <soapenv:Body>
    <tem:AccelerateCashStatusUpdateResponse>
      <!--Optional:-->
      <tem:AccelerateCashStatusUpdateResult>
        <!--Optional:-->
        <int:AccelerateCashStatusUpdateResponseDetails>
          <!--Zero or more repetitions:-->
          <fir:AccelerateCashStatusUpdateResponseDetails>
            <!--Optional:-->
            <fir:AccelerateCashStatusID></fir:AccelerateCashStatusID>
            <!--Optional:-->
            <fir:IsSuccess></fir:IsSuccess>
            <!--Optional:-->
            <fir:Message></fir:Message>
            <!--Optional:-->
            <fir:VendorId></fir:VendorId>
          </fir:AccelerateCashStatusUpdateResponseDetails>
        </int:AccelerateCashStatusUpdateResponseDetails>
        <!--Optional:-->
        <int:IsSuccess></int:IsSuccess>
        <!--Optional:-->
        <int:Message></int:Message>
      </tem:AccelerateCashStatusUpdateResult>
    </tem:AccelerateCashStatusUpdateResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

iv. Reference Materials

Reference Type	Description	Link

Revision History

Revision Date	Revision Description	Revised by	Approved by

Contact Information

For customer inquiries:
APEX Analytix
1501 Highwoods Boulevard
Suite 200
Greensboro, NC 27410

For technical support inquiries:
call: (877) 506-2739