

AIIData[®]

Web Services API Specifications

April 2021 Version 4.1



© 2019-2021 Fiserv, Inc. or its affiliates. All rights reserved. This work is confidential, and its use is strictly limited. Use is permitted only in accordance with the terms of the agreement under which it was furnished. Any other use, duplication, or dissemination without the prior written consent of Fiserv, Inc. or its affiliates is strictly prohibited. The information contained herein is subject to change without notice. Except as specified by the agreement under which the materials are furnished, Fiserv, Inc. and its affiliates do not accept any liabilities with respect to the information contained herein and are not responsible for any direct, indirect, special, consequential or exemplary damages resulting from the use of this information. No warranties, either express or implied, are granted or extended by this document.

<http://www.fiserv.com>

Fiserv is a registered trademark of Fiserv, Inc.

Other brands and their products are trademarks or registered trademarks of their respective holders and should be noted as such.

Contents

Overview.....	6
Scope and Target Audience	6
AllData Account Aggregation	7
Data Harvesting Techniques	7
Web service connectivity.....	7
File download	7
User credential-based harvesting	7
High-Level Architecture	7
AllData Service Integration.....	8
Full API WS Integration	8
UI Widgets and API Integration	8
UI Portal Integration (PFM).....	8
Full API Web Service Integration	9
User Management	9
Initial Financial Institution List Setup	10
Periodic Maintenance Refresh	10
External FI Seed Data	11
External FI Messages (Special Instructions)	11
Add External Account (Held Away Accounts)	11
Add Account workflow (without multi-factor authentication)	11
Add Account workflow (with multi-factor authentication)	12
Add Account workflow (OAuth)	14
Add Account workflow (PCI compliant FI).....	17
Delete Account workflow	18
Account Management.....	18
Harvest Account Data.....	18
Auto Update	18
On-Demand Update	19
Retrieve Account Data.....	19
Banking transactions.....	19
Credit Card transactions	20
Investment transactions	20
Other Account transactions.....	21
UI Widgets and API Integration.....	22

User Registration and Sign-On.....	22
UI Portal (AllData PFM) Integration	23
User Management Web Services	24
createUser	24
deleteUser	24
getUserProfile	24
updateUserProfile	25
updateUserPassword	25
signon	25
External FI Seed Data Web services	26
getFinancialInstInfo	26
searchFinancialInst.....	26
External FI Messages Web service	27
getFIMessageInfo	27
Account Management Web Services	28
initiateAddAccounts	28
getAddAccountStatus	28
getNewAccounts.....	29
createAccounts	29
deleteAccounts	29
updateAccountCredentials	29
maintainAccount	30
addOfflineAccount	30
updateOfflineAccount	30
addUpdateHostAccounts.....	31
HostAccountsStatus	31
Account Harvesting APIs	32
updateAccounts	32
getHarvestStatus	32
getAccountUpdateSummary.....	32
Account Data Pull APIs	33
getAccountsSummary	33
getAccountDetails.....	33
getInvestmentTrans	33
getCreditCardTrans	33

getBankingTrans.....	34
getOtherAccountTrans	34
getTransactions	34
getDeletedTrans	34
getInvestmentPos	35
getEmployerStockOptions	35
Transaction Categorization APIs	36
getTxnCategoriesInfo	36
categorizeTransaction	36
deleteSubCategory	36
Budgets	37
createBudget	37
deleteBudget.....	37
editBudget.....	37
findBudget.....	37
Goals.....	38
createGoal	38
editGoal	38
deleteGoal	38
findGoal	38
Bank Statement Download Services	39
bankStatementConsent	39
getBankStmtDetails	39
Miscellaneous	40
GetWMAccessKey.....	40
Web Services Setup and Usage	41
Web Services Setup	41
Concurrent Access	41
URLs	41
Appendix A: API Error Codes	42
Appendix B: Harvester Error Codes	48
Appendix C: MFA Image-Retrieving URL	53
Appendix D: Zillow (Zestimate) – Terms of Use.....	54

Overview

This document describes the web services API usage of the Fiserv AllData application. A RESTful API web service handles communication between the partner application and Fiserv, using HTTP protocol over SSL/TLS as the communication medium.

To extract data from Fiserv, the partner application sends a request with certain partner-identifying parameters and user authentication tokens. Upon receipt, the Fiserv server performs the requested task and responds with the results.

The AllData application supports two data formats: XML and JSON. Please contact your AllData sales representative for a link to the updated JSON schema/sandbox testing environment for the web services.

The AllData system provides web service access to:

- add financial institution (“FI”) seed data
- add, delete, and manage users
- add, delete, and manage user accounts with the FIs
- harvest user account information daily and on demand
- retrieve various kinds of transactions to provide accurate and up-to-date information about user assets and holdings

Scope and Target Audience

The purpose of this document is to provide information to partner architects, developers, and project/product managers to enable them make design decisions to connect to AllData web services.

The scope of this document is limited to AllData web services. Aspects of AllData services that include APIs are mentioned in the context of web services integration. Detailed information on other AllData PFM (personal finance manager) features such as UI portal, widgets, and batch file integrations for host account management are available in separate documents.

AlIData Account Aggregation

Account aggregation lets users view all their online financial accounts such as checking, savings, investment, retirement, insurance, and credit cards from a range of FIs on a single web page.

Data Harvesting Techniques

Aggregation uses a combination of techniques to gather the various account data:

- **Web service connectivity:** A direct connection to the FI using a web service
- **File download:** AlIData downloads files from FI websites for account data.
- **User credential-based harvesting:** AlIData harvests FI websites for client account balances and transactions.

Web service connectivity

Data collection uses a direct connection to the FI through web services. Due to the overhead in providing a dedicated service for aggregators, many small to mid-size institutions do not provide this service.

File download

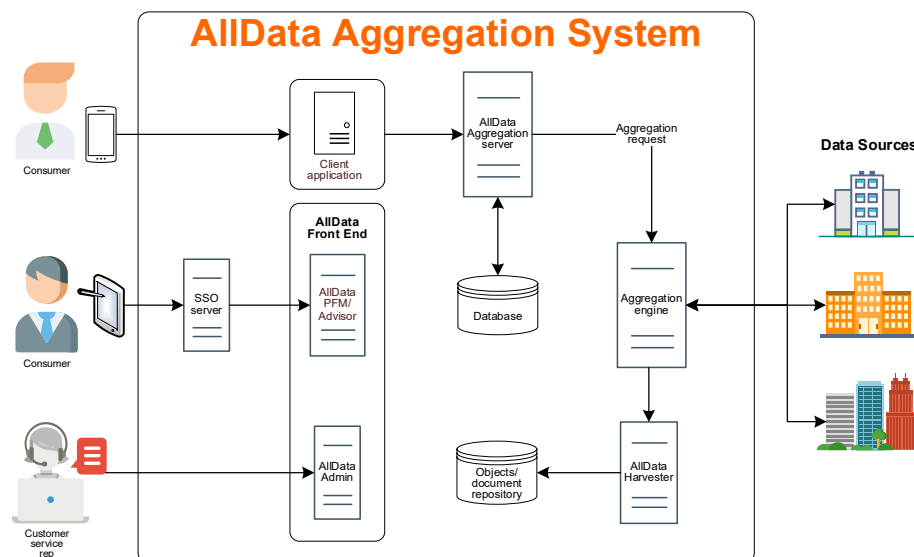
FIs make their user data available in downloadable QIF, OFX, CSV, and TXT files upon user authentication at login. AlIData uses this information to fetch the user account and transaction data.

User credential-based harvesting

This technique involves collecting customer account data from FI web pages with a sophisticated web crawling engine that uses customer login credentials. This approach has high success rates and can handle multi-factor authentication (MFA) scenarios using the AlIData API. A dedicated connectivity engineering team supports the underlying infrastructure to ensure data availability and validity.

High-Level Architecture

This diagram shows the high-level architecture and data flow between a partner, AlIData, and institutions.



AllData Service Integration

This section describes the different partner integration options with the AllData system.

Full API WS Integration

In this model, RESTful web services APIs handle the complete flow from user registrations to data pull and account management. Partners develop custom user interfaces for account setup and management using the web services. Partners can invoke the web services using XML or JSON requests. Partners can also sign up for optional nightly batch files that allow AllData to push data.

UI Widgets and API Integration

In this model, the AllData service is integrated through individual widgets embedded or overlaid in the partner web/native/mobile application. A widget is a web resource composed of web pages and accessed as any HTTP URL. Users authenticate with AllData to establish valid SSO sessions before accessing widgets. Partners can configure the widgets' functionality and look and feel using CSS.

UI Portal Integration (PFM)

In this model, the AllData PFM UI is used for account administration, and partners use data pull daily to retrieve the latest account information available from the Fiserv platform for all users.

The following table lists the different integration types and the API sets they typically use.

API set	AllData API integration	UI portal integration	UI widget / API integration
User management	✓	✓	✓
External FI seed data	✓		
Host account management	✓	✓	✓
Account management	✓		
Account harvesting	✓		
Account data pull API	✓		✓
Advisor management	✓		
Client management	✓		
Home setup	✓		
Transaction categorization	✓		

Full API Web Service Integration

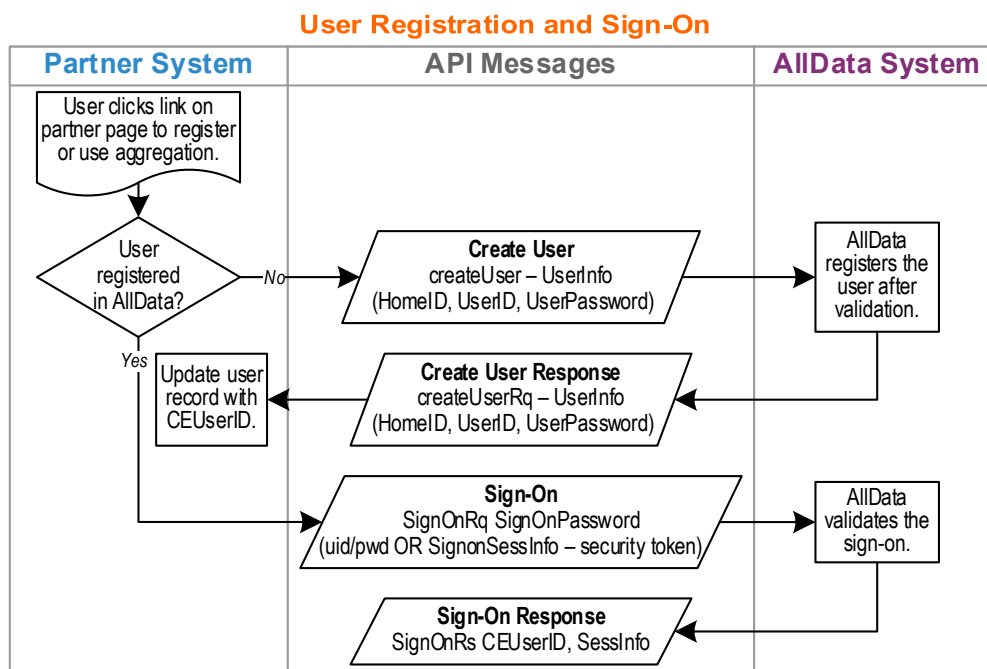
In this model, partners do not use the AllData UI for account setup and management. Instead, partners develop custom user interfaces that communicate with AllData using web service APIs. The following is a list of the different categories of APIs available in the AllData system, each with a brief description.

- **User management:** APIs that support activities to manage users
- **External FI seed data:** APIs for partners to fetch information about FI data sources
- **Account management:** APIs to support user account management activities like user creation
- **Account harvesting:** APIs to allow periodic and on-demand retrieval of user account details
- **Account data management:** APIs to allow clients to retrieve user account data from AllData
- **Client management:** APIs to manage activities related to client-advisor relationships
- **Transaction categorization:** APIs to support transaction categorization
- **Advisor management:** APIs to allow working with wealth management advisors (applies only to AllData Advisor platform)
- **Miscellaneous:** Other APIs, such as for PCI compliance

User Management

AllData user management APIs support various activities to manage users. Partners use the Add User API to add one or more users to the AllData platform. The API expects typical user profile information such as name and address. Once AllData adds a user to its database, it returns a unique CEUserID that must persist in the partner database. The ID accompanies any user actions with AllData.

The following diagram details the typical add user process.



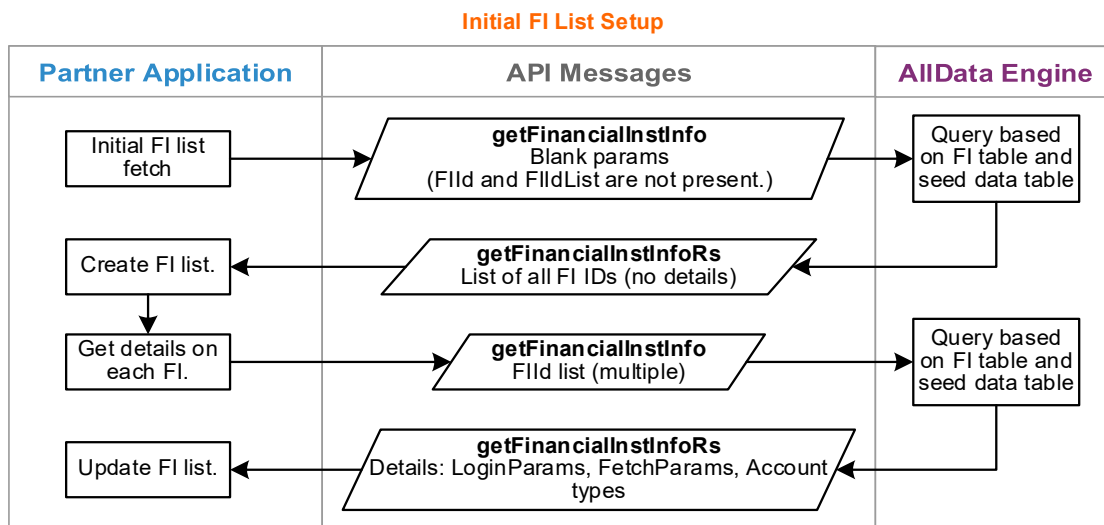
The User Management module uses the following web services:

- createUser
- deleteUser
- getUserProfile
- updateUserProfile
- updateUserPassword

Initial Financial Institution List Setup

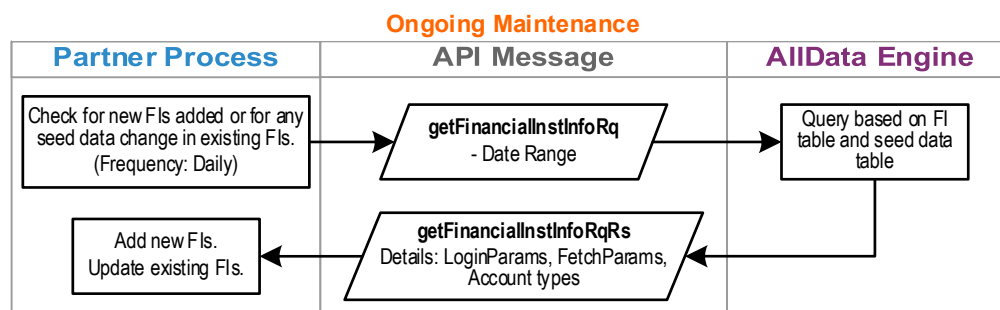
As part of the initial integration with AllData, partners must fetch a list of institutions Fiserv supports and store it in their systems using the web services. After this initial load, subsequent synchronizations fetch only newly added and updated institution data.

Note: The AllData FI list has over 12,500 entries. Given the size of the list, we recommend that partners batch getFIDetails calls, with a limit of 500 FIs returned per call. This process should be executed server to server, taking into consideration that the fetch times may be longer and should not abort with a short timeout.



Periodic Maintenance Refresh

Partners should check for new and updated institution data daily by passing the current time and previous refresh time to obtain the delta. The following diagram represents this periodic data maintenance process.



External FI Seed Data

The following APIs are involved in external FI seed data management:

- `getFinancialInstInfo`
- `searchFinancialInst`

External FI Messages (Special Instructions)

After fetching the supported institutions, partners must retrieve and store “FI Messages” (special instructions) associated with some institutions. Special instructions include actions users must perform before adding accounts. For example, some institution websites require users to provide consent to allow third-party aggregators to harvest account information; special instructions from such an institution would contain the user action to provide consent on its website.

After the initial load, partners perform periodic synchronizations that fetch only new and updated FI Message data.

The API used to fetch FI Messages is `getFIMessageInfo`.

Add External Account (Held Away Accounts)

AllData account management APIs support activities required to aggregate a user’s externally held (“held away”) accounts. Using the FI seed data and user login credentials, Fiserv fetches the user accounts at a given FI. The user chooses the specific accounts to aggregate and monitor, then Fiserv persists and updates the account information on a nightly basis. Partners can also refresh these accounts on demand.

Add Account workflow (without multi-factor authentication)

Partners use the following steps to add new accounts without multi-factor authentication (MFA). (See the [next section](#) for the same process with MFA.) Partners should cache the FI seed data daily before calling these web services.

1. `initiateAddAccounts`

This step initiates the process of adding external accounts. This request triggers a harvesting request using `HarvestAddRq` with FI login parameters (username/password) or `FILoginAcctId`. AllData connects to the FI website and harvests account information. Refer to [Account Management Web Services](#) for the details of the API.

2. `getAddAccountStatus`

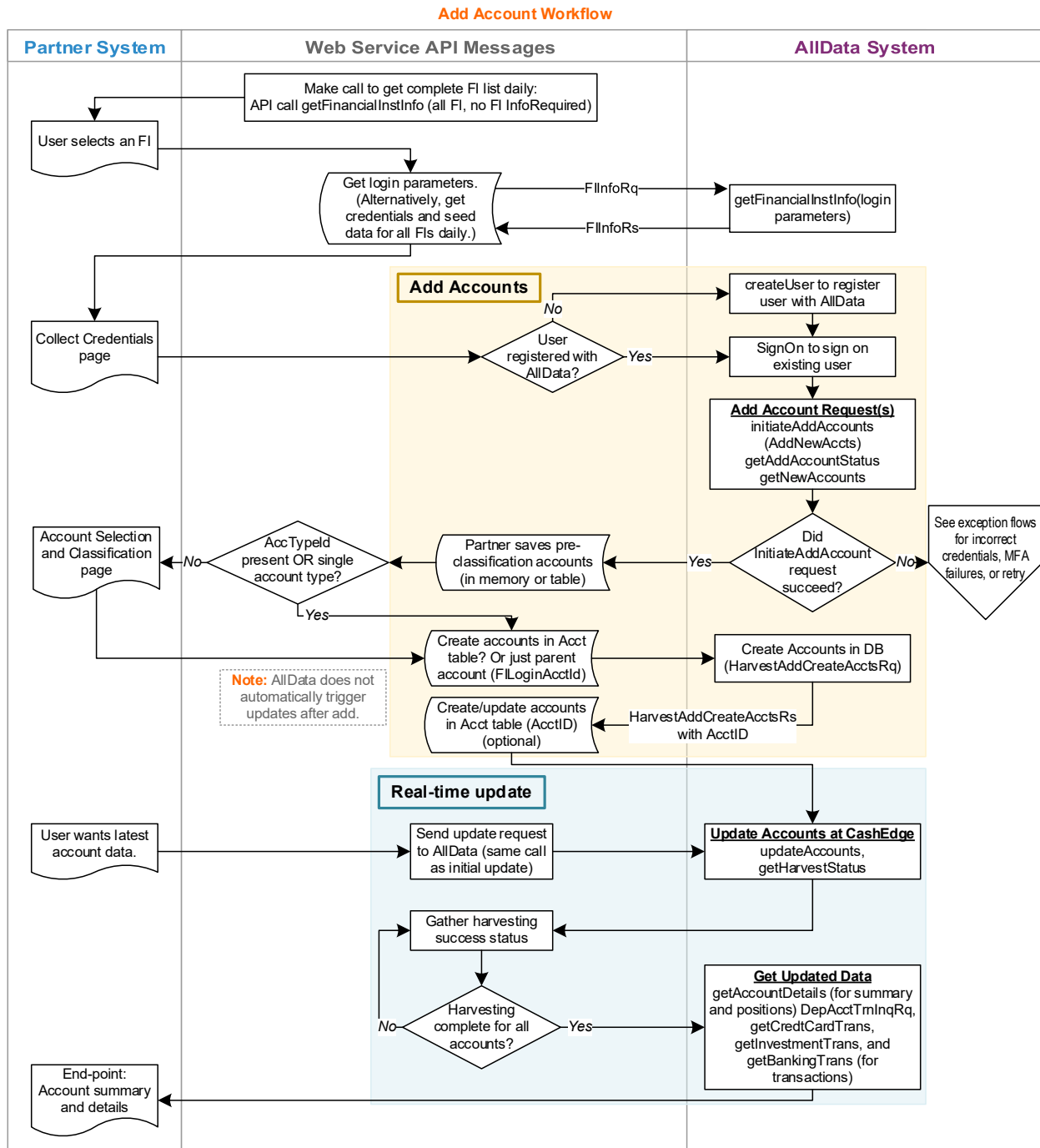
Invoke this API after the `initiateAddAccounts` request has started. It returns the status of `InitiateAddAccounts`. Invoke this API at a periodic interval to check the status. Refer to [Account Management Web Services](#) for the details of the API.

3. `getNewAccounts`

Invoke this API after the `initiateAddAccounts` request has completed. It returns the list of financial accounts found on the FI website. Currently these accounts are not created in the AllData system. Refer to [Account Management Web Services](#) for the details of the API.

4. `createAccounts`

Invoke this API to create the accounts in AllData system. Fiserv classifies the harvested accounts based on certain account classification rules. The user can either override this classification or provide the classification if not already set. After successfully adding accounts the partner must call update account APIs to run on-demand harvesting to retrieve account data. Refer to [Account Management Web Services](#) for API details. The following figure depicts the typical non-MFA Add Account workflow.



Add Account workflow (with multi-factor authentication)

Alldata system supports multi-factor authentication (MFA) as part of the Add Account workflow which some FIs may require. Alldata supports these with special handling of such accounts. When the first-time login credentials fail with MFA error (303 error code), Alldata APIs fetch the MFA details (question or image) and present it to the partner. The partner provides the answer to the question on behalf of the user and the response is saved in the Alldata system to access the user accounts in the future.

Below are the possible MFA scenarios. See the Alldata MFA Sample Responses document for more detailed information.

- One text question, one text answer
- One text question, multiple text answers
- One image question, one text answer
- One image question, multiple image answers
- One text question, multiple image answers

The following are the changes the partner will implement for image-based MFA.

1. Alldata will send the image ID in the FIMFAQuestions aggregate as part of the HarvestAddStsInqRs or HarvestStsInqRs.

```
<FIMFAQuestions>
  <FIMFAQuestion parameterName="ControlName">
    <QuestionPrompt type="image">image id</QuestionPrompt>
  </FIMFAQuestion>
</FIMFAQuestions>
```

2. The partner calls a URL with the image ID generated in the previous step. The base64-encoded image is transmitted to the client browser using the HTTPS connection. See [Appendix C: MFA Image-Retrieving URL](#) for the URL details. After decoding the payload, handle downloaded images in the Portable Network Graphics (.png) format.

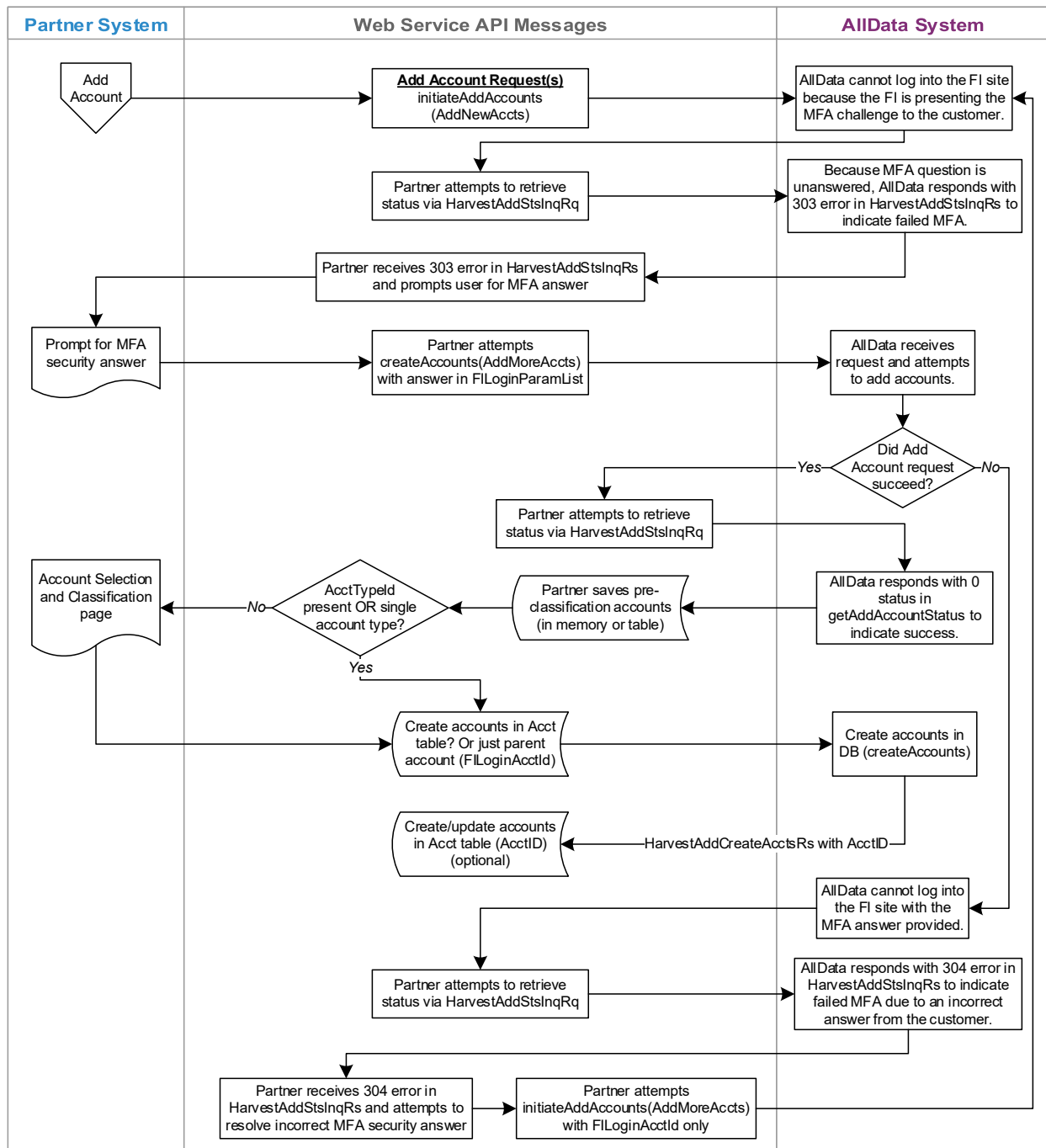
Note: Alldata provides an additional option to the partner to get the base64-encoded image in the HarvestAddStsInqRs or HarvestStsInqRs in lieu of image ID. Partners must contact the Alldata team to turn on the property to send the actual image as part of the API request.

Alldata sends the base64-encoded image in the FIMFAQuestions aggregate as part of the HarvestAddStsInqRs or HarvestStsInqRs.

```
<FIMFAQuestions>
  <FIMFAQuestion parameterName="ControlName">
    <QuestionPrompt type="image">Base64-encoded image</QuestionPrompt>
  </FIMFAQuestion>
</FIMFAQuestions>
```

When an attempt to add or update account is already initiated, and while checking the status of the request, the multi-factor authentication flow described in the following diagram occurs.

Add Account Workflow — Multi-Factor Authentication



Add Account workflow (OAuth)

FIs are implementing OAuth (Open Authorization) protocol to secure their users' sensitive information and to prevent their users' credentials from falling into third-party hands. AllData supports OAuth authorization in the Add Account workflow.

Unlike the other Add Account workflows, partners perform the following steps for OAuth-enabled FIs instead of gathering users' FI credentials.

1. FIInfoRqRs

The FIInfo is enhanced with a new element isOAuthFI to help partners determine whether the FI is OAuth-enabled. A value of "True" in this isOAuthFI element indicates the FI is OAuth-enabled.

```
<FIInfo>
  <FIId>505054</FIId>
  <FIName>Fiserv - Oauth Implementation</FIName>
  <Country>US</Country>
  <FIHarvestStatus>Enabled</FIHarvestStatus>
  <FICrawlable>Enabled</FICrawlable>
  <isOAuthFI>True</isOAuthFI>
  <URL>
    <ParamName>FILogin</ParamName>
    <ParamVal>https://testbank.wm.cashedge.com</ParamVal>
  </URL>
</FIInfo>
```

2. initiateAddAccounts

As in the standard approach to add new accounts, partners must invoke the initiateAddAccounts API to add accounts under OAuth-enabled institutions. The new parameter PartnerAppID is included in this API request and is a mandatory element when invoking the API to add accounts under OAuth-enabled institutions. Partners must send the PartnerAppID of the application from which the attempt is initiated and that PartnerAppID must be a registered one with the FI. Fiserv performs primary validation to confirm that the PartnerAppID is registered and returns an "Invalid Partner Application ID" response code if not registered (error code 4333). Partners must contact Fiserv to register any new application with the FI.

On invoking the initiateAddAccounts API request with the PartnerAppID, Fiserv sends back the response with the new aggregate OAuthInfo for the OAuth-enabled institutions.

```
<OAuthInfo>
  <OAuthRqID>11002</OAuthRqID>
  <OAuthUrl>https://testbank.wm.cashedge.com/OauthFI?type=code</OAuthUrl>
  <OAuthMethod>GET</OAuthMethod>
</OAuthInfo>
```

3. getAddAccountStatus

The partner shares the OAuthUrl to the user, who visits the institution-hosted site to provide login credentials and consent for the accounts to aggregate. Then the partner invokes the getAddAccountStatus API with the OAuthRqID that Fiserv sent in the initiateAddAccounts API response (instead of the RunID used in other Add Account workflows).

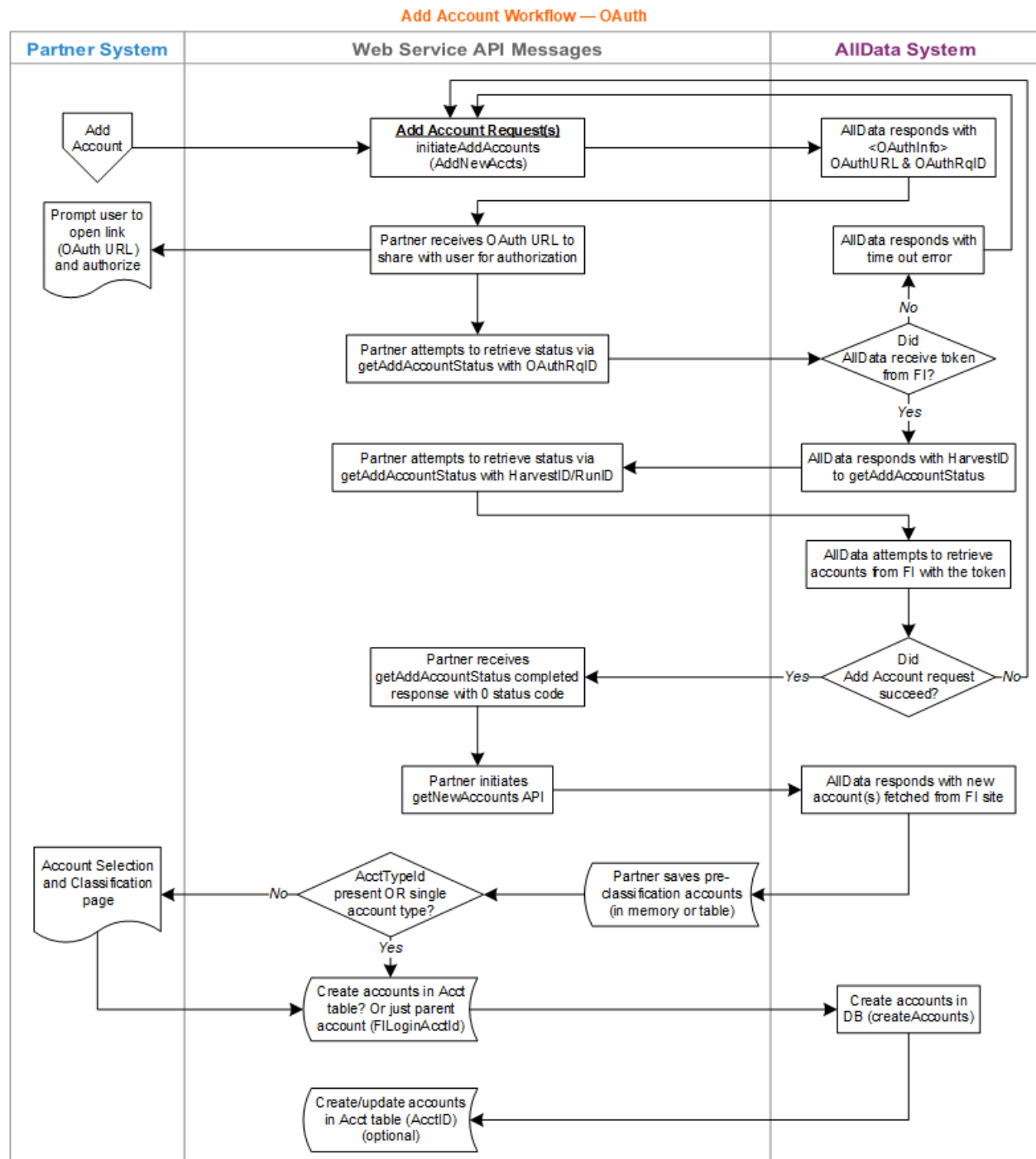
```
<HarvestAddStsInqRq>
  <OAuthRqID>11002</OAuthRqID>
  <HarvestAddID>3259574c50426432526f7a4a313</HarvestAddID>
</HarvestAddStsInqRq>
```

After the user action at the institution site is complete and the institution sends the token to Fiserv to access user account information, Fiserv sends back the response to getAddAccountStatus with the HarvestID. Then the partner uses the HarvestID as the RunID to invoke getAddAccountStatus and continues polling the status.

After getAddAccountStatus returns a "completed" response, the partner continues with the regular Add Account workflow, invoking the getNewAccounts and createAccounts APIs to complete the add process. The update process for accounts owned by users who have completed the OAuth procedure for OAuth-enabled FIs is the same as the account update process for FIs that do not use OAuth.

If the token shared by the FI expires or becomes invalid, Fiserv marks the institution with error code 301 (Login failure – invalid login credentials) and expects the user to re-authorize at the institution-hosted site to get the new token. To clear the error, the partner must invoke the initiateAddAccounts API with the 'AddMoreAccounts' option and include PartnerAppID. Fiserv sends a response with OAuthInfo and the partner follows the same steps above until getAddAccountStatus returns a "completed" response. Then the partner invokes the updateAccounts API to update the accounts and refresh their data.

The following diagram depicts the OAuth Add Account workflow.



Add Account workflow (PCI compliant FI)

If the FI is PCI compliant, partners must implement the following in addition to one of the Add Account workflows above.

1. FIInfoRqRs

The FIInfo is enhanced with a new element (HasCardData) to help the partner determine whether the FI is PCI compliant. A value of "Y" in this HasCardData element indicates PCI compliance.

```
<FIInfo>
  <FIId>88886650</FIId>
  <FIName>PFM Test Host FI 3</FIName>
  <Country>US</Country>
  <FIHarvestStatus>Enabled</FIHarvestStatus>
  <FICrawlable>Enabled</FICrawlable>
  <URL>
    <ParamName>FILogin</ParamName>
  </URL>
  <IsMFA>0</IsMFA>
  <HasCardData>Y</HasCardData>
</FIInfo>
```

2. GetWMAccessKey

Partners must call a **GetWMAccessKey** web service to generate the WMAccessKey. Refer to the [Miscellaneous chapter](#) for the details of this web service.

- Partners should not send card information to AllData web services system.
- Partners must send card info to Fiserv PCI system and get token to use in place of card number.
- The Fiserv PCI system requires the WMAccessKey for access.
- The WMAccessKey is valid for 15 minutes.

3. CardToken

Partners call a URL with the WMAccessKey generated in the previous step along with the "CardInfo" to generate the CardToken. See the AllData PCI Integration document for details.

4. HarvestAddRq:

Call HarvestAddRq with FI login parameters providing username/password of the FI along with the CardToken generated in step 3 in the <CryptVal> node.

```
<FILoginParam>
  <ParamName>2060549</ParamName>
  <CryptParamVal>
    <CryptType>None</CryptType>
    <CryptVal>WM##uDgOf9phLddGz+cu5s9f6IZKs1M=</CryptVal>
  </CryptParamVal>
</FILoginParam>
```

5. HarvestAddFetchRq:

Call HarvestAddFetchRq if the HarvestAddStsInqRq returned a "completed" status passing in a harvest ID.

6. HarvestAddCreateRq:

Call HarvestAddCreateRq and provide the accounts to create on the AllData side.

Delete Account workflow

To delete an account from AllData, partners invoke the Delete Account (FIDeleteRq) API on the AllData web services system when the account is deleted from the partner system.

This API is used to delete a connection to an FI established by the partner for a consumer (represented by FILoginAcctId) or a given financial account under that connection (represented by AcctId).

Account Management

Account Management uses the following web services APIs:

- **initiateAddAccounts:** Partners call this API with client FI login parameters to establish a connection. AllData returns a HarvestID and RunID which the partner uses asynchronously to check the operation status.
- **getAddAccountStatus:** Partners use this API with RunID and HarvestID to poll the status of the initiateAddAccounts API.
- **getNewAccounts:** This API returns the list of accounts within an FI connection once successfully authenticated.
- **createAccounts:** Partners call this API with a list of accounts to be added to AllData for harvesting.
- **deleteAccounts:** Partners use this API to remove accounts from AllData for an FI connection, and the connection itself. Deleting the accounts removes all the harvested data from AllData. Deleting the FI connection (FILoginAcctId) removes all the accounts harvested within the FI, along with user credentials.
- **updateAccountCredentials:** If the user credentials for a stored connection change outside of AllData, partners call this API to update the stored credentials within AllData for an FI connection.
- **maintainAccount:** Partners use this API call to change account classifications, account nicknames, etc.

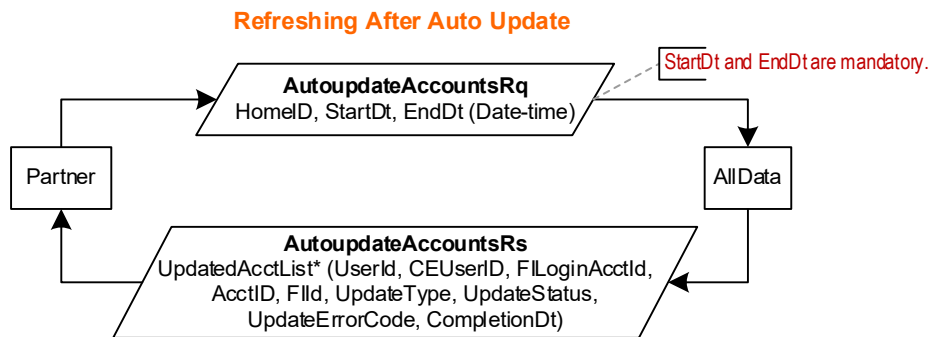
Harvest Account Data

Account Harvesting is a process that gathers the latest account information from the FI. There are two methods of update:

- Auto Update
- On-Demand Update

Auto Update

Auto Update is a process by which AllData updates all the accounts classified in automatic fashion on a nightly basis. AllData does not perform nightly updates for inactive users. Partners can elect to receive batch files on a customized schedule.

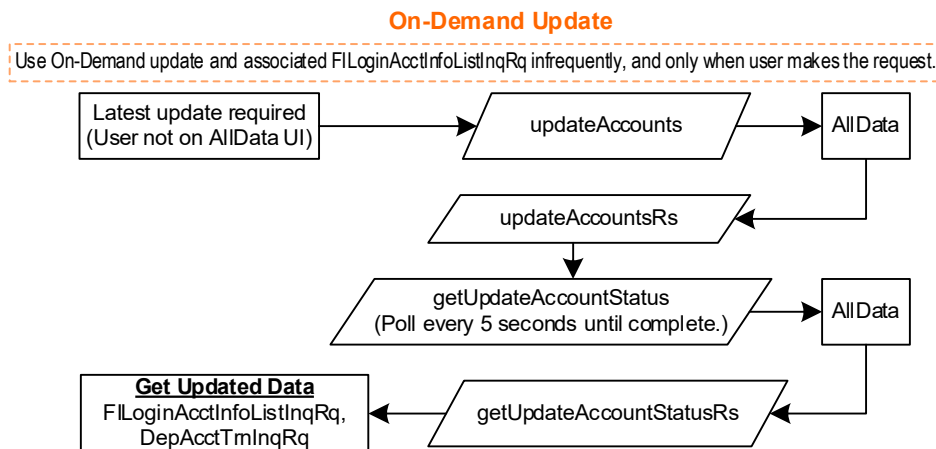


To get the data: Make FILoginAcctInfoListInqRq or FIAcctSummaryInqRq followed by transactions data pull requests.

- Pull data at user level (FILoginAcctInfoListInqRq with CEUserId) during primary auto update window.
- Pull data at login level (FILoginAcctInfoListInqRq with FILoginAcctId) during rest of day.

On-Demand Update

Users trigger On-Demand updates manually to get the latest account information.



The data refresh uses the following web services APIs:

- **updateAccounts:** This asynchronous API returns a RunID and a HarvestID. Partners use these values to poll for update status.
- **getHarvestStatus:** Partners use this API to poll the updateAccounts API call using RunID and HarvestID.

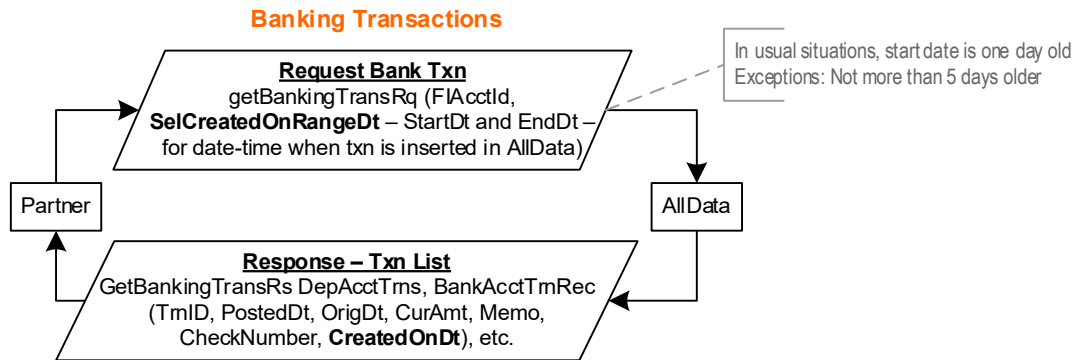
Retrieve Account Data

Once the accounts are added to AllData system, the AllData engine automatically refreshes the accounts data nightly. The information gathered as part of harvesting is made available through web service APIs.

The transaction data pull APIs can be used to retrieve transactions for each account type and time period. The different transactions APIs are given below.

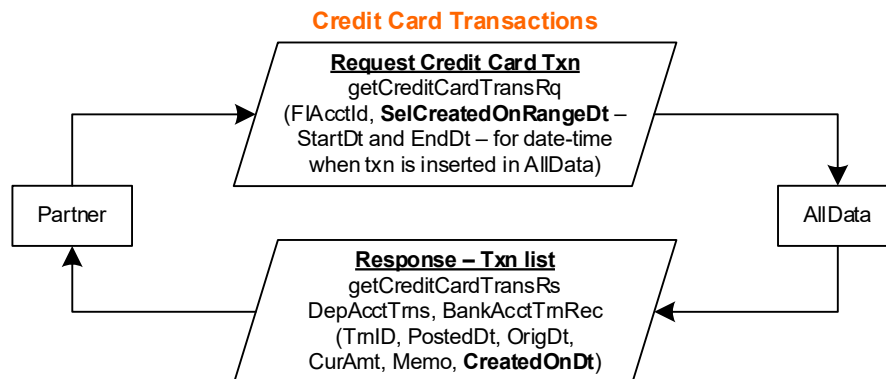
Banking transactions

The Banking transactions can be extracted from the AllData system for a certain time period using the getBankingTrans WS API. See [Account Data Pull APIs](#) for the details of the web service.



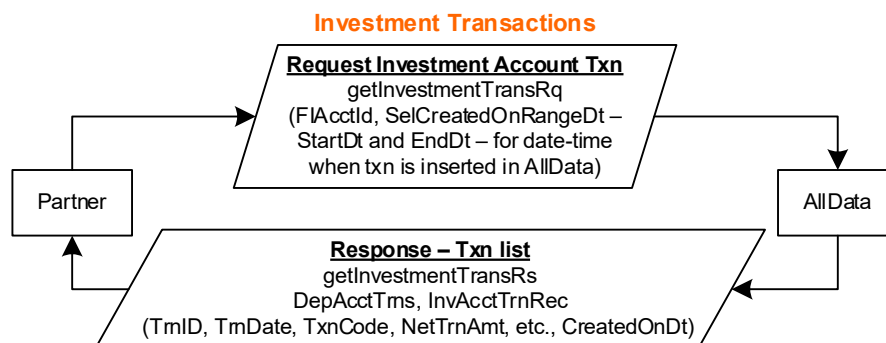
Credit Card transactions

The Credit Card transactions can be extracted from the AllData system for a certain time period using the getCreditCardTrans WS API. See [Account Data Pull APIs](#) for the details of the web service.



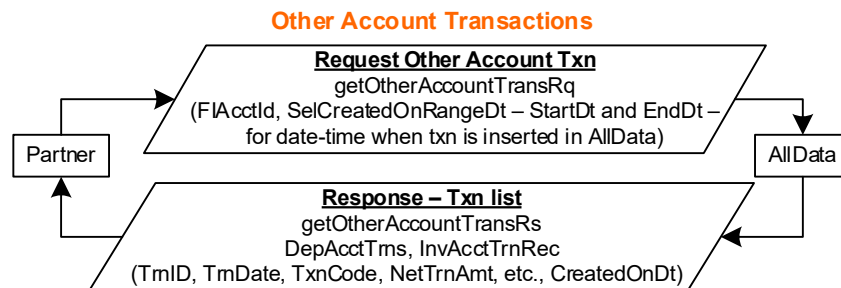
Investment transactions

The Investment transactions can be extracted from the AllData system for a certain time period using the getInvestmentTrans WS API. See [Account Data Pull APIs](#) for the details of the web service.



Other Account transactions

The asset/liability/biller account transactions can be extracted from the AllData system for a certain time period using the `getOtherAccountTrans` WS API. See [Account Data Pull APIs](#) for the details of the web service.



The account details and positions data pull APIs can be used to retrieve account summary and investment positions for each account type and time period.

Data pulls involve the following web services APIs:

- **getAccountsSummary:** This API returns metadata details for an individual account.
- **getAccountDetails:** This API returns a list of metadata for all harvested accounts within an FI connection.
- **getInvestmentTrans:** This API returns transaction details for an investment account.
- **getCreditCardTrans:** This API returns transaction details for a credit card account.
- **getBankingTrans:** This API returns transaction details for a banking account.
- **getOtherAccountTrans:** This API returns transaction details for liability accounts like billers, mortgages, loans, insurance, and other liability account types (except credit cards).
- **getInvestmentPos:** This API returns the investment positions within an investment account.

UI Widgets and API Integration

In this model, the Alldata Aggregation service is integrated as individual widgets embedded or overlaid in the partner web application or mobile application. See the Alldata Widget Integration Guide for complete partner widget documentation.

An Alldata widget is a web resource composed of web pages and accessed as any HTTP URL. Partners must use the user credentials registered with Alldata to invoke the sign-on API and get a valid session token before accessing any widgets.

Using the session token this way allows users to seamlessly access Alldata widgets from partner applications without entering additional identifiers, passwords, or other credentials.

Most API partners choose to integrate the **Alldata Add Accounts widget** into their applications. It enables adding financial institutions using their FI-issued credentials.

The image displays two sequential steps of the 'Add Accounts' widget interface. The first step, 'Find Your Institution', features a search bar and a grid of popular bank logos. The second step, 'Provide Sign-In Info', shows the login fields for a selected institution (Citi).

User Registration and Sign-On

User registration and sign-on involves the following APIs.

- **createUser:** Partners use this to create customer profiles within an Alldata home.
- **signon:** This API allows partners to use Alldata customer credentials to get an SSO token. This token is required to launch the Add Accounts Widget flow.

See [User Management Web Services](#) for more details on user registration and sign-on APIs.

UI Portal (AllData PFM) Integration

In this model, a partner uses a hosted AllData PFM UI for account administration and daily data updates.

The createUser WS API request includes performing AllData PFM user registration, and the AllData PFM UI is used for account administration to add, edit, and delete accounts in the user's login.

The user passes seamlessly from the partner's secure web application to the AllData PFM secure web portal without entering an additional identifier, password, or other credentials.

AllData PFM:

The screenshot shows the ABC Bank AllData PFM dashboard. The top navigation bar includes links for BANKING, CREDIT CARDS, LENDING, INVESTING, BUSINESS, and ALLDATA PFM. Below this is a secondary navigation bar with tabs for DASHBOARD, TRANSACTIONS, ANALYSIS, BUDGETS, GOALS, INVESTMENTS, DOCUMENTS, and SETTINGS. The main content area features a welcome message, a 'Learn more about this service' link, and three action cards: 'Add accounts from over 18,000 financial sources', 'Add real estate values with Zillow', and 'Set Goals, budgets and alerts to help manage your finances'. Below these are two sections: 'ACCOUNTS' and 'MONTHLY CASH FLOW'. The 'ACCOUNTS' section lists various account types with their balances, and the 'MONTHLY CASH FLOW' section shows a message about insufficient information for calculation. At the bottom, there is a 'RECENT TRANSACTIONS' table.

ACCOUNTS	Balance
Banking (+)	\$11,209
Credit Card (-)	\$1,882
Insurance (+)	\$35,810
Investment (+)	\$6,294,728
Mortgage (-)	\$173,952
Other Assets (+)	\$897,213
Other Liabilities (-)	\$32,165
Retirement (+)	\$60,009

RECENT TRANSACTIONS	Date	Description	Category	Amount
	05/26/20	VENMO DES: PAYMENT ID: XXXX81140	General Merchandise	-\$150.00
	05/20/20	DEKAKUMAR ANNAMA DES: Payment ID:	Finance Charges	\$100.00
	05/11/20	Monthly Maintenance Fee	Other Fees & Charges	-\$8.00
	05/10/20	Amazon Digital Svcs - Check #867890	Books & Other Media	-\$400.00
	05/07/20	Savings transaction FHB - Check #456789	Other Transfers & Adjustments	-\$136.92

AllData PFM – Account Administration:

The screenshot shows a dialog box titled 'USAA' with a warning icon. The text inside the dialog box reads: 'You are about to delete USAA and all associated accounts. All information related to this institution and its associated accounts will be permanently removed, including any links to budgets and goals. To continue, please click the "Delete Institution" button. If you would like to continue deleting this institution and all related accounts, please click the "Delete Institution" button.' There are two buttons at the bottom: 'Delete Institution' and 'Cancel'.

User Management Web Services

This chapter lists each User Management API in a table with a resource URL, descriptive information, and a link to Swagger documentation.

createUser

Web Service Name	createUser
Resource URL	<FiservWSUrl>/UserMgmt/createUser
Description	<p>This API is used to add/register the user with the AllData system.</p> <ul style="list-style-type: none"> • Minimum information required to register a user is username and password. • Username must be unique within Home and cannot be changed. • Username can be a maximum of 32 characters in length. • Username can be alphanumeric but does not allow special characters except underscore. • These reserved words are forbidden in usernames: admin, Fiserv, administrator, user • Password must be at least 8 characters and no more than 64 characters long. • Password is case-sensitive. • Password should not contain any spaces. • Password should not contain non-printable characters such as space or tab. • Returns numeric CEUserID if user is successfully created (useful for debugging issues) <p>Note: Users with "Admin" role cannot be added using this API; the AllData Professional Services team creates admin users during implementation.</p>
Swagger	createUser API docs

deleteUser

Web Service Name	deleteUser
Resource URL	<FiservWSUrl>/UserMgmt/deleteUser
Description	<p>This API is used to unregister a previously registered user.</p> <ul style="list-style-type: none"> • User data including accounts, transactions, etc., are deleted. • Admin users cannot be deleted using this service.
Swagger	deleteUser API docs

getUserProfile

Web Service Name	getUserProfile
Resource URL	<FiservWSUrl>/UserMgmt/getUserProfile
Description	This API is used to retrieve user profile information such as name, address, and contact details.
Swagger	getUserProfile API docs

updateUserProfile

Web Service Name	updateUserProfile
Resource URL	<FiservWSUrl>/UserMgmt/updateUserProfile
Description	This API is used to update user profile information such as name, address, and contact details. Note: Fiserv does not validate address or email.
Swagger	updateUserProfile API docs

updateUserPassword

Web Service Name	updateUserPassword
Resource URL	<FiservWSUrl>/UserMgmt/updateUserPassword
Description	This API is used to change the password of the current user identified by the credentials in this request's SignonRq section. <ul style="list-style-type: none">Admin user's password cannot be modified using this request. Note: This API only updates the user password for a partner's customer AllData credentials. To update FI credentials, use the updateUserCredentials API.
Swagger	updateUserPassword API docs

signon

Web Service Name	signon
Resource URL	<FiservWSUrl>/UserMgmt/SignonRq
Description	This API is used to sign on to the AllData service. It authenticates the user and generates a session token. The session token can be used for subsequent calls and this token will expire after 18 minutes of ideal time. This session token is also required to launch the Add Account widget flow.
Swagger	Signon API docs

External FI Seed Data Web services

This chapter lists each External FI Seed Data API in a table with a resource URL, descriptive information, and a link to Swagger documentation.

getFinancialInstInfo

Web Service Name	getFinancialInstInfo
Resource URL	<FiservWSUrl>/SeedDataInq/getFinancialInstInfo
Description	<p>This API can be used in three different ways:</p> <ul style="list-style-type: none"> • Retrieve FI IDs of all FIs Fiserv supports. • Retrieve FI details given a set of FI IDs. • Incremental refresh of FI seed data for a specific timeframe (typically every day). <p>This API also includes information about the extended services supported in an FI. When clients opt-in to such services, information on FI capability to support the service is included in this API response.</p>
Swagger	getFinancialInstInfo API docs

searchFinancialInst

Web Service Name	searchFinancialInstitution
Resource URL	<FiservWSUrl>/SeedDataInq/searchFinancialInst
Description	<p>Provide at least three characters to search for FIs. Valid values for 'FIInfoRequired' are:</p> <ul style="list-style-type: none"> ■ FIIDInfo - retrieves only IDs ■ FIIDInfoList - retrieves list of FIs with detail information ■ FINameInfoList – retrieves list of FIs with just ID and name • By default, this API returns FI IDs. • Use the 'FISearchSize' parameter to limit the number of result records.
Swagger	searchFinancialInstitution API docs

External FI Messages Web service

This chapter presents the External FI Messages API in a table with a resource URL, descriptive information, and a link to its Swagger documentation.

getFIMessageInfo

Web Service Name	getFIMessageInfo
Resource URL	<FiservWSUrl>/FIMessagesInq/getFIMessageInfo
Description	This API is used fetch the FI Messages (special instructions) connected to the institutions. It returns a collection of all FIs that have custom messages. Partners use this API on a periodic basis to collect FI-specific messages and display them to their customers during the login flow.
Swagger	getFIMessageInfo API docs

Account Management Web Services

This chapter lists each Account Management API in a table with a resource URL, descriptive information, and a link to Swagger documentation.

initiateAddAccounts

Web Service Name	initiateAddAccounts
Resource URL	<FiservWSUrl>/AccountMgmt/initiateAddAccounts
Description	<p>This API is also referred to as InitiateAddAccounts or HarvestAddRequest and is used to add new accounts or add more account to existing parent connection.</p> <ul style="list-style-type: none"> Requires FI ID, trust mode, and login parameters as an input. <p>This request is also used to send in MFA answers using the previously started HarvestID and additional login parameters requested.</p> <ul style="list-style-type: none"> If FI login parameters such as UserID or password are incorrect, error condition will occur with specific error code. Returns HarvestID (or run ID): Represents the running session at FI and is required while checking status. Returns FILoginAccountID: This identifier uniquely identifies the online login account of the user with an FI. If user tries to add the same login name again, "Account already exists" error (4300) occurs. <p>Refer to the Add Account workflow (OAuth) when invoking this API for OAuth-enabled FIs.</p>
Swagger	initiateAddAccounts API Docs

getAddAccountStatus

Web Service Name	getAddAccountStatus
Resource URL	<FiservWSUrl>/AccountMgmt/getAddAccountStatus
Description	<p>This API is used to poll the status of InitiateAddAccounts.</p> <ul style="list-style-type: none"> Requires HarvestID and RunID from 'initiateAddAccounts' response Response includes status of "completed," "harvest error," or "in progress" <p>Check status periodically until service successfully harvests accounts, errors out, or times out.</p> <p>Handle errors such as login failure (301), multi-factor authentication (303), and wrong MFA answer (304) using the HarvestAddRq-AddMoreAccts element.</p> <p>Note: Additional information is needed in case of MFA.</p> <p>Refer to the Add Account workflow (OAuth) when invoking this API for OAuth-enabled FIs.</p>
Swagger	getAddAccountStatus API Docs

getNewAccounts

Web Service Name	getNewAccounts
Resource URL	<FiservWSUrl>/AccountMgmt/getNewAccounts
Description	<p>This API is invoked after the initiateAddAccounts request has completed. It returns the list of financial accounts found on the FI website. At this point these accounts are not created in Fiserv database. The basic account information available after login to FI site is harvested, but information such as balance, transactions, and positions may not be available.</p> <p>This API also includes account ownership information (full account number, account owner name, routing number, and contact information – address, email, and phone number) when clients opt-in for that extended service.</p>
Swagger	getNewAccounts API Docs

createAccounts

Web Service Name	createAccounts
Resource URL	<FiservWSUrl>/AccountMgmt/createAccounts
Description	<p>User can choose accounts to add within AllData for harvesting.</p> <p>Fiserv classifies the harvested accounts based on certain account classification rules. The user can either override this classification or provide the classification if not already set.</p> <p>After successfully adding accounts, call update account APIs to run on-demand harvesting to retrieve account data.</p>
Swagger	createAccounts API docs

deleteAccounts

Web Service Name	deleteAccounts
Resource URL	<FiservWSUrl>/AccountMgmt/deleteAccounts
Description	<p>This API is used to delete either a parent CFI (FILoginAcctId) or a given financial account.</p> <ul style="list-style-type: none"> • If a parent CFI ID is given, all associated financial accounts are also deleted. • All account data including transactions and positions are deleted.
Swagger	deleteAccounts API Docs

updateAccountCredentials

Web Service Name	updateAccountCredentials
Resource URL	<FiservWSUrl>/AccountMgmt/updateAccountCredentials
Description	<p>This API can be used to update the user credentials information stored within AllData. After updating credentials in AllData, partner should validate the credentials by invoking on-demand account harvesting flow.</p> <p>Note: This API is not applicable for OAuth-enabled FIs.</p>
Swagger	updateAccountCredentials API docs

maintainAccount

Web Service Name	maintainAccount
Resource URL	<FiservWSUrl>/AccountMgmt/maintainAccount
Description	This API is used to modify nickname at Fiserv, account type, and properties such as retirement status. Provide extended account type while modifying account type.
Swagger	maintainAccount API Docs

addOfflineAccount

Web Service Name	addOfflineAccount
Resource URL	<FiservWSUrl>/ClientMgmt/addOfflineAccount
Description	This API is used to add offline accounts to client. Offline accounts are tangible assets not held in typical financial institutions. Examples of offline accounts are jewelry, automobiles, boats, RVs, rentals, paintings, and heirlooms.
Swagger	addOfflineAccount API Docs

updateOfflineAccount

Web Service Name	updateOfflineAccount
Resource URL	<FiservWSUrl>/ClientMgmt/updateOfflineAccount
Description	This API is used to modify client offline accounts.
Swagger	updateOfflineAccount API Docs

addUpdateHostAccounts

Web Service Name	addUpdateHostAccounts
Resource URL	<FiservWSUrl>/AccountMgmt/addUpdateHostAccounts
Description	<p>This API used to add and update client banking/credit card accounts from host FIs.</p> <p>Add: Account number, account name, and account type are mandatory elements to add client host accounts. Balance and transaction information are optional elements.</p> <p>Update: The host accounts are identified using either combination of (FIId + AcctNumber) or AcctId in the API request.</p> <ul style="list-style-type: none"> • If the combination of (FIId + AcctNumber) or AcctId matches between the AllData and the API request, the API updates the account attributes (balance, account name, transactions) in AllData. • If the combination of (FIId + AcctNumber) or AcctId does not match, the API adds a new host account in the AllData system. • The response API will notify the partner about the status of each host account processed as part of the request API.
Swagger	addUpdateHostAccounts API Docs

HostAccountsStatus

Web Service Name	HostAccountsStatus
Resource URL	<FiservWSUrl>/AccountMgmt/hostAccountsStatus
Description	<p>This API returns the status of host accounts transaction categorization.</p> <p>Possible return values are:</p> <ul style="list-style-type: none"> • InProgress • Completed • Failure
Swagger	hostAccountsStatus API Docs

Account Harvesting APIs

This chapter lists each Account Harvesting API in a table with a resource URL, descriptive information, and a link to Swagger documentation.

updateAccounts

Web Service Name	updateAccounts
Resource URL	<FiservWSUrl>/HarvestAccountData/updateAccounts
Description	<p>This API can be used to trigger on-demand account harvesting. Typically, all added accounts get harvested in nightly batch mode. If a user (or CSR) wants to see latest account data, partners can call this asynchronous API, poll the harvesting status and, after it completes, refresh the account data for the user.</p> <ul style="list-style-type: none"> • Updates all accounts of a user from list of FIs, all accounts under list of parent CFI IDs, or list of individual accounts • Update accounts under a single parent CFI ID after the initial add. • RunID and HarvestID returned in the response are required when checking call status. <p>This API also provides for requesting extended transactions history of the accounts on demand (any time) for the clients who opt in for that service. The request should include the ExtendedTrnsDays element with the number of days expected.</p>
Swagger	updateAccounts API Docs

getHarvestStatus

Web Service Name	getHarvestStatus
Resource URL	<FiservWSUrl>/HarvestAccountData/getHarvestStatus
Description	<p>The API returns the status of the update done in the updateAccounts API. Requires RunID and HarvestID from the updateAccounts response. This API has the IncludeDetail attribute that returns status based on the values passed. The typical values of this tag are "FIStatus" and "AcctStatus."</p> <ul style="list-style-type: none"> • FIStatus: Returns the status of FIs specified in the updateAccounts API • AcctStatus: Returns the status of accounts of FI. If the above values are not specified, API returns the status of all the accounts specified in the updateAccounts API.
Swagger	getHarvestStatus API Docs

getAccountUpdateSummary

Web Service Name	getAccountUpdateSummary
Resource URL	<FiservWSUrl>/Account Data Inquiry Service/getAccountUpdateSummary
Description	<p>This API is primarily for clients using widget implementation. After users add accounts using the Add Account widget and the return URL is received from AllData, invoke this API to get the status of accounts update, before invoking the data pull APIs.</p> <p>This API returns status of all accounts updated under that CEUserID within the last hour.</p>
Swagger	getAccountUpdateSummary API Docs

Account Data Pull APIs

This chapter lists each Account Data Pull API in a table with a resource URL, descriptive information, and a link to Swagger documentation.

getAccountsSummary

Web Service Name	getAccountsSummary
Resource URL	<FiservWSUrl>/AccountDataInq/getAccountsSummary
Description	This API is used to fetch the account summary information for a specified FI account.
Swagger	getAccountsSummary API Docs

getAccountDetails

Web Service Name	getAccountDetails
Resource URL	<FiservWSUrl>/AccountDataInq/getAccountDetails
Description	<p>This API can be used to fetch detail information about one or more FI login accounts of a user.</p> <ul style="list-style-type: none"> It can be used to retrieve information about one or more login accounts. If no ID is specified, it returns information about all of user's FI login accounts.
Swagger	getAccountDetails API Docs

getInvestmentTrans

Web Service Name	getInvestmentTrans
Resource URL	<FiservWSUrl>/AccountDataInq/getInvestmentTrans
Description	This API is used to fetch the transactions for an investment account. It returns the transactions for the specified date range.
Swagger	getInvestmentTrans API Docs

getCreditCardTrans

Web Service Name	getCreditCardTrans
Resource URL	<FiservWSUrl>/AccountDataInq/getCreditCardTrans
Description	This API is used to fetch the transactions for a credit card account. It returns the transactions for the specified date range.
Swagger	getCreditCardTrans API Docs

getBankingTrans

Web Service Name	getBankingTrans
Resource URL	<FiservWSUrl>/AccountDataInq/getBankingTrans
Description	This API is used to fetch the transactions for a banking account. It returns the transactions for the specified date range.
Swagger	getBankingTrans API Docs

getOtherAccountTrans

Web Service Name	getOtherAccountTrans
Resource URL	<FiservWSUrl>/AccountDataInq/getOtherAccountTrans
Description	This API is used to fetch the transactions for other asset/liability/biller account. It returns the transactions for the specified date range.
Swagger	getOtherAccountTrans API Docs

getTransactions

Web Service Name	getTransactions
Resource URL	<FiservWSUrl>/AccountDataInq/getTransactions
Description	<p>This API is used for searching for transactions in specified FI login accounts according to specified search criterion.</p> <p>TrnType can be used to either retrieve debit transactions, credit transactions or both</p>
Swagger	getTransactions API Docs

getDeletedTrans

Web Service Name	getDeletedTrans
Resource URL	<FiservWSUrl>/AccountDataInq/getDeletedTrans
Description	<p>This API is used to get the deleted transactions of all accounts in the home or by account group with a specified date range and row index of the batch.</p> <p>The client/partner accumulating the transaction data in their system uses this API to remove the deleted transactions to avoid duplication of transactions in their system.</p> <p>This process is referred to as “transaction deduping” and is performed to keep transaction data in sync between AllData and the FI. It updates the AllData system with any changes found in old transactions posted in the FI source.</p>
Swagger	getDeletedTrans API Docs

getInvestmentPos

Web Service Name	getInvestmentPos
Resource URL	<FiservWSUrl>/AccountDataInq/getInvestmentPos
Description	This API is used to retrieve investment positions for an investment account.
Swagger	getInvestmentPos API Docs

getEmployerStockOptions

Web Service Name	getEmployerStockOptions
Resource URL	<FiservWSUrl>/AccountDataInq/getEmployerStockOptions
Description	This API is used to retrieve the stock options details of an ESOP account.
Swagger	getEmployerStockOptions API Docs

Transaction Categorization APIs

Transaction categorization is an optional feature that can be turned on for partners that require AllData to categorize transactions. AllData can only perform this for banking and credit card account types.

This chapter lists each transaction categorization API in a table with a resource URL, descriptive information, and a link to Swagger documentation. See the AllData Transaction Categorization – API Specification document for more details.

getTxnCategoriesInfo

Web Service Name	getTxnCategoriesInfo
Resource URL	<FiservWSUrl>/SeedDataInq/getTxnCategoriesInfo
Description	<p>This API is used to get the transactions categories from the AllData system for a specific home.</p> <p>Use Admin User role to get all the standard transaction categories and subcategories AllData provides. Clients must pull this complete standard set initially and later make this request to get any new category added or will be intimated to make this request if any new category added.</p> <p>Use AllData User role to get custom subcategories created by the specific user.</p>
Swagger	getTxnCategoriesInfo API Docs

categorizeTransaction

Web Service Name	categorizeTransaction
Resource URL	<FiservWSUrl>/TxnMgmt/categorizeTransaction
Description	<p>This API is used to assign/update the subcategory of transactions and apply this user choice to similar transactions in future.</p> <p>This API can also be used to create a user-specific custom subcategory and assign it to transactions, and to apply the subcategory to similar future transactions.</p>
Swagger	CategorizeTransaction API Docs

deleteSubCategory

Web Service Name	deleteSubCategory
Resource URL	<FiservWSUrl>/TxnMgmt/deleteSubCategory
Description	This API is used to delete the custom subcategory created by specific user.
Swagger	DeleteSubCategory API Docs

Budgets

This chapter lists each Budget-related API in a table with a resource URL, descriptive information, and a link to Swagger documentation. Budgets are used in PFM use cases.

createBudget

Web Service Name	createBudget
Resource URL	<FiservWSUrl>/BudgetMgmt/createBudget
Description	This API is used to create budgets.
Swagger	createBudget API Docs

deleteBudget

Web Service Name	deleteBudget
Resource URL	<FiservWSUrl>/BudgetMgmt/deleteBudget
Description	This API is used to delete the budgets created by users.
Swagger	deleteBudget API Docs

editBudget

Web Service Name	editBudget
Resource URL	<FiservWSUrl>/BudgetMgmt/editBudget
Description	This API is used to edit the existing budgets.
Swagger	editBudget API Docs

findBudget

Web Service Name	findBudget
Resource URL	<FiservWSUrl>/BudgetMgmt/findBudget
Description	This API is used to find the status of user-created budgets and to track their progress.
Swagger	findBudget API Docs

Goals

This chapter lists each Goal-related API in a table with a resource URL, descriptive information, and a link to Swagger documentation. Goals are used in PFM use cases.

createGoal

Web Service Name	createGoal
Resource URL	<FiservWSUrl>/GoalMgmt/createGoal
Description	This API is used to create a goal.
Swagger	createGoal API Docs

editGoal

Web Service Name	editGoal
Resource URL	<FiservWSUrl>/GoalMgmt/editGoal
Description	This API is used to edit the goal, changing its target amount, target date, or the contribution percentage of the account.
Swagger	editGoal API Docs

deleteGoal

Web Service Name	deleteGoal
Resource URL	<FiservWSUrl>/GoalMgmt/deleteGoal
Description	This API is used to delete a goal.
Swagger	deleteGoal API Docs

findGoal

Web Service Name	findGoal
Resource URL	<FiservWSUrl>/GoalMgmt/findGoal
Description	This API is used to find a goal and to check the status of the goal and the account(s) associated with it.
Swagger	findGoal API Docs

Bank Statement Download Services

This chapter lists each Bank Statement Download Services-related API in a table with a resource URL, descriptive information, and a link to Swagger documentation. AllData can download bank statements for a select list of FIs. By default, AllData can download up to three historical statements once customer consent is received. See the PDF Statement Download Service document for more details.

bankStatementConsent

Web Service Name	bankStatementConsent
Resource URL	<FiservWSUrl>/AccountMgmt/bankStatementConsent
Description	<p>This API is used to provide user consent and FI consent to download bank account statements from the financial institution website.</p> <ul style="list-style-type: none"> User consent and FI consent should be provided separately using this API. <ul style="list-style-type: none"> User consent is the primary consent, which is to be provided first. On providing user consent, AllData will set FI consent to true for all eligible FIs under the user profile. FI consent can be revoked in one or more FIs when the user does not wish to obtain statements from any specific FIs. FI consent can be provided again for consent-revoked FIs.
Swagger	bankStatementConsent API Docs

getBankStmtDetails

Web Service Name	getBankStmtDetails
Resource URL	<FiservWSUrl>/AccountDataInq/getBankStmtDetails
Description	<p>This API is used to get user and FI consent details and the details of the FIs and bank accounts under the user profile that are eligible for downloading the bank account statements.</p> <p>It also includes the details of account statements available for download under each of those accounts.</p> <p>Note: The availability of statements is dependent on connectivity with the source site. Refer to institution(s) or account(s) errors and resolve any associated issues to get the statements.</p>
Swagger	getBankStmtDetails API Docs

Miscellaneous

This chapter lists each API in a table with a resource URL, descriptive information, and a link to Swagger documentation.

GetWMAccessKey

Web Service Name	GetWMAccessKey
Resource URL	<FiservWSUrl>/AccountMgmt/GetWMAccessKey
Description	This API is used to generate WMAccessKey key for PCI compliance requirements for the FI.
Swagger	GetWMAccessKey API Docs

Web Services Setup and Usage

Web Services Setup

The following are the prerequisites for web services API integration with Fiserv.

1. Environment setup in AllData system
2. Fiserv provides partner with admin user credentials
3. Server URL to submit web services API requests
4. Session handover URL (for single sign-on (SSO))

Concurrent Access

Currently the AllData system throttle supports up to 50 concurrent sessions of the WS APIs. Fiserv recommends partners do not exceed 50 sessions for optimal performance.

URLs

Please note that the server URLs differ between UAT and production environments. The URLs are provided as part of the setup.

Appendix A: API Error Codes

The following table lists all valid response status codes for the status aggregate defined earlier, their severities, their default text provided in the StatusDesc element, conditions that may trigger them, and, in some cases, information to help API partners resolve issues.

Code	Severity	StatusDesc default text	Condition	Partner resolution action
0	Info	Success	The service provider successfully processed the request.	
50	Warning	Partial success	Partial success	
100	Error	General Error	An error prevented the service provider from processing the transaction. No additional information is available.	If this error persists, please contact us with the CEUserID and timestamp.
600	Error	Unsupported message	The request received from the partner is not currently supported.	
1020	Error	Required element not included	Required element is missing in the request received from the partner.	Ensure API request provides all mandatory elements. Refer to Swagger documentation.
1740	Error	Authentication Failed	The customer could not be authenticated due to an incorrect HomeID, login ID, or password.	Verify the user login ID and password.
2740	Error	Invalid currency code	The currency code specified in the request is invalid.	
4000	Error	Invalid UserID	The UserID specified in the request is invalid.	
4010	Error	Invalid partnerID	The partnerID specified in the request is invalid.	
4012	Error	Invalid HomeID	The HomeID specified in the request is invalid.	
4020	Error	Invalid password	The password specified in the request is invalid.	
4030	Error	User already registered	The user identified by partnerID:HomeID:UserID is already registered with the server.	
4040	Error	User not registered	The user identified by partnerID:HomeID:UserID is not registered with the server.	
4050	Error	Invalid encryption scheme	The encryption scheme specified in the request is invalid.	
4060	Info	No data available	No data is available to satisfy the request. This could mean that the user has not registered any financial institutions, there are no transactions in the specified date range, and so on.	If this error occurs on invoking getAccountDetails API, it means user has not added any financial institutions. Confirm the user successfully added accounts under any financial institutions. If this error occurs on invoking transactions APIs, confirm the LastSuccessfulUpdate date is returned in getAccountDetails API for that specific account and the specified date range lies within the LastSuccessfulUpdate date.

Code	Severity	StatusDesc default text	Condition	Partner resolution action
4070	Error	Validation failure	One or more input fields were invalid.	Partner should make sure the mandatory parameters are sent in the request and in the defined format as in the corresponding Swagger documentation.
4080	Error	SessInfo creation failed	The Fiserv server was unable to create a SessInfo security token.	Reinvoke the signon API to get the new SessInfo token. If the error persists, please contact us with the CEUserID and timestamp.
4090	Error	Invalid SessInfo	The SessInfo token specified in the request is invalid. The session token is invalid or expired.	Generate a new SessInfo token and reinvoke the process flow with the new SessInfo token. If the error persists, please contact us with the CEUserID and timestamp.
4100	Error	Invalid FIAcctId	The FIAcctId specified in the request is invalid.	
4110	Error	AcctType mismatch	The account specified in the request is not of the type specified in the request.	Verify AcctType of the accounts from getAccountDetails API response.
4120	Error	Not owner	The FILogin Account or FI Account for which a user is requesting details does not belong to the user.	The FILoginAcctId or AcctId sent by partner does not exist in the Fiserv system or the IDs belong to a different CEUserID in the system.
4130	Error	Invalid role	A sign-on has been attempted by a user in a role other than the authorized role.	
4140	Error	Invalid FILoginAcctId	The FILoginAcctId specified in the request is invalid.	
4150	Error	Invalid AcctType	The AcctType or ExtAcctType specified in the request is invalid.	Verify AcctType or ExtAcctType of the accounts from getAccountDetails API response.
4210	Error	Invalid XML Document	The request received from the partner site is not a valid XML document.	
4220	Error	UserID unavailable	The UserID specified in the request is not available for registration.	
4240	Error	UserDataPush failed	An attempt to push a user's data failed.	
4250	Error	Invalid transaction type	The transaction type specified in the request is invalid.	Expected transaction type is Debit or Credit. Any other value returns this error.
4260	Error	Invalid date	The date specified in the request is invalid.	
4270	Error	Invalid amount	The amount specified in the request is invalid.	
4280	Error	Invalid search criterion	A search criterion specified in the request is invalid.	
4290	Error	Invalid BroadcastMsgID.	The BroadcastMsgID specified in the request is invalid.	

Code	Severity	StatusDesc default text	Condition	Partner resolution action
4300	Error	Account Already Exists	There was an attempt to add an account that already exists.	Do not send an AddNewAccounts request for a user with the same set of credentials in an FI. In case of an error with initial attempt with that set of credentials, refer to the Harvester Error Codes section of this document for information and resolution suggestions.
4310	Error	Harvesting Error	There was a harvesting error during the “account add” or “add more accounts” operation.	Research the returned UpdateErrorCode in the Harvester Error Codes section of this document for information and resolution suggestions.
4320	Error	FI Login Credentials Required	An “add more account” operation was attempted for a low-trust account without providing the login credentials.	
4322	Error	Avoid gathering and sending User Credentials in OAuth FI	User credentials sent in the request for adding accounts in an OAuth-enabled FI.	Partner should refer to Add Account workflow (OAuth) and follow the instructions when initiating add under an OAuth-enabled FI.
4330	Error	Invalid HarvestID	The harvest ID / run ID provided in the request is invalid.	
4332	Error	Invalid OAuth Request ID	The OAuth request ID sent in the request is invalid.	
4333	Error	Invalid Partner Application ID	The partner application ID is invalid.	
4335	Error	Partner Application ID is required	Partner application ID is missing in the request.	Partner should ensure mandatory parameters are sent in the request.
4336	Error	OAuth configuration is required	There is an error in OAuth configuration.	If this error persists, please contact us with the CEUserID and timestamp.
4350	Error	Update in progress.	An update is already in progress. A new update (update or add) request cannot be submitted until the existing update is completed.	Allow the running update to complete before invoking update or add.
4360	Error	Incomplete Data for Request Completion.	The data provided for Request completion is not enough. Typically, this means insufficient data was provided for “add new accounts,” “add more accounts,” or “account maintenance” operations, and so on.	
4370	Error	Invalid account classification attributes.	The account attributes combination required for account classification is invalid. <ul style="list-style-type: none"> Instrument AccountOwnership RetirementStatus 	Please contact us with CEUserID and timestamp.

Code	Severity	StatusDesc default text	Condition	Partner resolution action
4390	Error	Invalid combination of FILoginAcctId and HarvestAddFetch(s)	Data contained in HarvestAddFetchAcctList does not match the FILoginAcctId. Especially created for the HarvestCreateAcctsRq.	Confirm the FILoginAcctId sent in the request is correct and consistent through the harvest add flow.
4410	Error	Invalid FIId	The FIId provided is not supported or does not exist in the database or the number provided is null or non-numerical.	
4420	Error	Count of Accounts is high for this LoginAcctId	The number of accounts eligible for data pull submitted in AdvFILoginAcctInqRq is greater than the configured value.	
4430	Error	No accounts found or accounts not eligible for update	The query criteria did not match with any accounts in the database eligible for update.	The updateAccounts API is invoked for a user with no eligible accounts to update. That is, all of the user's accounts have invalid credentials or they exist under suspended FIs. Confirm that the user has eligible accounts for update.
4440	Error	Duration is more than 3 days	The maximum duration allowed for deleted transactions API is three days.	
4450	Error	No New Accounts	This is a warning message generated, when the harvest action during an "account add" or "add more accounts" operation was successful, but no new accounts were fetched from FI site.	This could be a scripting error or there could simply be no new accounts to add.
4460	Error	FI Login Credential exceed max length	Login credentials entered by user exceed the maximum length allowed at financial institution website.	
4470	Error	The FI Login Account is suspended and cannot be updated. Please update the login information to resume aggregation activity.	When harvesting account login credentials are rejected, the Account Harvest Status (AcctHarvestStatus) is disabled. The FI Login Account Info List Inquiry Request (FILoginAcctInfoListInqRq) returns this error message in the response (FILoginAcctInfoListInqRs) to convey that the account is locked in the Fiserv system.	Correct the institution login credentials.
4471	Error	This account is at a financial institution that is not supported within this service. The account data will not be able to be refreshed/updated because support is now unavailable for this financial institution.	Received in the status aggregate when attempting to update an account or add more accounts to an FI that is suspended in Fiserv.	Inform customers that the FI is temporarily unavailable.
4480	Error	Invalid AcctId	The AcctId in the request does not match with any accounts for this user or is null.	
4510	Error	Invalid Account Type Id	The account type id provided is invalid and does not exist in the database or is null.	
4530	Error	Invalid Balance Type	The balance type provided is invalid.	
4540	Error	Account Group specified in the request is invalid	The account group provided is invalid. Possible values are: Bill, Cash, Credit, Insurance, Investment, and Other	

Code	Severity	StatusDesc default text	Condition	Partner resolution action
4550	Error	Invalid Date Range	The Start Date specified in the date range greater than current date.	
4551	Error	Invalid date range. Modified date range returned no results.	Date range is a maximum of 90 days for all the transaction pull APIs except Deleted transactions API. No transactions are available in Fiserv for the modified date range.	The default maximum range is 90 days; this is configurable for partners requesting extended transaction history.
4560	Error	FI URL cannot be NULL	FI URL provided in the request is empty. FI URL is a mandatory field.	
4570	Error	FI Name cannot be NULL	FI Name provided in the request is empty. FI Name is a mandatory field.	
4580	Error	FI has already been requested and is under development	The Fiserv Data Operations team is already working on developing scripts for the requested FI.	
4590	Error	FI is already supported	The requested FI is already supported.	Search for this FI or refresh the FI seed data.
4591	Error	No FIs found matching the search criteria provided	The FI search criteria provided did not match with any FIs in the database.	
4600	Error	Request ID is required	Request ID (RqUID) is a mandatory field. Each request should have a unique request ID so partners can reconcile the responses they receive.	
4603	Error	Person Name is required	PersonName aggregate is mandatory in PersonInfo aggregate.	
4604	Error	First Name is required	FirstName is mandatory in PersonName aggregate.	
4605	Error	Last Name is required	LastName is mandatory in PersonName aggregate.	
4615	Error	E-Mail address passed in the request is invalid	The email address provided in the request is invalid.	
4620	Error	UserProfile is required	UserProfile aggregate is mandatory if the partner home is configured.	
4621	Error	PersonInfo is required	PersonInfo aggregate is mandatory if the partner home is configured.	
4622	Error	Contact Info is required	Contact Info aggregate is mandatory if the partner home is configured.	
4623	Error	Day Phone is required	DayPhone is mandatory if the partner home is configured.	
4624	Error	Evening Phone is required	EveningPhone is mandatory if the partner home is configured.	
4625	Error	Phone number has non-numeric data	DayPhone and/or EveningPhone provided are non-numerical.	
4626	Error	Total phone numbers do not match allowed occurrence	Telephone numbers provided should not exceed the configured occurrence value.	
4627	Error	Email Address is required	Email address is mandatory if the partner home is configured.	
4628	Error	Invalid Email Address	The email address provided in the request is invalid.	

Code	Severity	StatusDesc default text	Condition	Partner resolution action
4629	Error	Only one day phone value is allowed	DayPhone is provided more than once.	
4630	Error	Only one evening phone value is allowed	EveningPhone is provided more than once.	
4710	Error	User is locked	The requested user is locked in the Fiserv system.	Please contact us with the CEUserID.
5012	Error	Trust mode cannot be changed, as only high trust mode is allowed.	Trust mode cannot be changed if the partner home is configured for high-trust mode only.	Occurs when partners send other trust modes (low or medium trust) when editing the FI credentials.
5013	Error	Invalid AcctName	When creating or updating an offline account, an empty or null value was sent for account name element.	
7601	Error	Transaction Not Found	There is no transaction with given criteria.	Verify the transaction ID sent in the request.
7602	Error	Category Not Found	The category provided does not exist in the Categorization Engine seed data.	
7603	Error	Subcategory Not Found	The subcategory provided does not exist in the Categorization Engine seed data.	
7604	Error	Subcategory label already exist	The user tried to add a category that already exists.	
7605	Error	Not Processed	This error is related to category engine processing indicating that the transaction category update was not processed.	
7611	Error	Request originated from an invalid IP address	Request originated from unknown (not whitelisted) IP address.	Contact Fiserv to whitelist any new IPs or make the request from whitelisted IPs.
7613	Error	Unsupported End Date, is lesser than 5 days from today	EndDt provided in AutoHarvestStsRq is less than 5 days from today.	
7701	Error	This financial institution requires a multi-factor authentication process that is currently not enabled for this home	The FI requires an image-based multi-factor authentication that is currently not enabled for this home. Please contact Fiserv to enable it.	

Appendix B: Harvester Error Codes

The following table lists and describes error codes returned as part of harvesting.

Error code	Error	Message	Description	Partner resolution action
100	Internal software error	Internal software error. Please try updating again in a few minutes.	Fiserv tried harvesting information from an institution but could not harvest the information successfully despite getting response from the institution.	Temporary issue: Partner should send HarvestAddRq/HarvestRq again. If it still fails, then it may be a harvesting error. Partner should report the issue to Fiserv after several unsuccessful retries. These errors could be due to many reasons, such as FI server is busy or FI site is down. Retries usually resolve the problem.
103	Nonexistent URL or network failure.	Network failure. Please try updating again in a few minutes.	This is typically caused when there is a change in the hostname of the URL that is being accessed, but sometimes occurs due to temporary issues.	Temporary issue: Partner should send HarvestAddRq/HarvestRq again. If it still fails, then it may be a harvesting error. Partner should report the issue to Fiserv after several unsuccessful retries.
104	Time out error	We cannot establish a connection to your financial institution website at this time. Please try updating again in a few minutes.	Fiserv tried to connect to the institution where your account exists and was unable to reach the host institution due to network traffic. This is NOT a session timeout. When we fail to make a connection with the FI site after a few tries, it is classified as 104 error.	Temporary issue: Partner should send HarvestAddRq/HarvestRq again. If it still fails, then it may be a harvesting error. Partner should report the issue to Fiserv after several unsuccessful retries.
105	Target server error or server down	We are receiving a message that the institution's website is temporarily unavailable. We will update the account when the website becomes available	Fiserv tried to connect to the institution where your account exists, and host institution is returning server error.	Temporary Issue - The partner needs to send HarvestAddRq/HarvestRq again. If it still fails, then it may be a harvesting error. Partner should report the issue to Fiserv after several unsuccessful retries.
106	Connection failed	We are not able to update your account because of network traffic on the internet. Please try updating again in a few minutes.	Fiserv tried to connect to the institution where your account exists and was unable to reach the host institution due to connection failure.	Temporary issue: Partner should send HarvestAddRq/HarvestRq again. If it still fails, then it may be a harvesting error. Partner should report the issue to Fiserv after several unsuccessful retries.
107	SSL error	We cannot establish a connection to your financial institution website at this time. Please try updating again in a few minutes.	This is typically caused by a failure to establish SSL connection with the FI website. Usually it is a failure that AllData can fix.	Temporary issue: Partner should send HarvestAddRq/HarvestRq again. If it still fails, then it may be a harvesting error. Partner should report the issue to Fiserv after several unsuccessful retries.

Error code	Error	Message	Description	Partner resolution action
108	Server error (FI website maintenance)	Your financial institution website is not available. Please try updating again later.	This happens when the FI website returns a positive response but displays a message that the account information is unavailable. This is established based on a positive key text assertion.	Temporary issue: Partner should send HarvestAddRq/HarvestRq again. If it still fails, then it may be a harvesting error. Partner should report the issue to Fiserv after several unsuccessful retries.
109	Client error	We cannot establish a connection to your financial institution website at this time. Please try updating again in a few minutes.	This is an extremely rare error. It occurs due to a change in the site or some temporary issues.	Temporary issue: Partner should send HarvestAddRq/HarvestRq again. If it still fails, then it may be a harvesting error. Partner should report the issue to Fiserv after several unsuccessful retries.
110	Account information unavailable	We did not find any relevant account values displayed in the financial institution website for this account which could be aggregated. Please check your account status in financial institution website.	Account-specific issues such as account value unavailable to harvest fall in this category. In this case, there will not be any required values displayed in the FI website for an account to harvest and it's not related to temporary FI website maintenance issues. For example, no balance account could be on the lines of a closed account or inactive due to specific reasons. User should check the account status on FI website and then decide to drop or retain the account.	Partner should send HarvestAddRq/HarvestRq again when all the information is available in FI website is available. If it still fails, then it may be a harvesting error. Partner should report the issue to Fiserv after several unsuccessful retries.
121	Account not updated during nightly refresh	There is a delay in updating this account due to a system issue. Please check after some time or retry updating the account.	Due to technical reasons, there could be a delay in account update during nightly refresh. Fiserv will investigate this error to resolve it.	Temporary issue: Partner should send HarvestAddRq/HarvestRq again. If it still fails, then it may be a harvesting error. Partner should report the issue to Fiserv after several unsuccessful retries.
200	Site change	We are in the process of upgrading our product to rectify this problem. Please try again later.	This along with error code 100 is a blanket category for a change in the website layout. This is also a generic error code that will be presented before getting assigned with an appropriate error code.	Temporary issue: Partner should send HarvestAddRq/HarvestRq again. If it still fails, then it may be a harvesting error. Partner should report the issue to Fiserv after several unsuccessful retries.
201	Site change - account maintenance needed	We can no longer find this account in your online access at this financial institution website.	This is caused by a change in display of account identifier by the FI website leading to a mismatch in account identification process and inability of the AllData harvesting process to identify an account. This could also be caused by a user closing an account with the FI and therefore account is not available on the site.	Partner should send FIDeleteRq to delete the failing accounts and use HarvestAddRq-AddMoreAccts for adding new accts.

Error code	Error	Message	Description	Partner resolution action
202	Multiple matches found	We are not able to update this account at this time as we are currently upgrading our data collection process for this financial institution.	Failure at the time of identifying accounts during account update process because multiple matches were found for the account.	Partner should report the issue to Fiserv.
203	No account found	It appears that this account can no longer be located within the institution's website. Please confirm this by logging into the Institution's website and delete this account from Fiserv if it does not exist.	Account not found in FI website. Account's presence on FI website will be monitored for a few days and then marked as a missing account (204) if it is still not found.	Partner should send FIDeleteRq to delete the failing accounts and use HarvestAddRq-AddMoreAccts for adding new accts.
204	No account found	This account no longer exists in the institution's website. Please delete this account as it does not exist.	This error code is meant to clearly indicate to the user that the account is missing on the FI website and to seek user's intervention to delete the account from Fiserv application. These accounts are not harvested nightly.	Partner should send FIDeleteRq to delete the failing accounts and use HarvestAddRq-AddMoreAccts for adding new accts.
205	Empty fetch list	We are not able to update this account at this time as we are currently upgrading our data collection process for this financial institution.	Failure at the time of identifying accounts during account update process because no accounts were found.	Partner should report the issue to Fiserv.
208	Unable to determine type of account	We are unable to determine what type of an account you have at this financial institution.	This error code is meant to indicate that our account classification engine is unable to determine the type of this account using information our harvesting engine finds at the financial institution.	Partner should classify this account based on additional information they have about this account or may ask the user to classify it as the user will know the type of account.
209	Incorrect account type classification	Incorrect account type classification. Please reclassify the account type correctly to aggregate values for this account.	Account type incorrectly classified, hence driving the harvest failure during account update. For example, an investment account is classified as a checking account type.	Partner should send UserAcctModRq with correct account attributes and then send HarvestRq
300	Login failure	Please verify that the financial institution username/password that you entered is correct. If the login credentials are correct, please try again later.	This is a result of inability to establish a session due to login failure. This login failure is different from 301 (below) in that it excludes cases where the FI website displays an explicit message about any incorrect login credentials.	UserOFILoginCredentialsModRq followed by HarvestAddRq-AddMoreAcct
301	Login failure - invalid login credentials	We cannot login with username/password combination you provided. Please make sure the login information is correct.	The FI website displays a clear message that the login credentials are incorrect.	UserOFILoginCredentialsModRq followed by HarvestAddRq-AddMoreAcct

Error code	Error	Message	Description	Partner resolution action
302	Agreement page notification	Agreement page notification. Please log into your online account at this financial institution and follow the instructions presented by the financial institution.	This is typically caused when the FI site displays an intermediate page that cannot be skipped or there is some action that the user needs to take to proceed with online banking. It requires user intervention on the FI website.	User must clear the intermediate page by logging into the FI website. Then partner should send HarvestAddRq/HarvestRq.
303	Additional information required	Multi-factor authentication failure. Your financial institution website requires additional information to proceed.	The error is displayed when harvester detects a challenge question as part of MFA (multi-factor authentication).	HarvestAddRq-AddMoreAcct to answer the MFA question and pass the original RunID and session ID
304	Incorrect answer provided to the site key challenge question	Multi-factor authentication failure. Please make sure that you provide the right answer to the challenge question.	Displays after providing answer to the challenge question as part of MFA (multi-factor authentication) the FI site returns an identifiable message indicating that the answer was wrong.	HarvestAddRq-AddMoreAcct with FILoginAcctId only
305	Incorrect third parameter provided to the site	Invalid client or account identifier. Please make sure that you provide the correct client or account identifier.	This is applicable only for advisor access FIs and not for advisor aggregation FIs.	Treat similarly to 301.
306	Incorrect FI selection	Incorrect financial institution selection. Please select the correct financial institution to add your account.	Applies only to account setup process, where the user selects an incorrect FI for account addition. For example, advisors choosing Client Access FI to add. However, if a FI moves some of its accounts to another FI then it could occur during update.	Delete the account (FIDeleteRq) that caused this error. HarvestAddRq-AddNewAcct
307	Account locked	Your account is locked at this financial institution. Please contact your financial institution for more information.	Account is locked on FI website and the harvest process fails. User should contact financial institution to unlock the account.	User must call FI to unlock the logon. Then partner should send HarvestAddRq/HarvestRq.
311	Harvest failure	We are in the process of upgrading our product to rectify this problem. Please try again later.	Script failure during login process resulting in an add account or update failure. Fiserv will investigate and resolve this script failure. This is also a generic error code that is returned for all failures during login before getting assigned to an appropriate error code.	Temporary issue: Partner should send HarvestAddRq/HarvestRq again. If it still fails, then it may be a harvesting error. Partner should report the issue to Fiserv after several unsuccessful retries.
400	Database update failure	Database update failure. Please try updating again in a few minutes.	This is an internal error for AIIData database failures. This is typically caused by a database configuration issue and AIIData can usually fix it. This error is typically not specific to an FI script and is rare.	Temporary issue: Partner should send HarvestAddRq/HarvestRq again. If it still fails, there may be a harvesting error. Partner should report the issue to Fiserv after several unsuccessful retries.

Error code	Error	Message	Description	Partner resolution action
600	Institution on watch list	We cannot currently add your accounts because we are in the process of upgrading our support for this institution. This upgrade may take several more days. We will place this institution in your pending accounts list. Please try adding accounts again in a few days.	This error is encountered when a user tries to add an FI which is under development in AllData system. After the scripts are developed, the error code will be reset to enable user access.	Data Operations is in the process of developing the scripts for the FI. This error can be reported to Fiserv to escalate the completion of FI script development.
999	Internal Software Error – Unknown Error	Internal Software Error. Please try updating again in a few minutes.	This represents the residual category and is very minimal.	Temporary Issue - The partner needs to send HarvestAddRq/HarvesRq again. If it still fails, then it may be a harvesting error. Partner should report the issue to Fiserv after several unsuccessful retries.

Appendix C: MFA Image-Retrieving URL

The partner must call an image-retrieving URL in the HTML tag to get the image from the AllData system. The base64-encoded image is transmitted to the client browser via HTTPS connection to be rendered in the partner UI application. The partner can also request that Fiserv enable transmitting a base64-encoded version of the image directly in the web service response.

Details of Fiserv image-retrieving URL:

Prod URL:

```

```

UAT URL:

```

```

Method Type: Get

Input Param: imageId

Response:

- On Success: The base64-encoded image
- On Failure: Null stream is passed to the partner.

Note: The image will be deleted from the AllData system for that image ID once invoked using the image-retrieving URL.

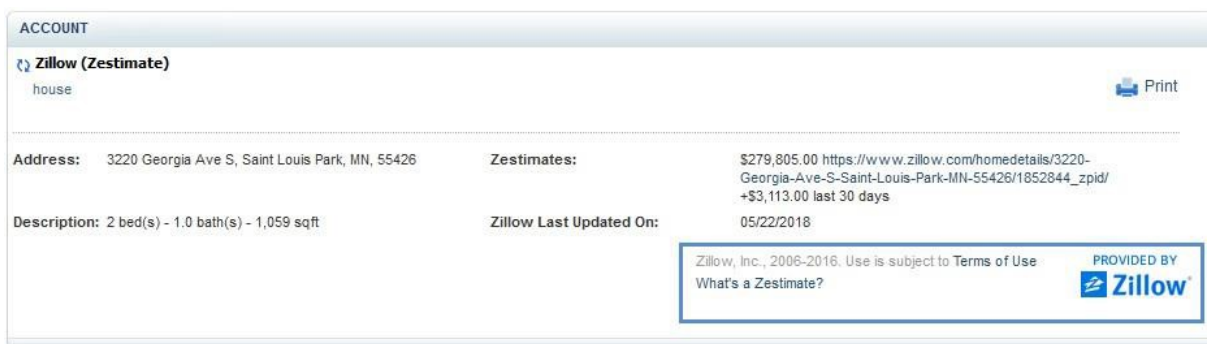
Appendix D: Zillow (Zestimate) – Terms of Use

To use Zillow (“Zestimate”) FI to obtain the property value, Fiserv and Zillow have agreed to brand Zillow as the information provider wherever the Zillow FI and Zestimate information appear. Clients that use the Zillow FI information through Fiserv should do the Zillow branding in their UI when displaying the Zillow FI information.

The below information should appear in the UI when displaying the Zillow (Zestimate) information.

- Zillow logo:
http://www.zillow.com/widgets/GetVersionedResource.htm?path=/static/logos/Zillowlogo_150x40.gif
- Zillow “Terms of Use” link: <http://www.zillow.com/corp/Terms.htm>
- “What’s a Zestimate” link: <http://www.zillow.com/zestimate/>

Sample:



The sample shows a Zillow widget within a light blue header labeled 'ACCOUNT'. Below the header, the text 'Zillow (Zestimate)' is followed by a small house icon and the word 'house'. A 'Print' button is in the top right. The main content area is divided into two columns. The left column contains 'Address: 3220 Georgia Ave S, Saint Louis Park, MN, 55426' and 'Description: 2 bed(s) - 1.0 bath(s) - 1,059 sqft'. The right column contains 'Zestimates: \$279,805.00' with a URL, '+\$3,113.00 last 30 days', and 'Zillow Last Updated On: 05/22/2018'. At the bottom right, there is a box with the text 'Zillow, Inc., 2006-2016. Use is subject to Terms of Use' and 'What's a Zestimate?', alongside a 'PROVIDED BY Zillow' logo.

ACCOUNT	
Zillow (Zestimate) house Print	
Address: 3220 Georgia Ave S, Saint Louis Park, MN, 55426	Zestimates: \$279,805.00 https://www.zillow.com/homedetails/3220-Gorgia-Ave-S-Saint-Louis-Park-MN-55426/1852844_zpid/ +\$3,113.00 last 30 days
Description: 2 bed(s) - 1.0 bath(s) - 1,059 sqft	Zillow Last Updated On: 05/22/2018
<div>Zillow, Inc., 2006-2016. Use is subject to Terms of Use What's a Zestimate?</div> <div>PROVIDED BY </div>	