FINANCIAL™ DATA EXCHANGE

Foundational Requirements for Data Providers

Legal Notice

FDX is a standards body and adopts this Foundational Requirements for Data Providers for general use among industry stakeholders. Many of the terms, however, are subject to additional guidance under prevailing laws, industry norms, and/or governmental regulations. While referencing certain laws that may be applicable, readers, users, members, or any other parties should seek legal advice of counsel relating to their particular practices and applicable laws in the jurisdictions where they do business. See FDX's complete Legal Disclaimer located at http://www.financialdataexchange.org for other applicable disclaimers.

Revision History

Document Version	Notes	Date
1.0	Initial Document Release This document was created as a result of FDX RFC 0039 and incorporates the full contents of the RFC for public release.	**December 2020**

Contents

Introduction

Foundational Requirements have been established to ensure that FDX Certification applicants meet standards that protect Data Recipients and ensure proper interoperability as expected by the End Users. While these requirements vary based on the provider's level of use, they exist for all in the areas of **Functional** and **Non-Functional** requirements. Functional requirements refer to the successful interaction between *end user authorization*, the *FDX API*, *minimum call compliance*, and *data*. Non Functional requirements refer to *availability*, *performance*, *scalability*, and *security*.

This document is part of a series of certification documents and explains foundational requirements. Note that meeting these requirements alone is not sufficient for certification. Please refer to the Certification Model for an understanding of methodology and testing.

1.1 Terminology

Please refer to Taxonomy for a complete definition of terms used in this document. The relationship between data provider, recipient, agent, and end user are key concepts throughout FDX implementation.

Specific to this process, the term **Certification Case** refers to a use case scenario on a pre-production environment that is only applicable to certification.

1.2 Scope

The scope includes the following areas which are described in detail in the Requirements section:

Functional Requirements

- *User Experience (UX)/End User Authorization:*
    - Authorization End User URL
    - Responsive pages
    - Authorization Code
    - State parameter
    - FDX User Experience - Providers should comply with recommendations specified by the FDX User Experience and Consent document
- *API Endpoints*
    - Authorization endpoint

    - FDX Data API endpoints
- *Minimum Call Compliance*
    - FDX Maintenance endpoints
    - Data API
- *Test Data*

Non-Functional Requirements (Formerly Operational)

- *Availability*
- *Performance (Response time and SLAs)*
- *Scalability*
    - Scalability will not be certified in pre-production
    - Providers are required to provision production capacity to handle requester requirements, provided that the requester follows usage best practices (see section 2.2.3)
- *Security*
    - End user authentication
    - End user authorization
    - API connection security for authentication, authorization, and data transfers

1.3 Prerequisites

An FDX certification applicant must meet these prerequisites to be considered for certification:

- A pre-production environment must be available with the following functionality that is equivalent to production:
    - Test profiles for all the supported certification cases and account types, with the supported data set (see “Representative Test Data” in Section 2.1 below)
    - Support for the OAuth 2.0 User Experience for relevant screens and substantially similar content
    - Support for all relevant error codes
    - Support for all relevant end-points
- The certification applicant must employ the latest published FDX API (currently v4.0) or current supported version within the last 12 months (v3.0). All base URI should include the version that is implemented.
    - *Note: OFX is not in scope for this document*

- Documentation for FiAttributes when utilized

As a general practice, the pre-production environment should maintain configuration and test data prescribed within the scope for requirements. Availability and other Non-Functional requirements for the pre-production environment are listed in the “Non-Functional Requirements” section below.

2.0 Requirements

This section describes all of the Functional and Non-Functional requirements. For reference, each requirement is numbered by function. For example, an End User requirement would be *EU 1* while an availability requirement would be *AV 3*. Any available best practices for each functional area are included.

2.1 Functional Requirements

The following set of requirements should be met in order to create functional tests with the supported FDX API version.

Requirement #	Requirement	Notes
FR 1: Setup	All necessary setup tasks should take place before beginning a compliance test, such as Mutual TLS, IP Filtering, client ID/certificate or ID/Secret, Callback URI, etc.	
FR 2: End User Authorization/User Experience	Establish the URL and required parameters for end user authorization. Applicants must be able to return the 'state' parameter used. The current recommendation is to create a v4 UUID style 'state' parameter of 36 character length.	This character length thwarts Cross Site Request Forgery (replays)
FR 3: API Tests of Authorization and Data	Perform schema validation using a compliant set of JSON request requirements and responses Test authorization parameters (list expiry for Authorization Code, Access Token, and Refresh Tokens) Test authorization functionality (exchange Authorization Code for Access Token, Refresh Token, and Revoke Token)	
FR 4: Representative Test data	Test data should strive to cover 80% of the account and transaction types that are found in the production environment. Test data should cover 90 days of transactions with representative transactions for the supported account types (credits, debits) from the day the Compliance Certification is performed. Note that some certification cases may require longer lookback ranges Test data should include all FiAttributes utilized by the provider	

2.1.1 Functionality Best Practices

- **Test Environment Availability** - Every effort shall be made by the Provider to guarantee a minimum of 'business hours' uninterrupted Test Environment access. Both stability

(predictable behavior) and availability of the Test Environment is a critical factor to meet requirements inside the allotted time frame.

- **Dedicated test data** - The Provider must make the necessary arrangements for the test data provided (online profiles - username & passwords, account data, and transactions, statements, documents, etc.) to be exclusively dedicated to the process during the specified time.
- **Environment Stability** - Provider released API and End User Authorization pages must be already or soon to be released for public use. Any issues (lack of documentation, missing/incomplete data, invalid JSON response, etc.) must be resolved within a reasonable period to allow any testing process to complete within the allotted time.
- **API Documentation** - A swagger file, OpenAPI Specification v3.0, or "Developer" portal detailing specific Provider implementation details must be provided and updated as needed during testing. This must include not only sample JSON responses, but a complete list of the HTTP Errors and all Provider specific error codes and error messages for each fault condition

2.2 Non-Functional Requirements

Non-Functional requirements comprise availability, response and operational performance, scalability, and security.

2.2.1 Provider Availability Requirements

Requirement #	Requirement	Notes
AV 1	Providers will have a distinct error code which specifies that the service is under maintenance for the following interfaces - See 3.1.7 Error for additional information Authorization/Consent from UI Flow The Provider Consent User Interface must be able to message the user with an error code specifying that the site is under maintenance and ask the user to return later Authorization Service Resource Service/API	
AV 2	If a provider must perform emergency maintenance (outside of the standard schedule) due to a service risk, the Provider should follow these guidelines: If the system is in a severely degraded state with failure rates or performance issues impacting the service-level agreement (SLA), or on the verge of becoming severely degraded, then the emergency maintenance should be executed immediately If the system is not operating outside of availability and/or performance SLAs and can continue to sustain until off peak hours, perform the maintenance from 1 AM to 5 AM ET Announce the emergency maintenance to providers	1AM to 5AM is a suggested maintenance window for US Providers as it represents the lowest consumer usage period during a week for Banking and Financial applications. A different maintenance period may be used if necessary, as long as it is published for the benefit of any reliant parties.
AV 3	Providers will always measure the availability of their services and make every attempt to continually strive for the highest availability possible	*Availability issues will be resolved directly between Requesters and Providers for all Data Exchange Services.*

Availability targets are a measure of service up-time that occur in 1-minute increments over a monthly schedule. Providers will measure availability for three distinct services:

- Authorization - Consent from the User Interface Flow
- Authorization Service - OAuth 2.0 token exchange
- Resource Server/API - FDX API access to protected Financial data

Availability Calculation

The service availability % calculation is measured by dividing the total number of minutes of up-time by the number of minutes in a month. The service availability calculation excludes any minutes associated with planned maintenance downtime. Thus only unplanned downtime should be subtracted from total monthly minutes to calculate up-time.

Availability = (Total monthly minutes – Number of Unplanned Downtime Minutes) / Total monthly Minutes

As an example, the following chart represents availability percentages for various downtimes in a 30-day month.

"Availability	Downtime (minutes)
99%	432
99.9%	43
99.99%	4
99.999%	<< 1

2.2.1.1 Provider Availability Best Practices

- Providers should provide the following Service Levels for both Consent Flow and OAuth 2.0/API services:
    - 99.9% availability or better (a maximum of 43 minutes per month of unplanned downtime)
    - Providers will announce maintenance to Requesters within an agreed upon time period of notice, via agreed upon support channels (email communication is acceptable)

A highly available pre-production environment is important in promoting speed of development. General guidance for pre-production environments is as follows:

- Providers should provide support for Requesters during API development to service availability issues
- Communicate maintenance periods to Requesters

2.2.1.2 Requester and Data Agent Best Practices

Requesters should follow appropriate usage guidelines to ensure that they do not inhibit the up-time of the service:

- 1 AM to 5 AM is a suggested maintenance window for US Providers as it represents the lowest consumer usage period during a week for Banking and Financial applications. A different maintenance period may be used if necessary, as long as it is published for the benefit of any reliant parties.

- When a Provider announces a scheduled maintenance, the requester should suppress any scheduled account refresh traffic during the scheduled time period (1 AM to 5 AM ET on Sunday)
    - A Requester UI with Provider Access must be able to:
        - Automatically detect that the Provider UI/service is unavailable if there is no response due to maintenance
        - Message the user that the Provider UI/service is currently under maintenance if there is no response for this reason
        - Handle the scenario where the Provider UI servers a maintenance page (e.g. “System is currently under maintenance. Please come back later”)
    - Any scheduled/routine API account refreshes should be re-scheduled outside of the maintenance window
    - If scheduled Authorization/Resource Server API requests are made to a Provider during the scheduled maintenance window that result in failure, the Requester should:
        - Schedule retries to occur at least 4 hours into the future
        - Ensure that traffic is spread out evenly over a window that matches the maintenance window. This is necessary to avoid a volume spike following a maintenance window.
    - Requesters should automatically handle any error conditions that occur during the standard maintenance window, regardless of a Provider announcement
- If the site is unexpectedly down, the retry for the services should be scheduled 4 or more hours into the future to avoid a situation where traffic is accelerated to provider services which are unavailable

2.2.2 Response Time

Service performance is not only dependent on the efficiency and scalability of the provider implementation, but also on appropriate usage from the requesters. This section provides the requirements for both Providers and Requesters for delivery of a great user experience with optimal response times for end customers.

Data API call responses shall be as good or better than the Data Provider guarantee on their End User secure access applications, such as Online Banking.

2.2.2.1 Provider Requirements – Response Time Expectations

The following table comprise the guidelines for transaction performance:

Service/API	Production Average Target	Production SLA Target	Pre-production Average. Target	Pre-production SLA Target
RT 1: Authorization/Consent Flow UI Redirect Landing page	1 Second	2 seconds	1 Seconds	2 Seconds
RT 2: Authorization/Consent Flow UI Page Views	1 Second	2 Seconds	1 Seconds	2 Seconds
RT 3: Authorization Service Token Exchange - Authorization Code	0.2 Seconds	0.5 Second	0.2 Second	0.5 Seconds
RT 4: Authorization Service Token Exchange - Refresh Token	0.2 Seconds	0.5 Second	0.2 Second	0.5 Seconds
RT 5: /accounts & /accounts/{accountId} (no Transactions)	0.2 Seconds	0.5 Second	0.2 Second	0.5 Seconds
RT 6: /accounts/{accountId}/transactions – 2 years	5 Seconds	10 Seconds	5 Seconds	10 Seconds
RT 7: /accounts/{accountId}/transactions – 90 Days	0.5 Seconds	1 Second	0.5 Seconds	1 Seconds
RT 8: /accounts/{accountId}/transactions – 30 Days	0.3 Seconds	1 Second	0.3 Second	1 Seconds
RT 9: /accounts/{accountId}/transactions – 7 Days	0.2 Seconds	0.5 Second	0.2 Second	0.5 Seconds
RT 10: RT 10 = /accounts/{accountId}/statements – List of Statements	0.2 Seconds	0.5 Second	0.2 Second	0.5 Seconds
RT 11: RT 11 = /accounts/{accountId}/statements/{statementId} (PDF / Statement Image)	5 Seconds	10 Seconds	5 Seconds	10 Seconds

2.2.2.2 Provider Best Practices for Transactions

- First time user experience will require a transaction retrieval that must span the minimum age of transactions required for the certification case. A typical historical transaction range is between 90 days to 2 years
- Daily account refresh requests are highly important for the PFM certification case. Refresh requests may occur for a transaction overlap of 7 days from the last successful refresh. If the last successful refresh was 2 days in the past, the transaction start date would be 9 days in the past and the end date would be the current date

- Providers should index (i.e. contain a unique and durable transactionId) and return transactions based on when the transaction was last updated. This will ensure that searches will be most efficient in returning the optimal number of transactions
- Providers must specify the format and guarantee consistency across all timestamps. Date timestamps should be ISO 8601 compliant and always contain a time component that is UTC or time zone specific (ex: “2020-03-23T16:51:52.769Z”)
- Providers should disclose what date is used for the transaction range (Last Updated, Posted Date, Transaction Date)
- Providers must specify the behavior for the presence or absence of startTime and/or endTime:

statTime	endTime	Response
null	null	last (most recent) X days of transactions
null	dateB	X days prior to [dateB] inclusive [dateB] (start date on or after **lookback**)
dateA	null	[dateA] thru today (**lookback** - X days)
dateA	dateB	[dateA] through [dateB] (not to exceed **Max Date Range** - X days)

- If the transaction range is based on the Posted Date, then the Consumer could execute a weekly or monthly true-up. For example, asking for 30 days of transactions to in order to receive transactions older than 7 days that may have changed or come in late

2.2.2.3 Performance Measurement

Providers and Requesters should conduct their own performance measurements, both during development and in production. Performance should be measured on the Application server. This encompasses all aspects of service processing, including application and database system processing, as well as network interfaces. Performance can be measured through Access log processing systems or APM systems.

Requester and producer measurements are expected to be within 10-50 ms, depending on data center proximity for requests with a "reasonable" data response (less than 100KB). If the discrepancy is larger, requesters and providers should work together to reconcile the issue.

2.2.3 Operational Metrics and Best Practices

2.2.3.1 Operational Metrics Definitions

Operational metrics are a function of the following items:

Metric #	Requirement	Notes
OM 1	The volume of end users consuming services	
OM 2	The refresh frequency requirement of a specific certification case	

OM 3	The duration/time period for which offline refresh activities take place
OM 4	The real-time API call volume driven by end customers

2.2.3.2 Operational Best Practices

The following best practices establish mutual expectations for the Data Provider, Recipient, and Agent. This allows end customers to receive expected services from Recipients and Agents. These best practices will also give Data Provider guidelines to protect the API's from abuse.

- Data Providers should make reasonable efforts to make API data, availability, and performance as good or better as other End User channels (e.g. online banking, mobile banking).
    - Data Providers are responsible for providing their services to all their mutual End Users with data refreshed at least once in a 24 hour period
    - End Users grant consent to allow the Data Recipient/Agent to request data on their behalf to power the Data recipients services. Depending on the given Use Case, those data requests will be more frequent than a single request during a 24-hour.
- Controls may be put into place between the Provider and the Agent to prevent abuse of the API caused by an excessive number of requests by a Data Recipient. As an example:
    - A Data Provider may place a limit requirement and/or preferred refresh timing on batch requests if there are infrastructure issues related to API access. Configured limits should be clearly documented in the Data Provider's Registry/Directory for all Data Recipient and Agents to view
    - If temporary limits are necessary, they should be communicated to Recipients/Aggregators, both via the API itself and direct communication channels.
    - Providers who choose to place limits, should reassess quarterly as possible volume changes or have a process in place to increase limits for Data Agents as needed.
- Data Agents and Data Providers should work together to optimize traffic over the API wherever possible, including but not limited to using "lastActivityDate" (prevent unnecessary calls to retrieve transaction history), webhooks notifications when new transactional data for a given end customer is available, etc.

2.2.4 Security Requirements

Control Considerations for Consumer Financial Account Aggregation Services (referred to henceforth as Control Considerations, or CC) provides a reference architecture for an improved method of data exchange between data recipients, data access platforms, and data providers. While the documented guidelines cover a broad range of security concerns covered by the six

domains - data, software, network, physical, operational, and supplier security - these foundational requirements reference the following sections in the document -

- Requirements section, Financial Institutions Actions section, and
- Financial Aggregation Security Model Standard Specification.

2.2.4.1 Authorization Security Requirements

Requirement #	Requirement	Required/Optional	Notes
AS 1	Delegated user authorizations are performed using the OAuth 2.0 framework	R	
AS 2	The OAuth 2.0 framework-specific security considerations must be followed	R	
AS 3	The Provider must require authorization to be re-validated periodically	R	
AS 4	The Provider must have the ability to, and be responsible for, defining the duration of authorization	R	
AS 5	The Provider must also support the ability for the customer to define the duration of authorization	O	There is currently no API in the specification to pass the consent duration. This may be added to a future version of the API and would become required at that point.
AS 6	The Provider must enforce the lower of the customer-defined or provider-defined duration of authorization	O	There is currently no API in the specification to pass the consent duration. This may be added to a future version of the API and would become required at that point.
AS 7	The Provider must require re-authentication to re-validate an authorization	R	
AS 8	The Provider must support the ability for end users to revoke authorization	R	
AS 9	The Provider must have the ability to revoke authorization	R	

AS 10	The solution must support the ability for the Data Access Platform to revoke authorization	O	There is currently no API in the specification to revoke an authorization. This will be added to a future version of the API and become required at that point.
AS 11	The solution must support ability for the Data Recipient to revoke authorization	O	There is currently no API in the specification to revoke an authorization. This will be added to a future version of the API and become required at that point.

2.2.4.2 Authentication Security Requirements

Requirement #	Requirement	Required/Optional	Notes
ID 1	Provide federated user authentication that is interoperable with OpenID Connect 1.0 (OIDC 1.0)	R	
ID 2	The Provider is responsible for authenticating a customer directly without including intermediaries, e.g., aggregators or strong customer authentication (SCA), which may require retrieving a one-time password (OTP) that may not be accessible to the Certification staff	R	
ID 3	Credentials must not be shared with any third party, e.g., aggregators, or recipient apps	R	
ID 4	The Provider will give authorization to a data agent or data recipient based on successful customer authentication	R	
ID 5	Support multi-factor authentication between the end user and the data provider or, For de-coupled authentication scenarios, the provider must support: Fast Identity Online 1.2 (FIDO) Universal Authentication Framework (UAF) Client Initiated Backchannel Authentication (CIBA)	R	
ID 6	The Data Provider must have the ability to generate unique access tokens	R	
ID 7	The Data Provider will provide an access token to the data agent or data recipient	R	
ID 8	OIDC, FIDO and CIBA specific security considerations must be followed	R	

2.2.4.3 Connectivity Security Requirements

Requirement #	Requirement	Required/Optional	Notes
CSR 1	All connections within the access control and API framework must use HTTPS 1.1 for the following: Message exchanges between the API and OAuth 2.0 authorization server authentications Token requests Resource server access of the Provider API	R	
CSR 2	Use of the latest secure, stable and generally adopted version of TLS is required, per NIST and IETF recommendations	R	

References

- Control Considerations for Consumer Financial Account Aggregation Services v3.1, section "Financial Institutions Actions"
- https://openid.net/specs/openid-connect-core-1_0.html#Security
- https://fidoalliance.org/specs/fido-u2f-v1.2-ps-20170411/fido-security-ref-v1.2-ps-20170411.html
- https://openid.net/specs/openid-client-initiated-backchannel-authentication-core-1_0.html#rfc.section.14
- https://oauth.net/2/
- https://tools.ietf.org/html/rfc6749#section-10

3.0 Certification Types

This section describes the required certification types and endpoints. Note that certification cases should be evaluated as specified under the Certification Model Task Force section, making use of approved Testing Tooling Task Force.

During the certification period, the Provider will apply to one of the following API Functional certification Types (*Note that all certification types include the Common Certification Cases below*):

- Financial Data Provider (Read only)
- Tax Forms Data Provider *(Read only or Read and write)*
- Money Movement (Transfer) Provider
- Full Provider *(*Account Data, Tax, and Money Movement)

3.1 Common Certification Cases

The following certification cases are the required basis for all Provider Certification Types. The Common Certification cases are in the areas of Maintenance (M), Authorization (A), Customer Information (C), and both Authorization (AE) and Data Errors (DE).

3.1.1 Certification Cases: Maintenance

M-1 /availability - This is a required call. Data Providers must provide a single endpoint that may be used as an 'FDX-heartbeat'; additional information (such as up coming or recurring planned outages) is optional

M-2 /capability - This is a required call. Short of providing a full 'well-known endpoint' discovery service, Data Providers must at least provide a list of the endpoints that have been implemented and are ready for production use

3.1.2 Certification Cases: Authorization

A-1 /authorize - This is the end User Authorization flow. All Providers will need to provide the necessary setup and security information necessary to invoke the URL to be used. Once this Provider URL is invoke, all the end User experience is determined by the Provider content (Authentication, Authorization, Account Selection, Data clusters, Speed bump before returning to the client app, etc. - See the FDX User Experience and Consent section). Once the end User journey is completed, control returns to the calling client along with a single Authorization Code and any other client required information (e.g. 'state' parameter)

A-2 Token Exchange - This is the Provider's Authorization server endpoint used to exchanging the recently granted Authorization Code for a pair of tokens Access (aka 'short-lived') for Data retrieval and Refresh (aka 'long-lived') token

A-3 Token Refresh - This is the Provider's Authorization server endpoint to request a new (data) Access token for a new data request (ex: daily 'batch' refresh)

A-4 Revoke Token - This is the Provider's Authorization server endpoint to invalidate the Authorization from the client. End Users have as a main resource the Provider's own secure session Account access (online banking) application to manage previously granted authorizations

3.1.3 Certification Case: Current Customer Information

C-1 /customers/current - This required call returns additional information about the Customer who authorized the retrieval of financial data

3.1.4 Certification Cases: Errors

A series of "negative test cases" will be executed to elicit HTTP 40X, 429, and 501 error codes to capture both the code and message that the Data Provider returns for each of the error conditions:

3.1.4.1 Certification Cases: Authorization Errors

AE-1 and **AE-2** - Data Provider shall respond appropriately to all Authorization Errors which can result in HTTP error codes 400 (Bad Request) and 401 (Invalid Request)

3.1.4.2 Certification Cases: Data Errors

DE-1 through DE-8 - Data Provider shall respond appropriately to all Data Errors 400, 401, 404, 406, 429, and 501.

3.2 Financial Data Provider (Read only)

Meets all of the requirements for Common Certification Cases as well as:

3.2.1 Financial Data - Accounts (Level A)

- Get a (lightweight) list of accounts (D-1), and
- Get a (single) account with details (D-2), or
- Get a list of accounts (with details) (D-3)

3.2.2 Financial Data - Transactions (Level B)

- Get a list of transactions for an account (D-5 or D-6)

3.2.3 Financial Data - Accounts and Transactions (Level C)

- Get a list of accounts (with details) and transactions (D-4), or
- Get a list of transactions for an account (D-5 or D-6)

3.2.4 Financial Data - Accounts, Transactions, and Statements (Level D)

- All Level A, B, and
- Get a list of Statements (D-8) and
- Get a (single) Statement (D-9)

3.2.5 Financial Data - All (Accounts, Transactions, Images, and Statements) (Level E)

- All Level D, and
- Get a transaction image (D-7)

3.3 Tax Forms Data Provider (Level F)

Meets all of the requirements for Common Certification Cases as well as:

3.3.1 Certification Cases: Tax Data Provider (Read Only)

- Get a list of Tax Documents (TD-1)
- Get a specific Tax Document (by Tax Document ID) (TD-2)

3.3.2 Certification Cases: Tax Data Provider (Write)

- Create a new Tax Document (TD-3)
- Update a specific Tax Document (TD-4)

3.4 Money Movement (Transfers) Provider (Level G)

Meets all of the requirements for Common Certification Cases as well as:

3.4.1 Certification Cases: Start Transfer

Create a transfer between two end User accounts.

3.4.1 Certification Cases: Get Transfer Status

Get the status of a specific transfer (by Transfer ID) for a given end User.

3.5 Full Provider (Account Data, Tax, and Money Movement) (Level H)

Meets all of the requirements for Common Certification Cases as well as all ALL Certification Cases (A through G)