令和 6 年度

教育データ連携の実現に向けた実証調査研究

標準仕様による教育データ連携実証（OneRoster REST API 連携）

＜実証事業報告書＞

令和 7 年 3 月 29 日

株式会社内田洋行

＜　目　　次　＞

1.　実証の概要 1

1.1.　背景 1
1.2.　目的 2
1.3.　実証スケジュール 2

2.　OneRoster REST API の定義 3

2.1.　OneRoster REST API について 3
2.2.　Japan Profile について 3
2.3.　OneRoster REST API Japan Profile 相当の技術仕様 3
2.3.1.　OpenAPI ドキュメンテーション 3
2.3.2.　実装対象の API エンドポイントの定義について 4
2.3.3.　Japan Profile における項目の拡張 5
2.4.　API アクセスに対する認証・認可の制御 6

3.　対象システムへの実装 7

3.1.　校務支援システム 7
3.1.1.　概要 7
3.1.2.　校務支援システムへの実装 7
3.2.　学習支援システム 8
3.2.1.　概要 8
3.2.2.　学習支援システムへの実装 8
3.1.　対象システムの同士の接続 9
3.1.1.　テスト概要 9
3.1.2.　テスト内容と結果 9

4.　実装上の留意点および、規格の取扱いについて 10

4.1.　標準におけるクエリパラメータの取り扱い 10
4.1.1.　コンフォーマンステストにおけるクエリパラメータの取扱い 10
4.1.2.　Filter クエリパラメータの演算子について 10
4.2.　REST API における差分更新 11
4.3.　情報の一貫性について 11
4.3.1.　sourcedId の一貫性について 11
4.3.2.　データの出力順序について 12
4.4.　CSV と REST API におけるシステムの役割の差異について 12
4.5.　REST API エンドポイントの実装範囲について 13

5.　まとめと今後の課題 14

5.1.　OneRoster REST API Japan Profile の定義について 14
5.2.　REST API による連携実現のメリット 14
5.3.　標準化に向けて検討すべきこと 15

1.　実証の概要

1.1.　背景

校務支援システムと学習支援システムの間のデータ連携においては、主体情報（名簿情報）の連携を行うことを目的として、OneRoster Japan Profile を利用することが、学習 e ポータル標準モデルに定義されている。Japan Profile は、OneRoster の機能のうち、名簿情報の連携（Rostering Service）を対象とし、CSV フォーマット（CSV バインディング）を前提に作成されている。

Japan Profile は、ICT CONNECT21 の校務系-学習系情報連携 SWG での意見交換や、日本 1EdTech 協会の検討を経て作成された。本報告書執筆時点では、1EdTech コンソーシアムにおいて、標準化作業が進行しており、正式なプロファイルとして公開される予定となっている。

技術標準上は、OneRoster の CSV ファイルを学習系システムなどの外部システムへ連携する際の、出力・搬送手段は規定されていない。自治体等の情報セキュリティポリシーなどによって、校務支援システムを利用できるネットワークが、その他一般のネットワークから分離されていることを前提と考えると、運用上、ファイルベースでの搬送手段を確保することが必要とされた。

一方で、現時点においては、教育情報セキュリティポリシーガイドラインにおいては、校務支援システムを含む各種システムに、境界防御型のセキュリティを必須としておらず、ネットワーク分離は要求されていない。このような環境のもと、次世代の校務支援システムがパブリッククラウド環境への移行が進行することを念頭におくと、OneRoster REST API の利用により、校務支援システムと学習支援システムが直接通信することを前提とした連携方式の導入が可能となる。

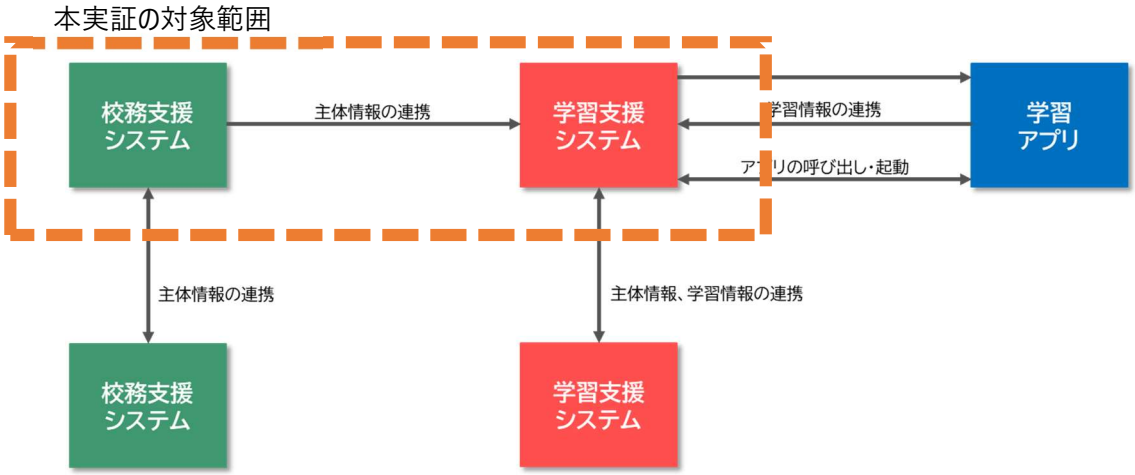

図 1-1　各システムの相互連携イメージと本実証の対象範囲[1]

[1] 図は「令和 4 年度教育関連データのデータ連携の実現に向けた実証調査研究 成果報告書」より引用

1.2. 目的

本実証では、OneRoster REST API の利用可能性や特性について調査検証を行うことを目的とし、以下の内容に取り組んだ。

CSV Japan Profile と同様のユースケース、同等の情報量を扱うことができることを前提として、OneRoster REST API における、Japan Profile 相当の技術仕様を定義する。

定義した Japan Profile 相当の REST API を校務支援システム（情報出力）と学習支援システム（情報入力）に実装した上で、それぞれ、1EdTech コンソーシアムの OneRoster REST API におけるコンフォーマンステスト（適合性テスト）を実施し、合格できる状態とする。

校務支援システムと学習支援システムの間の連携を実施し、実装上の課題などを整理する。

1.3. 実証スケジュール

本実証のスケジュールを図 1-2　本実証のスケジュール図 1-2 に示す。

校務支援システムに対しては、すでに REST API の実装は行われていたため、本実証では、コンフォーマンステストに関わる対応のみを実施した。

本実証では、システムに対する実装及び相互接続を行うだけでは無く、REST API の技術標準に対する調査も実施した。標準文書に記載されている内容の調査だけでは無く、日本 1EdTech 協会の支援や情報提供を受けながら、1EdTech コンソーシアムに対する問い合わせ等も実施している。この過程で、コンフォーマンステストの仕様と、実装の意図等も判明しており、本報告書の内容に反映している。

項目		10月	11月	12月	1月	2月	3月
校務支援システム	コンフォーマンステスト対応						
学習支援システム	設計						
	REST API実装・調整						
	コンフォーマンステスト対応						
相互接続	システム間相互接続						
調査・とりまとめ	REST APIに関する調査						
	成果とりまとめ						

図 1-2　本実証のスケジュール

2. OneRoster REST API の定義

2.1. OneRoster REST API について

OneRoster REST API は、REST API による通信を行うための OneRoster 仕様であり、HTTP による通信によって、JSON 形式のデータのやり取りを行うものである。

技術標準上は、サービス要求を受け付け、情報を送信する側を Provider と呼び、サービスを呼び出して情報を受信する側を Consumer と呼ぶ。Provider と Consumer の関係は、サービス構成上の位置づけを示すものであり、必ずしも特定の利用目的のシステムの存在を想定しているものではない。

本実証においては、校務支援システムを Provider として、学習支援システム（学習 e ポータル）を Consumer として実装を行っている。

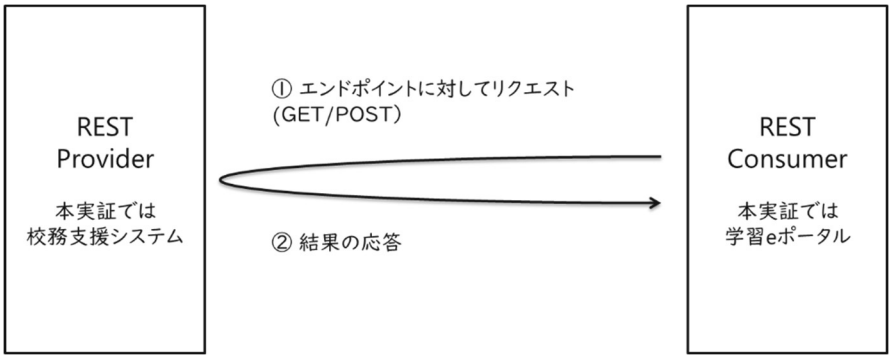

図 2-1　OneRoster REST API の構成

2.2. Japan Profile について

本実証で対象とする OneRoster REST API については、現在の Japan Profile（CSV フォーマット）のユースケースや仕様と同等のものを想定する。そのため、実装対象の仕様は、OneRoster における名簿連携（Rostering Service）を前提に、REST API バインディングにおける Japan Profile 相当の機能を想定したものを提案する。現時点では、Japan Profile における REST API の位置付けは存在していないため、Japan Profile の候補となる仕様として位置付ける。

2.3. OneRoster REST API Japan Profile 相当の技術仕様

2.3.1. OpenAPI ドキュメンテーション

本実証において定義および実装を行った技術仕様について、Open API によるドキュメントを成果物として公開する。OpenAPI により、API 仕様の全体を確認でき、コードの自動生成等を行うツールが利用可能となっている。

- OneRosterJapanProfile_v1.2_openapi.yaml （OpenAPI の仕様が定義されたファイル）
- OneRosterJapanProfile_v1.2_openapi.html （上記より生成した API ドキュメント）

参考：OpenAPI Generator

Open API Generator（https://openapi-generator.tech/）は、Open API によって定義された Specification ファイルから、コードテンプレートを自動生成するためのツールである。これにより、API サーバ部分のひな形が利用可能となる。100 以上の言語・フレームワークに対応したコードを生成可能である。

API 定義から OpenAPI generator で生成できるサーバサイドのコードは、API 処理部分のみで、データを抽出するロジックなどは含まれていない。完全な API サーバとして利用するためにはロジック部分を追加する必要があるが、API Specification を利用することで、開発効率を向上させることができる。

2.3.2. 実装対象の API エンドポイントの定義について

OneRoster の Rostering Service に定義のある API エンドポイントのうち、本実証においては、表 2-1 に示す 24 の API エンドポイントを実装対象として定義した。

OneRoster 標準において、必須の仕様であり、かつコンフォーマンステストの対象となる API エンドポイントが 22 存在している。これらの実装を行うことで、Japan Profile ではない、通常の OneRoster との互換性を維持することができる。

また、REST API において必須ではないものの、`/demographics` と、`/demographics/{sourcedId}` の 2 つのエンドポイントの実装を行った。これは、OneRoster CSV Japan Profile に、同様のデータ項目を扱う、demographics.csv が含まれていることから、この内容と整合性を保つために、実装対象としたものである。

その他のエンドポイントについては、Japan Profile の現在のユースケースに照らして必須ではないと整理し、実装対象には含めていない。

表 2-1　実装対象の API エンドポイントの定義

No.	API エンドポイント	実証対象	備考
1	/academicSessions	対象	仕様上必須
2	/academicSessions/{sourcedId}	対象	仕様上必須
3	/terms	対象	仕様上必須
4	/terms/{sourcedId}	対象	仕様上必須
5	/schools/{schoolSourcedId}/terms		
6	/gradingPeriods	対象	仕様上必須
7	/gradingPeriods/{sourcedId}	対象	仕様上必須
8	/terms/{termSourcedId}/gradingPeriods		
9	/classes	対象	仕様上必須
10	/classes/{sourcedId}	対象	仕様上必須
11	/courses/{courseSourcedId}/classes		
12	/schools/{schoolSourcedId}/classes		
13	/students/{studentSourcedId}/classes		
14	/teachers/{teacherSourcedId}/classes		
15	/terms/{termSourcedId}/classes		

No.	API エンドポイント	実証対象	備考
16	/users/{userSourcedId}/classes		
17	/courses	対象	仕様上必須
18	/courses/{sourcedId}	対象	仕様上必須
19	/schools/{schoolSourcedId}/courses		
20	/demographics	対象	CSV Japan Profile に項目が存在
21	/demographics/{sourcedId}	対象	CSV Japan Profile に項目が存在
22	/enrollments	対象	仕様上必須
23	/enrollments/{sourcedId}	対象	仕様上必須
24	/schools/{schoolSourcedId}/classes/{classSourcedId}/enrollments		
25	/schools/{schoolSourcedId}/enrollments		
26	/orgs	対象	仕様上必須
27	/orgs/{sourcedId}	対象	仕様上必須
28	/schools	対象	仕様上必須
29	/schools/{sourcedId}	対象	仕様上必須
30	/users	対象	仕様上必須
31	/users/{sourcedId}	対象	仕様上必須
32	/students	対象	仕様上必須
33	/students/{sourcedId}	対象	仕様上必須
34	/classes/{classSourcedId}/students		
35	/schools/{schoolSourcedId}/classes/{classSourcedId}/students		
36	/schools/{schoolSourcedId}/students		
37	/teachers	対象	仕様上必須
38	/teachers/{sourcedId}	対象	仕様上必須
39	/classes/{classSourcedId}/teachers		
40	/schools/{schoolSourcedId}/classes/{classSourcedId}/teachers		
41	/schools/{schoolSourcedId}/teachers		

2.3.3. Japan Profile における項目の拡張

CSV Japan Profile においては、日本の学校における名簿情報として、フリガナなどの情報項目（プロパティ）の追加が行われている。これらについては、`metadata.jp.`を接頭語とする階層型の構造を取っているが、これは、OneRoster 標準において定められている、拡張情報を格納するための規定に則ったものである。そのため、Japan Profile と、通常の OneRoster との互換性を維持した状態を維持することができる。

REST API にも同様の各情報に関する定義があるため、これを利用し、表 2-2 のように、CSV と同等の拡張を反映する。拡張した項目の内容、意味その他の規定については、CSV Japan Profile からの変更は行わないものとする。

表 2-2　Japan Profile における項目の拡張

CSV Japan Profile ファイル名	プロパティ	項目内容	対象の REST エンドポイント
Classes.csv	metadata.jp.specialNeeds	特別支援学級フラグ	/classes /classes/{sourcedId}
Enrollments.csv	metadata.jp.ShussekiNo	出席番号	/enrollments /enrollments/{sourcedId}
	metadata.jp.PublicFlg	公開フラグ	
Users.csv	metadata.jp.kanaGivenName	フリガナ名	/users /users/{sourcedId} /students /students/{sourcedId} /teachers /teachers/{sourcedId}
	metadata.jp.kanaFamilyName	フリガナ氏	
	metadata.jp.kanaMiddleName	フリガナミドルネーム	
	metadata.jp.homeClass	在籍クラス	
	metadata.jp.kanaPreferredFamilyName	フリガナ表示名	
	metadata.jp.kanaPreferredGivenName	フリガナ表示氏	
	metadata.jp.kanaPreferredMiddleName	フリガナ表示ミドルネーム	

2.4.　API アクセスに対する認証・認可の制御

API アクセスの認証および認可の制御は、REST API の標準仕様に準拠している。OAuth 2.0 Client Credentials Grant（OAuth 2.0 CC）と、Scope による制御を行うことを前提としている。

OAuth 2.0 CC では、クライアントアプリケーションが自身の資格情報でアクセストークンを取得し、API リクエストに使用する。

Scope についても標準仕様に準拠し、以下のスコープを利用することで、アクセス可能なリソースと操作の範囲を制御している。

https://purl.imsglobal.org/spec/or/v1p2/scope/roster.readonly

https://purl.imsglobal.org/spec/or/v1p2/scope/roster-core.readonly

https://purl.imsglobal.org/spec/or/v1p2/scope/roster-demographics.readonly

3.　対象システムへの実装

本実証において、校務支援システム、学習支援システムに対して、2 章で定義した、Japan Profile 相当の OneRoster REST API の実装を行った。いずれも 1EdTech コンソーシアムのコンフォーマンステストに合格しており、互換性に問題が無いことを確認した。

3.1.　校務支援システム

3.1.1.　概要

製品・サービス名：デジタル校務

ベンダー名：株式会社内田洋行

コンフォーマンステスト合格・認証日：2024 年 10 月 7 日

図 3-1　本報告書執筆時点における校務支援システムの認証状況
（1 EdTech TrustEd Apps Directory）

3.1.2.　校務支援システムへの実装

校務支援システムにおける OneRoster REST API 実装においては、2 章 において定義した API エンドポイントおよび、プロパティの拡張を実装した。

3.2. 学習支援システム

3.2.1. 概要

製品・サービス名：L-Gate

ベンダー名：株式会社内田洋行

コンフォーマンステスト合格・認証日：2025 年 2 月 27 日

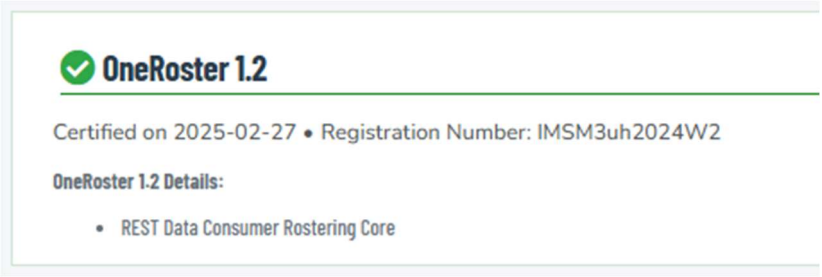

図 3-2 1 本報告書執筆時点における学習支援システムの認証状況
（1 EdTech TrustEd Apps Directory）

3.2.2. 学習支援システムへの実装

学習 e ポータル標準における、CSV での想定ユースケースに基づき、全件更新（Bulk）によるデータ更新を想定し、表 3-1 に示す、6 のエンドポイントに対するサービス呼び出しの実装を行った。

Provider とは異なり、Consumer は利用するエンドポイントが 1 つでもコンフォーマンステストの実施および合格が可能である。

表 3-1 Consumer が呼び出す API エンドポイントおよびサービス呼び出し

API エンドポイント	サービスコール	備考
/academicSessions	getAllAcademicSessions	全ての学期を取得
/classes	getAllClasses	全てのクラスを取得
/courses	getAllCourses	全てのコースを取得
/enrollments	getAllEnrollments	全ての所属情報を取得
/orgs	getAllOrgs	全ての組織を取得
/users	getAllUsers	全てのユーザーを取得

また、今回の実装では、一度に全件のデータ取得を行うのではなく、100 件ずつ、分割してデータ取得を行うことを目的として、以下のクエリパラメータの実装も行った。

表 3-2 Consumer が利用するクエリパラメータ

クエリパラメータ	備考
limit	取得データの件数（上限）を指定
offset	取得データのページングを行う
sort	取得データの並べ替えを行う
orderBy	取得データの並べ替え順序を指定

3.1. 対象システムの同士の接続

本実証において、対象システム同士の接続テストを以下の通り実施し、接続に成功した。

3.1.1. テスト概要

テスト実施期間

2 月 4 日～2 月 10 日

想定ユースケース

学習支援システムから校務支援システムに対して、名簿情報の全件取得を行う。

取得した名簿に基づいて、情報の更新および新規追加を行う。

使用データの概要

校務支援システムにダミーデータ（名簿）を登録し利用した。

3.1.2. テスト内容と結果

テスト 1：全件取得

校務支援システムのデータを各エンドポイントから全件取得するため、以下のリクエストを実施した。

表 3-3 テスト 1：全件取得

No.	取得リクエスト	結果
1	GET /academicSessions	合格
2	GET /classes	合格
3	GET /courses	合格
4	GET /enrollments	合格
5	GET /orgs	合格
6	GET /users	合格

テスト 2：100 件ずつのデータ取得

校務支援システムの各エンドポイントから、100 件ずつデータを取得するため、以下のリクエストを実施してデータを取得した。No.5 のリクエストでは、100 件以上のデータが登録されていた、enrollments に対して、offset クエリパラメータを利用して、101 件目から開始するデータの取得も行うことができた。

表 3-4 テスト 2：100 件ずつのデータ取得

No.	取得リクエスト	結果
1	GET /academicSessions?limit=100&sort=dateLastModifed&orderBy=asc	合格
2	GET /classes?limit=100&sort=dateLastModifed&orderBy=asc	合格
3	GET /courses?limit=100&sort=dateLastModifed&orderBy=asc	合格
4	GET /enrollments?limit=100&sort=dateLastModifed&orderBy=asc	合格
5	GET /enrollments?offset=100&limit=100&sort=dateLastModifed&orderBy=asc	合格
6	GET /orgs?limit=100&sort=dateLastModifed&orderBy=asc	合格
7	GET /users?limit=100&sort=dateLastModifed&orderBy=asc	合格

4. 実装上の留意点および、規格の取扱いについて

4.1. 標準におけるクエリパラメータの取り扱い

4.1.1. コンフォーマンステストにおけるクエリパラメータの取扱い

報告書執筆時点では、Provider のコンフォーマンステストでは、一部のクエリパラメータのテストを実施しないことが分かっている（表 4-1）。そのため、コンフォーマンステスト合格した Provider においても、クエリパラメータの利用においては制約が生まれる可能性があることは留意する必要がある。

表 4-1　コンフォーマンステストにおけるクエリパラメータのテスト状況

クエリパラメータ	備考	コンフォーマンステスト
limit	取得データの件数（上限）を指定	テスト対象外
offset	取得データのページングを行う	テスト対象外
sort	取得データの並べ替えを行う	対象
orderBy	取得データの並べ替え順序を指定	対象
filter	条件にもとづく取得データの絞り込みを実施	テスト対象外
fields	取得データのフィールドの範囲を指定する	テスト対象外

4.1.2. Filter クエリパラメータの演算子について

Filter クエリパラメータについては、条件にもとづいて、取得データにフィルタを行うための機能である。データ項目に対して、表 4-2 に示す演算子を利用して、検索を行うことができることが定義されている。

表 4-2　Filter クエリパラメータに対する演算子

演算子	意味
=	等しい
!=	等しくない
>	より大きい
>=	以上
<	より少ない
<=	以下
~	含む

コンフォーマンステストにおいては、Filter クエリパラメータについては、以下の 5 項目のテストのみが実施されている。規格上に記載されている演算子と、コンフォーマンステストにおけるテストで利用される演算子の範囲に違いがあり、曖昧さが残っている。

a. Filtering for students using the filter of "filter=roles.role~'student'" on the "getAllUsers()" endpoint
b. Filtering for teachers using the filter of "filter=roles.role~'teacher'" on the "getAllUsers()" endpoint
c. Filtering for grading periods using the filter of "filter=type='gradingPeriod'" on the "getAllAcademicSessions()" endpoint
d. Filtering for terms using the filter of "filter=type='term'" on the "getAllAcademicSessions()" endpoint
e. Filtering for schools using the filter of "filter=type='school'" on the "getAllAcademicSessions()" endpoint

本件については、日本 1EdTech 協会への問い合わせを行った。結果、コンフォーマンステストに定義されている Filter クエリパラメータの範囲が、仕様上の必須事項として解釈するのが適当であるとの見解を得ている。また、現時点における OneRoster REST API の普及状況および、ユースケース開発の状況に鑑みて、コンフォーマンステストにおける最小限の実装に留めることは否定されないとの認識も示された。

Filter クエリパラメータにおけるすべての演算子について、全てのエンドポイント、全てのプロパティに対して、完全なサポートを求めることは、実装する側から見ると、実装およびテスト等の工数が大きくなることが危惧される。

このようなことから、クエリパラメータの実装範囲は、想定されるユースケースやステークホルダーの議論により、Japan Profile で定義することも検討されてよい。

4.2. REST API における差分更新

OneRoster CSV においては、全ての情報を一括で出力する Bulk と、ある一定時間に発生した変更差分の情報だけを出力する Delta の 2 つのファイル形式が定義されている。Delta を利用することで、ファイルに含まれるデータ行数が減り、ファイルのデータサイズが少なくなる。これにより、システム連携におけるデータ処理時間の短縮を実現できる可能性がある。

REST API においては、Consumer によって、どのような問い合わせが行われるかによって情報出力の方法が変更されるため、例えば、情報の最終更新日（dateLastModified）に対してフィルタ処理をかけることで、過去 24 時間以内の更新情報を取得することができ、その問い合わせを毎日実施することで、差分更新を実現することができる。日本の自治体においては、児童生徒数で十万人を超える大規模な自治体もあることから、差分更新を行うことで、データ処理件数を大幅に縮小し、データ処理時間の短縮を実現することができる。

ただし、差分更新を実現するためのフィルタ処理については、設計等のミスなどによって、更新の取りこぼしを発生させる可能性もあるため、実装には注意が必要な機能である。また、差分更新を実現するためのフィルタ処理については、前項で触れたクエリパラメータの利用が不可欠であるが、クエリパラメータの利用については、コンフォーマンステストの対象が限られていることから、あらゆる状況で安定的に利用できるかは、現時点では不透明な機能と言わざるを得ない。

差分更新自体のユースケースやメリットは明確であることから、今後、標準そのものの改訂やコンフォーマンステストの厳格化などが行われる可能性も考慮に入れる必要がある。

4.3. 情報の一貫性について

4.3.1. sourcedId の一貫性について

OneRoster において、各種の情報の識別子となる、sourcedId の項目については、OneRoster CSV においては、一度に出力されるファイル（CSV）の範囲内で一意であることが明示的に定義されている。一方で、

複数回に分かれる出力時に、同じ情報に対して同じ sourcedId が一貫して出力される必要があるかどうかについては明示的な定義が存在していなかった。そのため、出力される sourcedId について一貫性を持って出力しないシステムも存在している。OneRoster 上には、soucedId 以外の識別子により識別可能な情報（例：users.csv）と、sourcedId 以外の情報で識別が難しい OneRoster 上の項目（例：classes.csv）が存在している。そのため、クラスの情報などの連携を行う際に情報の特定が難しい場合があった。

この点については、REST API においては、sourcedId に対して一貫性のある値が返されることが期待されていると解釈すべきである。

例えば、同じ情報（例：児童生徒）に対して、連続して 2 回の問い合わせが行われた場合で、かつ、この間に情報の更新が無かった場合、1 回目の問い合わせに含まれる情報と、2 回目の問い合わせに含まれる情報については、同一のサービスに対する同一の問い合わせである以上、同一の情報が出力されていることが自然である。また、sourcedId を用いて単一情報に対する問い合わせを行う API（例：/users/{sourcedId}）も存在していることから、この点についても、sourcedId に対する一貫性が必要である。

このように、REST API において sourcedId の一貫性が必要であることからすると、OneRoster CSV と REST API を同時にサポートする Provider においては、フォーマットが異なっていたとしても、同一の情報に対して、同一の sourcedId が出力されることが望ましい。

4.3.2. データの出力順序について

OneRoster REST API においては、sort・orderBy と言った情報を並び替えるためのクエリパラメータが存在している。これらの sort・orderBy の指定が無い場合において、どのような順序で情報が出力されるかについては規格上に特別な定義が無い。

しかしながら、limit や offset と言ったクエリパラメータによって、ページングを利用して、情報を複数回に分けて取得される状況を仮定した際に、情報の順序が呼び出しの都度異なっていては情報取得に困難が生じる。このことから、sort・orderBy が指定されない場合（システムデフォルト）の、並び順が存在していることが期待されている。

4.4. CSV と REST API におけるシステムの役割の差異について

REST API と CSV では、データの出力側である校務支援システムの役割が異なっている。REST API では、データの出力側は API エンドポイントを持つ Web サービスとして運営する必要がある。一方で CSV においては、データを出力するところまでが、出力側の役割である。

これらの違いにより、システムへの負荷やセキュリティ対策など、継続的に発生する運用に関する負担は、CSV より REST API の方が高く、CSV を実装した際とは異なる負担やノウハウが必要になる。例えば、API の

利用監視を行い、不正アクセスや異常なデータ取得を監視するなどのセキュリティ対策に加えて、可用性を確保するための負荷分散対策や、性能を維持するためのレスポンス時間の最適化のための工夫などが挙げられる。

これらを考慮した場合、REST API Provider となるベンダーから見た際に、CSV を実装する場合と比較して、実装だけでは無く、運用面に関わる負担が高くなることは否めない。

4.5. REST API エンドポイントの実装範囲について

REST API のエンドポイントの実装については、「2.3.2 実装対象の API エンドポイントの定義について」に示したように、技術標準上必須されているものと、CSV Japan Profile における仕様の整合性を保つように導入した項目である。

学習 e ポータル標準に定義されているユースケースを実現するのみの観点であれば、全てのエンドポイントは実用上必要では無く、実装が不要に見えるエンドポイントも多い。一方で、1EdTech 技術標準において必須されているエンドポイントの実装を Japan Profile において変更（縮小）した場合には、コンフォーマンステストなどのツールを日本独自に開発する必要が生まれてしまう。このような事情を踏まえた上で、ステークホルダー間での合意が必要である。

5. まとめと今後の課題

5.1. OneRoster REST API Japan Profile の定義について

本実証では、CSV Japan Profile と同様のユースケース、同等の情報量を扱うことができることを前提として OneRoster REST API における、Japan Profile 相当の技術仕様を定義した。

定義した Japan Profile 相当の REST API について、校務支援システム（情報出力）と学習支援システム（情報入力）に実装した上で、それぞれの製品に対して、1EdTech コンソーシアムの提供する、OneRoster REST API におけるコンフォーマンステスト（適合性テスト）を実施し、合格できる状態とした。このことから、定義した Japan Profile については、OneRoster における互換性を持った状態であると言える。

実装した校務支援システムと学習支援システムの間で、実際に情報の連携を実施することができたため、想定したユースケースにおいて、問題は生じないことが分かった。

5.2. REST API による連携実現のメリット

連携に関する作業を自動化する観点では、CSV より REST API の方が、連携の自動化が容易である。CSV は、任意のタイミングで全ての情報を出力することが前提であり、かつファイルの搬送手段も定義されていない。そのため、CSV に対応したシステム同士を採用したとしても、連携ファイルの投入を手動で実施する必要があり、自動で連携を行いたい場合は、連携手段の設計・実装を行う必要がある。CSV による連携を導入する自治体等はこれらの工数を負担する必要があった。これに対して、REST API では、データ搬送手段が標準として定義されているため、自動連携を行う場合に、連携手段の検討や設計を行う必要はない点がメリットとして挙げられる。

REST API ではデータ搬送手段が標準仕様に含まれるため、CSV と比較して、より厳密なセキュリティ管理が可能である。CSV では、出力されたファイルの管理や搬送手段は定義されていないため、出力したファイルの管理が各ユーザーに任されており、どのアプリケーションにファイルを投入するかなどは、アプリケーションが制御することはできない。これに対して、REST API では、OAuth 2.0 CC によって、API アクセスに対する認証・認可処理が存在するため、特定の Consumer に対してのみ API アクセスを許可することができる。また、データ通信先 IP アドレスの制御を行う、HTTPS による暗号化通信を必須とするなど、Provider 側でのセキュリティ機能の実装が可能な範囲が広い。このことから、REST API を採用する自治体等においては、セキュリティ管理に関する負担を下げることが可能である。

REST API の利用によって、同期に関するサービスレベルを向上させることができる点もメリットである。CSV の連携では、連携されるデータの種類やタイミングは、Provider 側（データ出力側）に依存するため、固定的なタイミングでの出力に限定される。例えば、学習システムにユーザーがログインするタイミングなど、システム利用のタイミングに合わせた、リアルタイムの連携処理などを実現することも可能である。

このような REST API の特性から、クラウドベースの多様なアプリケーションによる利用に適しており、多様なユースケースの登場を想定することができる。例えばデータ連携の関係性についても、校務支援システム→学習支援システムの連携だけでは無く、校務支援システム→学習ツールや、学習支援システム→学習ツールの関係性が想定される。また、学習ツール以外のアプリケーション、例えば、学習分析用のアプリケーションや保護者用アプリケーションなどとの連携も想定される。

5.3. 標準化に向けて検討すべきこと

REST API における Japan Profile を標準化し、普及させるプロセスとして、CSV と同じく、ICT CONNECT21 の校務系-学習系情報連携 SWG での意見交換や、日本 1EdTech 協会の検討が実施され、正式なプロファイルとして標準化されることが望ましい。

REST API における、Japan Profile の標準化プロセスにおいて、合意形成が図られるべき項目について、本報告書の内容を元に、以下のように整理する。（カッコ内は本報告書の項目番号）

- 実装すべきエンドポイントの範囲を決定する（2.3.2）
- Japan Profile において拡張する項目は CSV と同様で良いか確認する（2.3.3）
- 実装すべきクエリパラメータの範囲はどこまでの範囲とするか決定する（4.1.1）
- Filter クエリパラメータの演算子の実装範囲をどこまでにするか決定する（4.1.2）
- sourcedId の一貫性を維持することを確認する（4.3.1）
- データの出力順序の一貫性を維持することを確認する（4.3.2）