

Advanced Distributed Learning (ADL) Co-Laboratories

2014.2.18 日本語版

「Tin Canプロジェクト」

特定非営利活動法人日本イーラーニングコンソシアムとモバイルラーニングコンソシアムは、共同で「Tin Canプロジェクト」を発足しました。この文書は、「Tin Canプロジェクト」メンバーが、Experience API(v1.0.0)を日本語に翻訳したものです。

この文書についての問い合わせは、以下をお願いします。

tincan@elc.or.jp

「Tin Canプロジェクト」 日本語版制作メンバー（五十音順）

伊藤彰孝(株式会社ベネッセコーポレーション)

勝田祥生(株式会社レビックグローバル)

加藤泰久(NTTラーニングシステムズ株式会社)

熊澤 剛(株式会社ヒューマンサイエンス)

佐伯秀雄(NTTラーニングシステムズ株式会社)

田中頼人(早稲田大学)

徳弘太郎(ヤマハ株式会社)

藤井直人(TinCanプロジェクト・リーダー 株式会社ヌーサイト)

星野忠明(エスエイティーティー株式会社)

程田恭介(株式会社富士通ラーニングメディア)

前田 宏(株式会社ジンジャーアップ)

This document was authored by members of the Experience API Working Group (see list on pp. 7-8) in support of the Office of the Deputy Assistant Secretary of Defense (Readiness) Advanced Distributed Learning (ADL) Initiative. Please send all feedback and inquiries to helpdesk@adlnet.gov

この文書は、Experience API ワーキンググループのメンバー(2.2.1 の表を参照)が、国防次官補室(即応性担当)の Advanced Distributed Learning (ADL) イニシアティブを支援し作成しました。フィードバックやお問い合わせは、helpdesk@adlnet.gov までお送りください。

Table of Contents 目次

- 1.0. Revision History改定履歴
- 2.0. Role of the Experience APIExperience API の役割
 - 2.1. ADL's Role in the Experience APIxAPI における ADL の位置づけ
 - 2.2. Contributors貢献者
 - 2.2.1. Working Group Participantsワーキンググループの参加者
 - 2.2.2. Requirements Gathering Participants要求仕様収集への貢献者
 - 2.3 Reading Guidelines for the Non-Technically Inclined技術に詳しくない人への読み方ガイドライン
- 3.0. Definitions用語の定義
- 4.0. Statementステートメント
 - 4.1. Statement Propertiesステートメントプロパティ
 - 4.1.1. IDid (識別子)
 - 4.1.2. ActorActor (アクタ)
 - 4.1.3. VerbVerb (動詞)
 - 4.1.4. ObjectObject (目的語)
 - 4.1.5. ResultResult (結果)
 - 4.1.6. ContextContext (文脈)
 - 4.1.7. TimestampTimestamp (タイムスタンプ)
 - 4.1.8. StoredStored
 - 4.1.9. [AuthorityAuthority (権限)
] (#authority)
 - 4.1.10. VersionVersion (バージョン)
 - 4.1.11. AttachmentsAttachments (添付資料)
 - 4.1.12. Data Constraintsデータの制約
 - 4.2. Retrieval of Statementsステートメントの検索
 - 4.3. Voided無効
 - 4.4. Signed Statements署名付きステートメント
- 5.0. Miscellaneous Types各種タイプ
 - 5.1. Documentドキュメント
 - 5.2. Language Map言語マップ
 - 5.3. Extensions拡張
 - 5.4. Identifier Metadata識別子メタデータ
- 6.0. Runtime Communicationランタイム通信
 - 6.1. Encodingエンコーディング
 - 6.2. API VersioningAPI のバージョン管理
 - 6.3. Concurrency同時実行
 - 6.4. Securityセキュリティ
 - 6.4.1. Authentication Definitions各シナリオの取り扱い方法
 - 6.4.2. OAuth Authorization ScopeOAuth 認証スコープ
- 7.0. Data Transfer (REST)Data Transfer (REST)
 - 7.1. Error Codesエラーコード
 - 7.2. Statement APIステートメント API
 - 7.3. Document APIsドキュメント API
 - 7.4. State APIステート API
 - 7.5. Activity Profile APIアクティビティプロフィール API
 - 7.6. Agent Profile APIエージェントプロフィール API
 - 7.7. About resourceAbout リソース
 - 7.8. Cross Origin Requestsクロスオリジンリクエスト
 - 7.9. Validationバリデーション
 - 7.10. HTTP HEADHTTP HEAD
- Appendix A: Bookmarkletブックマークレット
- Appendix B: Creating an "IE Mode" RequestIE 用リクエスト作成
- Appendix C: Example definitions for Activities of type "cmi.interaction"アクティビティタイプ "cmi.interaction" のための体儀例
- Appendix D: Example Statementsステートメント例
- Appendix E: Converting Statements to 1.0.01.0.0 ヘステートメントを変換
- Appendix F: Example Signed Statement署名ステートメント例

1.0 Revision History 改定履歴

0.8 (Project Tin Can API Deliverable) to 0.9 (March 31, 2012) 0.8 (Project Tin Can API の配布) から 0.9 (2012/03/31)

Rustici Software, who delivered the Project Tin Can API, made modifications to the API prior to the April 2012 Kickoff Meeting. It was voted in this meeting to move those changes into the current specification and revision to 0.9.

Project Tin Can API を提供した Rustici Software が、2012年4月のキックオフ会議に先立ち、API に対する改定を行った。会議において投票が行われ、それらの改定を仕様書に取り込み、第0.9版とすることが議決された。

0.90 to 0.95 (August 31, 2012) 0.90 から 0.95 (2012/08/31)

"Core" Verbs and Activity types were removed from the specification. References to these verbs in results, context, interactions, and Activity Definitions were also removed. It was recommended that implementers prefer community defined verbs to creating their own verbs.

"コア"とされる動詞とアクティビティタイプが仕様書より削除された。また、結果、文脈、インタラクション、およびアクティビティ定義における、それらの動詞への参照も同時に削除された。実装者には、自ら動詞を定義するよりも、コミュニティが定義した動詞を利用することが推奨された。

- Verbs, Activity types, and extension keys are now URIs動詞、アクティビティタイプ、拡張キーを URI とする。
- Restructured and added language around some of the other implementation details and scope.いくつかの実装詳細と対象範囲を整理し、説明を追加した。
- Changed from using a person-centric view of Agents to a persona-centric view.エージェントについて、個人指向の視点からペルソナ指向の視点を用いるように変更。
- Friend of a Friend (FOAF) Agent merging requirement was removed.友達の友達 (FOAF) エージェントの結合要求を削除。
- Agent Objects must now have exactly 1 uniquely identifying property, instead of at least one.エージェントオブジェクトは、(1つ以上の一意 ID プロパティではなく)ただ1つの一意 ID プロパティを持つように変更。

0.95 to 1.0.0 (April 26, 2013) 0.95 から 1.0.0 (2013/4/26)

Various refinements and clarifications including:

以下のような多数の改善と明確化を行った。

- Adding attachments添付文書の概念を追加。
- Activity metadata is now stored as JSON rather than XMLアクティビティのメタデータを XML ではなく JSON に変更。
- Changes to voiding Statementsステートメントの無効化に関する変更。
- Clarification and naming of the Document APIsドキュメント API の追加。
- Changes to querying the Statement APIステートメント API の検索に関する変更。
- Signed Statements署名付きステートメントの追加。

2.0 Role of the Experience API Experience API の役割

The Experience API is a service that allows for statements of experience to be delivered to and stored securely in a Learning Record Store (LRS). These statements of experience are typically learning experiences, but the API can address statements of any kind of experience. The Experience API is dependent on Activity Providers to create and track these learning experiences; this specification provides a data model and associated components on how to accomplish these tasks.

Experience API (xAPI) は経験に関するステートメントを、Learning Record Store (LRS) に配送し、安全に記録するためのサービスである。これらの経験に関するステートメントは、典型的には学習経験を示す物であるが、API はどのような経験に関するステートメントも扱うことができる。xAPI は、アクティビティプロバイダがこれらの学習経験を生成、蓄積することを目的としており、本仕様はその動作を実現するためのデータモデルと関連するコンポーネントを提供する。

Specifically, the Experience API provides:

xAPI は以下を提供する。

- The structure and definition of Statement, State, learner, Activity and Objects, which are the means by which experiences are conveyed by a Activity Provider.アクティビティプロバイダによって経験が伝達される手段としての、ステートメント、状態、学習者、アクティビティ、オブジェクトの構造と定義
- Data Transfer methods for the storage and retrieval (but not validation) of these Objects to/from a Learning Record Store. Note that the systems storing or retrieving records need not be Activity Providers. LRSs may communicate with other LRSs, or reporting systems.上記のオブジェクトの LRS に対する記録と取り出し(検証は含まない)のためのデータ転送手段。情報を記録または取り出すシステムは、アクティビティプロバイダであ

る必要はないことに注意。LRS は他の LRS やレポーティングシステムと通信をすることがある

- Security methods allowing for the trusted exchange of information between the Learning Record Store and trusted sources.LRS とデータソースとの間での信頼のおける情報通信を可能にするセキュリティ手段

The Experience API is the first of many envisioned technologies that will enable a richer architecture of online learning and training. Authentication services, querying services, visualization services, and personal data services are some examples of additional technologies which the Experience API is designed to support. While the implementation details of these services are not specified here, the Experience API is designed with this larger architectural vision in mind.

xAPI はオンライン学習とトレーニングに関する高機能なアーキテクチャを提供する一連の技術提案の端緒となる。xAPI の適用例として、認証サービス、検索サービス、可視化サービス、個人データサービスなどが考えられる。本仕様には、これらのサービスの実装詳細については示されていないが、xAPI は大きなアーキテクチャ思想を前提に設計されている。

2.1 ADL's Role in the Experience API xAPI における ADL の位置づけ

The Advanced Distributed Learning (ADL) Initiative has taken on the roles of steward and facilitator in the development of the Experience API. The Experience API is seen as one piece of the ADL Training and Learning Architecture, which facilitates learning anytime and anywhere. ADL views the Experience API as an evolved version of SCORM that can support similar use cases, but can also support many of the use cases gathered by ADL and submitted by those involved in distributed learning that SCORM could not enable.

Advanced Distributed Learning (ADL) イニシアティブは xAPI の開発における、事務局やファシリテータとしての役割を担っている。xAPI は、学習をいつでもどこでも行えるような ADL のトレーニングと学習のためのアーキテクチャの一部と位置付ける。ADL は、xAPI を同じようなユースケースをサポートできる SCORM の拡張とみているだけでなく、ADL や分散学習に係る人たちによって提案された、SCORM が実現できないユースケースもサポートするものとする。

2.2 Contributors 貢献者

"My thanks to everyone who contributed to the Experience API project. Many of you have called into the weekly meetings and helped to shape the specification into something that is useful for the entire distributed learning community. Many of you assisted in releasing code samples, products, and documentation to aid those who are creating and adopting the specification. I'd also like to thank all of those who were involved in supplying useful, honest information about your organization's use of SCORM and other learning best practices. Through the use-cases, shared experiences, and knowledge you have shared, ADL and the community clearly identified the first step in creating the Training and Learning Architecture--the Experience API. You are truly the community leaders on which we depend to make our training and education the very best."

Experience API プロジェクトに貢献していただいた皆さんに感謝いたします。多くの皆さんが毎週のミーティングに参加し、本仕様書を分散学習コミュニティ全体に対して有用なものとなるように仕上げる支援をしてくれました。また、多くの皆さんが、仕様書を作成、編集している人たちの助けになるように、コードサンプルや製品、文書などを提供してくれました。各自の組織における SCORM の利用や学習に関するベストプラクティスの有用で正直な情報を提供していただいた皆様にも感謝したいと思います。皆さんから提供いただいたユースケース、経験と知識に基づいて、ADL とコミュニティはトレーニングと学習に関するアーキテクチャの第一歩 - Experience API を明確に定義することができました。皆さんこそが、最高のトレーニングと教育を提供するために私たちが頼りにできるコミュニティリーダーです。

Kristy S. Murray, Ed.D.
Director, ADL Initiative
OSD, Training Readiness & Strategy (TRS)

2.2.1 Working Group Participants ワーキンググループの参加者

_Name_名前	_Organization_組織
Aaron Silvers	ADL
Al Bejcek	NetDimensions
Ali Shahrazad	SaLTBOX
Andrew Downes	Epic
Andy Johnson	ADL
Andy Whitaker	Rustici Software
Anthony Altieri	American Red Cross
Anto Valan	Omnivera Learning Solutions

Avron Barr	Aldo Ventures, Inc.
Ben Clark	Rustici Software
Bill McDonald	Boeing
Brian J. Miller	Rustici Software
Chad Udell	Float Mobile Learning
Chris Sawwa	Meridian Knowledge Solutions
Dan Allen	Litmos
Dan Kuemmel	Sentry Insurance
Dave Mozealous	Articulate
David Ells	Rustici Software
David N. Johnson	Clear Learning Systems
Doug Hagy	Twin Lakes Consulting Corporation
Eric Johnson	Planning and Learning Technologies, Inc.
Fiona Leteney	Feenix e-learning
Greg Tatka	Menco Social Learning
Ingo Dahn	University Koblenz-Landau
Jason Haag	ADL
Jeff Place	Questionmark
Jennifer Cameron	Sencia Corporate Web Solutions
Jeremy Brockman	
Jhorlin De Armas	Riptide Software
Joe Gorup	CourseAvenue
John Kleeman	Questionmark
Jonathan Archibald	Brightwave
Jonathan Poltrack	ADL
Kris Miller	edcetra Training
Kris Rockwell	Hybrid Learning Systems
Lang Holloman	
Lou Wolford	ADL
Luke Hickey	dominKnow
Marcus Birtwhistle	ADL
Mark Davis	Exambuilder
Matteo Scaramuccia	
Megan Bowe	Rustici Software
Melanie VanHorn	ADL
Michael Flores	Here Everything's Better

Michael Roberts	vTrainingRoom
Mike Palmer	OnPoint Digital
Mike Rustici	Rustici Software
Nick Washburn	Riptide Software
Nikolaus Hruska	ADL
Pankaj Agrawal	Next Software Solutions
Patrick Kedziora	Kedzoh
Paul Esch	Nine Set
Paul Roberts	Questionmark
Rich Chetwynd	Litmos
Richard Fouchaux	Ontario Human Rights Commission
Richard Lenz	Organizational Strategies, Inc.
Rick Raymer	
Rob Chadwick	ADL
Robert Lowe	NetDimensions
Russell Duhon	SaLTBOX
Stephen Trevorrow	Problem Solutions, LLC.
Steve Baumgartner	
Steve Flowers	XPCconcept
Thomas Ho	
Tim Martin	Rustici Software
Tom Creighton	ADL
Walt Grata	ADL

2.2.2 Requirements Gathering Participants 要求仕様収集への貢献者

In collection of requirements for the Experience API, many people and organizations provided invaluable feedback to the Sharable Content Object Reference Model (SCORM®), distributed learning efforts, and learning technology efforts in general. While not an exhaustive listing, the white papers gathered in 2008 by the Learning Education and Training Standards Interoperability (LETSI) group, the Rustici Software *UserVoice* website, one-on-one interviews and various blogs were important sources from which requirements were gathered for the Experience API specification.

xAPI の要求の収集において、多くの人々や組織から、SCORM®、分散学習、および学習テクノロジー一般について、個別のフィードバックを得ることができた。全てを挙げることはできないが、学習教育研修システムの相互運用性 (LETSI) グループによって 2008 年にまとめられたホワイトペーパー、Rustici Software 社の *UserVoice* ウェブサイト、個別のインタビュー、およびさまざまなブログ記事は、xAPI の仕様をまとめるための重要な情報源となった。

2.3 Reading Guidelines for the Non-Technically Inclined 技術に詳しくない人への読み方ガイドライン

This is a definitive document which describes how the Experience API is to be implemented across a variety of systems. It is a technical document authored specifically for individuals and organizations implementing this technology with the intent of such individuals developing interoperable tools, systems and services that are independent of each other and interoperable with each other.

本文書は、さまざまなシステムに対して、xAPI の実装方法を示す正式文書である。また、本文書は、この技術を実装する個人と組織に向け、実装者が、独立で相互運用可能なツール、システム、およびサービスを開発できることを目指した技術文書である。

Whenever possible, the language and formatting used in this document is intended to be *considerate* of non-technical readers because

various tools, systems and services are based on the specification set described below. For this reason, sections that provide a *high-level overview* of a given facet of the Experience API are labeled **description** or **rationale**. Items in this document labeled as **requirements**, **details** or **examples** are more technical.

さまざまなツール、システムおよびサービスが、ここに定義される仕様に基づいて設計されるため、可能な限り、この文書中の文言や形式は、技術に詳しくない人にも配慮されている。そのために、xAPIの各部分の概要説明には、説明または背景という見出しをつけている。技術的な部分については、必要条件、詳細、または例という見出しをつけている。

3.0 Definitions 用語の定義

- Activity アクティビティ
- Activity Provider アクティビティプロバイダ (AP)
- Actor アクタ (Actor)
- Authentication 認証 (Authentication)
- Authorization 認可 (Authorization)
- Client クライアント
- Community of Practice 実践コミュニティ
- Experience API Experience API (xAPI)
- Immutable イミュータブル
- Internationalized Resource Identifier (IRI) IRI (Internationalized Resource Identifier)
- Inverse Functional Identifier 逆関数識別子 (Inverse Functional Identifier)
- Learning Management System (LMS) LMS
- Learning Record Store (LRS) LRS
- MUST / SHOULD / MAY ~しなければならぬ (MUST)/~すべきである (SHOULD)/~してもよい (MAY)
- Profile プロファイル
- Registration 登録事項 (Registration)
- Representational State Transfer (REST) REST (REpresentational State Transfer)
- Service サービス
- Statement ステートメント
- Tin Can API (TCAPI) Tin Can API (TCAPI)
- Verb 動詞 (Verb)

Activity アクティビティ: Something with which an Actor interacted. It can be a unit of instruction, experience, or performance that is to be tracked in meaningful combination with a Verb. Interpretation of Activity is broad, meaning that Activities can even be tangible Objects.

アクタが相互作用を行った、何らかの対象である。動詞との意味のある組合せによって記録される「教示や経験、成果」の単位となりうる。アクティビティは幅広く解釈でき、またアクティビティは具体物を指す場合もある。

Activity Provider (AP) アクティビティプロバイダ: The software object that is communicating with the LRS to record information about a learning experience. May be similar to a SCORM package in that it is possible to bundle learning assets with the software object that performs this communication, but an Activity Provider may also be separate from the experience it is reporting about.

LRSと通信し、学習経験についての記録を行うソフトウェアオブジェクト。学習アセットや通信可能なオブジェクトをまとめた SCORM パッケージに似ているが、アクティビティプロバイダは伝達しようとしている経験そのものから切り離されることがある。

Actor アクタ: An identity or persona of an individual or group tracked using Statements as doing an action (Verb) within an Activity.

個人やグループのアイデンティティや外的側面。それはステートメントを用い、アクティビティの中で動作しながら記録される。

Authentication 認証 (Authentication): The concept of verifying the identity of a user or system. Authentication allows interactions between the two "trusted" parties.

ユーザやシステムのアイデンティティを確認すること。認証によって、2つの「信頼された」対象どうし間のやり取りが可能になる。

Authorization 認可 (Authorization): The affordance of permissions based on a user or system's role; the process of making one user or system "trusted" by another.

ユーザやシステムの役割に応じ、何らかの利用許可を与えること。それはあるユーザやシステムを他者から信頼されるようにする過程である。

Client クライアント: - Refers to any entity that may interact with an LRS. A Client can be an Activity Provider, reporting tool, an LMS, or another LRS.

LRSとやり取りしうる全ての物。クライアントはアクティビティプロバイダや報告ツール、LMS や他の LRS にもなりうる。

Community of Practice 実践コミュニティ: A group, usually connected by a common cause, role or purpose, which operates in a common modality.

共通の動機や役割、目的によって結ばれることの多いグループである。またその集団は共通の様式のもとで行動する。

Experience API (xAPI)**Experience API (xAPI):** The API defined in this document, the product of "Project Tin Can". A simple, lightweight way for any permitted Actor to store and retrieve extensible learning records, learner and learning experience profiles, regardless of platform.

この文書の中で規定される API であり、Tin Can プロジェクトの成果物である。許可を受けたアクタが「拡張可能な学習記録や学習者プロフィール、学習経験プロフィール」を保存し、また取り出すための簡素で軽量な方法である。またそれはプラットフォームに依存しない。

Immutable**イミュータブル:** Adjective used to describe things which cannot be changed. With some exceptions, Statements in the xAPI are immutable. This ensures that when Statements are shared between LRSs, multiple copies of the Statement remain the same.

変わることはない事象を記述するための形容詞。若干の例外を除き、xAPI のステートメントはイミュータブル(不変)である。イミュータブルは、ステートメントが LRS 間で共有されるときに、複製されたステートメントの間の同一性を保つことを保証する。

Internationalized Resource Identifiers (IRI)**IRI (Internationalized Resource Identifiers):** A unique identifier which may be an IRL. In the xAPI, all IRIs should be a full absolute IRI including a scheme. Relative IRIs should not be used. IRIs should be defined within a domain controlled by the person creating the IRL.

IRL でありうる一意な識別子。xAPI では、全ての IRI はスキームを含む完全な絶対 IRI となるべきである。相対 IRI は使われるべきではない。IRL は IRL を作る者がコントロールするドメインの中で、定義されるべきである。

Inverse Functional Identifier**逆関数識別子 (Inverse Functional Identifier):** An identifier which is unique to a particular persona or group. Used to identify Agents and Groups.

特定の人物やグループに対する一意な識別子。行為者やグループを特定するために用いられる。

Learning Management System (LMS)**Learning Management System (LMS):** "A software package used to administer one or more courses to one or more learners. An LMS is typically a web-based system that allows learners to authenticate themselves, register for courses, complete courses and take assessments" (Learning Systems Architecture Lab definition). In this document the term will be used in the context of existing systems implementing learning standards.

Learning Systems Architecture Lab (訳注: カーネギーメロン大) の定義によれば、LMS は1人以上の学習者に1つ以上の学習コースを提供するためのソフトウェア・パッケージである。LMS は一般に学習者を認証し、コースに登録し、コースを修了させ、評価するための Web ベースのシステムである。本仕様書においては、標準規格を実装する既存のシステムという文脈のもとで、LMS という用語を用いる。

Learning Record Store (LRS)**Learning Record Store (LRS):** A system that stores learning information. Prior to the xAPI most LRSs were Learning Management Systems (LMSs); however this document uses the term LRS to be clear that a full LMS is not necessary to implement the xAPI. The xAPI is dependent on an LRS to function.

学習に関する情報を蓄えるためのシステム。xAPI が登場する前は、ほとんどの LRS は LMS だった。だが本仕様書で LRS という用語を用いる場合、「xAPI の実装のためにフル仕様の LMS が必要とは限らない」ということを、強調しておきたい。xAPI はその機能を果たすために、(訳注: LMS でなく) LRS のほうを必要とする。

MUST / SHOULD / MAY~しなければならない (MUST)/~すべきである (SHOULD)/~してもよい (MAY): Three levels of obligation with regards to conformance to the xAPI specification. A system that fails to implement a MUST (or a MUST NOT) requirement is non-conformant. Failing to meet a SHOULD requirement is not a violation of conformity, but goes against best practices. MAY indicates an option, to be decided by the developer with no consequences for conformity.

xAPI 仕様への適合性に関する、3つのレベルでの約束ごと。MUST 条件(または MUST NOT 条件)を満たさないシステムは、xAPI に適合しない。SHOULD 条件を満たさないシステムは適合性に反してはいないが、ベストプラクティスには相応しくない。MAY 条件は適合性を気にせずに開発者が選択できるオプションである。

Profile**プロフィール:** A construct where information about the learner or activity is kept, typically in name/document pairs that have meaning to an instructional system component.

一般的に「教育システムの要素として意味を持つ、名前と文書の組」によって学習者やアクティビティの情報を保持する構成体である。

Registration**登録事項 (Registration):** An instance of a learner experiencing a particular Activity.

特定のアクティビティを経験する学習者のインスタンスである。

Representational State Transfer (REST)**REST (REpresentational State Transfer):** An architecture for designing networked web Services. It relies on HTTP methods and uses current web best practices.

ネットワーク化された web サービスを接続するためのアーキテクチャである。REST は HTTP のメソッドを信頼し、現在の web のベストプラクティスを役立てる。

Service**サービス:** A software component responsible for one or more aspects of the distributed learning process. An LMS typically combines many services to design a complete learning experience.

分散学習環境の一つ以上の局面に責任を持つ、ソフトウェアの構成要素である。一般的に LMS は、学習経験全体をデザインするために複数のサービスを結合させ

る。

Statementステートメント: A simple construct consisting of `<actor (learner)>` `<verb>` `<object>`, with `<result>`, in `<context>` to track an aspect of a learning experience. A set of several Statements may be used to track complete details about a learning experience.

学習経験の一局面を記録する「文脈」において、「アクタ(学習者)、動詞、オブジェクト」の3つ組からなり「結果」を有する、単純な構成体である。いくつかのステートメントの組は、学習経験に関する完全な詳細情報を記録するために使われる。

Tin Can API (TCAPI)**Tin Can API (TCAPI)**: The previous name of the API defined in this document, often used in informal references to the Experience API.

本仕様書で定義される API の、以前の名称である。xAPI へのインフォーマルな呼称として用いられる。

Verb動詞 (Verb): Defines the action being done by the Actor within the Activity within a Statement.

ステートメント中のアクティビティにおける、アクタの行為を定義するものである。

4.0 Statement ステートメント

The Statement is the core of the xAPI. All learning events are stored as Statements. A Statement is akin to a sentence of the form "I did this".

ステートメントは xAPI の中心概念である。全ての学習イベントはステートメントとして記録される。ステートメントは、"I did this" といった文と似た構造を持つ。

4.1 Statement Properties ステートメントプロパティ

Actor, Verb, and Object are required, all other properties are optional. Properties can occur in any order, but are limited to one use each. Each property is discussed below.

アクタ、動詞、および目的語は必須で、それ以外のプロパティは任意である。プロパティの順序は自由であるが、それぞれ一度しか指定できない。以下に、各プロパティを示す。

Propertyプロパティ	Typeタイプ	Description説明
id	UUID	UUID assigned by LRS if not set by the Activity Provider. アクティビティプロバイダが設定しない場合に、LRS によって割り当てられた UUID。
actor	Object	Who the Statement is about, as an Agent or Group Object. Represents the "I" in "I Did This". ステートメントが誰に関するものか (Agent や Group として)。“I Did This” の中の “I” に対応。
verb	Object	Action of the Learner or Team Object. Represents the "Did" in "I Did This". 学習者やチームオブジェクトの行う行為。“I Did This” の “Did” に相当。
object	Object	Activity, Agent, or another Statement that is the Object of the Statement. Represents the "This" in "I Did This". Note that Objects which are provided as a value for this field should include an "objectType" field. If not specified, the Object is assumed to be an Activity. ステートメントの目的語となる アクティビティ、エージェント、または別のステートメント。“I Did This” の “This” に相当。この項目の値として提供される目的語には "objectType" フィールドを含むべきであることに注意。指定しない場合は目的語はアクティビティであるとみなされる。
result	Object	Result Object, further details representing a measured outcome relevant to the specified Verb. 動詞に関する測定結果を示す結果オブジェクト。
context	Object	Context that gives the Statement more meaning. Examples: a team the Actor is working with, altitude at which a scenario was attempted in a flight simulator. ステートメントに意味を補う文脈情報。例: アクタが所属するチームの情報。フライトシミュレータにおいて、あるシナリオが実行されたときの高度の情報。
timestamp	Date/Time	Timestamp (Formatted according to ISO 8601) of when the events described within this Statement occurred. If not provided, LRS should set this to the value of "stored" time. このステートメントで示されたイベントの発生時刻を示すタイムスタンプ (ISO 8601 形式に従う)。指定されない場合、LRS は "stored" のタイムスタンプをここに設定すべきである。
stored	Date/Time	Timestamp (Formatted according to ISO 8601) of when this Statement was recorded. Set by LRS. このステートメントが記録された時刻を示すタイムスタンプ (ISO 8601 形式に従う)。LRS によって設定される。
authority	Object	Agent who is asserting this Statement is true. Verified by the LRS based on authentication, and set by LRS if left blank. このステートメントが正しいものであると主張する Agent を示す。LRS の認証機構により確認され、指定がない場合は LRS によって設定される。

version	Version	The Statement's associated xAPI version, formatted according to Semantic Versioning 1.0.0. セマンティックバージョンング 1.0.0 形式で示したステートメントの xAPI バージョン。
attachments	Array of attachment Objects	Headers for attachments to the Statementステートメントに対する添付文書のヘッダー。

Aside from (potential or required) assignments of properties during LRS processing ("id", "authority", "stored", "timestamp", "version") Statements are immutable. Note that the content of Activities that are referenced in Statements is not considered part of the Statement itself. So while the Statement is immutable, the Activities referenced by that Statement are not. This means a deep serialization of a Statement into JSON will change if the referenced Activities change (see the Statement API's "format" parameter for details).

LRS によりプロパティ("id", "authority", "stored", "timestamp", "version") が割り当てられる場合を除き、ステートメントは不変である。ただし、ステートメント中で参照されているアクティビティの内容は、ステートメントそのものの一部とはみなさないことに注意が必要である。よって、ステートメントは不変だが、ステートメントによって参照されるアクティビティは不変ではない。これは、参照されるアクティビティが変更された時、ステートメントのディープコピーによって生成される JSON も変更されることを意味する(ステートメント API の "format" パラメータ参照)。

An example of the simplest possible Statement using all properties that MUST or SHOULD be used:

必須または推奨とされる全てのプロパティを用いる最も簡素なステートメントの例。

```
{
  "id": "12345678-1234-5678-1234-567812345678",
  "actor": {
    "mbox": "mailto:xapi@adlnet.gov"
  },
  "verb": {
    "id": "http://adlnet.gov/expapi/verbs/created",
    "display": {
      "en-US": "created"
    }
  },
  "object": {
    "id": "http://example.adlnet.gov/xapi/example/activity"
  }
}
```

See Appendix D: Example Statements for more examples. Appendix D: ステートメントの例 参照

4.1.1 id id (識別子)

Description説明

A UUID (see RFC 4122 for requirements, and the UUID must be in standard string form).

UUID(仕様は RFC 4122 参照。UUID は標準的な文字列形式でなければならない)

Requirements必要条件

- Ids MUST be generated by the LRS if a Statement is received without an id.受信ステートメントに ID が指定されていなかった場合、LRS が ID を生成しなければならない。
- Ids SHOULD be generated by the Activity Provider.ID はアクティブティブプロバイダによって生成されるべきである。

4.1.2 Actor Actor (アクタ)

Description説明

A mandatory Agent or Group Object.

必須の Agent または Group オブジェクト。

4.1.2.1 When the Actor ObjectType is Agent アクタのオブジェクト型が Agent の時

Description説明

An Agent (an individual) is a persona or system.

Agent(個人)は、人またはシステムである。

Details詳細

- An Agent MUST be identified by one (1) of the four types of Inverse Functional Identifiers (see 4.1.2.3 Inverse Functional Identifier);Agent は、逆関数識別子 (4.1.2.3 逆関数識別子 参照)の4つのタイプの1つによって定義されなければならない。
- An Agent MUST NOT include more than one (1) Inverse Functional Identifier;Agent に、1つを超える逆関数識別子を設定してはならない。
- An Agent SHOULD NOT use Inverse Functional Identifiers that are also used as a Group identifier.Agent は Group 識別子としても使われている逆関数識別子を使うべきではない。

The table below lists the properties of Agent Objects.

Agent オブジェクトのプロパティを以下の表に示す。

Propertyプロパティ	Typeタイプ	Description説明	Required必須
objectType	string	"Agent". This property is optional except when the Agent is used as a Statement's Object. "Agent". Agent が目的語の場合は必須である。	no
name	String	Full name of the Agent.Agent のフルネーム。	no
see 4.1.2.3 Inverse Functional Identifier 4.1.2.3 逆関数識別子参照		An Inverse Functional Identifier unique to the Agent.Agent に固有の逆関数識別子。	yes

4.1.2.2 When the Actor ObjectType is Group アクタのオブジェクト型が Group の時

Description説明

A Group represents a collection of Agents and can be used in most of the same situations an Agent can be used. There are two types of Groups, anonymous and identified.

Group は Agent の集合を意味し、Agent を指定できる状況の大部分において、使用することができる。Group は、匿名と指名の2種類がある。

Requirements必要条件

- A system consuming Statements MUST consider each Anonymous Group distinct even if it has an identical set of members.ステートメントを利用するシステムは、複数の匿名グループが同じメンバーで構成されている場合においても、それらを異なるものとして扱わなければならない。
- A system consuming Statements MUST NOT assume that Agents in the 'member' property comprise an exact list of Agents in a given anonymous or Identified Group.ステートメントを利用するシステムは、'member' プロパティに示された Agent が、与えられた匿名または指名のグループに属する Agent の厳密なリストであるとみなしてはならない。

Requirements for Anonymous Groups匿名グループに関する必要条件

- An Anonymous Group MUST include a 'member' property listing constituent Agents.匿名グループは、構成要員である Agent をリスト化した 'member' プロパティを持たなければならない。
- An Anonymous Group MUST NOT contain Group Objects in the 'member' property.匿名グループは、'member' プロパティにグループオブジェクトを含んではならない。
- An Anonymous Group MUST NOT include any Inverse Functional Identifiers.匿名グループは、いかなる逆関数識別子も含んではならない。

Requirements for Identified Groups指名グループに関する必要条件

- An Identified Group MUST include exactly one (1) Inverse Functional Identifier.指名グループは、厳密に1つの逆関数識別子を含まなければならない。
- An Identified Group MUST NOT contain Group Objects in the 'member' property.指名グループは、'member' プロパティにグループオブジェクトを含んではならない。
- An Identified Group SHOULD NOT use Inverse Functional Identifiers that are also used as Agent identifiers.指名グループは、Agent の識別子としても使われる逆関数識別子を使うべきではない。
- An Identified Group MAY include a 'member' property listing constituent Agents.指名グループは、構成要員である Agent をリスト化する 'member' プロパティを含んでもよい。

Details詳細

An Anonymous Group is used describe a cluster of people where there is no ready identifier for this cluster, e.g. an ad hoc team.

匿名グループは、一時的なチームなど、定められた名称が存在しない人々の集まりを示すために用いられる。

The table below lists all properties of an Anonymous Group.

匿名グループのすべてのプロパティを、下の表に示す。

Propertyプロパティ	Typeタイプ	Description説明	Required必須
objectType	String	"Group".	yes
name	String	Name of the group.グループの名称。	no
member	Array of Agent Objects	The members of this Group.このグループのメンバー。	yes

An Identified Group is used to uniquely identify a cluster of Agents.

指名グループは Agent の集合を一意に特定するために用いられる。

The table below lists all properties of an Identified Group.

指名グループのすべてのプロパティを、下の表に示す。

Propertyプロパティ	Typeタイプ	Description説明	Required必須
objectType	String	"Group".	yes
name	String	Name of the group.グループの名称。	no
member	Array of Agent Objects	The members of this Group.このグループのメンバー。	no
see 4.1.2.3 Inverse Functional Identifier 4.1.2.3 逆関数識別子参照		An Inverse Functional Identifier unique to the Group.グループを一意に示す逆関数識別子。	yes

4.1.2.3 Inverse Functional Identifier 逆関数識別子

Description説明

An "Inverse Functional Identifier" is a value of an Agent or Identified Group that is guaranteed to only ever refer to that Agent or Identified Group.

逆関数識別子は、その Agent や指名グループを参照することが、将来にわたり保証されている Agent や指名グループに関する値を示す。

Rationale背景

Learning experiences become meaningless if they cannot be attributed to identifiable individuals and/or groups. In an xAPI Statement this is accomplished with a set of Inverse Functional Identifiers loosely inspired on the widely accepted FOAF principle (see: Friend Of A Friend).

もし特定可能な個人やグループなどに関連付けられないのであれば、学習経験記録は意味のないものになってしまう。xAPI ステートメントでは、これを、広く受け入れられている FOAF 原則 (Friend Of A Friend 参照) を参考にした逆関数識別子の組み合わせによって実現する。

Details詳細

The table below lists all possible Inverse Functional Identifier properties.

逆関数識別子の取りうる全てのプロパティを以下の表に示す。

Propertyプロパティ	Typeタイプ	Description説明
mbox	mailto IRI	The required format is "mailto:email address". Only email addresses that have only ever been and will ever be assigned to this Agent, but no others, should be used for this property and mbox_sha1sum. 要求されるフォーマットは "mailto:email address" である。 この Agent に対して過去、未来を通じて割り当てられたメールアドレスのみを、このプロパティと mbox_sha1sum に利用すべきである。
mbox_sha1sum	String	The SHA1 hash of a mailto IRI (i.e. the value of an mbox property). An LRS MAY include Agents with a matching hash when a request is based on an mbox.mailto IRI (例:mbox プロパティの値)の SHA1 ハッシュ値。 LRS は、リクエストが mbox に関係する場合は、対応するハッシュ値をもつ Agent を含んでもよい。
openID	URI	An openID that uniquely identifies the Agent.Agent を一意に特定する OpenID。

account	Object	A user account on an existing system e.g. an LMS or intranet. LMS やイントラネットなどの既存のシステムにおけるユーザーアカウント。
---------	--------	--

Account Object Account オブジェクト

Description 説明

A user account on an existing system, such as a private system (LMS or intranet) or a public system (social networking site).

非公開のシステム(LMS またはイントラネット)や公開されたシステム(ソーシャルネットワーキングサイト)などの既存システムのユーザーアカウント。

Details 詳細

- If the system that provides the account Object uses OpenID, the Activity Provider SHOULD use the openID property instead of an account Object. アカウントオブジェクトを提供するシステムが OpenID を利用しているのであれば、アクティビティプロバイダはアカウントオブジェクトではなく、OpenID プロパティを利用すべきである。
- If the Activity Provider is concerned about revealing personally identifiable information about an Agent or Group, it SHOULD use an opaque account name (for example an account number) to identify all Statements about a person while maintaining anonymity. アクティビティプロバイダが Agent や Group について、個人を特定可能な情報を公開するのを懸念する場合は、匿名性を保ちつつその人についてのすべてのステートメントを識別可能にするために、意味を持たないアカウント名(例: アカウント番号)を利用すべきである。

The table below lists all properties of Account Objects.

Account オブジェクトのすべてのプロパティを以下の表に示す。

Property プロパティ	Type タイプ	Description 説明
homePage	IRL	The canonical home page for the system the account is on. This is based on FOAF's accountServiceHomePage. アカウントが利用されているシステムの正式のホームページ。これは、FOAF の accountServiceHomePage に基づく。
name	String	The unique id or name used to log in to this account. This is based on FOAF's accountName. このアカウントにログインするためのユニーク ID またはアカウント名。これは FOAF の accountName に基づく。

Example 例

This example shows an Agent identified by an opaque account:

意味を持たないアカウント名によって Agent を識別する例を示す。

```
{
  "objectType": "Agent",
  "account": {
    "homePage": "http://www.example.com",
    "name": "1625378"
  }
}
```

4.1.3 Verb Verb (動詞)

Description 説明

The Verb defines the action between Actor and Activity.

動詞はアクタとアクティビティとの間の行動を定義する。

Rationale 背景

The Verb in an xAPI Statement describes the action performed during the learning experience. The xAPI does not specify any particular Verbs. (With one exception, namely the reserved Verb 'http://adlnet.gov/expapi/verbs/voided'). Instead, it defines how to create Verbs so that communities of practice can establish Verbs meaningful to their members and make them available for use by anyone. A predefined list of Verbs would be limited by definition and might not be able to effectively capture all possible future learning experiences.

xAPI ステートメント中の動詞は、学習経験中に行われた行動を説明する。xAPI が動詞を特別に定義することはない。(例外として予約動詞 'http://adlnet.gov/expapi/verbs/voided' がある)。その代わりに、xAPI では実践コミュニティが、メンバー間での意思疎通のために動詞を作成し、さらには一般でも利用可能にするように動詞を作成する方法を定義している。あらかじめ定義された動詞をリスト化するという考えには制約があり、将来に亘って可能性のある学習経

験全てを効率的に扱うこともできないと考えられる。

Requirements 必要条件

Verbs appear in Statements as Objects consisting of an IRI and a set of display names corresponding to multiple languages or dialects which provide human-readable meanings of the Verb.

動詞はステートメント中にオブジェクトとして現れ、それは IRI と、動詞に関して人間が理解できる意味を複数の言語や方言に対応させた表示名とからなる。

- The display property MUST be used to illustrate the meaning which is already determined by the Verb IRI. display プロパティは、動詞 IRI により既に決められた意味を例示するために用いられなければならない。
- A system reading a Statement MUST use the Verb IRI to infer meaning. ステートメントを解釈するシステムは、意味の特定のために 動詞 IRI を利用しなければならない。
- The display property MUST NOT be used to alter the meaning of a Verb. display プロパティは、動詞の意味を変えるために利用されてはならない。
- A system reading a Statement MUST NOT use the display property to infer any meaning from the Statement. ステートメントを解釈するシステムは、ステートメントの意味の理解のために display プロパティを利用してはならない。
- A system reading a Statement MUST NOT use the display property for any purpose other than display to a human. Using the display property for aggregation or categorization of Statements is an example of violating this requirement. ステートメントを解釈するシステムは、display プロパティを、人間に対して提示する以外のいかなる用途にも用いてはならない。display プロパティを用いてステートメントを集約したりカテゴリ分けしたりすることは、この必要条件に対する違反の例となる。
- The display property SHOULD be used by all Statements. display プロパティは全てのステートメントに用意されるべきである。
- The IRI contained in the id SHOULD be human-readable and imply the Verb meaning. ID に含まれる IRI は人間が理解できる形で、動詞の意味を示すべきである。

Details 詳細

The table below lists all properties of the Verb Object.

以下の表に動詞オブジェクトの全てのプロパティを示す。

Property プロパティ	Type タイプ	Description 説明
id	IRI	Corresponds to a Verb definition. Each Verb definition corresponds to the meaning of a Verb, not the word. The IRI should be human-readable and contain the Verb meaning. 動詞の定義を示す。それぞれの動詞の定義は、単語ではなく動詞の意味に対応する。IRI は人間が理解できる形で、動詞の意味を含んでいるべきである。
display	Language Map 言語マップ	The human readable representation of the Verb in one or more languages. This does not have any impact on the meaning of the Statement, but serves to give a human-readable display of the meaning already determined by the chosen Verb. 一つ以上の言語で示された、人間が理解できる形での動詞の表現。ステートメントが表す意味について全く影響力を持たないが、選択された動詞によって既に決定されている意味を人間が理解できる形で表現する機能を提供する。

Example 例

```
{
  "verb" : {
    "id": "http://www.adlnet.gov/XAPIprofile/ran(travelled_a_distance)",
    "display": {
      "en-US": "ran",
      "es" : "corrió"
    }
  }
}
```

The Verb in the example above is included for illustrative purposes only. This is not intended to imply that a Verb with this meaning has been defined with this id. This applies to all example verbs given in this specification document, with the exception of the reserved Verb 'http://adlnet.gov/expapi/verbs/voided'.

上の例での動詞は例示目的でのみ示されている。これは、この意味を持った動詞がこの ID で定義されていることを意味するものではない。この原則は、予約動詞 ('http://adlnet.gov/expapi/verbs/voided') を除く、本仕様書のすべての動詞の例について適用される。

4.1.3.1 Use in Language and Semantics of Verbs 動詞の言語と意味に関する用法

Details 詳細

Semantics 意味

The IRI represented by the Verb id identifies the particular semantics of a word, not the word itself.

Verb ID によって表される IRI は、その単語そのものではなく、単語に関する特定の意味を示す。

For example, the English word "fired" could mean different things depending on context, such as "fired a weapon", "fired a kiln", or "fired an employee". In this case, an IRI MUST identify one of these specific meanings, not the word "fired".

例えば、英単語の "fired" は、"銃を撃つ(fired)" や "窯で焼く(fired)"、"従業員を解雇する(fired)" など、文脈に応じて異なる意味を持ち得る。この例においては、IRI は "fired" が単語として持ちうる意味ではなく、それらのうちの特定の1つの意味を示さなければならない。

The display property has some flexibility in tense. While the Verb IRIs are expected to remain in the past tense, if conjugating verbs to another tense (using the same Verb) within the Activity makes sense, it is allowed.

display プロパティでは時制に関してある程度の自由度を残している。動詞 IRI は過去形であることが期待されるが、対象のアクティビティについて(同じ動詞で)異なる時制にしたほうが妥当な場合は、そのようにしてもよい。

Language 言語

A Verb in the Experience API is an IRI, and denotes a specific meaning not tied to any particular language.

xAPI における動詞は IRI であり、いかなる言語にも依存しない固有の意味を示す。

For example, a particular Verb IRI such as <http://example.org/firearms#fire> might denote the action of firing a gun, or the Verb IRI <http://example.com/فعل/خواندن> might denote the action of reading a book.

例えば、<http://example.org/firearms#fire> のような特定の動詞 IRI は銃を撃つといった行動を意味し、<http://example.com/لعف/بندوخ> のような動詞 IRI は本を読むといった行動を意味する。

4.1.3.2 Use in Communities of Practice 実践コミュニティに関する用法

Requirements for Communities of Practice 実践コミュニティ向けの必要条件

- Anyone establishing a new Verb MUST own the IRI, or MUST have permission from the owner to use it to denote an xAPI Verb; 新しい動詞を定義する場合は IRI を所有しているか、xAPI の動詞を示すために IRI を利用する許可をその所有者から得なければならない。
- Anyone establishing a new Verb SHOULD make a human-readable description of the intended usage of the Verb accessible at the IRI. 新しい動詞を定義する場合は、動詞の想定される用途についての人間が理解できる定義を、IRI にてアクセス可能にしておくべきである。

Requirements for Activity Providers アクティビティプロバイダ向けの必要条件

- Activity Providers SHOULD use a corresponding existing Verb whenever possible. アクティビティプロバイダは、可能な限り既存の対応する動詞を利用すべきである。
- Activity Providers MAY create and use a Verb if no suitable Verb exists. アクティビティプロバイダは、適切な動詞が存在しない場合、動詞を作成し利用してもよい。

Details 詳細

Communities of practice will, at some point in time, need to establish new Verbs to meet the needs of their constituency.

実践コミュニティでは、その構成員の要求にこたえるために、どこかの段階で新たな動詞を定義する必要があるが出てくる。

Therefore, it is expected that xAPI generates profiles, lists, and repositories that become centered on Verb vocabularies. ADL is one such organization that is creating a companion document containing Verbs for xAPI. In fulfillment of the requirements above, a collection of IRIs of recommended Verbs exists. There are times when Activity Providers may wish to use a different Verb for the same meaning.

そのため、xAPI は動詞の語彙の中心となるプロフィール、リスト、リポジトリ等を作成することが望まれる。ADL もそのような組織の一つとして、xAPI における動詞についての解説文書を作成している。上の要求を満たすために、推奨される動詞の IRI 集が存在する。異なるアクティビティプロバイダ間で、同じ意味に対して、異なる動詞を利用したくなることも想定される。

4.1.4 Object (目的語)

Description 説明

The Object of a Statement can be an Activity, Agent/Group, Sub-Statement, or Statement Reference. It is the "this" part of the Statement, i.e. "I did this".

ステートメントにおける目的語は、アクティビティ、エージェント/グループ、サブステートメント、もしくはステートメントの参照などがあり得る。目的語は、ステートメントにおいて"対象"として表現される部分に当たる。例えば、"私はこれをした。(I did this)"のステートメントでは「これ(this)」にあたる。

Some examples:例:

- The Object is an Activity: "Jeff wrote an essay about hiking."目的語がアクティビティの場合:“ジェフはハイキングに関するエッセイを書いた”
- The Object is an Agent: "Nellie interviewed Jeff."目的語がエージェントの場合:“ネリーはジェフの面談を行った”
- The Object is a Sub-Statement or Statement Reference (different implementations, but similar when human-read): "Nellie commented on 'Jeff wrote an essay about hiking.'"目的語がサブステートメントもしくはステートメントの参照の場合(異なる手段ではあるが人が理解できる):“ジェフはハイキングに関するエッセイを書いた”についてネリーはコメントした。

Details詳細

Objects which are provided as a value for this field SHOULD include an "object

Type" field. If not specified, the objectType is assumed to be "Activity". Other valid values are: Agent, Group, Sub-Statement or StatementRef. The properties of an Object change according to the objectType.

このフィールドの値として提供されるオブジェクトは "objectType" フィールドを持つべきである。指定が無ければ、"objectType" は "Activity" と認識される。その他の有効な値は、Agent, Group, Sub-Statement もしくは StatementRef となるオブジェクトのプロパティは、objectType に応じて変わる。

4.1.4.1 When the ObjectType is Activity ObjectType がアクティビティの場合

A Statement may represent an Activity as the Object of the Statement. The following table lists the Object properties in this case.

ステートメントは、ステートメントの目的語としてアクティビティを示すことができる。本ケースにおけるオブジェクトのプロパティは以下の表の通り。

Propertyプロパティ	Typeタイプ	Description説明
objectType	String	MUST be "Activity" when present. Optional in all cases.利用する場合は“Activity”を設定しなければならない。全ての場合においてオプションとなる。
id	IRI	An identifier for a single unique Activity. Required.一意のアクティビティの識別子。必須。
definition	Object	Optional Metadata, See belowオプションのメタデータ、以下のアクティビティ定義を参照。

Activity Definitionアクティビティ定義

Details詳細

The table below lists the properties of the Activity Definition Object:アクティビティ定義オブジェクト

Propertyプロパティ	Typeタイプ	Use利用	Description説明
name	Language Map言語マップ	Recommended推奨	The human readable/visual name of the Activity可読/可視化したアクティビティの名称
description	Language Map言語マップ	Recommended推奨	A description of the Activityアクティビティの説明
type	IRI	Recommended推奨	The type of Activity.アクティビティのタイプ
moreInfo	IRL	Optional任意	SHOULD resolve to a document human-readable information about the Activity, which MAY include a way to 'launch' the Activity.アクティビティの「実施」の仕方を含むなど、アクティビティに関する(人が理解できる情報である)文書を指し示すべきである。
Interaction properties, See: Interaction ActivitiesInteraction propertiesについては、Interaction Activities を参照			
extensions	Object	Optional任意	A map of other properties as needed (see: Extensions)必要に応じて格納する他のプロパティに関するマップ (Extensions 参照)

Note注記: IRI fragments (sometimes called relative IRLs) are not valid IRIs. As with Verbs, it is recommended that Activity Providers look for and use established, widely adopted, Activity types.

IRI フラグメント (相対 IRL とも言う) は有効な IRI ではない。また、動詞と同じく、アクティビティプロバイダは、確立され広く適用されているアクティビティタイプを探して利用することが推奨される。

An LRS should update its internal representation of an Activity's definition upon receiving a Statement with the same Activity id, but with a different definition of the Activity from the one stored, but only if it considers the Activity Provider to have the authority to do so.

LRS は、同じアクティビティ ID のステートメントを受け取った場合で、記録されたものと異なるアクティビティの定義を受信した場合には、内部表現を更新するべきだが、アクティビティプロバイダがその権限を持つ場合に限る。

Activity id アクティビティ ID

Requirements for Activity Ids アクティビティ ID の必要条件

- An Activity id MUST be unique. アクティビティ ID は一意でなければならない。
- An Activity id MUST always reference the same Activity. アクティビティ ID は常に同じアクティビティを参照しなければならない。
- An Activity id SHOULD use a domain that the creator is authorized to use for this purpose. アクティビティ ID は作成者がこの目的のために許可されたドメインを使用すべきである。
- An Activity id SHOULD be created according to a scheme that makes sure all Activity ids within that domain remain unique. アクティビティ ID はそのドメイン内の全てのアクティビティ ID が一意になるようなスキームで作成されるべきである。
- An Activity id MAY point to metadata or the IRL for the Activity. アクティビティ ID はメタデータやアクティビティの IRL を示してもよい。

Requirements for the LRS LRS の必要条件

- An LRS MUST ignore any information which indicates two authors or organizations may have used the same Activity id. 2人の著者または2つの組織が同じアクティビティ ID を使用している可能性がある場合、LRS は全ての情報を無視しなければならない。
- An LRS MUST NOT treat references to the same id as references to different Activities. LRS は同じ ID への複数の参照を、異なるアクティビティへの複数の参照として取り扱ってはならない。
- Upon receiving a Statement with an Activity Definition that differs from the one stored, an LRS SHOULD decide whether it considers the Activity Provider to have the authority to change the definition and SHOULD update the stored Activity Definition accordingly if that decision is positive. 格納されたものとは異なるアクティビティ定義でステートメントを受信した場合、LRS はアクティビティプロバイダが定義を変更する権限を持っているかを判断すべきで、もし持っていると判断した場合は、記録されたアクティビティの定義を更新するべきである。
- An LRS MAY accept small corrections to the Activity's definition. For example, it would be okay for an LRS to accept spelling fixes, but it may not accept changes to correct responses. LRS は、アクティビティの定義に関する小修正を受け入れてもよい。例えば、スペルの訂正である。但し、正解を変更してはならない。

Requirements for the Activity Provider アクティビティプロバイダの必要条件

- An Activity Provider MUST ensure that Activity ids are not re-used across multiple Activities. アクティビティプロバイダは、アクティビティ ID が複数のアクティビティで重複利用されないことを保証しなければならない。
- An Activity Provider MUST only generate states or Statements against a certain Activity id that are compatible and consistent with states or Statements previously stored against the same id. アクティビティプロバイダは、以前に同じ ID に対して記録された状態またはステートメントと一致し、互換性のあるアクティビティ ID に対してのみ、その状態またはステートメントを生成しなければならない。
- An Activity Provider MUST NOT allow new versions (i.e. revisions or other platforms) of the Activity to break compatibility. アクティビティプロバイダは、アクティビティのバージョン更新時 (リビジョンや他のプラットフォームによる) に互換性を崩させてはならない。

Details 詳細

If it were possible to use the same id for two different Activities, the validity of Statements about these Activities could be questioned. This means an LRS may never treat (references to) the same Activity id as belonging to two different Activities, even if it thinks this was intended. Namely, when a conflict with another system occurs, it's not possible to determine the intentions.

2つの異なるアクティビティに同じ ID を使用するが可能であった場合、これらのアクティビティに関するステートメントの妥当性は疑問視される。これは、故意にそうされていたとしても、1つのアクティビティ ID を使用して2つの異なるアクティビティを区別できないを意味する。すなわち、他のシステムと矛盾が生じた場合、それが意図的かどうかを究明するはできない。

Activity Metadata アクティビティメタデータ

Requirements 必要条件

- If an Activity IRI is an IRL, an LRS SHOULD attempt to GET that IRL, and include in HTTP headers: "Accept: application/json, /". This SHOULD be done as soon as practical after the LRS first encounters the Activity id. もしアクティビティの IRI が IRL だった場合、"Accept: application/json, /" を HTTP ヘッダーに入れて、その IRL の GET を試みるべきである。これは LRS がアクティビティ ID を検知したら直ぐに実施しなければならない

ない。

- Upon loading JSON which is a valid Activity Definition from an IRL used as an Activity id, an LRS SHOULD incorporate the loaded definition into its internal definition for that Activity, while preserving names or definitions not included in the loaded definition. アクティビティ ID として使用した IRL から有効なアクティビティ定義の JSON をロードした際、LRS はロードした定義にない名前や定義を残しながら、ロードした定義をアクティビティの内部的な定義に組み込むべきである。
- An Activity with an IRL identifier MAY host metadata using the Activity Definition JSON format which is used in Statements, with a Content-Type of "application/json" IRL 識別子を持つアクティビティは、ステートメントで利用される Activity Definition JSON フォーマットを用いて Content-Type を "application/json" に設定したメタデータを提供してもよい。
- Upon loading any document from which the LRS can parse an Activity Definition from an IRL used as an Activity id, an LRS MAY consider this definition when determining its internal representation of that Activity's definition. アクティビティ定義の内部表現を決定する際に LRS はアクティビティ ID として使われる IRL からアクティビティ定義を解析することができ、そこから任意のドキュメントを読み込むときに、LRS はこの定義を考慮することができる。

Interaction Activities インタラクション アクティビティ

Rationale 背景

Traditional e-learning has included structures for interactions or assessments. As a way to allow these practices and structures to extend Experience API's utility, this specification includes built-in definitions for interactions, which borrows from the SCORM 2004 4th Edition Data Model. These definitions are intended to provide a simple and familiar utility for recording interaction data. These definitions are simple to use, and consequently limited. It is expected that communities of practice requiring richer interactions definitions will do so through the use of extensions to an Activity's type and definition.

従来の eラーニングはインタラクションとアセスメントが組み込まれた構造をもっている。それらの実情と構造を xAPI にも拡張するために、本仕様では SCORM 2004 第4版のデータモデルを参考にしたインタラクションの定義を含んでいる。これらの定義はインタラクションのデータを記録するためのシンプルで使い慣れた仕組みを提供することを目的としている。これらの定義はシンプルで使いやすいが、制約もある。より高機能なインタラクションの定義が必要な実践コミュニティは、アクティビティのタイプと定義の拡張を利用することによりそれを実現できる。

Requirements 必要条件

- Interaction Activities MUST have a valid interactionType. インタラクションアクティビティには、有効な interactionType が含まれていなければならない
- Interaction Activities SHOULD have the Activity type `http://adlnet.gov/expapi/activities/cmi.interaction`. インタラクションアクティビティは、アクティビティタイプとして、`http://adlnet.gov/expapi/activities/cmi.interaction` を持つべきである。
- An LRS, upon consuming a valid interactionType, MAY validate the remaining properties as specified in the table below and MAY return HTTP 400 "Bad Request" if the remaining properties are not valid for the Interaction Activity. 有効な interactionType を受信した場合、LRS は残りのプロパティを下記の表に沿って確認し、もし残りのプロパティがインタラクションアクティビティに対して有効でない場合は、HTTP 400 "Bad Request" を返してもよい。

Details 詳細

The table below lists the properties for Interaction Activities.

以下の表はインタラクションアクティビティのプロパティを表す。

Property プロパティ	Type タイプ	Description 説明
interactionType	String	As in "cmi.interactions.n.type" as defined in the SCORM 2004 4th Edition Run-Time Environment. SCORM2004 4th 版の Run-Time Environment で定義している "cmi.interactions.n.type"
correctResponsesPattern	An array of strings	Corresponds to "cmi.interactions.n.correct_responses.n.pattern" as defined in the SCORM 2004 4th Edition Run-Time Environment, where the final <i>n</i> is the index of the array. SCORM2004 4th 版の Run-Time Environment で定義している "cmi.interactions.n.correct_responses.n.pattern" に対応し、最後の <i>n</i> は配列のインデックスとなる
choices scale source target steps	Array of interaction components	Specific to the given interactionType (see below). interactionType (後述参照)にて特定

Interaction Components インタラクションのコンポーネント

Requirements 必要条件

- Within an array of interaction components, all id values MUST be distinct. インタラクションコンポーネントの配列中で、全ての ID は異なる値でなければならない。
- An interaction component's id value SHOULD not have whitespace. インタラクションコンポーネントの ID には空白があってはならない

Details 詳細

Interaction components are defined as follows:

インタラクションコンポーネントの定義は以下の通り

Property プロパティ	Type タイプ	Description 説明
id	String	A value such as used in practice for "cmi.interactions.n.id" as defined in the SCORM 2004 4th Edition Run-Time Environment SCORM 2004 4th 版の Run-Time Environment で定義している "cmi.interactions.n.id" で実際に使用する値
description	Language Map 言語マップ	A description of the interaction component (for example, the text for a given choice in a multiple-choice interaction) インタラクションコンポーネントの表現 (例えば、複数選択インタラクションで選択されたテキスト)

The following table shows the supported lists of CMI interaction components for an interaction Activity with the given interactionType.

インタラクションアクティビティにおける interactionType でサポートしている CMI インタラクションコンポーネントは以下の表の通り。

interactionType インタラクションタイプ	supported component list(s) サポートしているコンポーネントのリスト
choice, sequencing	choices
likert	scale
matching	source, target
performance	steps
true-false, fill-in, numeric, other	[No component lists defined]

See Appendix C for examples of Activity Definitions for each of the cmi.interaction types.

各 cmi.interaction タイプのアクティビティ定義は Appendix C のサンプルを参照。

4.1.4.2 When the "Object" is an Agent or a Group 目的語が単体のエージェントもしくはグループだった場合

Requirements 必要条件

- Statements that specify an Agent or Group as an Object MUST specify an 'objectType' property. 単体のエージェントもしくはグループを参照しているステートメントは必ず 'objectType' プロパティを明示しなくてはならない

See Section 4.1.2 Actor for details regarding Agents.

エージェントの詳細については Section 4.1.2 Actor (アクタ) を参照

4.1.4.3 When the "Object" is a Statement 目的語がステートメントだった場合

Rationale 背景

There are two possibilities for using a Statement as an Object. First, an Object can take on the form of a Statement that already exists by using a Statement Reference. A common use case for Statement References is grading or commenting on an experience that could be tracked as an independent event. The special case of voiding a Statement would also use a Statement Reference. Second, an Object can be brand new Statement by using a Sub-Statement. A common use case for Sub-Statements would be any experience that would be misleading as its own Statement. Each type is defined below.

目的語としてステートメントを取るのには、2つの場合がある。第一に、目的語は、既に存在するステートメントの参照を用いることにより、ステートメントの形をとることができる。ステートメント参照の一般的な例としては、独立のイベントとして扱うことができる経験に対する評価やコメントの付与があげられる。第二に、サブステートメントを利用することにより、目的語は独立したステートメントの形をとることができる。サブステートメントの一般的な利用法は、単独のステートメントでは誤解されるような経

験についてである。それぞれのタイプの定義は以下の通り。

Statement Referencesステートメント参照

Description説明

A Statement Reference is a pointer to another pre-existing Statement.

ステートメント参照は、他の既存ステートメントへのポインタである。

Requirements必要条件

- A Statement Reference MUST specify an "objectType" property with the value "StatementRef".ステートメント参照は、objectType プロパティとして "StatementRef" を指定しなければならない。
- A Statement Reference MUST set the "id" property to the UUID of a Statement.ステートメント参照は、ステートメントの UUID を "id" プロパティにセットしなければならない。

The table below lists all properties of a Statement Reference Object:

以下の表では、ステートメント参照オブジェクトの全プロパティを一覧表示している

Propertyプロパティ	Typeタイプ	Description説明
objectType	String	In this case, MUST be "StatementRef".この場合、"StatementRef" としなければならない
id	UUID	The UUID of a Statement. ステートメントの UUID

Example例

Assuming that some Statement has already been stored with the id 8f87ccde-bb56-4c2e-ab83-44982ef22df0, the following example shows how a comment could be issued on the original Statement, using a new Statement:

あるステートメントが ID 8f87ccde-bb56-4c2e-ab83-44982ef22df0 として既に記録されていると仮定したとき、以下の例では、どのようにして新しいステートメントで元のステートメントにコメントを発行するかを示している。

```
{
  "actor" : {
    "objectType": "Agent",
    "mbox": "mailto:test@example.com"
  },
  "verb" : {
    "id": "http://example.com/commented",
    "display": {
      "en-US": "commented"
    }
  },
  "object" : {
    "objectType": "StatementRef",
    "id": "8f87ccde-bb56-4c2e-ab83-44982ef22df0"
  },
  "result" : {
    "response": "Wow, nice work!"
  }
}
```

Sub-Statementsサブステートメント

A Sub-Statement is a new Statement included as part of a parent Statement.

サブステートメントは、親ステートメントの一部として含まれる新しいステートメントである

Requirements必要条件

- A Sub-Statement MUST specify an "objectType" property with the value "SubStatement".サブステートメントは、"objectType" プロパティで "SubStatement" として明示しなくてはならない
- A Sub-Statement MUST be validated as a Statement in addition to other Sub-Statement requirements.サブステートメントは、他のサブステートメントの必要要件に加えて、ステートメントとして評価されなければならない
- A Sub-Statement MUST NOT have the "id", "stored", "version" or "authority" properties.サブステートメントは、"id", "stored", "version" or

"authority"プロパティを持つてはならない

- A Sub-Statement MUST NOT contain a Sub-Statement of their own i.e. cannot be nested.サブステートメント内に、サブステートメントを含んではならない。すなわち、入れ子にはできない。

Example例

One interesting use of Sub-Statements is in creating Statements of intention. For example, using Sub-Statements we can create Statements of the form "`<I> <planned> (<I> <did> <this>)`" to indicate that we've planned to take some action. The concrete example that follows logically states that "I planned to visit 'Some Awesome Website'".

サブステートメントの興味深い使い方の一つは、意図を示すステートメントの構築である。例えば、サブステートメントを使って、何らかのアクションを起こそうとしたことを示す目的で "`<I> <planned> (<I> <did> <this>)`" といった形式のステートメントを作成することができる。次の具体例では、"I planned to visit 'Some Awesome Website'" を示している。

```
{
  "actor": {
    "objectType": "Agent",
    "mbox": "mailto:test@example.com"
  },
  "verb": {
    "id": "http://example.com/planned",
    "display": {
      "en-US": "planned"
    }
  },
  "object": {
    "objectType": "SubStatement",
    "actor": {
      "objectType": "Agent",
      "mbox": "mailto:test@example.com"
    },
    "verb": {
      "id": "http://example.com/visited",
      "display": {
        "en-US": "will visit"
      }
    }
  },
  "object": {
    "id": "http://example.com/website",
    "definition": {
      "name": {
        "en-US": "Some Awesome Website"
      }
    }
  }
}
```

4.1.5 Result Result(結果)

Description説明

An optional field that represents a measured outcome related to the Statement in which it is included.

ステートメントの取込み結果を計るためのオプションフィールド

Details詳細

The following table contains the properties of the Results Object.

以下の表には結果オブジェクトのプロパティが一覧表示されている

Propertyプロパティ	Typeタイプ	Description説明
score	Object	The score of the Agent in relation to the success or quality of the experience. See: Score エージェントのスコア。Score を参照
		Indicates whether or not the attempt on the Activity

success	Boolean	was successful. アクティビティの試みが成功したかどうかを表す
completion	Boolean	Indicates whether or not the Activity was completed. アクティビティが完了したかどうかを表す
response	String	A response appropriately formatted for the given Activity. アクティビティのために適切にフォーマットされたレスポンス
duration	Formatted according to ISO 8601 with a precision of 0.01 seconds ISO 8601 に基づいたステートメントが発生してからの経過時間 0.01秒の精度のフォーマット	Period of time over which the Statement occurred. ステートメントが発生してからの経過時間
extensions	Object	A map of other properties as needed. See: Extensions 必要に応じて追加される他のプロパティを表すマップ。Extensions を参照

4.1.5.1 Score スコア (Score)

Description 説明

An optional numeric field that represents the outcome of a graded Activity achieved by an Agent.

エージェントによって達成された、類別されたアクティビティの結果を表わすオプションの数字フィールド

Requirements 必要条件

- The Score Object SHOULD include 'scaled' if a logical percent based score is known. 論理的なパーセント基準のスコアが既知の場合は、スコアオブジェクトは 'scaled' を組み込むべきである。
- The Score Object SHOULD NOT be used for scores relating to progress or completion. Consider using an extension from an extension profile instead. スコアオブジェクトは、進捗もしくは完了に関連するスコアのために使用すべきではない。代わりに拡張プロファイルの拡張機能の利用を検討すること。

Details 詳細

The table below defines the Score Object.

以下の表は、スコアオブジェクトの定義である。

Property プロパティ	Type タイプ	Description 説明
scaled	Decimal number between -1 and 1, inclusive -1から1までの十進数	Cf. 'cmi.score.scaled' in SCORM 2004 4th Edition SCORM2004 4th 版の 'cmi.score.scaled' を参照
raw	Decimal number between min and max (if present, otherwise unrestricted), inclusive min と max の間の十進数 'cmi.score.raw' を参照 (現状、そうでなければ無制限)	Cf. 'cmi.score.raw' 'cmi.score.raw' を参照
min	Decimal number less than max (if present) max よりも小さい十進数	Cf. 'cmi.score.min' 'cmi.score.min' を参照
max	Decimal number greater than min (if present) min よりも大きい十進数	Cf. 'cmi.score.max' 'cmi.score.max' を参照

4.1.6 Context Context (文脈)

Description 説明

An optional field that provides a place to add contextual information to a Statement. All properties are optional.

文脈依存の情報をステートメントに付加するための任意のフィールド。全てのプロパティは任意である。

Rationale 背景

The "context" field provides a place to add some contextual information to a Statement. It can store information such as the instructor for an

experience, if this experience happened as part of a team Activity, or how an experience fits into some broader activity.

context フィールドは文脈依存の情報をステートメントに付加する機会を提供する。それは経験が集団活動の一部として行われた場合には、その経験の教授者の名前や、経験がより大きな活動にどのように組み込まれるか、といった情報を記録することができる。

Requirements 必要条件

- The revision property MUST NOT be used if the Statement's Object is an Agent or Group.
- The platform property MUST NOT be used if the Statement's Object is an Agent or Group.
- The language property MUST NOT be used if not applicable or unknown.
- The revision property SHOULD be used to track fixes of minor issues (like a spelling error).
- The revision property SHOULD NOT be used if there is a major change in learning objectives, pedagogy, or assets of an Activity. (Use a new Activity id instead).
- ステートメントの目的語がエージェントやグループであるなら、revision プロパティは使用してはいけない。
- ステートメントの目的語がエージェントやグループであるなら、platform プロパティは使用してはいけない。
- 適用できないか未知であるなら、language プロパティは使用してはいけない。
- revision プロパティは(綴り間違いのような)軽微な問題の修正を記録する際に用いるべきである。
- revision プロパティは、アクティビティの学習目標、教授法、または資産における大きな変化があるならば、用いるべきでない。そのような場合は新しいアクティビティを用いてほしい。

Details 詳細

Property プロパティ	Type タイプ	Description 説明
registration	UUID	The registration that the Statement is associated with. ステートメントに関連している登録
instructor	Agent (may be a group)	Instructor that the Statement relates to, if not included as the Actor of the statement. ステートメントのアクタに含まれない場合、ステートメントが関係する教授者
team	Group	Team that this Statement relates to, if not included as the Actor of the statement. ステートメントのアクタに含まれない場合、ステートメントが関係する集団
contextActivities	contextActivities Object	A map of the types of learning activity context that this Statement is related to. Valid context types are: "parent", "grouping", "category" and "other". このステートメントが関係する、学習活動コンテキストのタイプを持つマップ。有効なコンテキストタイプは "parent"、"grouping"、"category"、そして "other" である。
revision	String	Revision of the learning activity associated with this Statement. Format is free. このステートメントに関わる学習活動のリビジョン。形式は任意である。
platform	String	Platform used in the experience of this learning activity. この学習活動の経験で用いられるプラットフォーム。
language	String (as defined in RFC 5646)	Code representing the language in which the experience being recorded in this Statement (mainly) occurred in, if applicable and known. 利用可能または既知である場合、このステートメントで記録される経験が(主に)発生した際の言語コード。
statement	[Statement Reference] (#stmtref)	Another Statement which should be considered as context for this Statement. このステートメントのコンテキストと見なされる、他のステートメント。
extensions	Object	A map of any other domain-specific context relevant to this Statement. For example, in a flight simulator altitude, airspeed, wind, attitude, GPS coordinates might all be relevant (See Extensions) このステートメントに関連する、他の、ドメイン特化コンテキストのマップ。例えばフライトシミュレータにおける高度、対気速度、風、飛行姿勢、GPS 座標の全ては関連を持ちうる。(5.3 『拡張機能』参照)

Note 注記

Revision has no behavioral implications within the scope of xAPI. It is simply stored, so that it is available for reporting tools.

リビジョンは xAPI の範囲内で、何らかの動作に影響を与えることはない。それは、単に報告ツールが利用できるように記録される。

4.1.6.1 Registration Property Registration プロパティ

Description説明

An instance of a learner undertaking a particular learning activity.

特定の学習活動を行っている学習者のインスタンス。

Details詳細

When an LRS is an integral part of an LMS, the LMS likely supports the concept of registration. The xAPI applies the concept of registration more broadly. A registration could be considered to be an attempt, a session, or could span multiple Activities. There is no expectation that completing an Activity ends a registration. Nor is a registration necessarily confined to a single Agent.

LRS が LMS の不可欠な部分である場合、LMS は登録の概念を持つことが予想される。xAPI では、登録という概念はより広く適用される。登録は学習の試行や学習期間に関するものと見なすことができ、また一つの記録が登録のアクティビティにわたることも可能である。また、学習活動の完了が登録の終了を意味するとは限らない。そして登録は、一つのエージェントには限定されない。

4.1.6.2 ContextActivities Property ContextActivities プロパティ

Description説明

A map of the types of learning activity context that this Statement is related to.

このステートメントが関連する、学習活動コンテキストのタイプのマップ。

Rationale背景

Many Statements do not just involve one Object Activity that is the focus, but relate to other contextually relevant Activities. "Context activities" allow for these related Activities to be represented in a structured manner.

多くのステートメントは現在対象となる一つのオブジェクト・アクティビティのみを含むわけではなく、文脈上関連する他の活動にも関わる。"Context activities" はこれらの関連するアクティビティを構造化して表すことを可能にする。

Requirements必要条件

- Every key in the contextActivities Object MUST be one of parent, grouping, category, or other.
- Every value in the contextActivities Object MUST be either a single Activity object or an array of Activity objects.
- The LRS MUST return every value in the contextActivities Object as an array, even if it arrived as a single Activity object;
- The LRS MUST return single Activity Objects as an array of length one containing the same Activity.
- The Client SHOULD ensure that every value in the contextActivitiesObject is an array of Activity objects instead of a single Activity object.
- contextActivities オブジェクトの全てのキーは、parent、grouping、category、あるいは other の中の一つでなければならない。
- contextActivities オブジェクトの全ての値は、一つの Activity オブジェクトか、Activity オブジェクトの配列でなければならない。
- LRS は contextActivities オブジェクトの値は、それを単一の Activity オブジェクトとして受信した場合でも、配列として返さなければならない。
- LRS は単一の Activity オブジェクトを返す場合、その Activity を含む要素数1の配列として返さなければならない。
- クライアントは、contextActivities オブジェクト中の全ての値が Activity オブジェクトの配列であり、単独の Activity オブジェクトの形を取らないようにすべきである。

Details詳細

There are four valid context types. All, any or none of these MAY be used in a given Statement:

4つの有効なコンテキストタイプがある。対象のステートメント中で、これらの全てが使われたり、どれかが使われたり、あるいはどれも使われないことがある。

1. Parent: an Activity with a direct relation to the activity which is the Object of the Statement. In almost all cases there is only one sensible parent or none, not multiple. For example: a Statement about a quiz question would have the quiz as its parent Activity. 2. Grouping: an Activity with an indirect relation to the activity which is the Object of the Statement. For example: a course that is part of a qualification. The course has several classes. The course relates to a class as the parent, the qualification relates to the class as the grouping. 3. Category: an Activity used to categorize the Statement. "Tags" would be a synonym. Category SHOULD be used to indicate a "profile" of xAPI behaviors, as well as other categorizations. For example: Anna attempts a biology exam, and the Statement is tracked using the CMI-5 profile. The Statement's Activity refers to the exam, and the category is the CMI-5 profile. 4. Other: a context Activity that doesn't fit one of the other fields. For example: Anna studies a textbook for a biology exam. The Statement's Activity refers to the textbook, and the exam is a context Activity of type "other".

1. Parent: ステートメントの目的語であるアクティビティへの、直接の関わりを持つアクティビティ。ほとんどの場合、ただ一つの明確な親を持つか、あるいは一つも持たない。複数持つことはない。例えば、クイズの設問についてのステートメントは parent アクティビティとして「クイズ」を持つ。
2. Grouping: ステートメントの目的語であるアクティビティへの、間接的な関わりを持つアクティビティ。例: 資格取得の一部をなす講座。講座は複数回の講義で構成される。講座は講義に対して parent の関係にあり、資格取得は講義に対して grouping の関係にある。

3. Category: ステートメントを分類するためのアクティビティ。「タグ」と同義である。カテゴリは(他の分類と同様に) xAPI の挙動の "profile" を示すために用いられるべきである。例えば、アンナは生物学の試験に挑戦する。そしてステートメントは(訳注: AICC の) CMI-5 プロファイルを用いて記録される。そのステートメントのアクティビティは試験を参照し、そのカテゴリは CMI-5 プロファイルとなる。
4. Other: 他のフィールドのいずれにも合わない context アクティビティ。例えば、アンナは生物学の試験のために教科書を使い学んでいる。このステートメントのアクティビティは教科書を参照し、試験は other タイプのコンテキスト・アクティビティとなる。

Single Activity Objects are allowed as values so that 0.95 Statements will be compatible with 1.0.0.

0.95 版のステートメントが、1.0.0 版と互換性を保つために、単独の Activity オブジェクトを値として使っても構わない。

Note 注記

The values in this section are not for expressing all the relationships the Statement Object has. Instead, they are for expressing relationships appropriate for the specific Statement (though the nature of the Object will often be important in determining that). For instance, it is appropriate in a Statement about a test to include the course the test is part of as parent, but not to include every possible degree program the course could be part of in the grouping value.

この節は、ステートメントオブジェクトが持つ全ての関係性を説明するためのものではない。説明しているのは、(オブジェクトの性質がそれを決定するのにしばしば重要であるのだが)特定のステートメントに適合する関係性についてである。例えば、テストについてのステートメントに、そのテストを利用している講座を parent として含めるのは妥当だが、grouping の値として、関連の可能性がある全ての学位プログラムを並べるのは妥当とは言えない。

Example 例

Consider the following hierarchical structure: "Questions 1 to 6" are part of "Test 1" which in turn belongs to the course "Algebra 1". The six questions are registered as part of the test by declaring "Test 1" as their parent. Also they are grouped with other Statements about "Algebra 1" to fully mirror the hierarchy. This is particularly useful when the Object of the Statement is an Agent, not an Activity. "Andrew mentored Ben with context Algebra 1".

次のような階層構造を考えてほしい。「設問1から6」は「テスト1」に含まれ、同様にテスト1は「代数学1」講座に属している。6つの設問は「テスト1」を親と宣言することで、テストの一部として登録される。また、それらは「代数学1」の他のステートメントと共に grouping され、階層高層が完全に複写される。ステートメントの目的語が(アクティビティでなく)エージェントであるときに、これは特に有用である。「アンドリューは、代数学1のコンテキストでベンを指導した」。

```
... { "parent" : [ { "id" : "http://example.adlnet.gov/xapi/example/test 1" } ], "grouping" : [ { "id" : "http://example.adlnet.gov/xapi/example/Algebra1" } ] }
```

...

4.1.7 Timestamp Timestamp (タイムスタンプ)

Description 説明

The time at which a Statement was generated.

ステートメントが生成された時刻。

Requirements 必要条件

- A timestamp MUST be formatted according to ISO 8601.
- A timestamp SHOULD include the time zone.
- A timestamp SHOULD be the current or a past time when it is outside of a Sub-Statement.
- A timestamp MAY be truncated or rounded to a precision of at least 3 decimal digits for seconds (millisecond precision MUST be preserved).
- A timestamp MAY be a moment in the future, to denote a deadline for planned learning, provided it is included inside a Sub-Statement.
- timestamp は ISO 8601 に従う形式でなければならない。
- timestamp はタイムゾーンを含むべきである。
- サブ・ステートメントの外部である場合、timestamp は現在または過去の時刻となるべきである。
- timestamp は、秒を少なくとも3桁の精度で切り捨てられるか、あるいは丸められてもよい(ミリ秒の精度は保たなければならない)。
- timestamp は、それがサブ・ステートメントに含まれるならば、計画した学習の期限を示すために未来の時刻としてもよい。

Details 詳細

A timestamp in a Statement related to learning that occurs outside of the system can differ from Stored (the system time of the event). Namely, there can be delays between the occurrence of the experience and the reception of the corresponding Statement by the LRS. Another cause is when Statements are propagated to other systems.

システムの外部で発生する学習に関わるステートメントのタイムスタンプは、システムが保存していた時刻とは異なる場合がある。たとえば、経験が発生してから LRS がステートメントを受信するまでの間に、遅延が発生することがある。もう一つの理由として、ステートメントが他のシステムに伝播されるときに遅延がある。

4.1.8 Stored Stored

Description説明

The time at which a Statement is stored by the LRS.

ステートメントが LRS に記録された時刻。

The stored property is the literal time the Statement was stored. Use Timestamp to track a time at which the Statement was generated.

stored プロパティは、ステートメントが記録されたときの正確な時刻である。ステートメントが発生したときの時刻を記録する際は、Timestamp を使ってほしい。

Requirements必要条件

- The stored property MUST be formatted according to ISO 8601.
- The stored property SHOULD include the time zone.
- The stored property SHOULD be the current or a past time.
- The stored property MAY be truncated or rounded to a precision of at least 3 decimal digits for seconds (millisecond precision MUST be preserved).
- stored プロパティは ISO 8601 に従う形式でなければならない。
- stored プロパティはタイムゾーンを含むべきである。
- stored プロパティは現在または過去の時刻となるべきである。
- stored プロパティは、秒を少なくとも3桁の精度で切り捨てられるか、あるいは丸められてもよい(ミリ秒の精度は保たなければならない)。

4.1.9 AuthorityAuthority (権限)

Description説明

The authority property provides information about whom or what has asserted that this Statement is true.

authority プロパティは「誰、または何がこのステートメントが真であると主張するか」の情報を提供する。

Requirements必要条件

- Authority MUST be an Agent, except in 3-legged OAuth, where it must be a Group with two Agents. The two Agents represent an application and user together.
- The LRS MUST include the user as an Agent as the entire authority if the user connects directly (using HTTP Basic Authentication) or is included as part of a Group.
- The LRS MUST ensure that all Statements stored have an authority.
- The LRS SHOULD overwrite the authority on all stored received Statements, based on the credentials used to send those Statements.
- The LRS MAY leave the submitted authority unchanged but SHOULD do so only where a strong trust relationship has been established, and with extreme caution.
- The LRS MAY identify the user with any of the legal identifying properties if a user connects directly (using HTTP Basic Authentication) or a part of a 3-legged OAuth.
- Authority は 3-legged OAuth を除き、2つのエージェントによってグループが形成されてしまう場面においては、1個のエージェントが該当しなければならない。この2つのエージェントは、アプリケーションとユーザを表す。
- ユーザが HTTP ベーシック認証を用いて直接接続する場合やグループの一員である場合、LRS は全体の権限を持つエージェントとしてユーザを取り込まなければならない。
- LRS は記録された全てのステートメントが権限を持っていることを確認しなければならない。
- LRS はこれらのステートメントを送信するための資格情報に基づいて、全ての記録された受信済ステートメントにおける権限を上書きすべきである。
- LRS は受信した権限を変更せずに残してもよいが、強い信頼関係が確立された場合に限るべきであり、また残す際には細心の注意を払う必要がある。
- ユーザが HTTP ベーシック認証を用いて直接接続する場合、あるいは 3-legged OAuth の一部である場合、LRS は正当な識別プロパティによってユーザを識別してもよい。

Details詳細

The asserting authority represents the authenticating user of some system or application.

権限を主張するということは、システムやアプリケーション上でのユーザを認証することを意味する。

OAuth Credentials as Authority権限としての OAuth 資格情報

Description説明

This is a workflow for use of OAuth. 2-legged and 3-legged OAuth are both supported.

これは OAuth 利用のフローである。2-legged と 3-legged の OAuth がサポートされる。

Requirements 必要条件

- The authority MUST contain an Agent Object that represents the OAuth consumer, either by itself, or as part of a group in the case of 3-legged OAuth.
- The Agent representing the OAuth consumer MUST be identified by account.
- The Agent representing the OAuth consumer MUST use the consumer key as the "account name" field.
- If the Agent representing the OAuth consumer is a registered application, the token request endpoint MUST be used as the account homePage.
- If the Agent representing the OAuth consumer is not a registered application, the temporary credentials endpoint MUST be used as the account homePage.
- An LRS MUST NOT trust the application portion of the authority in the event the account name is from the same source as the unregistered application. (Multiple unregistered applications could choose the same consumer key. As a result, there is no consistent way to verify this combination of temporary credentials and the account name.)
- Each unregistered consumer SHOULD use a unique consumer key.
- authority は OAuth consumer を示すエージェントオブジェクトか、エージェントオブジェクトそのもの、もしくは 3-legged OAuth の場合にはグループの一部としてのエージェントオブジェクトを含まなければならない。
- OAuth consumer を表すエージェントは、アカウントによって識別されなければならない。
- OAuth consumer を示すエージェントは、「アカウント名」のフィールドとして consumer キーを用いなければならない。
- OAuth consumer を示すエージェントが登録済アプリケーションであるならば、トークンリクエストエンドポイントはアカウントのホームページとして用いられなければならない。
- OAuth consumer を示すエージェントが登録済アプリケーションでない場合は、一時的な資格情報のエンドポイントがアカウントのホームページとして用いられなければならない。
- LRS は、アカウント名が未登録のアプリケーションと同一のソースによる権限のアプリケーション部分を、信頼してはならない。(複数の未登録アプリケーションは、同じ consumer キーを選択できる。その結果として、一時的な資格情報とアカウント名の組合せを検証するための、信頼できる方法は存在しない。)
- 未登録のそれぞれの consumer は、一意な consumer キーを使うべきである。

Details 詳細

This workflow assumes a Statement is stored using a validated OAuth connection and the LRS creates or modifies the authority property of the Statement.

このワークフローは、ステートメントが検証済 OAuth コネクションを用いて格納され、LRS がステートメントの authority プロパティを作成、または変更することを仮定している。

In a 3-legged OAuth workflow, authentication involves both an OAuth consumer and a user of the OAuth service provider. For instance, requests made by an authorized Twitter plug-in on their Facebook account will include credentials that are specific not only to Twitter as a Client application, or them as a user, but the unique combination of both.

3-legged OAuth ワークフローでは、認証は "OAuth consumer" と「OAuth サービスプロバイダのユーザ」の両方を含む。例えば、Facebook アカウントにおける、許可された Twitter プラグインからの要求は、クライアントアプリケーションとしての Twitter やユーザに対するだけでなく、両者の固有の結びつきに対する資格情報を含んでいる。

Example 例

The pairing of an OAuth consumer and a user. OAuth consumer とユーザのペア。

```
... "authority": { "objectType": "Group", "member": [ { "account": { "homePage": "http://example.com/xAPI/OAuth/Token",  
"name": "oauthconsumerx75db" } }, { "mbox": "mailto:bob@example.com" } ] }
```

...

4.1.10 VersionVersion (バージョン)

Description 説明

Version information in Statements helps systems that process data from an LRS get their bearings. Since the Statement data model is guaranteed consistent through all 1.0.x versions, in order to support data flow among such LRSs the LRS is given some flexibility on Statement versions that are accepted.

LRS からのデータを処理するシステムは、ステートメントのバージョン情報によって、それらの挙動を決めることができる。ステートメントデータモデルが全てのバージョン 1.0.x を通じて一貫性が保証されるので、LRS 間のデータフローをサポートするために、LRS には受け入れられるステートメントのバージョンにおける柔軟性が与えられる。

Requirements 必要条件

- Version MUST be formatted as laid out for the API version header in API Versioning.バージョンは API Versioning 仕様中の API バージョンヘッダのレイアウト形式に従わなければならない。

LRS Requirements LRS の必要条件

- An LRS MUST accept all Statements where their version starts with "1.0." if they otherwise validate.
- An LRS MUST reject all Statements with a version specified that does not start with "1.0."
- Statements returned by an LRS MUST retain the version they are accepted with. If they lack a version, the version MUST be set to 1.0.0
- 有効なステートメントについて、LRS は "1.0." で始まるバージョンの全てのステートメントを受け入れなければならない。
- "1.0." から始まらないように指定されたバージョンの全てのステートメントを、LRS は拒否しなければならない。
- LRS によって返されるステートメントは、受け入れられたときのバージョンを保持しなければならない。バージョン情報が存在しないなら、バージョンは 1.0.0 に設定されなければならない。

Client Requirements クライアントの必要条件

- If Clients set the Statement version, they MUST set it to 1.0.0
- Clients SHOULD NOT set the Statement version.
- クライアントがステートメントのバージョンを設定するならば、それは 1.0.0 でなければならない。
- クライアントはステートメントのバージョンを設定すべきではない。

4.1.11 Attachments Attachments (添付資料)

Description 説明

A digital artifact providing evidence of a learning experience. 学習経験の証跡を提供するデジタル文書

Rationale 背景

In some cases an attachment may logically be an important part of a learning record. Think of a simulated communication with ATC, an essay, a video, etc. Another example of such an attachment is (the image of) a certificate that was granted as a result of an experience. It is useful to have a way to store these attachments in and retrieve them from an LRS. 場合により、添付文書は論理的に学習記録の重要な部分となる可能性がある。航空管制との通信シミュレーション、エッセイ、ビデオなどを考えてみてほしい。添付文書の他の例としては、経験の結果として与えられた修了証書(の画像)がある。これらの添付文書を LRS に記録したり LRS から読み出したりする方法があることは有益である。

Requirements for Attachment Statement Batches 添付文書ステートメントバッチのための必要条件

A Statement batch, Statement results, or single Statement that includes attachments: 添付文書を含むステートメントバッチ、ステートメントの結果、または、1つのステートメントは、以下の条件を満たす:

- MUST be of type "application/json" and include a fileUrl for every attachment EXCEPT for Statement results when the attachments filter is false or 添付文書フィルタが false の場合のステートメントの結果を除き、"application/json" タイプで、全ての添付文書毎に fileUrl を含まなければならない。または
- MUST conform to the definition of multipart/mixed in RFC 1341 and: RFC1341 における multipart/mixed の定義に準拠しなければならない。かつ、以下の条件を満たさなければならない:
 - The first part of the multipart document MUST contain the Statements themselves, with type "applicaton/json". multipart 文書の最初にはステートメント自身が "application/json" タイプで含まれる。
 - Each additional part contains the raw data for an attachment and forms a logical part of the Statement. This capability will be available when issuing PUT or POST against the Statement resource. 他の追加部分は、添付文書の生のデータを含み、ステートメントの論理部分を形成する。この機能はステートメントリソースに対して PUT や POST が発行されたときに利用可能である。
 - MUST include a X-Experience-API-Hash field in each part's header after the first (statements) part. 最初の(ステートメント)部分に続く各部分のヘッダーには X-Experience-API-Hash フィールドを含まなければならない。
 - This field MUST be set to match the "sha2" property of the attachment declaration corresponding to the attachment included in this part. このフィールドは、この部分に含まれる添付文書に一致した添付文書宣言の "sha2" プロパティと一致していなければならない。
 - MUST include a Content-Transfer-Encoding field with a value of "binary" in each part's header after the first (statements) part. 最初の(ステートメント)部分に続くそれぞれのパートのヘッダーに、"Content-Transfer-Encoding" の値として "binary" を含めなければならない。
 - SHOULD only include one copy of an attachment's data when the same attachment is used in multiple Statements that are sent

together. 複数のステートメントが同時に送られ、同じ添付文書が使われた時は、1つの添付文書のデータだけを含むべきである。

- SHOULD include a Content-type field in each part's header, for the first part this MUST be "application/json". 各部分のヘッダーには Content-type フィールドを含むべきであり、最初の部分は application/json タイプにしなければならない。

Requirements for the LRSLRSの必要条件

- An LRS MUST include attachments in the Transmission Format described above when requested by the Client (see Section 7.2 "Statement API"). クライアントから要求があった場合は、LRS は上記に示す転送形式で添付文書を含まなければならない。(7.2 ステートメント API参照)
- An LRS MUST NOT pull Statements from another LRS without requesting attachments. LRS は添付文書の要求なしでは他の LRS からステートメントを引き出してはならない。
- An LRS MUST NOT push Statements into another LRS without including attachment data received, if any, for those attachments. LRS は、受け取った添付文書データがある場合は、それらを添付することなく、ステートメントを他の LRS に送ってはならない。
- When receiving a PUT or POST with a document type of "application/json", "application/json" のドキュメントタイプの文書を PUT や POST で受け取る場合、
 - An LRS MUST accept batches of Statements which contain either no attachment Objects, or only attachment Objects with a populated fileUrl. LRS は、添付文書オブジェクトを含まないか、あるいは投入された fileUrl を持つ添付文書オブジェクトのみを含む場合に、ステートメントのバッチを受入れなければならない。
- Otherwise: 上記以外の時に:
 - An LRS MUST accept batches of Statements via the Statements resource PUT or POST that contain attachments in the Transmission Format described above. LRS は、上記で示した転送フォーマットでの添付文書を含むステートメントリソースの PUT または POST 経由でステートメントのバッチを受入れなければならない。
 - An LRS MUST reject batches of Statements having attachments that neither contain a fileUrl nor match a received attachment part based on their hash. LRS は、fileUrl を含まないか、ハッシュに基づき受け取った添付文書の部分と一致しない添付文書を持つステートメントのバッチを拒否しなければならない。
 - An LRS SHOULD assume a Content-Transfer-Encoding of binary for attachment parts. LRS は添付文書の部分に対してバイナリの Content-Transfer-Encoding を前提とすべきである。
- An LRS MAY reject (batches of) Statements that are larger than the LRS is configured to allow. LRS は、その LRS が許可された設定より大きいステートメント(のバッチ)を拒否することができる。

Note注記: There is no requirement that Statement batches using the mime/multipart format contain attachments. mime/multipart フォーマットを使うステートメントバッチが添付文書を含む場合は、何の必要条件もない。

Requirements for the Clientクライアントの必要条件

- The Client MAY send Statements with attachments as described above. クライアントは上記で記述された添付文書を含むステートメントを送ることができる。
- The Client MAY send multiple Statements where some or all have attachments if using "POST". クライアントは、POST を利用するときに、そのうちのいくつかは添付文書を含むかあるいはすべてが添付文書を含む複数のステートメントを送ることができる。
- The Client MAY send batches of type "application/json" where every attachment Object has a fileUrl, ignoring all requirements based on the "multipart/mixed" format. クライアントは、"multipart/mixed" 形式に基づく全ての必要条件を無視し、全ての添付文書オブジェクトが fileUrl を持つ "application/json" タイプのバッチを送ることができる。

Details詳細

The table below lists all properties of the Attachment Object. 以下のテーブルは添付文書オブジェクトのすべてのプロパティを示す。

Propertyプロパティ	Typeタイプ	Description説明	Required必須
usageType	IRI	Identifies the usage of this attachment. For example: one expected use case for attachments is to include a "completion certificate". A type IRI corresponding to this usage should be coined, and used with completion certificate attachments. この添付文書の利用方法を規定する。例えば、添付文書の期待されるユースケースの1つに「修了証明」を含むことがある。この用途に対応するタイプ IRI を作成し、修了証明添付文書と一緒に利用されるべきである。	yes
display	Language Map	Display name (title) of this attachment.この添付文書の名前(タイトル)を表示する。	yes
description	Language Map	A description of the attachment添付文書の説明	no

contentType	Internet Media Type	The content type of the attachment. 添付文書のコンテンツタイプ	yes
length	integer	The length of the attachment data in octets オクテットで示した添付文書データの長さ	yes
sha2	base64	The SHA-2 hash of the attachment data. A minimum key size of 256 bits is recommended. 添付文書データの SHA-2 ハッシュ。最低256ビット以上のキーが推奨される。	yes
fileUrl	IRL	an IRL at which the attachment data may be retrieved, or from which it used to be retrievable. 添付文書データが取り出される IRL、あるいは、取り出しが可能であった IRL。	no

Procedure for the exchange of attachments 添付文書交換の手続き

1. A Statement including an attachment is construed according to the Transmission Format described below. 添付文書を含むステートメントは、以下に記述する転送フォーマットにより構成される。
2. The Statement is sent to the receiving system using a Content-Type of "multipart/mixed". The attachments are placed at the end of such transmissions. ステートメントは、"multipart/mixed" のコンテンツタイプで、受け側のシステムに送られる。添付文書は転送の最後に配置される。
3. The receiving system decides whether to accept or reject the Statement based on the information in the first part. 受け側のシステムは、最初の部分にある情報に基づき、ステートメントを受入れるか拒否するかを決定する。
4. If it accepts the attachment, it can match the raw data of an attachment with the attachment header in a Statement by comparing the SHA-2 of the raw data to the SHA-2 declared in the header. It MUST not do so any other way. 仮に添付文書を受入れたときに、生データの SHA-2 とヘッダーで宣言された SHA-2 とを比較することで、ステートメントにある添付文書ヘッダーと生のデータを照合することができる。他の方法で照合してはならない。

Example 例

Below is an example of a very simple Statement with an attachment. Please note the following: 添付文書を含むステートメントの非常に簡単な例である。以下の点に注意してほしい:

- The boundary in the sample was chosen to demonstrate valid character classes; 以下の例の boundary は有効な文字クラスを例示するために選択されたものである。
- The selected boundary does not appear in any of the parts; 選択された boundary は、エンコードされた各添付文書のどの部分にもマッチしない。
- For readability the sample attachment is text/plain. Even if it had been a 'binary' type like 'image/jpeg' no encoding would be done, the raw octets would be included; 読みやすくするために、添付文書の例は text/plain にしている。仮に、image/jpeg のようにバイナリタイプであったとしても、符号化されず、生のオクテットデータがそのまま含まれる。
- Per RFC 1341, the boundary is followed by -- followed by the boundary string declared in the header. RFC 1341 より、boundary は <CRLF> のあとに、さらにその後のヘッダーで定義された boundary 文字列を続けたものとなる。

Don't forget the <CRLF> when building or parsing these messages. これらのメッセージを構築あるいは構文解析する場合に<CRLF>を忘れてはいけない。

Headersヘッダー:

```
Content-Type: multipart/mixed; boundary=abcABC0123'()+_,-./:=?
X-Experience-API-Version:1.0.0
```

Contentコンテンツ:

```
--abcABC0123'()+_,-./:=?
Content-Type:application/json
{
  "actor": {
    "mbox": "mailto:sample.agent@example.com",
    "name": "Sample Agent",
    "objectType": "Agent"
  },
  "verb": {
    "id": "http://adlnet.gov/expapi/verbs/answered",
    "display": {
      "en-US": "answered"
    }
  },
  "object": {
```

```

    "id": "http://www.example.com/tincan/activities/multipart",
    "objectType": "Activity",
    "definition": {
      "name": {
        "en-US": "Multi Part Activity"
      },
      "description": {
        "en-US": "Multi Part Activity Description"
      }
    }
  },
  "attachments": [
    {
      "usageType": "http://example.com/attachment-usage/test",
      "display": { "en-US": "A test attachment" },
      "description": { "en-US": "A test attachment (description)" },
      "contentType": "text/plain; charset=ascii",
      "length": 27,
      "sha2": "495395e777cd98da653df9615d09c0fd6bb2f8d4788394cd53c56a3bfdcd848a"
    }
  ]
}
--abcABC0123'()+_,-./:=?
Content-Type:text/plain
Content-Transfer-Encoding:binary
X-Experience-API-Hash:495395e777cd98da653df9615d09c0fd6bb2f8d4788394cd53c56a3bfdcd848a
here is a simple attachment
--abcABC0123'()+_,-./:=?--

```

4.1.12 Data Constraints データの制約

Description説明

All the properties used in Statements are restricted to certain types, and those types constrain the behavior of systems processing Statements. For clarity, certain key requirements are documented here, emphasizing where compliant systems have a responsibility to act in certain ways. ステートメントで使われる全てのプロパティはあるタイプに制限されている。また、それらのタイプはステートメントを処理するシステムの動作を制約している。明確化のために、どの部分において、準拠するシステムが特定の振る舞いを行う責任があるかを強調し、いくつかの主な必要条件をここに記述する。

Requirements for the Clientクライアントの必要条件

The following requirements reiterate especially important requirements already included elsewhere, to emphasize, clarify, and provide implementation guidance. 実行ガイダンスを強調し、明確化し、提供するために、以下の必要条件は、他にすでに含まれている特に重要な必要条件を繰り返している。

- Values requiring IRIs MUST be sent with valid IRIs. Please use a library to construct them instead of string concatenation. Complete IRI validation is extremely difficult, so much of the burden for ensuring data portability is on the Client. 値として送信される IRI は有効なものでなければならない。IRI の生成には、文字列結合ではなく、ライブラリを利用すべきである。IRI の完全な検証は非常に難しいため、データの可搬性を担保する責任の大部分はクライアント側にある。
- Keys of language maps MUST be sent with valid RFC 5646 language tags, for similar reasons. 同様の理由から、言語マップのキーは、有効な RFC 5646 言語タグと一緒に送られなければならない。

Requirements for the LRS LRS の必要条件

- The LRS MUST reject Statements 以下の場合、LRS はステートメントを拒否しなければならない。
 - with any null values (except inside extensions). 値がない場合 (拡張の中を除く)
 - with strings where numbers are required, even if those strings contain numbers. 数値が必要な場所に文字列が与えられている場合。(その文字列に数字が含まれる場合でも)
 - with strings where booleans are required, even if those strings contain booleans. 2値が必要な場所に文字列が与えられている場合。(その文字列に2値が含まれる場合でも)
 - with any non-format-following key or value, including the empty string, where a string with a particular format (such as mailto, UUID, or IRI) is required. 特別な形式 (mailto, IRI, UUID, または IRI) の文字が必要な場所に、空文字列を含む、形式に合わないキーまたは値が与えられている場合。
 - where the case of a key does not match the case specified in the standard. あるキーのケースが標準で規定されたケースに合わない場合。

- where the case of a value restricted to enumerated values does not match an enumerated value given in the standard exactly. 値のケースが数字の値に制限されていて、標準で示された数字の値に一致しない場合。
- The LRS MUST reject Statements containing IRL, IRI, or IRI values without a scheme. LRS はスキーマを持たない、IRL、IRI、または IRI の値を含むステートメントを拒否しなければならない。
- The LRS MUST at least validate that the sequence of token lengths for language map keys matches the RFC 5646 standard. LRS は少なくとも、言語マップキーのトークンの長さのシーケンスが RFC 5646 標準に一致することを検証しなければならない。
- The LRS MUST process and store numbers with at least the precision of IEEE 754 32-bit floating point numbers. LRS は、数字を最低限 IEEE754 32ビットの精度で処理して記録しなければならない。
- The LRS MUST validate parameter values to the same standards required for values of the same types in Statements. LRS はステートメントにある同じタイプの値の検証と同じ基準で、パラメータの値の検証を行わなければならない。 **Note 注記:** string parameter values are not quoted as they are in JSON. JSONとは異なり、文字列パラメータの値は引用符でくられない。
- The LRS MAY use best-effort validation for IRL, IRI, and IRI formats to satisfy the non-format-following rejection requirement. LRS が形式不適合による拒絶の必要条件を満たすためには、IRL, IRI, および IRI フォーマットについてベストエフォートの検証をすればよい。
- The LRS MAY use best-effort validation for language map keys to satisfy the non-format-following rejection requirement. LRS は、言語マップキーが形式に合わない要求拒否を満たすために、ベストエフォート検証を使うことができる。

4.2 Retrieval of Statements ステートメントの検索

Description説明

A collection of Statements can be retrieved by performing a query on the "statements" endpoint, see Section 7.2 "Statement API" for details. 一連のステートメントは検索クエリを "statements" とすることで、検索が可能となる。詳細は7.2 ステートメント APIを参照。

Details詳細

The following table shows the data structure for the results of queries on the Statement API. 下の表はステートメント API の検索結果のためのデータ構造を示している。

Propertyプロパティ	Typeタイプ	Description説明
statementsstatements	Array of Statements Array of Statements	List of Statements. If the list returned has been limited (due to pagination), and there are more results, they will be located at the "statements" property within the container located at the IRL provided by the "more" element of this Statement result Object. ステートメントのリスト。さらに結果があるのに、戻ってきたリストが(ページ制限により)制限されていたら、コンテンツに含まれる "statements" プロパティに配置されるだろう。コンテンツは、ステートメント結果オブジェクトのさらなる要素によって提供される IRL に置かれる。
more	IRL	Relative IRL that may be used to fetch more results, including the full path and optionally a query string but excluding scheme, host, and port. Empty string if there are no more results to fetch. 追加の結果を取り込むために用いられる相対 IRL。フルパスとオプションのクエリ文字列で、スキーム、ホスト、ポートの情報を含まない。 This IRL must be usable for at least 24 hours after it is returned by the LRS. In order to avoid the need to store these IRLs and associated query data, an LRS may include all necessary information within the IRL to continue the query, but should avoid generating extremely long IRLs. The consumer should not attempt to interpret any meaning from the IRL returned. この IRL は LRS によって戻された後、少なくとも24時間は利用可能でなければならない。これらの IRL と関連するクエリデータを保存しておく必要性を避けるために、LRS はクエリを実行するのに必要な全ての情報を IRL に含めてもよいが、非常に長い IRL の生成は避けるべきである。利用者は IRL の戻り値から意味を解釈すべきではない。

4.3 Voided 無効

Rationale背景

The certainty that an LRS has an accurate and complete collection of data is guaranteed by the fact that Statements cannot be logically changed or deleted. This immutability of Statements is a key factor in enabling the distributed nature of Experience API. ステートメントが論理的に変更されたり削除されたりすることは無い、という事実によって、LRS が正確で完全なデータを収集しているという確実性が保証される。このステートメントの不変性が、xAPI の分散性を可能にする重要な要因である。

However, not all Statements are perpetually valid once they have been issued. Mistakes or other factors could require that a previously made Statement is marked as invalid. This is called "voiding a Statement" and the reserved Verb "http://adlnet.gov/expapi/verbs/voided" is used for

this purpose. Any Statement that voids another cannot itself be voided. しかしながら、すべてのステートメントが、一度発行されると永久に有効である、ということではない。誤りや他の要因で、以前に作られたステートメントを無効化とマークすることが必要になることがある。これを、“ステートメントの無効化”と呼び、予約された動詞 “http://adlnet.gov/expapi/verbs/voided” はこの目的で使用される。他のステートメントを無効化するステートメント自身を無効化することはできない。

Requirements 必要条件

- When issuing a Statement that voids another, the Object of that voiding Statement MUST have the "objectType" field set to "StatementRef". 他のステートメントを無効化するステートメントを発行するときは、無効化ステートメントの目的語は "objectType" フィールドを "StatementRef" に設定しなければならない
- When issuing a Statement that voids another, the Object of that voiding Statement MUST specify the id of the statement-to-be-voided by its "id" field. 他のステートメントを無効化するステートメントを発行するとき、無効化ステートメントの目的語では、無効化すべきステートメントの ID を ID フィールドで指定しなければならない。
- Upon receiving a Statement that voids another, the LRS SHOULD reject the entire request which includes the voiding Statement with HTTP 403 'Forbidden' if the request is not from a source authorized to void Statements. 他のステートメントを無効化するステートメントを受け取ったとき、LRS は、その要求がステートメントの無効化を認められたソースからの要求でない場合は、HTTP 403 'Forbidden' にて、無効化ステートメントを含むリクエスト全体を拒否すべきである。
- Upon receiving a Statement that voids another, the LRS SHOULD return a descriptive error if the target Statement cannot be found. 他のステートメントを無効化するステートメントを受け取ったとき、LRS は、もし、対象ステートメントが見つからなかったときは、説明を含むエラーを返すべきである。
- Upon receiving a Statement that voids another, the LRS MAY roll back any changes to Activity or Agent definitions which were introduced by the Statement that was just voided; 他のステートメントを無効化するステートメントを受け取ったとき、LRS は無効化されたステートメントによって、もともと導入されたアクティビティやエージェントの定義に対する変更をロールバックしてもよい。
- An Activity Provider that wants to "unvoid" a previously voided Statement SHOULD issue that Statement again under a new id 以前の無効ステートメントを有効化するアクティビティプロバイダは新しい id で再度ステートメントを発行すべきである。
- A reporting system SHOULD NOT show voided or voiding Statements by default. レポートシステムは無効化された、または無効化するためのステートメントを標準では表示すべきではない。

Note: See "Statement References" in Section 4.1.4.3 When the "Object" is a Statement

for details about making references to other Statements. To see how voided statements behave when queried, See [StatementRef](#) in 7.2 Statement API). 注記: 他のステートメントの参照作成に関する詳細については、4.1.4.3 目的語がステートメントだった場合のステートメント参照を参照してほしい。呼ばれた際にいかに無効化ステートメントが振舞うかについては、7.2 ステートメント APIの [StatementRef](#) を参照してほしい。

Example 例

```
{
  "actor" : {
    "objectType": "Agent",
    "name" : "Example Admin",
    "mbox" : "mailto:admin@example.adlnet.gov"
  },
  "verb" : {
    "id": "http://adlnet.gov/expapi/verbs/voided",
    "display": {
      "en-US": "voided"
    }
  },
  "object" : {
    "objectType": "StatementRef",
    "id" : "e05aa883-acaf-40ad-bf54-02c8ce485fb0"
  }
}
```

This example Statement voids a previous Statement which it identifies with the Statement id "e05aa883-acaf-40ad-bf54-02c8ce485fb0". このステートメントの例は、ステートメント id が "e05aa883-acaf-40ad-bf54-02c8ce485fb0" で示される以前のステートメントを無効化する。

4.4 Signed Statements 署名付きステートメント

Description 説明

A Statement may include a [digital signature](#) to provide strong and durable evidence of the authenticity and integrity of the Statement. ステートメントは、強固で耐久性のあるステートメントの信頼性と完全性を提供するためにデジタル署名を含むことができる。

Rationale 背景

Some Statements will have regulatory or legal significance, or otherwise require strong and durable evidence of their authenticity and integrity. It may be necessary to verify these Statements without trusting the system they were first recorded in, or perhaps without access to that system. Digital signatures will enable a third-party system to validate such Statements. ステートメントの中には規制上のあるいは法的に重要なも

ものもあるだろう。また、強固で耐久性のある信頼性と完全性の証拠を必要とするだろう。これらのステートメントを最初に記録されたシステムを信頼せずに、あるいはシステムにアクセスせずに確認する必要があるかもしれない。デジタル署名は、第三者システムがそのようなステートメントを確認することを可能にする。

Requirements 必要条件

- A Signed Statement MUST include a JSON web signature (JWS) as defined here: <http://tools.ietf.org/html/draft-ietf-jose-json-web-signature>, as an attachment with a usageType of "http://adlnet.gov/expapi/attachments/signature" and a contentType of "application/octet-stream". 署名付きステートメントは、usageType が "http://adlnet.gov/expapi/attachments/signature" で、contentType が "application/octet-stream" である、添付文書として定義される、JSON web signature (JWS) を含まなければならない: <http://tools.ietf.org/html/draft-ietf-jose-json-web-signature>
- The JWS signature MUST have a payload of a valid JSON serialization of the Statement generated before the signature was added. 署名が付加される前に、JWS 署名は、生成されたステートメントの有効な JSON の連続のデータ本体を持たなければならない。
- The JWS signature MUST use an algorithm of "RS256", "RS384", or "RS512". JWS 署名は "RS256"、"RS384"、または "RS512" のアルゴリズムを使用しなければならない。
- The JWS signature SHOULD have been created based on the private associated with an X.509 certificate. JWS 署名は、X.509 証明と関連した秘密鍵に基づいて生成されるべきである。
- If X.509 was used to sign, the JWS header SHOULD include the "x5c" property containing the associated certificate chain. 仮に、X.509 が署名に使われたなら、JWS のヘッダーは関連した証明書チェーンを内包する "x5c" プロパティを含むべきである。
- The LRS MUST reject requests to store Statements that contain malformed signatures, with HTTP 400 and SHOULD include a message describing the problem in the response. In order to verify signatures are well formed, the LRS MUST do the following: LRS は、HTTP 400 で不正な形式の署名を内蔵するステートメントを記録する要求を拒否しなければならない、応答には問題を説明するメッセージを含むべきである。署名が正しい形式であることを検証するために、LRS は以下のことを実行しなければならない。
 - Decode the JWS signature, and load the signed serialization of the Statement from the JWS signature payload. JWS 署名を復号化して、JWS の署名データから、一連の署名付きステートメントを取り出す。
 - Validate that the "original" Statement is logically equivalent to the received Statement. オリジナルステートメントは論理的に受理したステートメントと同じである、ということを証明する。
 - When making this equivalence check, differences which could have been caused by allowed or required LRS processing of "id", "authority", "stored", "timestamp", or "version" MUST be ignored. この同等性をチェックする時に、許可または必要とされる LRS の "id", "authority", "stored", "timestamp" または "version" の処理がもたらす差は無視されなければならない。
 - If the JWS header includes an X.509 certificate, validate the signature against that certificate as defined in JWS. JWS ヘッダーが X.509 証明書を含むならば、JWS で定義されている手法で、その署名を証明書で認証する。
- Clients MUST NOT assume a signature is valid simply because an LRS has accepted it. クライアントは、単に LRS が受入れたという理由で、署名が有効であると仮定してはならない。

Details 詳細

Signed Statements include a JSON web signature (JWS) as an attachment. This allows the original serialization of the Statement to be included along with the signature. For interoperability, the "RSA + SHA" series of JWS algorithms have been selected, and for discoverability of the signer X.509 certificates SHOULD be used. 署名付きステートメントは、JSON web 署名 (JWS) を添付文書として含む。このことにより、ステートメントの原本をシリアル化したものが、署名とともに添付されることが可能となる。相互運用性のために、JWS のアルゴリズムの "RSA+SHA" シリーズが選択され、署名者の発見可能性のために、X.509 証明書が使われるべきである。

See [Appendix F: Example Signed Statement](#) for an example. 例は [Appendix F: 署名ステートメント例](#) を参照。

Note 注記: The step of validating against the included X.509 certificate is intended as a way to catch mistakes in the signature, not as a security measure. Clients MUST NOT assume a signature is valid simply because an LRS has accepted it. The steps to authenticate a signed Statement will vary based on the degree of certainty required and are outside the scope of this specification.

含まれる X.509 証明書に対して認証するステップは、安全対策のためではなく、署名の中に誤りを発見するためのものである。クライアントは、LRS が受入れたという理由だけで、署名が有効であると仮定してはならない。署名付きステートメントを認証するステップは、要求される確実性の度合いにより変化するものであり、この仕様の規定外である。

5.0 Miscellaneous Types 各種タイプ

5.1 Document ドキュメント

Description 説明

The Experience API provides a facility for Activity Providers to save arbitrary data in the form of documents, which may be related to an Activity, Agent, or combination of both.

xAPIにより、アクティビティプロバイダは、アクティビティやエージェント、もしくは、それら双方の組合せに関連したドキュメント形式による任意のデータを記録することが容易になる。

Details詳細

Propertyプロパティ	Typeタイプ	Description説明
ID	String	Set by AP, unique within state scope (learner, activity). AP(アクティビティプロバイダ)によって規定される。それは状態の範囲(学習者、アクティビティ)内において一意である。
updated	Timestamp	When the document was most recently modified. ドキュメントが最後に修正された時刻
contents	Arbitrary binary data	The contents of the document. ドキュメントの内容

Note注記

In the REST binding, State is a document, not an Object. The id is stored in the IRL, updated is HTTP header information, and contents is the HTTP document itself.

RESTバインディングにおいて、状態はドキュメントでありオブジェクトでは無い。そのIDはIRLに指定され、updatedはHTTPヘッダ情報として、contentsはHTTPのドキュメントそのものとして与えられる。

5.2 Language Map 言語マップ

Description説明

A language map is a dictionary where the key is an RFC 5646 Language Tag, and the value is a string in the language specified in the tag. This map should be populated as fully as possible based on the knowledge of the string in question in different languages.

言語マップはRFC 5646言語タグをキーとし、そのタグで指定された言語で記述された文字列を値とする辞書である。このマップは、その対象となる異なる言語において、その文字列の知識が及ぶ限り出来るだけ充実させるべきである。

5.3 Extensions 拡張

Description説明

Extensions are defined by a map. The keys of that map MUST be IRLs, and the values MAY be any JSON value or data structure. The meaning and structure of extension values under an IRL key are defined by the person who coined the IRL, who SHOULD be the owner of the IRL, or have permission from the owner. The owner of the IRL SHOULD make a human-readable description of the intended meaning of the extension supported by the IRL accessible at the IRL. A learning record store MUST NOT reject an Experience API Statement based on the values of the extensions map.

拡張はマップによって定義される。マップのキーはIRLでなければならないこと、そして、その値はJSONの値か、もしくはデータ構造であってよい。IRLキー下における拡張が持つ値の意味と構造は、IRLの作成者(そのIRLの所有者であるべき者)、または所有者から権限を与えられている者によって定義される。IRLの所有者は、IRLでサポートされた拡張が意図する意味の人が理解できる形式での説明を、IRLが指定する場所においてアクセス可能にするべきである。LRSは拡張マップの値に基づくxAPIのステートメントを拒絶してはならない。

Extensions are available as part of Activity Definitions, as part of statement context, or as part of some statement result. In each case, they're intended to provide a natural way to extend those elements for some specialized use. The contents of these extensions might be something valuable to just one application, or it might be a convention used by an entire community of practice.

拡張は、アクティビティ定義の一部として、ステートメント文脈の一部として、もしくは、あるステートメントの結果の一部として利用可能なものである。それぞれのケースにおいて、それらは、ある特定の利用のための要素を拡張する自然な方法を提供しようとするものである。こうした拡張のコンテンツは、ある一つのアプリケーションに役立つものであったり、もしくは、実践コミュニティ全体における規定かもしれない。

Extensions should logically relate to the part of the statement where they are present. Extensions in Statement context should provide context to the core experience, while those in the result should provide elements related to some outcome. For Activities, they should provide additional information that helps define an Activity within some custom application or community.

拡張は、それらが現れるステートメントの対応部分と論理的に関連づけられるべきである。ステートメント文脈における拡張は、中心的な経験に対する文脈を提供すべきであり、一方で、結果の文脈では、いくつかの成果に関連する要素を提供すべきである。アクティビティにおいては、拡張は、あるカスタムアプリケーションやコミュニティ

内におけるアクティビティを定義することの手助けとなる追加情報を提供すべきである。

Note 注記

A statement should not be totally defined by its extensions, and be meaningless otherwise. Experience API Statements should be capturing experiences among Actors and Objects, and SHOULD always strive to map as much information as possible into the built in elements, in order to leverage interoperability among Experience API conformant tools.

一つのステートメントが、その拡張によって全体的に定義されてしまい、拡張なしには無意味になってしまうようにすべきではない。xAPI のステートメントはアクタとオブジェクト間の経験を記録すべきであり、そして、xAPI 準拠のツール間の相互運用性を大きく高めるために、できるだけ多くの情報を既存の要素にマッピングするよう、常に努力すべきである。

5.4 Identifier Metadata 識別子メタデータ

Description 説明

There are several types of IRI identifiers used in this specification:

この仕様において利用される、いくつかの IRI 識別子のタイプが存在する:

- Verb 動詞
- Activity id アクティビティ ID
- Activity type アクティビティタイプ
- extension key 拡張キー
- attachment usage type 添付文書使用タイプ

For Activity ids, see Activity Definition.

アクティビティ ID については、[アクティビティ定義](#) を参照のこと。

For all other identifiers, metadata MAY be provided in the following JSON format:

すべての他の識別子については、メタデータは次の JSON フォーマットで提供される:

Property プロパティ	Type タイプ	Description 説明
name	Language Map	The human readable/visual name 人が読める / 見られる 名前
description	Language Map	description 説明

Requirements 必要条件

If metadata is provided, both name and description SHOULD be included.

メタデータが提供されている場合、名前と説明の双方が含まれるべきである。

*For any of the identifier IRIs above, if the IRI is an IRL that was coined for use with this specification, the owner of that IRL SHOULD make this JSON metadata available at that IRL when the IRL is requested and a Content-Type of "application/json" is requested.

- 上記、識別子としての IRI において、もし、IRI が本仕様によって作られた IRL であるなら、その IRL の所有者は、その IRL が "Content-Type: application/json" がリクエストされた際に、この JSON メタデータを IRL の場所において利用可能にすべきである。

* If this metadata is provided as described above, it is the canonical source of information about the identifier it describes.

- 上記に記述されたメタデータが提供された場合、それは、その識別子説明の情報の正規のソースである。

* Other sources of information MAY be used to fill in missing details, such as translations, or take the place of this metadata entirely if it was not provided or cannot be loaded. This MAY include metadata in other formats stored at the IRL of an identifier, particularly if that identifier was not coined for use with this specification.

- 翻訳等不足する詳細情報を埋めるために、あるいは提供されていなかったり読み込めない場合は、このメタデータを完全に置き換えるために、他の情報源を使ってもよい。特にその識別子が本仕様で利用する目的で作成されなかった場合においては、このメタデータは、識別子の IRL に記録された他のフォーマット内のメタデータを含んでもよい。

As with Verbs, we recommend that Activity Providers look for and use established, widely adopted identifiers for all types of IRI identifier other than Activity id. Where an identifier already exists, the Activity Provider:

動詞と同様、他のアクティビティ ID 以外のあらゆるタイプの IRI 識別子のために既に使われていて広く採用されている識別子を、アクティビティ・プロバイダは探して利用することを推奨する。識別子が既に存在する場合は、アクティビティ・プロバイダは:

* SHOULD use the corresponding existing identifier;

- 対応する既存の識別子を使用すべきであり;

* MAY create and use their own Verbs where a suitable identifier does not already exist.

- 適切な既存の識別子が存在していなければ、独自の動詞を作成して使用してもよい。

6.0 Runtime Communication ランタイム通信

Sections 6 and 7 detail the more technical side of the Experience API, dealing with how Statements are transferred between Activity Provider and LRS. A number of libraries are under development for a range of technologies (including JavaScript) which handle this part of the specification. It therefore may not be necessary for content developers to fully understand every detail of this part of the specification.

6章、7章では、アクティビティプロバイダと LRS の間でどのようにステートメントが転送されるかについて取扱いながら、xAPI のより技術的な側面を詳しく説明する。いくつかのライブラリは、この章の仕様を取り扱う一連の技術 (Java Script を含む) に対して開発中である。そのため、コンテンツ開発者がこの章の仕様を全て詳細に理解しなくてもよい。

6.1 Encoding エンコーディング

Requirement 必要事項

- All strings MUST be encoded and interpreted as UTF-8. 全ての文字列は UTF-8 形式でエンコードし解釈しなければならない。

6.2 API Versioning API のバージョン管理

Rationale 背景

Future revisions of the specification may introduce changes such as properties added to Statements.

本仕様の将来の改訂版においては、ステートメントにプロパティを追加するような変更を導入するかもしれない。

Systems retrieving Statements may then receive responses that include Statements of different versions. The version header allows for these version differences to be handled correctly, and to ascertain that no partial or mixed LRS version implementations exist.

そのため、ステートメントを読み出すシステムは、異なるバージョンのステートメントを含むレスポンスを受け取るかもしれない。このようなバージョンの違いをバージョンヘッダにより正確に取り扱うことができ、不完全または混在状態のバージョンの実装が存在していないことを確かめることができる。

Using Semantic Versioning will allow Clients and LRSs to reliably know whether they're compatible or not as the specification changes.

セマンティックバージョンングを利用することで、仕様に変更が加えられても、クライアントと LRS は相互に互換性があるかどうかを確実に知ることができる。

Requirements 必要条件

Every request from a Client and every response from the LRS must include an HTTP header with the name "X-Experience-API-Version" and the version as the value. Starting with 1.0.0, xAPI will be versioned according to [Semantic Versioning 1.0.0](#)

クライアントからのすべてのリクエストや LRS からの全てのレスポンスは、"X-Experience-API-Version" を名前、バージョンを値とする HTTP ヘッダーを持たなければならない。1.0.0版より、xAPI は [セマンティックバージョンング 1.0.0](#) に従いバージョン付けされる。

Example例: X-Experience-API-Version : 1.0.0

Requirements for the LRSLRS に対する必要条件:

- MUST include the "X-Experience-API-Version" header in every response. 全てのレスポンスに "X-Experience-API-Version" ヘッダを含めなければならない。
- MUST set this header to "1.0.0". このヘッダには "1.0.0" を設定しなければならない。
- MUST accept requests with a version header of "1.0" as if the version header was "1.0.0". バージョンヘッダに "1.0" が設定されたリクエストは、バージョンヘッダが "1.0.0" であるとみなして受入れなければならない。
- MUST reject requests with version header prior to "1.0.0" unless such requests are routed to a fully conformant implementation of the prior version specified in the header. "1.0.0" より前のバージョンヘッダが設定されたリクエストは、ヘッダ内に指定された以前のバージョンに完全に準拠した実装へ渡されない限りは拒否しなければならない。
- MUST reject requests with a version header of "1.1.0" or greater. "1.1.0" もしくはそれ以上のバージョンヘッダが設定されたリクエストは拒否しなければならない。
- MUST make these rejects by responding with an HTTP 400 error including a short description of the problem. これらの拒否は、問題に関する短い説明を含んだ HTTP 400 エラーで応答しなければならない。

Requirements for the Clientクライアントに対する必要条件:

- SHOULD tolerate receiving responses with a version of "1.0.0" or later. "1.0.0" もしくはそれ以降のバージョンが設定された応答を受けとることを許容すべきである。
- SHOULD tolerate receiving data structures with additional properties. 追加されたプロパティを含むデータ構造を受けとることを許容すべきである。
- SHOULD ignore any properties not defined in version 1.0.0 of the spec. バージョン "1.0.0" 仕様で規定されていないプロパティはすべて無視すべきである。

Converting Statements to other versions他のバージョンへのステートメントの変換:

- Systems MUST NOT convert Statements of newer versions into a prior version format, e.g., in order to handle version differences. システムは、例えばバージョンの違いを取り扱うために新しいバージョンのステートメントを前のバージョン形式に変換してはならない。
- Systems MAY convert Statements of older versions into a newer version only by following the methods described in [Appendix E: Converting Statements to 1.0.0](#). システムは、[Appendix E: 1.0.0へステートメントを変換](#)に記載された方法に従った場合のみ 古いバージョンのステートメントを新しいバージョン形式に変換してもよい。

6.3 Concurrency 同時実行

Description説明

Concurrency control makes certain that an API consumer does not PUT changes based on old data into an LRS.

同時実行制御は API 利用者が LRS に古いデータに基づく変更を PUT しないことを確実にする。

Details詳細

xAPI will use HTTP 1.1 entity tags (ETags) to implement optimistic concurrency control in the portions of the API where PUT may overwrite existing data, being:

xAPI は、PUT が既存のデータ実体を上書きする API の一部において、楽観的な同時実行制御を実装するために、HTTP1.1 エンティティタグ (ETags)を使用する予定である。その部分は次の通り。

- State API
- Agent Profile API
- Activity Profile API

The State API will permit PUT Statements without concurrency headers, since state conflicts are unlikely. The requirements below only apply to Agent Profile API and Activity Profile API.

状態の矛盾がほとんど発生しないことから、ステート API は同時実行ヘッダなしの PUT ステートメントを許容する予定である。以下の必要条件是、エージェントプロフィール API とアクティビティプロフィール API にのみ適用される。

Client requirementsクライアントに対する必要条件

An xAPI Client using either Agent Profile API or Activity Profile API...

Agent Profile API または Activity Profile API を使用する xAPI クライアントは

- MUST include the `If-Match` header or MUST include the `If-None-Match` header. `If-Match` ヘッダを含むか `If-None-Match` ヘッダを含まなければならない。

LRS requirementsLRS に対する必要条件

The LRS that responds to a GET request

LRS が GET リクエストに応答する際には

- MUST add an ETag HTTP header to the response. ETag HTTP ヘッダをレスポンスに追加しなければならない。
- MUST calculate the value of this header to be a hexadecimal string of the SHA-1 digest of the contents. このヘッダの値がコンテンツ部分の SHA-1 ダイジェストの 16 進数文字列となるように計算しなければならない。
- MUST enclose the header in quotes. ヘッダを引用符で囲わなければならない。

The reason for specifying the LRS ETag format is to allow API consumers that can't read the ETag header to calculate it themselves.

LRS ETag 形式を設定する理由は、ETag ヘッダを読み取れない API 利用者が自らその値を計算することを可能とするためである。

The LRS that responds to a PUT request

LRS が PUT リクエストに回答する際には

- MUST handle the `If-Match` header as described in RFC2616, HTTP 1.1 if it contains an ETag, in order to detect modifications made after the consumer last fetched the document. RFC2616 (HTTP 1.1) にて説明されるとおり、`If-Match` ヘッダが ETag を含む場合は、利用者が文書を最後に取り込んだあとに修正が行われたか検出するために `If-Match` ヘッダを取り扱わなければならない。
- MUST handle the `If-None-Match` header as described in RFC2616, HTTP 1.1 if it contains "", in order to detect when there is a resource present that the consumer is not aware of. RFC2616 (" " を含む場合は HTTP1.1) で定める `If-None-Match` ヘッダを、利用者が気づかないリソース提供があるときに検出するために取り扱わなければならない。

If the header precondition in either of the above cases fails, the LRS

上記のいずれの場合もヘッダの前提条件に当てはまらない場合、LRS は

- MUST return HTTP status 412 "Precondition Failed". HTTP status 412 "Precondition Failed" を返さなければならない。
- MUST NOT make a modification to the resource. リソースを変更させてはならない。

If a PUT request is received without either header for a resource that already exists, the LRS

PUT リクエストをいずれのヘッダもなく既存のリソースに対して受け取った場合は、LRS は

- MUST return HTTP status 409 "Conflict". HTTP status 409 "Conflict" を返さなければならない。
- MUST return a plain text body explaining that the consumer SHOULD 利用者がすべきことを説明するテキスト形式の本文を返さなければならない。
 - check the current state of the resource. リソースの現在の状態をチェックすべきである。
 - set the "If-Match" header with the current ETag to resolve the conflict. 競合を解消するために現在の ETag を設定した "If-Match" ヘッダをセットすべきである。
- MUST NOT make a modification to the resource. リソースを変更してはならない。

6.4 Security セキュリティ

Rationale 背景

In order to balance interoperability and the varying security requirements of different environments, several authentication options are defined.

相互運用性と異なる環境における様々なセキュリティの必要条件とのバランスを取るために、いくつかの認証オプションを定める。

Requirement 必要条件

The LRS MUST support authentication using at least one of the following methods:

LRS は次の手段のうち少なくとも一つを使用した認証をサポートしなければならない。

- OAuth 1.0 (rfc5849), with signature methods of "HMAC-SHA1", "RSA-SHA1", and "PLAINTEXT" OAuth 1.0 (RFC 5849) ("HMAC-SHA1"、"RSA-SHA1" および "PLAIN TEXT" の署名方式を利用する)
- HTTP Basic Authentication HTTP 基本認証
- Common Access Cards (implementation details to follow in a later version) Common Access Cards (実装は今後のバージョンでフォローされ詳細化される)
- The LRS MUST handle making, or delegating, decisions on the validity of Statements, and determining what operations may be performed based on the credentials used. LRS はステートメントの有効性に関して判断するか委任するかを決め、使用された証明書に基づいて、どのような処理を行うかを決定しなければならない。

Authentication scenarios 認証シナリオ

The below matrix describes the possible authentication scenarios.

以下の表において、想定される認証のシナリオを示す。

A **registered application** is an application that will authenticate to the LRS as an OAuth consumer that has been registered with the LRS. A **known user** is a user account on the LRS, or on a system which the LRS trusts to define users.

登録されたアプリケーションとは、LRS に既に登録された OAuth consumer として LRS に認証されているアプリケーションである。**認識された利用者**とは LRS または LRS がユーザを定義することを委託しているシステム上のユーザアカウントである。

	Known user 認識された利用者	User unknown 認識されていない利用者
Application is registered アプリ	Standard workflow for OAuth. OAuth に対応する標準ワー	LRS trusts application to access xAPI without additional user credentials. OAuth token steps are not invoked. LRS は付加的な利用者証明書を持たない xAPI アク

ケーションが登録される場合	クフローが適用される。	アクセスを行うアプリケーションを信頼する。OAuth トークンステップは呼び出されない。
Application is not registered アプリケーションが登録されていない場合	The application Agent is not identified as a registered Agent and the LRS cannot make assumptions on its identity.アプリケーションエージェントは登録されたエージェントとして識別されないため、LRS はその識別を前提とすることはできない。	
No application アプリケーションがない場合	HTTP Basic Authentication is used instead of OAuth, since no application is involved.アプリケーションが呼び出されないため、HTTP 基本認証が OAuth の代わりに利用される。	
No authentication 認証なしの場合	MAY be supported by the LRS, possibly for testing purposes.テストを目的とする場合などで LRS によりサポートされてもよい。	

6.4.1 How To Handle Each Scenario 各シナリオの取り扱い方法

General 概要

- The LRS must record the application's name and a unique consumer key (identifier). LRS はアプリケーションの名前と一意な consumer key (識別子)を記録しなければならない。
- The LRS must provide a mechanism to complete this registration, or delegate to another system that provides such a mechanism. The means by which this registration is accomplished are not defined by OAuth or the xAPI. LRS はこの登録を完了するためのメカニズムを提供するか、同様のメカニズムを提供する別システムに委譲しなければならない。この登録を実現する方法は OAuth または xAPI で定義されていない。

Application registered + known user 登録されたアプリケーション + 認識された利用者

- Use endpoints below to complete the standard workflow. 標準ワークフローを完了するために以下のエンドポイントを利用する。
- If this form of authentication is used to record Statements and no authority is specified, the LRS should record the authority as a group consisting of an Agent representing the registered application, and an Agent representing the known user. この認証形式がステートメントを記録する際に何も権限が指定されずに利用される場合は LRS は登録されたアプリケーションを表すエージェントと認識されたユーザを表すエージェントからなるグループとして権限を記録すべきである。

Application registered + user unknown 登録されたアプリケーション + 認識されていない利用者

- LRS will honor requests that are signed using OAuth with the registered application's credentials and with an empty token and token secret. LRS は登録されたアプリケーションの証明書と空のトークンとトークンシークレットで OAuth を使用して署名されたリクエストを受け取るだろう。
- If this form of authentication is used to record Statements and no authority is specified, the LRS should record the authority as the Agent representing the registered application. もしこの認証形式がステートメントの記録の際に何も権限が指定されずに使用される場合は、LRS は登録されたアプリケーションを表すエージェントとしての権限を記録すべきである。

Application not registered + known user 登録されていないアプリケーション + 認識された利用者

- Use a blank consumer secret; 空白の consumer secret を利用すること。
- Call "Temporary Credential" request; "Temporary Credential" リクエストを呼び出すこと。
- Specify "consumername" and other usual parameters; User will then see "consumername" plus a warning that the identity of the application requesting authorization cannot be verified. "consumer name" と他の一般的なパラメータを設定すること。そのため、利用者は "consumer name" と認証をリクエストしたアプリケーションの識別子が確認できないという警告を目にするだろう。
- the LRS MUST record an authority that includes both that application and the authenticating user, as a group, since OAuth specifies an application. OAuth がアプリケーションを指定するため、LRS はアプリケーションと認証しているユーザの両方をグループとして含む権限を記録しなければならない。

No application + known user アプリケーションがない + 認識された利用者

- Use username/password combination that corresponds to an LRS login. LRS ログインに対応するユーザ名 / パスワードの結合情報を用いること。
- Authority to be recorded as the Agent identified by the login, **unless...** 権限はログインにより識別されるエージェントとして記録される。但し、以下の場合を除く。
 - other Authority is specified **and...** 他の権限情報が指定される。かつ、
 - LRS trusts the known user to specify this Authority. LRS はこの権限情報を指定する認識された利用者を信頼する。

No authorization 認証なし

- Requests should include headers for HTTP Basic Authentication based on a blank username and password, in order to distinguish an explicitly unauthenticated request from a request that should be given a HTTP Basic Authentication challenge. 明らかに認証されていないリクエストと HTTP 基本認証チャレンジが提供されなければならないリクエストを区別するために、リクエストには空白のユーザ名とパスワードに基づく HTTP 基本認証のためのヘッダを含めるべきである。

Requirements 必要条件

The LRS:

LRS を

- MUST be able to be configured for complete support of the xAPI: xAPI へ完全に対応させるためには以下のとおり設定できなければならない。
 - With any of the above methods. 上記のいずれかの手順によること。
 - In any of the workflow scenarios above. 上記のいずれかのワークフローシナリオによること。
- MAY (for security reasons): (セキュリティ上の理由で) 以下のとおり設定してもよい。
 - Support a subset of the above methods. 上記の手順のサブセットをサポートすること。
 - Limit the known users or registered applications. 認識された利用者または登録されたアプリケーションを制限すること。
- SHOULD at a minimum supply OAuth with "HMAC-SHA1" and "RSA-SHA1" signatures. "HMAC-SHA1" および "RSA-SHA1" による署名を用いた OAuth を最低でも提供すべきである。

6.4.2 OAuth Authorization Scope OAuth 認証スコープ

Description 説明

These are recommendations for scopes which should enable an LRS and an application communicating using the xAPI to negotiate a level of access which accomplishes what the application needs while minimizing the potential for misuse. The limitations of each scope are in addition to any security limitations placed on the user account associated with the request.

LRS と xAPI を利用して通信するアプリケーションが、間違った利用の可能性を最小限にしつつアプリケーションの要求に応えるアクセスレベルを調整できるようにするためにスコープに対する推奨を示す。それぞれのスコープの制限は、リクエストに関連するユーザアカウントへの任意のセキュリティ制限に加わるものである。

Requirements 必要条件

The LRS:

LRS は

- MUST accept a scope parameter as defined in [OAuth 2.0](#); [OAuth 2.0](#) で定義されたスコープパラメータを受入れなければならない。
- MUST assume a requested scope of "statements/write" and "statements/read/mine" if no scope is specified; もしスコープが指定されていない場合は "statements/write" と "statements/read/mine" のリクエストスコープと仮定しなければならない。
- MUST support the scope of "all" as a minimum; 最低限として "all" のスコープをサポートしなければならない。
- MAY support other scopes. その他のスコープをサポートしてもよい。

An xAPI Client:

xAPI クライアントは

- SHOULD request only the minimal needed scopes, to increase the chances that the request will be granted. リクエストが許可される機会を増やすために、最低限必要なスコープのみ要求すべきである。

Details 詳細

The following table lists xAPI scope values:

以下の表で xAPI スコープ値を一覧にする。

Scope スコープ	Permission 許可
statements/write	write any Statement 任意のステートメントを書き込む
statements/read/mine	read Statements written by "me", that is with an authority matching what the LRS would assign if writing a Statement with the current token. "自分" が書いたステートメントを読み込む、つまり、現在のトークンでステートメントを書くと仮定した場合に LRS が割り当てるであろう権限に一致した権限を所有していることに相当する。
statements/read	read any Statement 任意のステートメントを読み込む
	read/write state data, limited to Activities and Actors associated with the current token to the extent it is

state	possible to determine this relationship.この関係を決定できる範囲で現在のトークンに関するアクティビティとアクタに制限された状態データを読み込む／書込む。
define	(re)Define Activities and Actors. If storing a Statement when this is not granted, ids will be saved and the LRS may save the original Statement for audit purposes, but should not update its internal representation of any Actors or Activities.アクティビティとアクタを(再)定義する。もし認められないままステートメントを保存した場合、ids が保存され LRS は調査を目的としたステートメントの原文を保存するかもしれないが、アクタまたはアクティビティの内部表現で上書きすべきではない。
profile	read/write profile data, limited to Activities and Actors associated with the current token to the extent it is possible to determine this relationship.この関係を決定できる範囲で現在のトークンに関するアクティビティとアクタに制限されたプロフィールデータを読み込む／書込む。
all/read	unrestricted read access制限なしの読み込みアクセス
all	unrestricted access制限なしのアクセス

OAuth Extended Parameters OAuth 拡張パラメータ

Note that the parameters "consumer_name" and "scope" are not part of OAuth 1.0, and therefore if used should be passed as query string or form parameters, not in the OAuth header.

"consumer_name" や "scope" といったパラメータは OAuth1.0 に含まれないので注意すること。そのためもし使われていた場合は OAuth ヘッダとしてではなくクエリ文字列やフォームパラメータとみなして引き渡すべきである。

OAuth Endpoints OAuth エンドポイント

Name	Endpoint	Example
Temporary Credential RequestTemporary Credential リクエスト	OAuth/initiate	http://example.com/xAPI/OAuth/initiate
Resource Owner AuthorizationResource Owner 認可	OAuth/authorize	http://example.com/xAPI/OAuth/authorize
Token Requestトークンリクエスト	OAuth/token	http://example.com/xAPI/OAuth/token

Example 例

The list of scopes determines the set of permissions that is being requested. For example, an instructor might grant "statements/read" to a reporting tool, but the LRS would still limit that tool to Statements that the instructor could read if querying the LRS with their credentials directly (such as Statements relating to their students).

スコープのリストは、リクエストされている許可の設定を決定する。例えば、インストラクタは "statements/read" をレポートツールのために許可するかもしれない。しかし、LRS はインストラクタが LRS に対して直接認証をして問合せをすることによって読み取るステートメント(たとえば自分の生徒に関するステートメント)のみにツールを制限することができる。

7.0 Data Transfer (REST)

This section describes that the xAPI consists of 4 sub-APIs: Statement, State, Agent, and Activity Profile. These four sub-APIs of the Experience API are handled via RESTful HTTP methods. The Statement API can be used by itself to track learning records.

このセクションでは xAPI を構成する4つのサブ API (ステートメント、ステート、エージェント プロファイル、アクティビティ プロファイル)について説明する。4つのサブ API は RESTful な HTTP メソッドによって処理される。ステートメント API は学習記録を追跡するために単独で利用することができる。

Note: In all of the example endpoints given in the specification, "http://example.com/xAPI/" is the example IRL of the LRS and everything after this represents the endpoint which MUST be used.

注:この仕様書では、LRS の IRL (エンドポイント)を "http://example.com/xAPI/" としている。実装時には、任意のエンドポイントを必ず用意する必要がある。

LRS RequirementsLRS の必要条件

The LRS MUST reject with `HTTP 400 Bad Request` status (see below) any request to any of these APIs using any parameters: * the LRS does not recognize (**Note:** LRSs may recognize and act on parameters not in this specification). * that match parameters described in this specification in all but case.

LRS はこれらの API に対するあらゆるリクエストについて、パラメータが以下の条件に該当する場合、`HTTP 400 Bad Request` ステータスで拒絶しなければならない。

- LRS が認識できないパラメータである(注釈: :LRS はこの仕様書に記載のないパラメータを認識して、動作する可能性がある)。

- 本仕様に記述されるパラメータと一致するが、大文字小文字の別が異なる。

7.1 Error Codes エラーコード

The list below offers some general guidance on HTTP error codes that could be returned from various methods in the API. An LRS MUST return the error code most appropriate to the error condition based on the list below, and SHOULD return a message in the response explaining the cause of the error.

以下のリストは API の様々なメソッドから返される HTTP エラーコードに関する一般的なガイドである。LRS はエラーの状況に最も合致するエラーコードを返さなければならない。また LRS はエラーの原因を説明するメッセージを返すべきである。

* `400 Bad Request` - Indicates an error condition caused by an invalid or missing argument. The term "invalid arguments" includes malformed JSON or invalid Object structures.

- `400 Bad Request` - 引数が無効もしくは欠落していることを起因とするエラー状態を表す。「引数が無効」とは、不正な JSON または無効なオブジェクト構造を含む。

* `401 Unauthorized` - Indicates that authentication is required, or in the case authentication has been posted in the request, that the given credentials have been refused.

- `401 Unauthorized` - 認証が必要、もしくはリクエスト中の認証資格情報が拒否されたことを表す。

* `403 Forbidden` - Indicates that the request is unauthorized for the given credentials. Note this is different than refusing the credentials given. In this case, the credentials have been validated, but the authenticated Client is not allowed to perform the given action.

- `403 Forbidden` - 与えられた認証資格情報ではリクエストを拒否されたことを表す。認証資格情報自体の拒否とは異なるので注意すること。このケースでは、認証資格情報は確認されたものの、認証されたクライアントでは指定されたアクションを実行することが許可されなかったことを表す。

* `404 Not Found` - Indicates the requested resource was not found. May be returned by any method that returns a uniquely identified resource, for instance, any State or Agent Profile or Activity Profile API call targeting a specific document, or the method to retrieve a single Statement.

- `404 Not Found` - 要求されたリソースが見つからなかったことを表す。その例としては特定の文書を対象とする ステート、エージェントプロフィール、アクティビティプロフィール API call や、単一のステートメントを返すメソッドなどが挙げられる。

* `409 Conflict` - Indicates an error condition due to a conflict with the current state of a resource, in the case of State API, Agent Profile or Activity Profile API calls, or in the Statement PUT call. See Section 6.3 concurrency for more details.

- `409 Conflict` - ステート API やエージェントプロフィール API、アクティビティプロフィール API、もしくはステートメント PUT 呼び出しの際に発生する、リソースの現在の状態との競合を起因とするエラー状態を表す。詳細は6.3 同時実行を参照のこと。

* `412 Precondition Failed` - Indicates an error condition due to a failure of a precondition posted with the request, in the case of State or Agent Profile or Activity Profile API calls. See Section 6.3 Concurrency for more details.

- `412 Precondition Failed` - State API や Agent Profile API、Activity Profile API 呼び出しの際に発生する、リクエストともにポストされた前提条件の失敗に起因するエラーを表す。詳細は6.3 同時実行を参照のこと。

* `413 Request Entity Too Large` - Indicates that the LRS has rejected the Statement or document because its size is larger than the maximum allowed by the LRS. The LRS is free to choose any limit and MAY vary this limit on any basis, e.g., per authority, but MUST be configurable to accept Statements of any size.

- `413 Request Entity Too Large` - LRS が許容するサイズを超過していることを理由に、LRS がステートメントもしくはドキュメントを拒否したことを表す。LRS はサイズの制限値を自由に設定できる。また LRS は任意の条件ごと(例: 権限単位)にこの制限値を変更してもよい。ただしステートメントについてはあらゆるサイズのものを受け入れるよう設定できなければならない。

* `500 Internal Server Error` - Indicates a general error condition, typically an unexpected exception in processing on the server.

- `500 Internal Server Error` - サーバ上に予期しない例外処理が発生した場合など、一般的なエラー状態を表す。

7.2 Statement API ステートメント API

The basic communication mechanism of the Experience API.

xAPI の基本的な通信メカニズム

PUT Statements PUT ステートメント

Example endpoint: <http://example.com/xAPI/statements>

エンドポイントの例: <http://example.com/xAPI/statements>

Stores Statement with the given id.

与えられた id でステートメントを格納する。

Returns: `204 No Content`

戻り値: `204 No Content`

Parameterパラメータ	Typeタイプ	Defaultデフォルト値	Description説明
statementId	String		Id of Statement to record記録するためのステートメント id

POST Statements POST ステートメント

Example endpoint: <http://example.com/xAPI/statements>

エンドポイントの例: <http://example.com/xAPI/statements>

Stores a Statement, or a set of Statements. Since the PUT method targets a specific Statement id, POST must be used rather than PUT to save multiple Statements, or to save one Statement without first generating a Statement id. An alternative for systems that generate a large amount of Statements is to provide the LRS side of the API on the AP, and have the LRS query that API for the list of updated (or new) Statements periodically. This will likely only be a realistic option for systems that provide a lot of data to the LRS.

ステートメントもしくはステートメント群を格納する。PUT メソッドは特定のステートメント id をターゲットにするので、複数のステートメントを保存する場合や、ステートメント ID を最初に生成せずに単一のステートメントを保存する場合には PUT ではなく POST を使用しなければならない。大量のステートメント群を生成するシステムのための代替策は AP 上の API の LRS 側を提供することである。そして定期的にステートメント群を更新(もしくは新規)するよう LRS からその API を照会する。これは大量のデータを LRS に提供するようなシステムにおいてのみ現実的なオプションといえる。

Returns: `200 OK`, Statement id(s) (UUID).

戻り値: `200 OK`, ステートメント id(s) (UUID).

Common requirements for PUT and POST PUT と POST に対する一般的な必要条件

An LRS MUST NOT make any modifications to its state based on a receiving a Statement with a statementID that it already has a Statement for. Whether it responds with `409 Conflict` or `204 No Content`, it MUST NOT modify the Statement or any other Object.

既にステートメントの割り当てられているステートメント ID をもったステートメントを受信した場合でも、LRS はその状態にいかなる変更も行ってはならない。`409 Conflict`もしくは`204 No Content`で応答するにかかわらず、ステートメントもしくはその他のあらゆるオブジェクトを変更してはならない。

If the LRS receives a Statement with an id it already has a Statement for, it SHOULD verify the received Statement matches the existing one and return `409 Conflict` if they do not match.

もし LRS が受信したステートメントの ID がすでに LRS に存在していれば、受信したステートメントが既存のものと合致するか検証し、合致しない場合には `409 Conflict` を返すべきである。

The LRS MAY respond before Statements that have been stored are available for retrieval.

LRS は格納されたステートメントが検索可能となる前に応答してもよい。

GET Statements

Example endpoint: <http://example.com/xAPI/statements>

エンドポイントの例: <http://example.com/xAPI/statements>

This method may be called to fetch a single Statement or multiple Statements. If the `statementId` or `voidedStatementId` parameter is specified a single Statement is returned.

このメソッドは単一のステートメントもしくは複数のステートメントを取得するために呼び出すことができる。ステートメント ID もしくは `voidedStatementId` がパラメータとして指定されている場合には、単一のステートメントが戻り値として返される。

Otherwise returns: A StatementResult Object, a list of Statements in reverse chronological order based on "stored" time, subject to permissions and maximum list length. If additional results are available, an IRL to retrieve them will be included in the StatementResult Object.

それ以外の戻り値: 権限と最大リスト長の制約の中で、"stored" 時間の新しい順に並べたステートメントのリストである ステートメントの結果 オブジェクト。追加の結果が利用可能な場合には、それらを取得するための IRL は StatementResult オブジェクトに含まれる。

Returns: 200 OK, Statement or Statement Result (See Section 4.2 for details)

戻り値: 200 OK, ステートメントもしくはステートメントの結果 (詳細は4.2参照)

Parameterパラメータ	Typeタイプ	Defaultデフォルト値	Description説明
statementId	String		ID of Statement to fetch取得するためのステートメントの ID
voidedStatementId	String		ID of voided Statement to fetch. see Voided Statements 取得するための無効なステートメントの ID。詳細は VoidedStatement を参照。
agent	Agent or Identified Group Object (JSON)		<p>Filter, only return Statements for which the specified Agent or group is the Actor or Object of the Statement. 指定されたエージェントやグループが、ステートメントのアクタやオブジェクトであるステートメントのみをフィルタして返す。</p> <ul style="list-style-type: none"> Agents or identified groups are equal when the same Inverse Functional Identifier is used in each Object compared and those Inverse Functional Identifiers have equal values. 各オブジェクトで同じ逆機能識別子が利用され、それらの逆機能識別子が等しい値を持つ場合、エージェントもしくは識別されたグループは等しい。 For the purposes of this filter, groups that have members which match the specified Agent based on their Inverse Functional Identifier as described above are considered a matchこのフィルタの目的のために、上記逆機能識別子に基づいて同一であるとみなされた特定のエージェントと一致するメンバを持つグループは等しいとみなされる。 <p>See agent/group Object definition for details. 詳細は エージェント/グループオブジェクトの定義 参照。</p>
verb	Verb id (IRI)		Filter, only return Statements matching the specified verb id. フィルタし、特定の verb id とマッチしたステートメントのみを返す。
activity	Activity id (IRI)		Filter, only return Statements for which the Object of the Statement is an Activity with the specified id. フィルタし、指定された id をもつアクティビティをオブジェクトとするステートメントのみを返す。
registration	UUID		Filter, only return Statements matching the specified registration id. Note that although frequently a unique registration id will be used for one Actor assigned to one Activity, this should not be assumed. If only Statements for a certain Actor or Activity should be returned, those parameters should also be specified. フィルタし、指定した registration id に一致するステートメントを返す。あるアクティビティに割り当てられるあるアクタに対して、一意の登録 ID が割り当てられることが多いが、それを前提とすべきでないことに注意が必要である。特定のアクタもしくはアクティビティのためのステートメントのみが返されるべき場合には、それらのパラメータもあわせて指定すべきである。
related_activities	Boolean	False	Apply the Activity filter broadly. Include Statements for which the Object, any of the context Activities, or any of those properties in a contained SubStatement match the Activity parameter, instead of that parameter's normal behavior. Matching is defined in the same way it is for the 'Activity' parameter. "アクティビティフィルタを広く適用する。オブジェクトや、あらゆる文脈のアクティビティ、もしくはパラメータ本来の正常な動作ではなく、アクティビティパラメータにマッチするサブステートメントを含むプロパティをもつステートメントを含む。マッチングはアクティビティパラメータと同様の方式で定義される。
related_agents	Boolean	False	Apply the Agent filter broadly. Include Statements for which the Actor, Object, authority, instructor, team, or any of these properties in a contained SubStatement match the Agent parameter, instead of that parameter's normal behavior. Matching is defined in the same way it is for the 'agent' parameter. エージェントフィルタを広く適用する。パラメータ本来の正常な動作ではなく、エージェントパラメータにマッチするアクタやオブジェクト、権限、インストラクタ、チームもしくはそれらのあらゆるプロパティをもつプロパティを含む。マッチングはエージェントパラメータと同様の方式で定義される。
			Only Statements stored since the specified timestamp (exclusive) are returned. 指

since	Timestamp		定された timestamp よりあと(timestamp の時刻は含まない)に記録されたステートメントのみを返す。
until	Timestamp		Only Statements stored at or before the specified timestamp are returned.指定された timestamp と同時またはそれ以前に記録されたステートメントのみを返す。
limit	Nonnegative Integer	0	Maximum number of Statements to return. 0 indicates return the maximum the server will allow.返すステートメントの最大数。0 はサーバが許容する最大値を返すことを表す。
format	String: ("ids", "exact", or "canonical")	exact	<p>If "ids", only include minimum information necessary in Agent, Activity, and group Objects to identify them. For anonymous groups this means including the minimum information needed to identify each member. If "exact", return Agent, Activity, and group Objects populated exactly as they were when the Statement was received。「ids」の場合、エージェントやアクティビティ、そしてグループオブジェクトを識別するために最低限必要な情報のみを含む。匿名グループにおいては各メンバを識別するために必要な最低限の情報を意味する。「exact」の場合、ステートメントが受理された際と完全に同一なエージェントやアクティビティ、そしてグループオブジェクトを返す。</p> <p>If "canonical", return Activity Objects populated with the canonical definition of the Activity Objects as determined by the LRS, after applying the language filtering process defined below, and return the original Agent Objects as in "exact" mode. Activity Objects contain Language Map Objects for name and description. Only one language should be returned in each of these maps。「canonical」の場合、以下に定義する言語フィルタを適用し、オリジナルのエージェント オブジェクトを「exact」モードで返したのち、LRS により判断され、正規の定義を含んだアクティビティ オブジェクトを返す。アクティビティ オブジェクトは名前と説明のために Language Map オブジェクトを含む。これらのマップではひとつの言語のみが返されるべきである。</p> <p>In order to provide these strings in the most relevant language, the LRS will apply the Accept-Language header as described in RFC 2616 (HTTP 1.1), except that this logic will be applied to each language map individually to select which language entry to include, rather than to the resource (list of Statements) as a whole. An LRS requesting Statements for the purpose of importing them SHOULD use a format of "exact".これらの文字列を最も関連度の高い言語で提供するために LRS は、RFC 2616 (HTTP 1.1)で解説されている通り、Accept-Language ヘッダを適用する。ただし、全体としてリソース(ステートメントのリスト)に適用される場合ではなく、このロジックが各言語マップに個別に適用される場合はこの限りではない。LRS がこれらをインポートすることを目的としてステートメントを要求する場合には、「exact」フォーマットを用いるべきである。</p>
attachments	Boolean	False	If true LRS MUST use the multipart response format and include all attachments as described in 4.1.11. Attachments, otherwise the LRS MUST NOT include attachment raw data and MUST send the prescribed response with Content-Type application/json.trueの場合、LRS は4.1.11. Attachmentsで解説している通り、マルチパートレスポンスフォーマットを用いなければならない、あらゆる添付文書を含めなければならない。それ以外の場合には LRS は添付文書の生データを含めてはならず、Content-Type application/json の形式で所定の応答を送信しなければならない。
ascending	Boolean	False	If true, return results in ascending order of stored time true の場合、格納された時間の昇順で結果を返す。

The LRS MUST reject with an HTTP 400 error any requests to this resource which:

以下の場合、LRS はこれらリソースに対するリクエストに対してHTTP 400エラーで拒絶しなければならない。

* contain both statementId and voidedStatementId parameters * ステートメント ID と voidedStatementId 両方のパラメータを含む場合。

* contain statementId or voidedStatementId parameters, and also contain any other parameter besides "attachments" or "format".

- ステートメント ID または voidedStatementId のパラメータを含み、また「 attachments 」もしくは「 format 」以外のパラメータを含む場合。

The LRS MUST include the header "X-Experience-API-Consistent-Through", in ISO 8601 combined date and time format, on all responses to Statements requests, with a value of the timestamp for which all Statements that have or will have a "stored" property before that time are known with reasonable certainty to be available for retrieval. This time SHOULD take into account any temporary condition, such as excessive

load, which might cause a delay in Statements becoming available for retrieval.

LRS はステートメントリクエストに対するあらゆる応答に対して「X-Experience-API-Consistent-Through」ヘッダを ISO 8601 combined date and time フォーマットを含まなければならない。その値は、指定されたタイプスタンプ以前の "stored" プロパティを持つ全てのステートメントが、妥当な確からしさで取り出し可能になっていることを示すタイムスタンプである。この時間は、ステートメントが検索可能となるまでの遅延をもたらす過剰負荷などの任意の一時的な状態を考慮すべきである。

Note 注記

Due to query string limits, this method MAY be called using POST and form fields if necessary. The LRS MUST differentiate a POST to add a Statement or to list Statements based on the parameters passed.

クエリ文字列の制限により、このメソッドは POST と form field を使用して呼び出してもよい。LRS は、ステートメントを追加するための POST と、複数のステートメントをリストする POST とを、渡されるパラメータに応じて区別しなければならない。

Filter Conditions for StatementRefs StatementRefs のためのフィルタ条件

For filter parameters which are not time or sequence based (that is, other than since, until, or limit), Statements which target another Statement (by using a StatementRef as the Object of the Statement) will meet the filter condition if the targeted Statement meets the condition. The time and sequence based parameters must still be applied to the Statement making the StatementRef in this manner. This rule applies recursively, so that "Statement a" is a match when a targets b which targets c and the filter conditions described above match for "Statement c".

フィルタパラメータが時間やシーケンスベースでない場合（つまり since、until もしくは limit 以外の場合）で、（ステートメントのオブジェクトとして StatementRef を使用して）他のステートメントを参照する時、その参照先のステートメントが条件を満たす場合は、参照元のステートメント自身も条件を満たす。時間やシーケンスベースのパラメータはこの方法で StatementRef を行うステートメントに対しても適用する必要がある。このルールは再帰的に適用される。すなわち、ステートメント A が B を参照し、B が C を参照している状況で、上に示すフィルタ条件がステートメント C に当てはまるなら、ステートメント A もフィルタ対象となる。

For example, consider the Statement "Ben passed explosives training", and a follow up Statement: "Andrew confirmed \<StatementRef to original Statement>". The follow up Statement will not mention "Ben" or "explosives training", but when fetching Statements with an Actor filter of "Ben" or an Activity filter of "explosives training", both Statements match and will be returned so long as they fall into the time or sequence being fetched. たとえば、「ベン」は爆発物のトレーニングに合格したというステートメントと、「アンドリューは\<StatementRef to original Statement>を確認した」というフォローアップのステートメントがあったとする。フォローアップステートメントでは、「ベン」にも「爆発物のトレーニング」にも触れていないが、アクタのフィルタで「ベン」またはアクティビティのフィルタで「爆発物のトレーニング」を指定してステートメントを取り出した場合、時間とシーケンスの条件に合致している限り、両方のステートメントがマッチし返答される。

This section does not apply when retrieving Statements with statementId or voidedStatementId. これらの条件はステートメントをステートメント ID や voidedStatementId で検索した場合には適用されされない。

Note 注記

StatementRefs used in the statement field in context do not affect how Statements are filtered. コンテキストのステートメント フィールドで用いられる StatementRefs は、ステートメントのフィルタに対して影響を与えない。

Voided Statements 無効なステートメント

The LRS MUST not return any Statement which has been voided, unless that Statement has been requested by voidedStatementId. The LRS MUST still return any Statements targetting the voided Statement when retrieving Statements using explicit or implicit time or sequence based retrieval, unless they themselves have been voided, as described in the section on filter conditions for StatementRefs. This includes the voiding Statement, which cannot be voided. Reporting tools can identify the presence and statementId of any voided Statements by the target of the voiding Statement. Reporting tools wishing to retrieve voided Statements SHOULD request these individually by voidedStatementId.

LRS は voidedStatementId によりリクエストされたステートメントでない限り、無効となったステートメントを返してはならない。LRS は、StatementRefs によるフィルタ条件で示した通り、時間またはシーケンス基準による直接・間接の取り出しの場合には、ステートメント自体が無効化されている場合を除き、無効化されたステートメントを参照しているステートメントも返さなければならない。これは、自分自身を無効化することのできない、無効化ステートメントを含む。レポートツールは無効化ステートメントのターゲットを用いてあらゆる無効化ステートメントの存在とステートメント ID を識別することができる。レポートツールは voidedStatementId を用いて個別に無効化されたステートメントをリクエストすべきである。

7.3 Document APIs ドキュメント API

The three Document APIs provide document storage for learning activity providers and Agents. The details of each API are found in the following sections, and the information in this section applies to all three APIs.

3つのドキュメント API はアクティビティプロバイダとエージェントのためにドキュメントストレージを提供する。各 API の詳細は次のセクションで説明する。このセクションで説明する内容はこれら3つの API すべてに適用される。

New Agents and Activities 新規エージェントとアクティビティ

An Activity Provider MAY send documents to any of the document APIs for Activities and Agents that the LRS does not have prior knowledge of. The LRS MUST NOT reject documents on the basis of not having prior knowledge of the Activity and/or Agent.

アクティビティプロバイダは、LRS が事前知識を持たないアクティビティとエージェントに関する文書を、いずれの文書 API に送ってもよい。LRS はアクティビティおよび/またはエージェントの予備知識を持っていないことを理由にしてドキュメントを拒絶してはならない。

POST to store application/json arrays of variables application/json 配列を変数に格納するための POST メソッド

API	Methodメソッド	Endpointエンドポイント	Example例
State API	POST	activities/state	http://example.com/xAPI/activities/state
Activity Profile API	POST	activities/profile	http://example.com/xAPI/activities/profile
Agent Profile API	POST	agent/profile	http://example.com/xAPI/agents/profile

APs MAY use Documents of content type "application/json" to store sets of variables. For example a document contains:

AP は変数セットを格納するためにコンテンツタイプ application/json のドキュメントを利用してもよい。例えばドキュメントは次の情報を含むとする:

```
{
  "x" : "foo",
  "y" : "bar"
}
```

When an LRS receives a POST request with content type application/json for an existing document also of content type application/json, it MUST merge the posted document with the existing document. In this context merge is defined as:

LRS がコンテンツタイプが application/json の既存のドキュメントについて、POST リクエストをコンテンツタイプ application/json で受け取った場合、ポストされてきたドキュメントを既存ドキュメントにマージしなければならない。この文脈において「マージ」とは次のように定義する。

* de-serialize the Objects represented by each document

- 複数の文書に分割されたオブジェクトをデシリアライズする

* for each property directly defined on the Object being posted, set the corresponding property on the existing Object equal to the value from the posted Object.

- ポストされたオブジェクトに直接定義されたプロパティごとに、既存オブジェクトに対して対応するプロパティの値をセットする。

* store any valid json serialization of the existing Object as the document referenced in the request

- リクエストの参考ドキュメントとして、既存オブジェクトの有効な json のシリアライズ性を格納します。

Note that only top-level properties are merged, even if a top-level property is an Object the entire contents of each original property are replaced with the entire contents of each new property.

最上位階層のプロパティのみがマージされることに注意が必要である。たとえ最上位階層のプロパティ(の値)がオブジェクトであっても、それぞれの元のプロパティの値すべてが、新しいプロパティの値に完全に置き換えられる。

For example, this document is POSTed with the same id as the existing document above:

例えば、既に存在する上記ドキュメントを POST したものと同一 ID で、再度このドキュメントが POST された場合:

```
{
  "x" : "bash",
  "z" : "faz"
}
```

the resulting document stored in the LRS is:

LRS に格納されるドキュメントの結果は次の通りとなる。

```
{
  "x" : "bash",
  "y" : "bar",
  "z" : "faz"
}
```

If the original document exists, and the original document or the document being posted do not have a Content-Type: of "application/json", or

if either document can not be parsed as JSON Objects, the LRS MUST respond with HTTP status code 400 "Bad Request", and MUST NOT update the target document as a result of the request.

オリジナルのドキュメントが存在し、元のドキュメントもしくは post されてきたドキュメントのコンテンツタイプが application/json でない場合、あるいはどちらのドキュメントも JSON オブジェクトとして解析できない場合、LRS は HTTP のステータスコード400 "Bad Request" で応答しなければならず、リクエストに対してターゲットとするドキュメントをアップデートしてはならない。

If the original document does not exist, the LRS MUST treat the request the same as it would a PUT request and store the document being posted.

元のドキュメントが存在しない場合、LRS は PUT リクエストと同様に取り扱い、POST されてきたドキュメントを保存しなければならない。

If the merge is successful, the LRS MUST respond with HTTP status code 204 "No Content".

マージが成功した場合、LRS は HTTP ステータスコード 204 "No Content" で応答しなければならない。

If an AP needs to delete a property, it SHOULD use a PUT request to replace the whole document as described below.

AP がプロパティを削除する必要がある場合、後述するように PUT リクエストを用いてドキュメント全体を差し替えるべきである。

7.4 State API ステートAPI

Generally, this is a scratch area for Activity Providers that do not have their own internal storage, or need to persist state across devices. When using the State API, be aware of how the stateId parameter affects the semantics of the call. If it is included, the GET and DELETE methods will act upon a single defined state document identified by "stateId". Otherwise, GET will return the available ids, and DELETE will delete all state in the context given through the other parameters.

一般に、これはアクティビティプロバイダが独自の内部記憶領域を持たない、あるいは、複数の端末間で状態の共有が必要な場合などに利用されるスクラッチ領域である。ステート API を使用するとき、stateId パラメータが呼び出しの意味にどのような影響を与えるかということに注意すべきである。もし stateId パラメータが含まれる場合、GET と DELETE メソッドは、"stateId" によって指定される単独のステート文書に対して実行される。そうでない場合は、GET は利用可能な ID を返し、DELETE はその他のパラメータを通じて与えられた文脈の全てのステートを削除する。

PUT | POST | GET | DELETE activities/state PUT | POST | GET | DELETE アクティビティ/ステート

Example endpoint: <http://example.com/xAPI/activities/state>

エンドポイント例: <http://example.com/xAPI/activities/state>

Stores, fetches, or deletes the document specified by the given stateId that exists in the context of the specified Activity, Agent, and registration (if specified).

アクティビティ、エージェント、そして登録(存在すれば)からなる文脈に与えられた stateId によって指定される文書を記録、取り込み、又は削除する。

Returns: (PUT | POST | DELETE) 204 No Content, (GET) 200 OK - State Content

Parameterパラメータ	Typeタイプ	Required必須	Description説明
activityId	String	yes	The Activity id associated with this state.このステートと関連したアクティビティID
agent	JSON	yes	The Agent associated with this state.このステートと関連したエージェント
registration	UUID	no	The registration id associated with this state.このステートと関連した登録
stateId	String	yes	The id for this state, within the given context.与えられた文脈における、このステートのID

GET activities/state GET アクティビティ/ステート

Example endpoint: <http://example.com/xAPI/activities/state>

エンドポイント例: <http://example.com/xAPI/activities/state>

Fetches ids of all state data for this context (Activity + Agent \[+ registration if specified\]). If "since" parameter is specified, this is limited to entries that have been stored or updated since the specified timestamp(exclusive).

この文脈(アクティビティ+エージェント[指定された場合の登録])における全てのステートデータのidを取り込む。もし、"since" パラメータが指定されている場合、指定された timestamp (その時刻を含まない)よりあとに記録または更新されたエントリのみ制限される。

Returns: 200 OK, Array of ids

戻り値: 200 OK, Array of ids

Parameterパラメータ	Typeタイプ	Required必須	Description説明
activityId	String	yes	The Activity id associated with these states. これらのステートと関連したアクティビティ ID
agent	JSON	yes	The Actor associated with these states. これらのステートと関連したエージェント
registration	UUID	no	The registration id associated with these states. これらのステートと関連した登録
since	Timestamp	no	Only ids of states stored since the specified timestamp (exclusive) are returned. 指定された timestamp (その時刻を含まない) よりあとに記録されたステート ID のみが返却される。

DELETE activities/state DELETE アクティビティ/ステート

Example endpoint: <http://example.com/xAPI/activities/state>

エンドポイント例: <http://example.com/xAPI/activities/state>

Deletes all state data for this context (Activity + Agent [+ registration if specified]).

この文脈(アクティビティ+エージェント[指定された場合は登録])における全てのステートデータを削除する。

Returns: 204 No Content

戻り値: 204 No Content

Parameterパラメータ	Typeタイプ	Required必須	Description説明
activityId	String	yes	The Activity id associated with this state.このステートと関連したアクティビティ ID
agent	JSON	yes	The Actor associated with this state.このステートと関連したエージェント
registration	UUID	no	The registration id associated with this state.このステートと関連した登録 ID

7.5 Activity Profile API アクティビティ プロファイル API

The Activity Profile API is much like the State API, allowing for arbitrary key / document pairs to be saved which are related to an Activity. When using the Activity Profile API for manipulating documents, be aware of how the profileId parameter affects the semantics of the call. If it is included, the GET and DELETE methods will act upon a single defined document identified by "profileId". Otherwise, GET will return the available ids, and DELETE will delete all state in the context given through the other parameters.

アクティビティ プロファイル API は、ステート API と似ているものであり、任意のキー / 文書のペアをアクティビティに関連させて保存することを可能とする。文書操作のためにアクティビティ プロファイル API を使用するとき、profileId パラメータが呼び出しの意味にどのような影響を与えるかということに注意すべきである。もし profileId パラメータが含まれる場合、GET と DELETE メソッドは、"profileId" によって定義される単独の文書に従って実行される。そうでない場合は、GET は利用可能な id を返し、DELETE はその他のパラメータを通じて与えられた文脈の全てのステートを削除する。

The Activity Profile API also includes a method to retrieve a full description of an Activity from the LRS.

アクティビティ プロファイル API もまた、LRS からアクティビティの全ての説明を読みだすメソッドを含んでいる。

GET activities GET アクティビティ

Example endpoint: <http://example.com/xAPI/activities>

エンドポイント例: <http://example.com/xAPI/activities>

Loads the complete Activity Object specified.

指定された全アクティビティオブジェクトをロードする。

Returns: 200 OK - Content

戻り値: 200 OK, Content

Parameterパラ	Typeタイ	Required必
-------------	--------	-----------

Parameter/パラメータ	Type/タイプ	Required/必須	Description/説明
activityId	String	yes	The id associated with the Activities to load. ロードするアクティビティと関連したアクティビティ ID

PUT | POST | GET | DELETE activities/profile PUT | POST | GET | DELETE アクティビティ/プロフィール

Example endpoint: <http://example.com/xAPI/activities/profile>

エンドポイント例: <http://example.com/xAPI/activities/profile>

Saves/retrieves/deletes the specified profile document in the context of the specified Activity.

指定されたアクティビティの文脈で特定のプロフィール文書を保存 / 読み出し / 削除する。

Returns: (PUT | POST | DELETE) 204 No Content, (GET) 200 OK - Profile Content

戻り値: (PUT | POST | DELETE) 204 No Content, (GET) 200 OK, Profile Content

Parameter/パラメータ	Type/タイプ	Required/必須	Description/説明
activityId	String	yes	The Activity id associated with this profile. このプロフィールと関連したアクティビティ ID
profileId	String	yes	The profile id associated with this profile. このプロフィールと関連したプロフィール ID

GET activities/profile GET アクティビティ/プロフィール

Example endpoint: <http://example.com/xAPI/activities/profile>

エンドポイント例: <http://example.com/xAPI/activities/profile>

Loads ids of all profile entries for an Activity. If "since" parameter is specified, this is limited to entries that have been stored or updated since the specified timestamp (exclusive).

アクティビティのために全てのプロフィールエントリの id をロードする。もし、"since" パラメータが指定されている場合、指定された timestamp (その時刻を含まない) よりあとに記録または更新されたエントリだけに制限される。

Returns: 200 OK - List of ids

戻り値: 200 OK, List of ids

Parameter/パラメータ	Type/タイプ	Required/必須	Description/説明
activityId	String	yes	The Activity id associated with these profiles. これらのプロフィールと関連したアクティビティ ID
since	Timestamp	no	Only ids of profiles stored since the specified timestamp (exclusive) are returned. 指定された timestamp (その時刻を含まない) よりあとに記録されたプロフィール ID のみが返却される

7.6 Agent Profile API エージェント プロファイル API

The Agent Profile API is much like the State API, allowing for arbitrary key / document pairs to be saved which are related to an Agent. When using the Agent Profile API for manipulating documents, be aware of how the profileId parameter affects the semantics of the call. If it is included, the GET and DELETE methods will act upon a single defined document identified by "profileId". Otherwise, GET will return the available ids, and DELETE will delete all state in the context given through the other parameters.

エージェントプロフィール API はステート API と似ているものであり、任意のキー / 文書のペアをエージェントと関連させて保存することを可能とする。文書操作のためにエージェントプロフィール API を使用するとき、profileId パラメータが呼び出しの意味にどのような影響を与えるかということに注意すべきである。もし profileId パラメータが含まれる場合、GET と DELETE メソッドは、"profileId" によって定義される単独の文書に従って実行される。そうでない場合は、GET は利用可能な id を返し、DELETE はその他のパラメータを通じて与えられた文脈の全てのステートを削除する。

The Agent Profile API also includes a method to retrieve a special Object with combined information about an Agent derived from an outside service, such as a directory service.

エージェントプロフィール API もまた、ディレクトリサービスなどの外部サービスから派生したエージェントに関する結合情報と一体となった、特殊なオブジェクトを読みだすメ

ソッドを含んでいる。

GET agents GET エージェント

Example endpoint: <http://example.com/xAPI/agents>

エンドポイント例: <http://example.com/xAPI/agents>

Return a special, Person Object for a specified Agent. The Person Object is very similar to an Agent Object, but instead of each attribute having a single value, each attribute has an array value, and it is legal to include multiple identifying properties. Note that the argument is still a normal Agent Object with a single identifier and no arrays. Note that this is different from the FOAF concept of person, person is being used here to indicate a person-centric view of the LRS Agent data, but Agents just refer to one persona (a person in one context).

指定されたエージェントに対して特別の、Person オブジェクトを返却する。Person オブジェクトはエージェントオブジェクトと非常に似ているが、それぞれの要素が単一の値を持つ代わりに、それぞれの要素が配列の値を持ち、複数の識別プロパティを含むことが許されている。ここで留意すべきは、引数が単一の識別子を持つ通常のエージェントオブジェクトであり、配列ではないということである。これは FOAF の person 概念とは異なることに注意してほしい。ここでは person は LRS エージェントデータの person 中心の観点を示す存在として使用されるのであり、しかし、エージェントは、ただ一つのペルソナ(一つの文脈における person)を参照しているのである。

An LRS capable of returning multiple identifying properties for a Person Object SHOULD require the connecting credentials have increased, explicitly given permissions. An LRS SHOULD reject insufficiently privileged requests with 403 "Forbidden". If an LRS does not have any additional information about an Agent to return, the LRS MUST still return a Person when queried, but that Person Object will only include the information associated with the requested Agent.

Person オブジェクトに複数の識別プロパティを返却する能力がある LRS は、明示的にパーミッションを与えて、連結する資格情報が増加することを要求すべきである。LRS は 403 "Forbidden" で、権限の不十分なリクエストを拒絶すべきである。LRS がエージェントに関して返答すべき追加情報を持たない場合、照会時に Person を返却しなければならないが、Person オブジェクトはリクエストされたエージェントと関連した情報のみを含む。

Person properties Person プロパティ

All array properties must be populated with members with the same definition as the similarly named property from Agent Objects.

全ての配列プロパティは、エージェントオブジェクトからの類似命名されたプロパティと同じ定義で要素を追加しなければならない。

Propertyプロパティ	Typeタイプ	Description説明
objectType	String	"Person". Required. 必須
name	Array of strings.	Optional. List of names of Agents to retrieve. 任意. 読み出しするエージェント名リスト
mbox	Array of IRIs in the form "mailto:email address".	List of e-mail addresses of Agents to retrieve. 読み出しするエージェントのメールアドレス リスト
mbox_sha1sum	Array of strings.	List of the SHA1 hashes of mailto IRIs (such as go in an mbox property). mailto IRIs (例えば mbox プロパティなど)の SHA1 ハッシュリスト
openid*	Array of strings.	List of openids that uniquely identify the Agents to retrieve. 読み出しするエージェントを一意的に識別する openid リスト
account*	Array of account objects.	List of accounts to match. Complete account Objects (homePage and name) must be provided. 適合するアカウントリスト。完全なアカウントオブジェクト(ホームページや名前)が提供されなければならない。

See also: Section 4.1.2.1 Agent.

参照: Section 4.1.2.1 Agent.

Returns: 200 OK - Expanded Agent Object

戻り値 200 OK, Expanded Agent Object

Parameterパラメータ	Typeタイプ	Required必須	Description説明
agent	Object (JSON)	yes	The Agent representation to use in fetching expanded Agent information. 拡張されたエージェント情報を取り込む際に使用するエージェント表現

PUT | POST | GET | DELETE agents/profile PUT | POST | GET | DELETE エージェント/プロフィール

Example endpoint: <http://example.com/xAPI/agents/profile>

エンドポイント例: <http://example.com/xAPI/agents/profile>

Saves/retrieves/deletes the specified profile document in the context of the specified Agent.

指定されたエージェントの文脈で特定のプロフィール文書を保存/読み出し/削除する。

Returns: (PUT | POST | DELETE) 204 No Content, (GET) 200 OK - Profile Content

戻り値:(PUT | POST | DELETE) 204 No Content, (GET) 200 OK - Profile Content

Parameterパラメータ	Typeタイプ	Required必須	Description説明
agent	Object (JSON)	yes	The agent associated with this profile. このプロフィールに関連したエージェント
profileid	String	yes	The profile id associated with this profile. このプロフィールに関連したプロフィール ID

GET agents/profile GET エージェント/プロフィール

Example endpoint: <http://example.com/xAPI/agents/profile>

エンドポイント例: <http://example.com/xAPI/agents/profile>

Loads ids of all profile entries for an Agent. If "since" parameter is specified, this is limited to entries that have been stored or updated since the specified timestamp (exclusive).

エージェントのために全てのプロフィールエントリの ID をロードする。"since" パラメータが指定されている場合、指定された timestamp (その時刻を含まない)よりあとに記録または更新されたエントリのみ制限される。

Returns: 200 OK - List of ids

戻り値: 200 OK , List of IDs

Parameterパラメータ	Typeタイプ	Required必須	Description説明
agent	Object (JSON)	yes	The Agent associated with this profile. このプロフィールに関連したエージェント
since	Timestamp	no	Only ids of profiles stored since the specified timestamp (exclusive) are returned 指定された timestamp (その時刻を含まない)よりあとに記録されたプロフィール ID のみが返却される。

7.7. About resource About リソース

GET about GET about

Example endpoint: <http://example.com/xAPI/about>

エンドポイント例: <http://example.com/xAPI/about>

Description説明

Returns JSON Object containing information about this LRS, including the xAPI version supported.

この LRS に関する情報 (サポートする xAPI のバージョンを含む) を返答する。

Rationale背景

Primarily this resource exists to allow Clients that suport multiple xAPI versions to decide which version to use when communicating with the LRS. Extensions are included to allow other uses to emerge.

主に、このリソースは、複数の xAPI バージョンをサポートするクライアントに、LRS との通信時にどのバージョンを使用するかを決定させるために存在する。拡張機能は他の用途が現れることを見越して含まれている。

Details詳細

Returns: 200 OK - Single 'about' JSON document.

戻り値: 200 OK - Single 'about' JSON document.

propertyプロパティ	typeタイプ	description説明
version	array of version strings	xAPI versions this LRS supports この LRS がサポートする xAPI バージョン
Extensions	Object	A map of other properties as needed. 必要に応じた他のプロパティのマッピング

LRS RequirementLRS 要件

* MUST return the JSON document described above, with a version property that includes the latest minor and patch version the LRS conforms to, for each major version.

- 各々のメジャーバージョンに対して、その LRS が準拠する最新のマイナーおよびパッチバージョンを示すバージョンプロパティを含んだ、上述した JSON 文書を送らなければならない。
 - For version 1.0.0 of this specification, this means that "1.0.0" MUST be included; "0.9" and "0.95" MAY be included. (For the purposes of this requirement, "0.9" and "0.95" are considered major versions.)
 - この仕様書のバージョン 1.0.0 では、"1.0.0" が含まなければならないことを意味する。"0.9" や "0.95" は含まれてもよい。(この要件の目的のため、"0.9" や "0.95" はメジャーバージョンと考えられている。)
- SHOULD allow unauthenticated access to this resource

このリソースへの認証されていないアクセスを許可すべきである。

- MUST NOT reject requests based on their version header as would otherwise be required by 6.2 API Versioning.

6.2 API Versioning によって必要とされない限り、バージョンヘッダーに基づいてリクエストを拒絶してはならない。

7.8 Cross Origin Requests クロス オリジン リクエスト

One of the goals of the xAPI is to allow cross-domain tracking, and even though xAPI seeks to enable tracking from applications other than browsers, browsers still need to be supported. Internet Explorer 8 and 9 do not implement Cross Origin Resource Sharing, but rather use their own Cross Domain Request API, which cannot use all of the xAPI as described above due to only supporting "GET" and "POST", and not allowing HTTP headers to be set.

xAPI のゴールの一つは、クロスドメイントラッキングを許可することであり、xAPI がブラウザ以外のアプリケーションからトラッキングができるようにしようとしても、ブラウザはサポートされる必要がある。Internet Explorer 8 と 9 は、クロスオリジンリソースシェアリングを実装しておらず、自身のクロスドメインリクエスト API を利用する。"GET" と "POST" をサポートしているのみで、HTTP ヘッダーへのセットを許可されていないため、上述した xAPI の全てを利用することはできない。

The following describes alternate syntax for consumers to use only when unable to use the usual syntax for specific calls due to the restrictions mentioned above. All LRSs must support this syntax.

上記の制限により特定の呼び出しのための通常の構文を使用できないときにのみ消費者が使用するべき別の構文について、以下に述べる。全ての LRS はこの構文をサポートしなければならない。

Method: All xAPI requests issued must be POST. The intended xAPI method must be included as the only query string parameter on the request.

メソッド: 発行された全 xAPI リクエストは POST されなければならない。対象とする xAPI メソッドはリクエスト上の唯一の問合せ文字列パラメータとして含まなければならない。

(example: <http://example.com/xAPI/statements?method=PUT>)

(例: <http://example.com/xAPI/statements?method=PUT>)

Headers: Any required parameters which are expected to appear in the HTTP header must instead be included as a form parameter with the same name.

ヘッダー: HTTP ヘッダーに記述すべき必須のパラメータは、同名の form パラメータとして与えなければならない。

Content: If the xAPI call involved sending content, that content must now be encoded and included as a form parameter called "content". The LRS will interpret this content as a UTF-8 string. Storing binary data is not supported with this syntax.

コンテンツ: xAPI の呼び出しがコンテンツの送信を伴う場合、"content" と呼ばれるフォームパラメータとしてコンテンツはエンコードされ、含まなければならない。LRS は UTF-8 文字列としてこのコンテンツを解釈する。バイナリデータの記録は、この構文でサポートされない。

Attachments: Sending attachment data requires sending a multipart/mixed request, therefore sending attachment data is not supported with this syntax. See 4.1.11. Attachments

添付文書: 添付文書データの送信は、multipart/mixed リクエストを送信する必要があり、それゆえ添付データの送信はこの構文でサポートされない。4.1.11. Attachments を参照。

See Appendix B for an example function written in JavaScript which transforms a normal request into one using this alternate syntax.

通常リクエストをこの代替構文を使用して変換した JavaScript で記述した関数の例については Appendix B を参照。

It should also be noted that versions of Internet Explorer lower than 10 do not support Cross Domain Requests between HTTP and HTTPS. This means that for IE9 and lower, if the LRS is on an HTTPS domain, the Client sending the Statement must also be on HTTPS. If the LRS is on HTTP, the Client must be too.

Internet Explorer 10 より低いバージョンが、HTTP と HTTPS 間でクロスドメインリクエストをサポートしていないことに留意すべきである。これは IE9 とそれ以下を意味しており、LRS が HTTPS ドメイン上にある場合、ステートメントを送信するクライアントも HTTPS 上になければならない。LRS が HTTP 上にある場合は、クライアントも HTTP 上になければならない。

There may be cases where there is a requirement for the Client Activity Provider to support IE8 and IE9 where the Client code is hosted on a different scheme (HTTP or HTTPS) from the LRS. In these cases, proxy is needed to communicate to the LRS. Two simple solutions might be to 1) set up a proxy pass through on the same scheme as the Client code to the LRS or 2) to host an intermediary server side LRS on the same scheme as the Client code to route Statements to the target LRS. An LRS MAY choose to provide both HTTP and HTTPS endpoints to support this use case. HTTP is inherently less secure than HTTPS, and both LRS and Client should consider the security risks before making the decision to use this scheme.

クライアントアクティビティプロバイダにとって、クライアントコードが LRS とは異なるスキーム(HTTP or HTTPS)上でホストされている IE8 と IE9 をサポートすることが要求されるケースがある。これらのケースにおいて、プロキシは LRS と通信するために必要とされる。二つの簡単な解決策として:

1) クライアントコードから LRS へ同じスキームのプロキシパススルーを構成する。あるいは、2) ターゲット LRS へのステートメント経路をクライアントコードと同じスキームとした、仲介サーバーサイド LRS をホストする

このユースケースをサポートするために、LRS は HTTP と HTTPS エンドポイント両方の提供を選んでよい。HTTP は HTTPS よりも本質的に安全性が低くなるため、LRS とクライアントの両方がこのスキームを利用する決定を下す前にセキュリティリスクを考慮すべきである。

7.9 Validation パリテーション

The function of the LRS within the xAPI is to store and retrieve Statements. As long as it has sufficient information to perform these tasks, it is expected that it does them. Validation of Statements in the Experience API is focused solely on syntax, not semantics. It SHOULD enforce rules regarding structure, but SHOULD NOT enforce rules regarding meaning. Enforcing the rules that ensure valid meaning among Verb definitions, Activity types, and extensions is the responsibility of the Activity Provider sending the Statement.

xAPI 内での LRS の機能はステートメントを記録することや読み出すことである。これらのタスクを実行するための十分な情報がある限り、ステートメントの記録や読み出しが期待される。Experience API でのステートメントのパリテーションは、もっぱら構文に集中され、意味には及ばない。構造に関するルールを施行すべきであり、意味に関するルールを施行すべきではない。動詞の定義、アクティビティタイプ、そして拡張機能の中で有効な意味を保証するようなルールを施行することは、ステートメントを送信するアクティビティプロバイダの責任である。

7.10. HTTP HEAD HTTP HEAD

Description 説明

The LRS will respond to HEAD requests by returning the meta information only, using the HTTP headers, and not the actual document.

LRS は、実際のドキュメントを使用せず、HTTP ヘッダーを使用してメタ情報を返却することだけで HEAD リクエストに応答する。

Rationale 背景

Clients accessing the LRS may need to check if a particular Statement exists, or determine the modification date of documents such as state or Activity or Agent profile. Particularly for large documents it's more efficient not to get the entire document just to check its modification date.

LRS にアクセスするクライアントは、特定のステートメントが存在する場合にチェックを必要とし、ステート、アクティビティ、エージェントプロファイルのドキュメント修正日を決定することができる。特に大規模なドキュメントでは、ドキュメント修正日を確認する際、全体ドキュメントを取得しないため、より効率的である。

LRS Requirements LRS 必要条件

* The LRS MUST respond to any HTTP HEAD request as it would have responded to an otherwise identical HTTP GET request except: * The message-body MUST be omitted. * The Content-Length header MAY be omitted, in order to avoid wasting LRS resources.

- LRS は(下記を)除外して、他の同一の HTTP GET リクエストに対する応答と同じように、どの HTTP HEAD リクエストにも応答しなければならない。
 - *メッセージボディは、省略しなければならない。
 - *LRS リソースの浪費を避けるために、Content-Length ヘッダーは省略される。

Appendix A: Bookmarkletブックマークレット

An xAPI Bookmarklet enables individual user tracking with base authentication. Examples could be an "I think this," "I learned this," "I like this," or "I don't like this" Statement that allows self-reporting. The following is an example of such a bookmarklet, and the Statement that this bookmarklet would send if used on the page: <http://adlnet.gov/xapi>.

xAPI ブックマークレットは、基本認証によって個々のユーザをトラッキングすることを可能にする。例をあげると、自己申告を許す「私はこれを考える」、「私はこれを学ぶ」、「私はこれが好き」、もしくは「私はこれが嫌い」というステートメントである。以下は、ブックマークレットの一例であり、ページ (<http://adlnet.gov/xapi>) 上で利用した場合、このブックマークレットが送るであろうステートメントである。

The bookmarklet MAY be provided by the LRS to track a specific user for behavior analytics.

ブックマークレットは、特定のユーザの行動分析のために追跡をするような LRS で提供されてもよい。

Therefore the LRS IRL, authentication, and Actor information is hard coded into the bookmarklet. Note that since the authorization token must be included in the bookmarklet, the LRS should provide a token with limited privileges, Ideally the token should enable the storage of self-reported learning Statements only.

したがって、LRS の IRL、認証、およびアクタの情報は、ブックマークレット内にハードコードされる。認証トークンがブックマークレットに含まれるため、LRS はトークンに対して限定された権限、理想的には、自己申告の学習に関するステートメントの記録のみの権限、を提供すべきことに注意が必要である。

The UUID SHOULD be included as part of the bookmarklet PUT Statement. If a Statement is POSTed without a UUID, the LRS MUST generate one.

UUID は、ブックマークレットの PUT ステートメントの一部として含まれるべきである。もしステートメントが UUID のないステートメントが POST された場合、LRS が生成しなければならない。

In order to allow cross-domain reporting of Statements, a browser that supports the "Access-Control-Allow-Origin" and "Access-Control-Allow-Methods" headers must be used, such as IE 8+, FF 3.5+, Safari 4+, Safari iOS Chrome, or Android browser. Additionally the server must set the required headers.

ステートメントのクロスドメイン報告を可能にするためには、「 Access-Control-Allow-Origin 」と「 Access-Control-Allow-Methods 」のヘッダーをサポートするブラウザを使う必要がある (IE8 以上、Firefox3.5 以上、Safari4 以上、iOS 上の Chrome、もしくは Android ブラウザ)。さらに、サーバは必要とされるヘッダーを付けなければならない。

In the example below, the following values in the first few lines should be replaced with your own values. All other values should be left as they are.

以下の例について、先頭から数行の値は実際の値に置き換えて利用すること。それ以外の値は変更しないこと。

Value in example例文中に用いる値	Explanation説明
<code>http://localhost:8080/xAPI/</code>	Endpoint of the LRS to send the Statements to.ステートメントを送信する LRS のエンドポイント
<code>dGVzdDpwYXNzd29yZA==</code>	Base 64 encoded username and password, usually in the form "username : password".通常 "username : password" の形式で Base 64 エンコードされるユーザ名とパスワード。
<code>learner@example.adlnet.gov</code>	Email address of the learner using the bookmarklet.ブックマークレットを使用する学習者のメールアドレス

```
var url = "http://localhost:8080/xAPI/statements?statementId="+_ruuid();
var auth = "Basic dGVzdDpwYXNzd29yZA==";
var statement = {
  "actor" : {
    "objectType" : "Agent",
    "mbox" : "mailto:learner@example.adlnet.gov"
  },
  "verb" : {
    "id" : "",
    "display" : {}
  },
  "object" : {
    "id" : "",
    "definition" : {}
  }
};
var definition = statement.object.definition;
```

```

statement.verb.id = 'http://adlnet.gov/expapi/verbs/experienced';
statement.verb.display = { "en-US" : "experienced" };
statement.object.id = window.location.toString();
definition.type = "http://adlnet.gov/expapi/activities/link";

var xhr = new XMLHttpRequest();
xhr.open("PUT", url, true);
xhr.setRequestHeader("X-Experience-API-Version", "1.0");
xhr.setRequestHeader("Content-Type", "application/json");
xhr.setRequestHeader("Authorization", auth);
xhr.onreadystatechange = function() {
  if(xhr.readyState == 4) {
    alert(xhr.status + " : " + xhr.responseText);
  }
};
xhr.send(JSON.stringify(statement));

/*!
Modified from: Math.uuid.js (v1.4)
http://www.broofa.com
mailto:robert@broofa.com

Copyright (c) 2010 Robert Kieffer
Dual licensed under the MIT and GPL licenses.
*/
function _ruuid() {
  return 'xxxxxxxx-xxxx-4xxx-yxxx-xxxxxxxxxxxx'.replace(/[xy]/g, function(c) {
    var r = Math.random()*16|0, v = c == 'x' ? r : (r&0x3|0x8);
    return v.toString(16);
  });
}

```

Example Statement Using Bookmarklet ブックマークレットを使用したステートメント例

Headers ヘッダ

```

{
  "X-Experience-API-Version": "1.0.0",
  "Content-Type": "application/json; charset=UTF-8",
  "Authorization": "d515309a-044d-4af3-9559-c041e78eb446",
  "Referer": "http://adlnet.gov/xapi/",
  "Content-Length": "###",
  "Origin": "http://adlnet.gov"
}

```

Method Pathメソッドパス

PUT : /xAPI/Statements/?statementId=ed1d064a-eba6-45ea-a3f6-34cdf6e1dfd9

Body:

```

{
  "actor": {
    "objectType": "Agent",
    "mbox": "mailto:learner@example.adlnet.gov"
  },
  "verb" : {
    "id": "http://adlnet.gov/expapi/verbs/experienced",
    "display": {
      "en-US": "experienced"
    }
  },
  "object": {
    "id": "http://adlnet.gov/xapi/",
    "definition": {
      "type": "http://adlnet.gov/expapi/activities/link"
    }
  }
}

```

Appendix B: Creating an "IE Mode" RequestIE用リクエスト作成

```
function getIEModeRequest(method, url, headers, data){

    var newUrl = url;

    // Everything that was on query string goes into form varsクエリストリングにある情報は全てフォーム変数に

    var formData = new Array();
    var qsIndex = newUrl.indexOf('?');
    if(qsIndex > 0){
        formData.push(newUrl.substr(qsIndex+1));
        newUrl = newUrl.substr(0, qsIndex);
    }

    // Method has to go on query string, and nothing elseメソッドはクエリストリングに入れる。それ以外は入れない
    newUrl = newUrl + '?method=' + method;

    // Headersヘッダ

    if(headers !== null){
        for(var headerName in headers){
            formData.push(
                headerName + "=" +
                encodeURIComponent(
                    headers[headerName]));
        }
    }

    // The original data is repackaged as "content" form varオリジナルデータは "content" フォーム変数に再パックされる

    if(data !== null){
        formData.push('content=' + encodeURIComponent(data));
    }

    return {
        "method":"POST",
        "url":newUrl,
        "headers":{},
        "data":formData.join("&")
    };
}
```

Appendix C: Example definitions for Activities of type "cmi.interaction"アクティビティタイプ「 cmi.interaction 」のための定義例

true-falsetrue/false

```
"definition": {
  "description": {
    "en-US": "Does the xAPI include the concept of statements?"
  },
  "type": "http://adlnet.gov/expapi/activities/cmi.interaction",
  "interactionType": "true-false",
  "correctResponsesPattern": [
    "true"
  ]
}
```

choice

```
"definition": {
  "description": {
```

```

"en-US": "Which of these prototypes are available at the beta site?"
},
"type": "http://adlnet.gov/expapi/activities/cmi.interaction",
"interactionType": "choice",
"correctResponsesPattern": [
  "golf[,]tetris"
],
"choices": [
  {
    "id": "golf",
    "description": {
      "en-US": "Golf Example"
    }
  },
  {
    "id": "facebook",
    "description": {
      "en-US": "Facebook App"
    }
  },
  {
    "id": "tetris",
    "description": {
      "en-US": "Tetris Example"
    }
  },
  {
    "id": "scrabble",
    "description": {
      "en-US": "Scrabble Example"
    }
  }
]
}
...

```

fill-in

```

"definition": {
  "description": {
    "en-US": "Ben is often heard saying: "
  },
  "type": "http://adlnet.gov/expapi/activities/cmi.interaction",
  "interactionType": "fill-in",
  "correctResponsesPattern": [
    "Bob's your uncle"
  ]
}
...

```

likert

```

"definition": {
  "description": {
    "en-US": "How awesome is Experience API?"
  },
  "type": "http://adlnet.gov/expapi/activities/cmi.interaction",
  "interactionType": "likert",
  "correctResponsesPattern": [
    "likert_3"
  ],
  "scale": [
    {
      "id": "likert_0",
      "description": {
        "en-US": "It's OK"
      }
    }
  ]
}
...

```

```

    }
  },
  {
    "id": "likert_1",
    "description": {
      "en-US": "It's Pretty Cool"
    }
  },
  {
    "id": "likert_2",
    "description": {
      "en-US": "It's Damn Cool"
    }
  },
  {
    "id": "likert_3",
    "description": {
      "en-US": "It's Gonna Change the World"
    }
  }
]
}
...

```

matching

```

"definition": {
  "description": {
    "en-US": "Match these people to their kickball team:"
  },
  "type": "http://adlnet.gov/expapi/activities/cmi.interaction",
  "interactionType": "matching",
  "correctResponsesPattern": [
    "ben[.]3[,],chris[.]2[,],troy[.]4[,],freddie[.]1"
  ],
  "source": [
    {
      "id": "ben",
      "description": {
        "en-US": "Ben"
      }
    },
    {
      "id": "chris",
      "description": {
        "en-US": "Chris"
      }
    },
    {
      "id": "troy",
      "description": {
        "en-US": "Troy"
      }
    },
    {
      "id": "freddie",
      "description": {
        "en-US": "Freddie"
      }
    }
  ],
  "target": [
    {
      "id": "1",
      "description": {
        "en-US": "Swift Kick in the Grass"
      }
    }
  ],
}

```

```

{
  "id": "2",
  "description": {
    "en-US": "We got Runs"
  }
},
{
  "id": "3",
  "description": {
    "en-US": "Duck"
  }
},
{
  "id": "4",
  "description": {
    "en-US": "Van Delay Industries"
  }
}
]
}
...

```

performance

```

"definition": {
  "description": {
    "en-US": "This interaction measures performance over a day of RS sports:"
  },
  "type": "http://adlnet.gov/expapi/activities/cmi.interaction",
  "interactionType": "performance",
  "correctResponsesPattern": [
    "pong[.]1:[,][dg[.]:10[,],lunch[.]"
  ],
  "steps": [
    {
      "id": "pong",
      "description": {
        "en-US": "Net pong matches won"
      }
    },
    {
      "id": "dg",
      "description": {
        "en-US": "Strokes over par in disc golf at Liberty"
      }
    },
    {
      "id": "lunch",
      "description": {
        "en-US": "Lunch having been eaten"
      }
    }
  ]
}

```

sequencing

```

"definition": {
  "description": {
    "en-US": "Order players by their pong ladder position:"
  },
  "type": "http://adlnet.gov/expapi/activities/cmi.interaction",
  "interactionType": "sequencing",
  "correctResponsesPattern": [

```

```

"tim[,]mike[,]ells[,]ben"
],
"choices": [
{
  "id": "tim",
  "description": {
    "en-US": "Tim"
  }
},
{
  "id": "ben", "description": {
    "en-US": "Ben"
  }
},
{
  "id": "ells",
  "description": {
    "en-US": "Ells"
  }
},
{
  "id": "mike",
  "description": {
    "en-US": "Mike"
  }
}
]
}
...

```

numeric

```

"definition": {
  "description": {
    "en-US": "How many jokes is Chris the butt of each day?"
  },
  "type": "http://adlnet.gov/expapi/activities/cmi.interaction",
  "interactionType": "numeric",
  "correctResponsesPattern": [
    "4:"
  ]
}

```

other

```

"definition": {
  "description": {
    "en-US": "On this map, please mark Franklin, TN"
  },
  "type": "http://adlnet.gov/expapi/activities/cmi.interaction",
  "interactionType": "other",
  "correctResponsesPattern": [
    "(35.937432,-86.868896)"
  ]
}

```

Appendix D: Example statementsステートメント例

Example of a simple statement (line breaks are for display purposes only): 簡単なステートメント例 (改行は表示目的)

```

...
{
  "id":"fd41c918-b88b-4b20-a0a5-a4c32391aaa0",
  "actor":{
    "objectType": "Agent",
    "name":"Project Tin Can API",
    "mbox":"mailto:user@example.com"
  },
  "verb":{
    "id":"http://adlnet.gov/expapi/verbs/created",
    "display":{
      "en-US":"created"
    }
  },
  "object":{
    "id":"http://example.adlnet.gov/xapi/example/simplestatement",
    "definition":{
      "name":{
        "en-US":"simple statement"
      },
      "description":{
        "en-US":"A simple Experience API statement. Note that the LRS
        does not need to have any prior information about the Actor (learner), the
        verb, or the Activity/object."
      }
    }
  }
}
}
}
...

```

Typical simple completion with verb "attempted": Verbの「attempted」を用いた一般的で簡単な完了

```

...
{
  "actor":{
    "objectType": "Agent",
    "name":"Example Learner",
    "mbox":"mailto:example.learner@adlnet.gov"
  },
  "verb":{
    "id":"http://adlnet.gov/expapi/verbs/attempted",
    "display":{
      "en-US":"attempted"
    }
  },
  "object":{
    "id":"http://example.adlnet.gov/xapi/example/simpleCBT",
    "definition":{
      "name":{
        "en-US":"simple CBT course"
      },
      "description":{
        "en-US":"A fictitious example CBT course."
      }
    }
  },
  "result":{
    "score":{
      "scaled":0.95
    },
    "success":true,
    "completion":true
  }
}
}
...

```

Appendix E: Converting Statements to 1.0.0.0へステートメントを変換

Rationale 背景

This is a 1.0.0 specification, and as such implementers should not have to consider prior versions of the specification. However, prior versions did see notable adoption. This data conversion is specified in order to preserve the data tracked using earlier versions, and make it available to new implementers in a consistent manner.

これは、1.0.0の仕様であり、実装者が、従来のバージョンの仕様を考慮する必要がないように努めた。しかしながら、従来のバージョンは、注目に値する普及を示した。このデータコンバージョンは、旧バージョンを用いて記録されたデータを継続して活用でき、新しい実装者も同様の方式で利用できることを目指して作成した。

Details 詳細

Conversion of Statements created based on version 0.9 バージョン0.9を元に作られたステートメントのコンバージョン

A 1.0.0 system converting a Statement created in 0.9 MUST follow the steps below:

0.9から作成されたステートメントを変換する1.0.0システムは以下のステップに沿っていなければならない。

- If the Statement has been voided or uses Verbs, Activity types, or properties not included in the 0.9 specification, do not convert it. ステートメントが無効化された、または 0.9 版の仕様に含まれない動詞、アクティビティタイプまたはプロパティである場合は、変換をしない。
- Prefix "verb" with "http://adlnet.gov/expapi/verbs/". Verb に http://adlnet.gov/expapi/verbs/ の prefix をつける。
- Prefix any Activity ids which are not full absolute IRIs with "tag:adlnet.gov,2013:expapi:0.9:activities:". 絶対表記の IRI でないアクティビティ ID には、"tag:adlnet.gov,2013:expapi:0.9:activities:" の prefix をつける。
- Prefix any extension keys which are not full absolute IRIs with "tag:adlnet.gov,2013:expapi:0.9:extensions:". 完全な絶対 IRI でない拡張キーには、"tag:adlnet.gov,2013:expapi:0.9:activities:" の prefix をつける。
- Prefix Activity types with "http://adlnet.gov/expapi/activities/" アクティビティタイプに "http://adlnet.gov/expapi/activities/" の prefix を付ける。
- for each Agent (Actor): すべてのエージェント(アクタ)について:
 - Search for Inverse Functional Identifiers in this order: "mbox, mboxsha1sum, openId, account". *Keep the first populated Inverse Functional Identifier found and discard the rest.* 「mbox, mboxsha1sum, openId, account」, この順序で逆関数識別子を検索する。最初に見つけられた逆関数識別子を保持し、残りは破棄する。
 - For the above Inverse Functional Identifier, take the first element in the array and use that as the value of that Inverse Functional Identifier property, discarding any remaining elements. 上記の逆関数識別子については、配列で最初の要素を得る。そしてその逆関数識別子プロパティの値としてそれを利用し、残りの要素は破棄する。
 - If the "name" property is present, set it equal to the first element in the "name" array, discard the remaining elements. もしnameプロパティが存在する場合には、name配列の最初の要素にそれが等しくなるよう設定をし、残りの要素を破棄する。
 - Remove all remaining properties. 残りすべてのプロパティを削除する。
- Remove the "voided" property from the Statement, if present. Remember, if the value of the voided property is true, then the Statement MUST NOT be converted. もし存在するなら、ステートメントから " voided " プロパティを削除する。もし、" voided " プロパティの値が true なら、ステートメントは変換してはならない。
- Add "version": "1.0.0" 「version": "1.0.0"」を追加する。
- If an authority was not previously set, set the authority to an Agent identified by an account with a homePage set to the home page corresponding to the system performing the conversion and an accountName of "unknown". 権限が前もって設定されていない場合は、homePage に変換を実行したシステムに対応するホームページ、accountName に "unknown" を設定したアカウントによって定義されるエージェントを権限として設定する。
- if the Statement field in context was set, remove it from the Statement. コンテキストにおけるステートメントフィールドがセットされていた場合、ステートメントから削除すること。
- Preserve all other fields without modification, including "stored". Stored should still be updated if the Statement is passed to another system. 他すべてのフィールドは stored を含め、変更せずに保持すること。stored はステートメントが他システムへ渡った場合についても最新化されるべきである。

Conversion of Statements created based on version 0.95 バージョン0.95を元に作られたステートメントのコンバージョン

A 1.0.0 system converting a Statement created in 0.95 MUST follow the steps below: 0.95で作成されたステートメントを変換する1.0.0のシステムは、以下のステップに従わなければならない。

- If the Statement is voided, do not convert it. 無効となったステートメントは変換しないこと。
- Remove the "voided" property from the Statement, if present. Remember, if the value of the voided property is true, then the Statement MUST NOT be converted. 存在する場合、ステートメントから voided プロパティを削除すること。voided プロパティの値が true である場合、ステートメントを変換してはならない。
- Add "version": "1.0.0" 「 version ": "1.0.0"」を追加する。
- If an authority was not previously set, set the authority to an Agent identified by an account with a homePage set to the home page corresponding to the system performing the conversion and an accountName of "unknown". もし権限が前もって与えられてないなら、コンバージョンと "unknown" というアカウント名を実行するシステムと対応するホームページに、ホームページ設定と一緒にアカウントによって識別されたエージェントに権限を与える。
- if the Statement field in context was set to anything other than a StatementRef, remove it from the Statement. もしコンテキストのステートメントフィールドが StatementRef 以外に設定されたら、ステートメントからフィールドを削除する。
- Preserve all other fields without modification, including "stored". Stored should still be updated if the Statement is passed to another system. stored を含めたその他すべてのフィールドを変更なしで保存する。ステートメントが他のシステムに送られた場合は、stored は更新される。

Example 例

A 0.9 Statement: 0.9ステートメント

```

...
{
  "id": "d1eec41f-1e93-4ed6-acbf-5c4bd0c24269",
  "actor": {
    "objectType": "Person",
    "name": [
      "Joe Schmoe",
      "Joseph Schmoseph"
    ],
    "mbox": [
      "mailto:joe@example.com"
    ],
    "openid": [
      "http://openid.com/joe-schmoe"
    ]
  },
  "verb": "completed",
  "inProgress": false,
  "object": {
    "objectType": "Activity",
    "id": "http://www.example.com/activities/001",
    "definition": {
      "name": {
        "en-US": "Example Activity"
      },
      "type": "course"
    }
  },
  "result": {
    "completion": true
  },
  "context": {
    "instructor": {
      "objectType": "Person",
      "lastName": [
        "Dad"
      ],
      "firstName": [
        "Joe's"
      ],
      "mbox": [
        "mailto:joesdad@example.com"
      ]
    },
    "contextActivities": {
      "parent": {
        "objectType": "Activity",

```

```

        "id": "non-absolute-activity-id",
        "definition": {
          "name": {
            "en-US": "Another Activity"
          }
        }
      }
    },
    "timestamp": "2012-06-01T19:09:13.245Z",
    "stored": "2012-06-29T15:41:39.165Z"
  }
}

```

Converted to 1.0.0: 1.0.0に变换

```

{
  "version": "1.0.0",
  "id": "d1eec41f-1e93-4ed6-acbf-5c4bd0c24269",
  "actor": {
    "objectType": "Agent",
    "name": "Joe Schmo",
    "mbox": "mailto:joe@example.com"
  },
  "verb": {
    "id": "http://adlnet.gov/expapi/verbs/completed",
    "display": {
      "en-US": "completed"
    }
  },
  "object": {
    "objectType": "Activity",
    "id": "http://www.example.com/activities/001",
    "definition": {
      "name": {
        "en-US": "Example Activity"
      }
    },
    "type": "http://adlnet.gov/expapi/activities/course"
  },
  "result": {
    "completion": true
  },
  "context": {
    "instructor": {
      "objectType": "Agent",
      "mbox": "mailto:joesdad@example.com"
    },
    "contextActivities": {
      "parent": [
        {
          "objectType": "Activity",
          "id": "tag:adlnet.gov,2013:expapi:0.9:activities:non-absolute-activity-id",
          "definition": {
            "name": {
              "en-US": "Another Activity"
            }
          }
        }
      ]
    }
  },
  "timestamp": "2012-06-01T19:09:13.245Z",
  "stored": "2012-06-29T15:41:39.165Z",
  "authority": {
    "objectType": "Agent",
    "account": {
      "homePage": "http://www.example.com",
      "name": "unknown"
    }
  }
}

```

```
}
}
}
...
```

Appendix F: Example Signed Statement 署名ステートメント例

An example signed Statement, as described in: 4.4 Signed Statements.

4.4 署名ステートメントに記載された証明ステートメント例

The original Statement serialization to be signed. New lines in this example are included via CR+LF (0x0D + 0x0A).

署名されるべきオリジナルステートメントのリスト。この例では新しい行が「CR+LF (0x0D + 0x0A)」を含められる。

```
{
  "version": "1.0.0",
  "id": "33cfff416-e331-4c9d-969e-5373a1756120",
  "actor": {
    "mbox": "mailto:example@example.com",
    "name": "Example Learner",
    "objectType": "Agent"
  },
  "verb": {
    "id": "http://adlnet.gov/expapi/verbs/experienced",
    "display": {
      "en-US": "experienced"
    }
  },
  "object": {
    "id": "https://www.youtube.com/watch?v=xh4kIiH3Sm8",
    "objectType": "Activity",
    "definition": {
      "name": {
        "en-US": "Tax Tips & Information : How to File a Tax Return "
      },
      "description": {
        "en-US": "Filing a tax return will require filling out either a 1040, 1040A or 1040EZ form"
      }
    }
  },
  "timestamp": "2013-04-01T12:00:00Z"
}
...
```

Example private key for X.509 certificate that will be used for signing:

署名に使用されるX.509証明書の秘密鍵例

```
....
-----BEGIN RSA PRIVATE KEY-----
MIICXAIBAABgQDjxvZXF30wL4oKjZYXgR0ZyaX+u3y6+JqTqiNkFa/VTnet6Ly2
OT6ZmmcJEPnq3UnepHoOQ+Gfhhtkkl13j06j5iNn4obcCVWTL9yXNVJH+Ko+Xu4Y
1/ySPRrIPyTjtHdG0M2XzI1mmlqm+CAS+KCbJeH4tf543kIWC5pC5p3cVQIDAQAB
AoGA0ejdvGq2XKuddu1kWXl0Aphn4YmdPpPyCNTaxp1U6PBVMRjY0aNgLQE6b02p
/HJiU4Y4PkgzkEgCu0xf/mOq5DnSkX32ICoQS6jChABAe20ErPfm5t8h9YKsTfn9
401AouuwD9ePrteizd4YvHtiMMwmh5PtUoCbqLefawNApAECQQD1mdBW3zL0okUx
2pc4ttn2qArCG4CsEZMLlGRDd3FwPWJz3ZPNEEgZWxGSpA9F1QTZ6JYXIfejjRo
UuvRMWeBAkEA7WvzDBNcv4N+xeUKvH8ILti/BM58LraTtqJlZjQSOvek0srxtmDg
5of+rxN6IM4p7yvQa+7YOU0ukrVXjG+1QJBAI2mBrjzXgm9xTa5odn97JD7UMFA
/WHj1Me/Nx/35V52qaav1sZbluw+TvKMcqApYj5G2SUpSNudHLDGkmd2nQECQFfc
lBRK8g7ZncekbGW3aRLVGV0xClnLLTzw01amBKOUm8V6XxsMHQ6TE2D+fkJoNUY1
2HGpk+Fwvy2D1hrGuoUCQAXfaLSxtawdPt1ZTPVueF7ZikQDsVg+vtTFgpuHlor2
6EVc1RbHHZm32yvGDY8IkcoMfJQqLONDdLfS/05yoNU=
```


