

# 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](mailto: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)

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](mailto:helpdesk@adlnet.gov) までお送りください。

# 目次

---

- 1.0. 改定履歴
- 2.0. Experience API の役割
  - 2.1. xAPI における ADL の位置づけ
  - 2.2. 貢献者
    - 2.2.1. ワーキンググループの参加者
    - 2.2.2. 要求仕様収集への貢献者
  - 2.3 技術に詳しくない人への読み方ガイドライン
- 3.0. 用語の定義
- 4.0. ステートメント
  - 4.1. ステートメントプロパティ
    - 4.1.1. id (識別子)
    - 4.1.2. Actor (アクタ)
    - 4.1.3. Verb (動詞)
    - 4.1.4. Object (目的語)
    - 4.1.5. Result (結果)
    - 4.1.6. Context (文脈)
    - 4.1.7. Timestamp (タイムスタンプ)
    - 4.1.8. Stored
    - 4.1.9. [Authority (権限)  
] (#authority)
    - 4.1.10. Version (バージョン)
    - 4.1.11. Attachments (添付資料)
    - 4.1.12. データの制約
  - 4.2. ステートメントの検索
  - 4.3. 無効
  - 4.4. 署名付きステートメント
- 5.0. 各種タイプ
  - 5.1. ドキュメント
  - 5.2. 言語マップ
  - 5.3. 拡張
  - 5.4. 識別子メタデータ
- 6.0. ランタイム通信
  - 6.1. エンコーディング
  - 6.2. API のバージョン管理
  - 6.3. 同時実行
  - 6.4. セキュリティ
    - 6.4.1. 各シナリオの取り扱い方法
    - 6.4.2. OAuth 認証スコープ
- 7.0. Data Transfer (REST)
  - 7.1. エラーコード
  - 7.2. ステートメント API
  - 7.3. ドキュメント API
  - 7.4. ステート API
  - 7.5. アクティビティプロファイル API
  - 7.6. エージェントプロファイル API
  - 7.7. About リソース
  - 7.8. クロスオリジンリクエスト
  - 7.9. パリテーション
  - 7.10. HTTP HEAD
- Appendix A: ブックマークレット
- Appendix B: IE 用リクエスト作成
- Appendix C: アクティビティタイプ "cmi.interaction" のための体儀例
- Appendix D: ステートメント例
- Appendix E: 1.0.0 ヘステートメントを変換
- Appendix F: 署名ステートメント例

# 1.0 改定履歴

---

## 0.8 (Project Tin Can API の配布) から 0.9 (2012/03/31)

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 から 0.95 (2012/08/31)

"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。

- 動詞、アクティビティタイプ、拡張キーを URI とする。
- 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。
- 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を用いるように変更。
- 友達の友達 (FOAF) エージェントの結合要求を削除。
- 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 から 1.0.0 (2013/4/26)

以下のような多数の改善と明確化を行った。

- 添付文書概念を追加。
- アクティビティのメタデータを XML ではなく JSON に変更。
- ステートメントの無効化に関する変更。
- ドキュメント API の追加。
- ステートメント API の検索に関する変更。
- 署名付きステートメントの追加。

# 2.0 Experience API の役割

---

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。

xAPI は以下を提供する。

- 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
- 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 やレポートシステムと通信をすることがある
- 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

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 xAPI における ADL の位置づけ

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々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考えている。

## 2.2 貢献者

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。

## 2.2.1 ワーキンググループの参加者

名前	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 Foucaux	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 要求仕様収集への貢献者

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 (LESTI) 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 技術に詳しくない人への読み方ガイドライン

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。

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 用語の定義

---

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

### アクティビティ:

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。

### アクティビティプロバイダ:

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。

### アクタ:

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。

### 認証 (Authentication):

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。

### 認可 (Authorization):

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。

### クライアント: -

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。

#### 実践コミュニティ:

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):

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。

#### イミュータブル:

変わる事の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。

#### IRI (Internationalized Resource Identifiers):

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。

#### 逆関数識別子 (Inverse Functional Identifier):

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 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):

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。

#### ～しなければならない (MUST)/～すべきである (SHOULD)/～してもよい (MAY):

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。

#### プロフィール:

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。

#### 登録事項 (Registration):

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。

#### REST (REpresentational State Transfer):

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。

#### サービス:

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。

#### ステートメント:

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):

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。

#### 動詞 (Verb):

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 ステートメント

---

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 ステートメントプロパティ

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。

プロパティ	タイプ	説明
id	UUID	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	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” 中の “I” に対応。
verb	Object	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	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	動詞に関する測定結果を示す結果オブジェクト。
context	Object	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	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	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	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	セマンティックバージョンing 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	ステートメントに対する添付文書のヘッダー。

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)。

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"
  }
}
```

Appendix D: ステートメントの例 参照

### 4.1.1 id (識別子)

#### 説明

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)

## 必要条件

- 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。
- ID はアクティブタイププロバイダによって生成されるべきである。

### 4.1.2 Actor (アクタ)

#### 説明

必須の Agent または Group オブジェクト。

#### 4.1.2.1 アクタのオブジェクト型が Agent の時

#### 説明

Agent(個人)は、人またはシステムである。

#### 詳細

- 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。
- Agent に、1つを超える逆関数識別子を設定してはならない。
- 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。

Agent オブジェクトのプロパティを以下の表に示す。

プロパティ	タイプ	説明	必須
objectType	string	"Agent"。Agent が目的語の場合は必須である。	no
name	String	Agent のフルネーム。	no
4.1.2.3 逆関数識別子参照		Agent に固有の逆関数識別子。	yes

#### 4.1.2.2 アクタのオブジェクト型が Group の時

#### 説明

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。

#### 必要条件

- 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。
- 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。

#### 匿名グループに関する必要条件

- 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。
- 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。
- 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。

#### 指名グループに関する必要条件

- 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。
- 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。
- 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。
- 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。

#### 詳細

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。

匿名グループのすべてのプロパティを、下の表に示す。

プロパティ	タイプ	説明	必須
-------	-----	----	----

objectType	String	"Group".	yes
name	String	グループの名称。	no
member	Array of Agent Objects	このグループのメンバー。	yes

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。

指名グループのすべてのプロパティを、下の表に示す。

プロパティ	タイプ	説明	必須
objectType	String	"Group".	yes
name	String	グループの名称。	no
member	Array of Agent Objects	このグループのメンバー。	no
4.1.2.3 逆関数識別子参照		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

##### 説明

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。

##### 背景

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。

##### 詳細

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。

プロパティ	タイプ	説明
mbox	mailto IRI	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	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	Agent を一意に特定する OpenID。
account	Object	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 オブジェクト

##### 説明

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。

##### 詳細

- 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。
- 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。

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。

プロパティ	タイプ	説明
homePage	IRL	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	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。
------	--------	---

#### 例

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 (動詞)

#### 説明

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。

#### 背景

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。

#### 必要条件

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。

- 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。
- 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。
- 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。
- 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。
- 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。
- 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。
- 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。

#### 詳細

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。

プロパティ	タイプ	説明
id	IRI	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	言語マップ	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。

#### 例

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

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 動詞の言語と意味に関する用法

##### 詳細

##### 意味

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。

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、それらの中の特定の1つの意味を示さなければならない。

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。

##### 言語

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。

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/بندقية 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 実践コミュニティに関する用法

##### 実践コミュニティ向けの必要条件

- 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。
- 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。

##### アクティビティプロバイダ向けの必要条件

- 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。
- 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。

##### 詳細

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。

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)

##### 説明

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。

例:

- 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"
- 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"
- 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。

##### 詳細

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。

##### ObjectType がアクティビティの場合

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。

プロパティ	タイプ	説明
objectType	String	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	一意のアクティビティの識別子。必須。
definition	Object	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。

## アクティビティ定義

### 詳細

#### アクティビティ定義オブジェクト

プロパティ	タイプ	利用	説明
name	言語マップ	推奨	可読／可視化したアクティビティの名称
description	言語マップ	推奨	アクティビティの説明
type	IRI	推奨	アクティビティのタイプ
moreInfo	IRL	任意	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については、Interaction Activities を参照			
extensions	Object	任意	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）

### 注記:

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。

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。

## アクティビティ ID

### アクティビティ ID の必要条件

- アクティビティ ID は一意でなければならない。
- 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。
- 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。
- 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。
- 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。

### LRS の必要条件

- 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。
- 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。
- 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。
- 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。

### アクティビティプロバイダの必要条件

- 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。
- 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

ステートメントを生成しなければならない。

- 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。

## 詳細

2つの異なるアクティビティに同じ ID を使用すること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。

## アクティビティメタデータ

### 必要条件

- 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。
- 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。
- 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。
- 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。

## インタラクション アクティビティ

### 背景

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。

### 必要条件

- 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
- 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。
- 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。

## 詳細

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。

プロパティ	タイプ	説明
interactionType	String	SCORM2004 4th 版のRun-Time Environmentで定義している"cmi.interactions.n.type"
correctResponsesPattern	An array of strings	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 n は配列のインデックスとなる
choices   scale   source   target   steps	Array of interaction components	interactionType (後述参照)にて特定

## インタラクションのコンポーネント

### 必要条件

- 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。
- 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

## 詳細

インタラクションコンポーネントの定義は以下の通り

プロパティ	タイプ	説明
id	String	SCORM2004 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	言語マップ	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)

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。

インタラクションタイプ	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]

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 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

#### 必要条件

- 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

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 目的語がステートメントだった場合

#### 背景

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。

#### ステートメント参照

##### 説明

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。

#### 必要条件

- 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。
- 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。

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

プロパティ	タイプ	説明
objectType	String	この場合、"StatementRef" としなければならない
id	UUID	ステートメントの UUID

#### 例

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、どの様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!"
  }
}
```

## サブステートメント

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

### 必要条件

- 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
- 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
- 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
- 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。

### 例

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(結果)

##### 説明

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

##### 詳細

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

プロパティ	タイプ	説明
score	Object	エージェントのスコア。Score を参照
success	Boolean	アクティビティの試みが成功したかどうかを表す
completion	Boolean	アクティビティが完了したかどうかを表す
response	String	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	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	ステートメントが発生してからの経過時間
extensions	Object	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 スコア (Score)

##### 説明

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

##### 必要条件

- 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。
- 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。

##### 詳細

以下の表は、スコアオブジェクトの定義である。

プロパティ	タイプ	説明
scaled	-1から1までの十進数	SCORM2004 4th 版の 'cmi.score.scaled' を参照
raw	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)	'cmi.score.raw' を参照
min	maxよりも小さい十進数	'cmi.score.min' を参照
max	minよりも大きい十進数	'cmi.score.max' を参照

#### 4.1.6 Context (文脈)

##### 説明

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。

##### 背景

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。

#### 必要条件

- 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。

#### 詳細

プロパティ	タイプ	説明
registration	UUID	ステートメントが関連している登録
instructor	Agent (may be a group)	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	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	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	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	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 )	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)	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	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)

#### 注記

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 プロパティ

##### 説明

特定の学習活動を行っている学習者のインスタンス。

##### 詳細

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 プロパティ

##### 説明

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。

##### 背景

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。

## 必要条件

- 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。

## 詳細

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: 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。

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。

## 注記

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、関連の可能性の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。

## 例

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製される。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 (タイムスタンプ)

#### 説明

ステートメントが生成された時刻。

#### 必要条件

- 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。

#### 詳細

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際の遅延がある。

### 4.1.8 Stored

#### 説明

ステートメントが LRS に記録された時刻。

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。

#### 必要条件

- 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 Authority (権限)

#### 説明

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。

#### 必要条件

- 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。

#### 詳細

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。

#### 権限としての OAuth 資格情報

#### 説明

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。

#### 必要条件

- 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 を示すエージェントは、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。

#### 詳細

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。

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。

#### 例

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 Version (バージョン)

##### 説明

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。

##### 必要条件

- 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 の必要条件

- 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。

##### クライアントの必要条件

- 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 (添付資料)

##### 説明

学習経験の証跡を提供するデジタル文書

##### 背景

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。

##### 添付文書ステートメントバッチのための必要条件

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:

- 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
- 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:
  - 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。
  - 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。
  - 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。
  - 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。
  - 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。
  - 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。
  - 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。

##### LRSの必要条件

- 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)
- 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。

- 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。
- "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、
  - 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。
- 上記以外の時に：
  - 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。
  - 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。
  - 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。
- 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。

**注記:** 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。

#### クライアントの必要条件

- 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。
- 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。
- 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。

#### 詳細

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。

プロパティ	タイプ	説明	必須
usageType	IRI	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	この添付文書の名前(タイトル)を表示する。	yes
description	Language Map	添付文書の説明	no
contentType	Internet Media Type	添付文書のコンテンツタイプ	yes
length	integer	オクテットで示した添付文書データの長さ	yes
sha2	base64	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	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

#### 添付文書交換の手続き

1. 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. 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. 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. 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。

#### 例

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:

- 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。
- 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。

- 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。
- 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。

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。

ヘッダー:

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

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 データの制約

##### 説明

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。

##### クライアントの必要条件

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。

- 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。

- 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。

#### LRS の必要条件

- 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。
  - 値がない場合 (拡張の中を除く)
  - 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)
  - 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)
  - 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。
  - 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。
  - 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。
- 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。
- 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認証しなければならない。
- 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。
- 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。注記: JSONとは異なり、文字列パラメータの値は引用符でくくられない。
- 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。
- 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 ステートメントの検索

### 説明

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。

### 詳細

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。

プロパティ	タイプ	説明
statements	Array of Statements	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、コンテナに含まれる "statements" プロパティに配置されるだろう。コンテナは、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	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。  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 無効

### 背景

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。

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。

### 必要条件

- 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
- 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。
- 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。

- 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。
- 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。
- 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。
- 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。

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 を参照して欲しい。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。

#### 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"
  }
}
```

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 署名付きステートメント

### 説明

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。

### 背景

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。

### 必要条件

- 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>
- 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。
- JWS 署名は "R256"、"RS384"、または "RS512" のアルゴリズムを使用しなければならない。
- 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。
- 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。
- 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。
  - 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。
  - 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、ということを確認する。
    - 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。
  - 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。
- 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。

### 詳細

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。

例はAppendix F: 署名ステートメント例を参照。

#### 注記:

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 各種タイプ

### 5.1 ドキュメント

#### 説明

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。

#### 詳細

プロパティ	タイプ	説明
ID	String	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	ドキュメントが最後に修正された時刻
contents	Arbitrary binary data	ドキュメントの内容

#### 注記

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 言語マップ

#### 説明

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 拡張

#### 説明

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。

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。

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助けとなる追加情報を提供すべきである。

## 注記

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 識別子メタデータ

### 説明

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：

- 動詞
- アクティビティ ID
- アクティビティタイプ
- 拡張キー
- 添付文書使用タイプ

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。

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：

プロパティ	タイプ	説明
name	Language Map	人が読める／見られる 名前
description	Language Map	説明

### 必要条件

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。

- 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。
- 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。
- 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。

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：

- 対応する既存の識別子を使用すべきであり；
- 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 ランタイム通信

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 エンコーディング

#### 必要事項

- 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 のバージョン管理

#### 背景

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。

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。

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。

#### 必要条件

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。

例: X-Experience-API-Version : 1.0.0

#### LRS に対する必要条件:

- 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。
- このヘッダには "1.0.0" を設定しなければならない。
- 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。
- "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。
- "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。
- 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。

#### クライアントに対する必要条件:

- "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。
- 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。
- 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。

#### 他のバージョンへのステートメントの変換:

- 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。
- 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 同時実行

### 説明

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。

### 詳細

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

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。以下の必要条件は、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。

#### クライアントに対する必要条件

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

- 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 に対する必要条件

LRS が GET リクエストに回答する際には

- ETag HTTP ヘッダをレスポンスに追加しなければならない。
- 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。
- ヘッダを引用符で囲わなければならない。

LRS ETag 形式を設定する理由は、ETag ヘッダを読込めない 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。

LRS が PUT リクエストに回答する際には

- 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。
- 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。

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

- HTTP status 412 "Precondition Failed" を返さなければならない。
- リソースを変更させてはならない。

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

- HTTP status 409 "Conflict" を返さなければならない。
- 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。
  - リソースの現在の状態をチェックすべきである。
  - 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。
- リソースを変更してはならない。

## 6.4 セキュリティ

### 背景

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。

### 必要条件

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 ( RFC 5849 ) ( "HMAC-SHA1"、"RSA-SHA1" および "PLAIN TEXT" の署名方式を利用する)
- HTTP 基本認証
- 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)
- 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。

### 認証シナリオ

以下の表において、想定される認証のシナリオを示す。

**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。

	認識された利用者	認識されていない利用者
アプリケーションが登録される場合	OAuth に対応する標準ワークフローが適用される。	LRS は付加的な利用者証明書を持たない xAPI 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。
アプリケーションが登録されていない場合	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ので、LRS はその識別を前提とすることはできない。	
アプリケーションがない場合	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。	
認証なしの場合	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 各シナリオの取り扱い方法

#### 概要

- 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。
- 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。

#### 登録されたアプリケーション + 認識された利用者

- 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。
- 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。

#### 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 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。
- 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。

#### 登録されていないアプリケーション+ 認識された利用者

- 空白の consumer secret を利用すること。
- "Temporary Credential" リクエストを呼び出すこと。
- "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。
- 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。

#### アプリケーションがない+ 認識された利用者

- 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。
- 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。
  - 他の権限情報が指定される。かつ、
  - 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。

#### 認証なし

- 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。

#### 必要条件

LRS を

- 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。
  - 上記のいずれかの手順によること。
  - 上記のいずれかのワークフローシナリオによること。
- (セキュリティ上の理由で)以下のとおり設定してもよい。
  - 上記の手順のサブセットをサポートすること。
  - 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。
- "HMACS-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 認証スコープ

#### 説明

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。

#### 必要条件

LRS は

- 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。
- 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。
- 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。
- その他のスコープをサポートしてもよい。

xAPI クライアントは

- 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。

#### 詳細

以下の表で xAPI スコープ値を一覧にする。

スコープ	許可
statements/write	任意のステートメントを書き込む
statements/read/mine	"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 が割り当てて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	任意のステートメントを読み込む
state	この関係を決定できる範囲で現在のトークンに関係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／書き込む。
define	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	この関係を決定できる範囲で現在のトークンに関係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／書き込む。
all/read	制限なしの読み込みアクセス
all	制限なしのアクセス

### OAuth 拡張パラメータ

"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 エンドポイント

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

### 例

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)

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。

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 の必要条件

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)。
- 本仕様に記載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 エラーコード

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` - 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` - 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` - 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` - 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` - 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` - 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` - 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` - 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 ステートメント API

### xAPI の基本的な通信メカニズム

#### PUT ステートメント

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

与えられた id でステートメントを格納する。

戻り値: `204 No Content`

パラメータ	タイプ	デフォルト値	説明
statementId	String		記録するためのステートメント id

#### POST ステートメント

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

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。

戻り値: `200 OK`, ステートメント id(s) (UUID).

#### PUT と POST に対する一般的な必要条件

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`で応答するかにかかわらず、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。

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同致するか検証し、合致しない場合には `409 Conflict` を返すべきである。

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

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

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 voidedStatement id 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。

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。

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)

		デフォ	
--	--	-----	--

パラメータ	タイプ	ルト 値	説明
statementId	String		取得するためのステートメントの ID
voidedStatementId	String		取得するための無効なステートメントの ID。詳細は <a href="#">VoidedStatement</a> を参照。
agent	Agent or Identified Group Object (JSON)		<p>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"> <li>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。</li> <li>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。</li> </ul> <p>詳細は<a href="#">エージェント/グループオブジェクトの定義</a>参照。</p>
verb	Verb id (IRI)		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)		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		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	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	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。
since	Timestamp		指定された 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		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	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>「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>「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>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	trueの場合、LRS は4.1.11. <a href="#">Attachments</a> 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	true の場合、格納された時間の昇順で結果を返す。
-----------	---------	-------	----------------------------

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。

\* ステートメント ID と voidedStatementId 両方のパラメータを含む場合。

- 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。

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。

#### 注記

クエリ文字列の制限により、このメソッドは POST と form filed 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。

#### StatementRefs のためのフィルタ条件

フィルタパラメータが時間やシーケンススペースでない場合（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。時間やシーケンススペ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。

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。

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。

#### 注釈

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。

#### 無効なステートメント

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。レポーティング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。レポーティングツールは 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 ドキュメント API

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。

#### 新規エージェントとアクティビティ

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。

#### application/json 配列を変数に格納するための POST メソッド

API	メソッド	エンドポイント	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

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"
```

```
}
```

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。

- 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
- 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。
- 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。

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。たとえば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。

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"  
}
```

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"  
}
```

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ない、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。

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。

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。

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。

## ステートAPI

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 アクティビティ/ステート

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

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

パラメータ	タイプ	必須	説明
activityId	String	yes	このステートと関連したアクティビティID
agent	JSON	yes	このステートと関連したエージェント
registration	UUID	no	このステートと関連した登録
stateId	String	yes	与えられた文脈における、このステートのID

GET アクティビティ/ステート

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

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。

戻り値: 200 OK, Array of ids

パラメータ	タイプ	必須	説明
activityId	String	yes	これらのステートと関連したアクティビティ ID
agent	JSON	yes	これらのステートと関連したエージェント
registration	UUID	no	これらのステートと関連した登録
since	Timestamp	no	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 アクティビティステート

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

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。

戻り値: 204 No Content

パラメータ	タイプ	必須	説明
activityId	String	yes	このステートと関連したアクティビティ ID
agent	JSON	yes	このステートと関連したエージェント
registration	UUID	no	このステートと関連した登録 ID

## アクティビティ プロファイル API

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。

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 アクティビティ

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

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。

戻り値: 200 OK, Content

パラメータ	タイプ	必須	説明
activityId	String	yes	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 アクティビティプロフィール

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

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。

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

パラメータ	タイプ	必須	説明
activityId	String	yes	このプロフィールと関連したアクティビティ ID
profileId	String	yes	このプロフィールと関連したプロフィール ID

#### GET アクティビティプロフィール

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

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のみ制限される。

戻り値: 200 OK, List of ids

パラメータ	タイプ	必須	説明
activityId	String	yes	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	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

## エージェント プロファイル API

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。

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 エージェント

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

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。

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增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 プロパティ

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。

プロパティ	タイプ	説明
objectType	String	"Person". 必須
name	Array of strings.	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".	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.	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.	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.	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。

参照: Section 4.1.2.1 Agent.

戻り値 200 OK, Expanded Agent Object

パラメータ	タイプ	必須	説明
agent	Object (JSON)	yes	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

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

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。

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

パラメータ	タイプ	必須	説明
agent	Object (JSON)	yes	このプロフィールと関連したエージェント
profileid	String	yes	このプロフィールと関連したプロフィール ID

#### GET エージェントプロフィール

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

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だけに制限される。

戻り値: 200 OK , List of IDs

パラメータ	タイプ	必須	説明
agent	Object (JSON)	yes	このプロフィールと関連したエージェント
since	Timestamp	no	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。

## About リソース

#### GET about

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

#### 説明

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。

#### 背景

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。

#### 詳細

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

プロパティ	タイプ	説明
version	array of version strings	この LRS がサポートする xAPI バージョン
Extensions	Object	必要に応じた他のプロパティのマップ

#### LRS 要件

- 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 文書を返さなければならない。
  - 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。)

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。

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。

## クロス オリジン リクエスト

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。

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。

**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。

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

**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。

**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。

**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。

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。

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。

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。

## パリデーション

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デーションは、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。

## HTTP HEAD

### 説明

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。

### 背景

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 必要条件

- 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: 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: IE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 querystring, 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: 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": [
    "1",
    "2",
    "3",
    "4",
    "5"
  ]
}
```

```

"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: 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: 1.0.0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: 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": "33cff416-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-----  
MIICXAIIBAAKBgQDjxvZXF30WL4oKjZYXgR0ZyaX+u3y6+JqTqiNkFa/VTnet6Ly2  
OT6ZmmcJEPnq3UnewpHoOQ+Gfhhtkw13j06j5iNn4obcCVWTL9yXNvJH+Ko+xu4Y  
l/ySPRrIPYtjtHdG0M2XzI1mmLqm+CAS+KCbJeH4tf543kIWC5pC5p3cVQIDAQAB  
AoGA0e7jdvGq2XKuddu1kWX10Aphn4YmdPpPyCNTaxp1U6PBVMRjY0aNgLQE6b02p  
/HjiU4Y4PkgzkEgCu0xf/mOq5DnSkX32ICoQS6jChABAE20ErPfm5t8h9YKsTfn9  
401AouuwD9ePRteizd4YvHtiMMwmh5PtUoCbqLefawNAPAEQCQD1mdBW3zL0okUx  
2pc4ttn2qArCG4CsEZML1GRDd3FwPWJz3ZPNEEGZWXGSpA9F1QTZ6JYXIfejjRo  
UuvRMWeBAKEA7WvzDBNcv4N+xeUKvH8ILti/BM58LraTtqJlZjQSOvek0srxtmDg  
5of+xrXN6IM4p7yvQa+7YOUUokrVXjG+1QJBAI2mBrjzXgm9xTa5odn97JD7UMFA  
/WHjlMe/Nx/35V52qaav1sZbluw+TvKMcqApYj5G2SupSNudHLDGkmd2nQECQFfc  
1BRK8g7ZncekbGW3aRLVGVOxC1nLLTzwO1amBKOUm8V6XxsMHQ6TE2D+fKJONUy1  
2HGpk+FwWY2D1hRGuoUCQAXfaLSxtawdPtLZTPVueF7ZikQDsVg+vtTFgpuH1oR2  
6EVc1RbHHZm32yvGDY8IkcoMfJQqLONDdLfS/05yoNU=  
-----END RSA PRIVATE KEY-----  
...
```

## Example public X.509 certificate 公開X.509証明書例

```
...
-----BEGIN CERTIFICATE-----
MIIDATCCAmqgAwIBAgIJAMB1csNuA6+kMA0GCSqGSIb3DQEBBQUAMHExCzAJBgNV
BAYTA1VTMRIwEAYDVQQIEw1UZW5uZXNzZWUxGDAWBgNVBAoTD0V4YW1wbGUgQ29t
cGFueTEQMA4GA1UEAxMHRXhhbXBsZTEiMCAGCSqGSIb3DQEJARYTZXhhbXBsZUB1
eGFtcGx1LmNvbTAeFw0xMzA0MDQxNTI4MzBaFw0xNDA0MDQxNTI4MzBaMIGWQsw
CQYDVQQGEwJVUzESBAGA1UECBMjVGVubmVzc2V1MREwDwYDVQHEwGcmFua2xp
bjEYMBYGA1UEChMjPRXhhbXBsZSBD21wYw55MRAwDgYDVQQLLWwEwGFtcGx1MRAw
DgYDVQQDEwEwGFtcGx1MSIwIAYJKoZIhvcNAQkBFhNleGFtcGx1QGV4YW1wbGUu
Y29tMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDjxvZXF30WL4oKjZYXgR0Z
yaX+u3y6+JqTqiNkFa/VTnet6Ly2OT6ZmmcJEPnq3UnewpHoOQ+GfhhTkW13j06j
5iNn4obcCVWTL9yXNvJH+Ko+xu4Y1/ySPRrIPyTjtHdG0M2XzI1mmlqm+CAS+KCb
JeH4tf543kIWC5pC5p3cVQIDAQAB03sweTAJBgNVHRMEAjaAMCwGCWGSAGG+EIB
DQOQFfh1PcGVuU1NMIEd1bmVvYXRlZCBDZj0aWZpY2F0ZTAdBgNVHQ4EFgQUUVs3v
5afEd0eoYeVaJAQE4v0S1QwHwYDVR0jBBGwFoAUyV1c3yvra4EBz20I4BF39IAi
xBkwDQYJKoZIhvcNAQEFBQADgYEAgs/FF5D0Hnj44rvT6kgn3KJAvv21j/fyjtK
IrYS331jXGn6GyA4qtXA23Pr04uc/wYCSIDpPobh62xTCd9qObKhgW0i05P
SBLqUu3mwfAe15LJBJBqPVZ4K0kppePBu8m6aIZoH57L/9t40oaL8yKs/qjKFeI1
OFWZxvA=
-----END CERTIFICATE-----
...
```

## Example certificate authority certificate CA証明書例

```
...
-----BEGIN CERTIFICATE-----
MIIDNzCCAqCgAwIBAgIJAMB1csNuA6+jMA0GCSqGSIb3DQEBBQUAMHExCzAJBgNV
BAYTA1VTMRIwEAYDVQQIEw1UZW5uZXNzZWUxGDAWBgNVBAoTD0V4YW1wbGUgQ29t
cGFueTEQMA4GA1UEAxMHRXhhbXBsZTEiMCAGCSqGSIb3DQEJARYTZXhhbXBsZUB1
eGFtcGx1LmNvbTAeFw0xMzA0MDQxNTI4MzBaFw0yMzA0MDIxNTI4MzBaMHEXcCzAJ
BgNVBAYTA1VTMRIwEAYDVQQIEw1UZW5uZXNzZWUxGDAWBgNVBAoTD0V4YW1wbGUg
Q29tcGFueTEQMA4GA1UEAxMHRXhhbXBsZTEiMCAGCSqGSIb3DQEJARYTZXhhbXBs
ZUB1eGFtcGx1LmNvbTBcnzANBkqkqhkIG9w0BAQEFAAOBjQAwGyKCGYEA1sBnBWPZ
0f7WJUFtJy5+01S1S5Z6DD6Uye9vK9AycgV5B3+WC8HC5u5h91MuJAC1ARPVUOT
svPRs45qKNFIgIGRKKPAWzjawEI2sCJRSKV47i6B8bdv4WkuGvQaveZGI0q1mN5R
1Eim2gUItRj1hgcc9rQavj1nFKDY2r1XGukCAwEAa0B1jCB0zAdBgNVHQ4EFgQU
yV1c3yvra4EBz20I4BF39IAixBkwaMGA1UdIwSBmzCBmIAUyV1c3yvra4EBz20I
4BF39IAixBmhdarzMHEXcCzAJBgNVBAYTA1VTMRIwEAYDVQQIEw1UZW5uZXNzZWUx
GDAWBgNVBAoTD0V4YW1wbGUgQ29tcGFueTEQMA4GA1UEAxMHRXhhbXBsZTEiMCAG
CSqGSIb3DQEJARYTZXhhbXBsZUB1eGFtcGx1LmNvbYIJAMB1csNuA6+jMAwGA1Ud
EwQFMAMBAF8wDQYJKoZIhvcNAQEFBQADgYEAhWTeBtGk735yKhM8DqCxcvNnEZ0Nx
sYEY0jgRG1yXT1W5pE691fSH5AZ+T6fpwpZcWY5QYkoN6Dnwj0xGkSfQC3/yGmcU
DKBPwiZ502s9C+fe1kUEnrX2Xea4agVngMzR8DQ6oOauLWqehDB+g2ENWRLoVgS+
ma5/Ycs0GTyrECY=
-----END CERTIFICATE-----
...
```

JWS Header. Note that along with specifying the algorithm, the certificate chain for the signing certificate has been included.

JWSヘッダー。アルゴリズムを記すの一緒に、署名証明書のために証明書チェーンが含まれることに注意下さい。

```
...
{
  "alg": "RS256",
  "x5c": [
    "MIIDATCCAmqgAwIBAgIJAMB1csNuA6
    +kMA0GCSqGSIb3DQEBBQUAMHExCzAJBgNVBAYTA1VTMRIwEAYDVQQIEw1UZW5uZXNzZWUxGDAWBgNVBAoTD0V4YW1wbGUgQ29tcGFueTEQMA4GA1UEAxMHRXhhbXBs
    "MIIDNzCCAqCgAwIBAgIJAMB1csNuA6
    +jMA0GCSqGSIb3DQEBBQUAMHExCzAJBgNVBAYTA1VTMRIwEAYDVQQIEw1UZW5uZXNzZWUxGDAWBgNVBAoTD0V4YW1wbGUgQ29tcGFueTEQMA4GA1UEAxMHRXhhbXBs
    yGmcUDKBPwiZ502s9C+fe1kUEnrX2Xea4agVngMzR8DQ6oOauLWqehDB
```

```
+g2ENWRLoVgS+ma5/Ycs0GTyrECY="
  ]
}
```

#### JWS signature JWS 署名

```
ew0KICAgICJhbGciOiAiUmluYXN0KICAgICJ4NWMiOiBbdQogICAgICAgICJNSU1EQVRDQ0FtcWdBd0lCQWdJskFNQjFjc051QTYra01BMEduDU3FHU0liM0RRRU
...
```

#### Signed Statement 署名されたステートメント

```
...
{
  "version": "1.0.0",
  "id": "33cff416-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",
  "attachments": [
    {
      "usageType": "http://adlnet.gov/expapi/attachments/signature",
      "display": { "en-US": "Signature" },
      "description": { "en-US": "A test signature" },
      "contentType": "application/octet-stream",
      "length": 4235,
      "sha2": "dc9589e454ff375dd5dfd6f556d2583e231e8cafe55ef40102ddd988b79f86f0"
    }
  ]
}
...
```

#### 注記

添付した署名が表示されない場合、添付資料で添付型メッセージのフォーマットを確認すること。