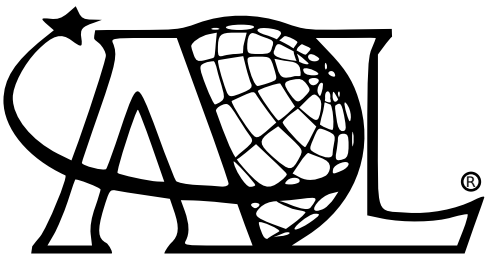

EXPERIENCE API™

Спецификация Experience API

Версия 1.0.1

Методология построения взаимодействия между клиентскими и серверными приложениями для передачи достигнутых результатов обучения в распределённых информационно-образовательных средах.

Санкт-Петербург

2015

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

УНИВЕРСИТЕТ ИТМО

Experience API
Версия 1.0.1

Перевод
А.Д. Копилов
А.В. Лямин

Спецификация

Санкт-Петербург

2015

Спецификация Experience API. Версия 1.0.1 / Перевод с английского А.Д. Копилов, А.В. Лямин – СПб: Университет ИТМО, 2015. – 120 с.

Изложены правила организации взаимодействия между клиентскими и серверными приложениями для передачи информации об учебных достижениях в распределённых информационно-образовательных средах.

Предназначено для бакалавров и магистрантов направления «Информационные системы и технологии»

Рекомендовано к печати Учебно-методическим советом университета.

Университет ИТМО – ведущий вуз России в области информационных и фотонных технологий, один из немногих российских вузов, получивших в 2009 году статус национального исследовательского университета. С 2013 года Университет ИТМО – участник программы повышения конкурентоспособности российских университетов среди ведущих мировых научно-образовательных центров, известной как проект «5 в 100». Цель Университета ИТМО – становление исследовательского университета мирового уровня, предпринимательского по типу, ориентированного на интернационализацию всех направлений деятельности.

Данное издание содержит перевод, документа, расположенного в свободном доступе по URL https://github.com/adlnet/xAPI-Spec/blob/1.0.1/xAPI.md

Оригинальный документ написан участниками команды разработчиков Experience API (см. список в разделе 2.2.1) при поддержке Заместителя помощника секретаря обороны ADL. Пожалуйста, отправляйте отзывы и вопросы по Experience API на адрес helpdesk@adlnet.gov

Предложения и замечания по переводу отправляйте на адрес de@mail.ifmo.ru

Experience API specification 1.0.1

Advanced Distributed Learning
(ADL) Initiative

Спецификация Experience API
Версия 1.0.1

Университет ИТМО

Авторы перевода:

Копилов А. Д.

Лямин А. В.

УНИВЕРСИТЕТ ИТМО

Содержание

1.0 История изменений

0.8 (Первый выход проекта Tin Can API) - 0.9 (31 марта 2012)

Компания Rustici Software, разработавшая проект Tin Can API, вносила изменения в API до ознакомительной встречи в апреле 2012 года. На этой встрече было решено использовать изменения в текущей спецификации, версии 0.9.

0.90 - 0.95 (31 августа 2012)

«Ядровые» действия (Verb) и типы задач (Activity) удалены из спецификации. Ссылки на эти команды так же удалены. При реализации API рекомендуется использовать команды, определённые сообществом, а не создавать свои собственные.

- Действия, типы задач и ключи расширений теперь в виде URI.
- Переписано и уточнено о некоторых других деталях и объёме реализации.
- Переход от человек-ориентированного представления субъекта (Agent) к образ-ориентированному.
- Требование объединения Друга Друга (Friend of a Friend, FOAF) субъекта удалено.
- Субъект должен иметь ровно одно уникальное идентифицирующее свойство, а не хотя бы одно.

0.95 - 1.0.0 (26 апреля 2013)

Различные усовершенствования и оптимизация, в том числе:

- Вложения
- Данные о задачах сохраняются в JSON, а не в XML
- Изменения в утверждениях аннулирования (voiding Statements)
- Оптимизация и именование API документа (Document)
- Изменения в вызывании API утверждений
- Подписанные утверждения

1.0.0 - 1.0.1 (1 октября 2013)

Оптимизация и примеры, в том числе:

- Исправление опечаток
- Дополнительные примеры в приложениях

2.0 Роль Experience API

Experience API – сервис, который позволяет передавать и размещать в хранилище учебных достижений (Learning Record Store, LRS) результаты обучения или какой-либо другой деятельности. Информация, которая размещается в хранилище, создаётся и отслеживается провайдерами задач (Activity Providers), а спецификация Experience API описывает модель данных и сопутствующие решению основных задач компоненты. В частности, Experience API обеспечивает:

- Структуру и определение классов Statement (утверждение), State (состояние), Learner (учащийся), Activity (задача) и Objects (объекты), с помощью которых провайдер задач представляет информацию о приобретённом опыте.
- Методы передачи данных для сохранения и поиска (но не валидации) экземпляров этих классов в хранилище учебных достижений. Необходимо отметить, что не только провайдеры задач могут сохранять данные и осуществлять поиск записей. Каждое хранилище может взаимодействовать с другими хранилищами учебных достижений или системами для формирования отчётов.
- Способы защиты, обеспечивающие безопасный обмен информацией между хранилищем и доверенными источниками.

Experience API – одна из многих технологий, предусмотренных новой архитектурой онлайн-обучения. Сервисы аутентификации, сервисы запросов, сервисы визуализации и сервисы персональных данных – только некоторые из дополнительных технологий, поддержка которых предусмотрена Experience API. Хотя детали реализации этих сервисов здесь не рассматриваются, спецификация Experience API разработана с расчётом на расширенные возможности новой архитектуры.

2.1 Роль ADL в разработке Experience API

Некоммерческая организация ADL (Advanced Distributed Learning – продвинутое (перспективное) распределённое обучение, передовые технологии распределённого обучения) сыграла роль координатора и посредника в разработке спецификации. Experience API рассматривается как часть архитектуры обучения ADL, которая ориентирована на возможность обучения когда угодно и где угодно. ADL представляет Experience API как улучшенную версию SCORM (Sharable Content Object Reference Model – эталонная модель совместного доступа к контенту), поддерживающую аналогичные варианты использования, а также другие, отобранные ADL, относящиеся к электронному обучению, но не включённые в SCORM.

2.2 Авторы

> «Я благодарен всем, кто принимал участие в этом проекте. Многие из вас участвовали в еженедельных совещаниях и помогали сделать спецификацию полезной для всего сообщества дистанционного обучения. Многие из вас помогали в создании примеров кода, продукции и документации, чтобы поддержать тех, кто разрабатывает и подтверждает спецификацию. Я так же хочу поблагодарить всех тех, кто поделился полезной, честной информацией об использовании в их компаниях SCORM и других технологий обучения. По предложенным требованиям, предоставленному опыту и Вашим знаниям ADL и сообщество чётко определило первый шаг в создании Experience API – Архитектуры тренинга и обучения. Вы настоящие лидеры сообщества, на вас мы опираемся, делая наше обучение наилучшим.»

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

2.2.1 Команда участников

Имя	Организация
Aaron Silvers	ADL
Al Bejcek	NetDimensions
Ali Shahrazad	SaLTBOX
Andrew Downes	Epic
Andy Johnson	ADL
Andy Whitaker	Rustici Software
Anthony Altieri	American Red Cross
Anto Valan	Omnivera Learning Solutions
Avron Barr	Aldo Ventures, Inc.
Ben Clark	Rustici Software
Bill McDonald	Boeing
Brian J. Miller	Rustici Software
Chad Udell	Float Mobile Learning
Chris Sawwa	Meridian Knowledge Solutions
Dan Allen	Litmos

Dan Kuemmel	Sentry Insurance
Dave Mozealous	Articulate
David Ells	Rustici Software
David N. Johnson	Clear Learning Systems
Doug Hagy	Twin Lakes Consulting Corporation
Eric Johnson	Planning and Learning Technologies, Inc.
Fiona Leteney	Feenix e-learning
Greg Tatka	Menco Social Learning
Ingo Dahn	University Koblenz-Landau
Jason Haag	ADL
Jeff Place	Questionmark
Jennifer Cameron	Sencia Corporate Web Solutions
Jeremy Brockman	
Jhorlin De Armas	Riptide Software
Joe Gorup	CourseAvenue
John Kleeman	Questionmark
Jonathan Archibald	Brightwave
Jonathan Poltrack	ADL
Kris Miller	edcetra Training
Kris Rockwell	Hybrid Learning Systems
Lang Holloman	
Lou Wolford	ADL
Luke Hickey	DominKnow
Marcus Birtwhistle	ADL
Mark Davis	Exambuilder
Matteo Scaramuccia	
Megan Bowe	Rustici Software
Melanie VanHorn	ADL
Michael Flores	Here Everything's Better
Michael Roberts	VtrainingRoom
Mike Palmer	OnPoint Digital

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

Авторы перевода

Копилов Александр Дмитриевич	Университет ИТМО
Лямин Андрей Владмирович	Университет ИТМО

2.2.2 Участники сбора требований

В коллекции требований к Experience API многие люди и компании предоставляли бесценные отзывы о SCORM, попытках проведения дистанционного обучения и образования в целом. Хотя список и не полный, документы, собранные в 2008 году группой LETSI (Learning Education and Training Standards Interoperability –

Совместимость стандартов обучения), на сайте Rustici Software UserVoice, в личных интервью и различных блогах были важными источниками требований к спецификации Experience API.

2.3 Указания к прочтению для нетехнического уклона

Это точный документ, описывающий, как спецификация Experience API должна быть реализована в различных системах. Это технический документ, написанный специально для лиц и организаций, реализующих данную технологию, чтобы они могли разрабатывать совместимые инструменты, системы и сервисы, независимые друг от друга и совместимые между собой.

Насколько возможно, язык и форматирование в этом документе целенаправленно сделаны *дружественными* к нетехническим читателям, поскольку многие инструменты, системы и сервисы основаны на описанной ниже спецификации. Поэтому, разделы, предоставляющие *общий обзор* аспектов помечены, как **описание** или **пояснение**. Элементы, помеченные как **требования**, **детали** или **примеры** более технические.

3.0 Определения

Activity (задача): Класс, который определяет предмет деятельности. Это то, с чем взаимодействует экземпляр класса Actor (действующее лицо) в процессе обучения, приобретения опыта или выполнения какой-либо работы, т.е. всё то, что образует значимую комбинацию с некоторым экземпляром класса Verb (глагол). Класс Activity (задача) имеет много интерпретаций, он может соответствовать даже вещам, таким как стул (реальный или виртуальный). В утверждении «Анна опробовала рецепт торта» классом Activity (задача) служит рецепт торта. Книга, электронный курс, экскурсия или встреча – всё это примеры класса Activity (задача).

Activity Provider (AP, провайдер задач): Программный объект, который записывает в LRS информацию об учебном процессе. Аналогичен SCORM-пакету, т.к. можно связать учебные ресурсы (assets) с программным объектом, который выполнит запись, однако провайдер задач может функционировать независимо от ресурса, о работе с которым создаётся запись.

Actor (действующее лицо): Лицо или группа лиц, действия которых с классом Activity (задача) отслеживаются с помощью экземпляров класса Statement (утверждение).

Authentication (аутентификация): Концепция проверки подлинности пользователя или системы. Аутентификация разрешает взаимодействие между пользователями или системами, находящимися в доверительных отношениях.

Authorization (авторизация): Определение прав на основе роли пользователя или системы; процесс установления доверительных отношений с пользователем или системой.

Client (клиент): Любая сущность, которая может взаимодействовать с LRS. Клиентом может быть провайдер задач, инструмент для построения отчётов, LMS, LRS.

Community of Practice (профессиональное сообщество): Группа, объединённая общим делом, ролью или целью, которая действует в общей модальности (в единой среде).

Experience API (xAPI): Программный интерфейс, описанный в данном документе, – результат проекта «Tin Can». Механизм, позволяющий любому действующему лицу вне зависимости от платформы сохранять и восстанавливать записи об учебном процессе, обучающемся и его достижениях.

Immutable (неизменяемый): Прилагательное, используемое для описания того, что невозможно изменить. За некоторыми исключениями класс Statement (утверждение) в xAPI является неизменяемым. Это гарантирует, что когда экземпляр класса Statement (утверждение) совместно используются несколькими LRS, все копии этого экземпляра совпадают.

Internationalized Resource Identifier (IRI, интернационализированный идентификатор ресурса): Уникальный идентификатор, в качестве которого может выступать IRL. В xAPI IRI должны быть абсолютными и включать название схемы. Относительные IRI не используются. IRL должен быть определён в домене, управляемом тем, кто создаёт IRL.

Internationalized Resource Locator (IRL, интернационализированный указатель ресурса): В контексте этого документа IRL – это IRI, который при переводе в URI становится URL. Некоторые профессиональные сообщества используют термин URL вместо IRI, что технически некорректно в xAPI.

Inverse Functional Identifier (обратный функциональный идентификатор): Идентификатор, который является уникальным для агента или группы. Используется для идентификации агентов или групп.

Learning Management System (LMS, система управления обучением): "Программный пакет, используемый для назначения одного или более курсов на одного или более обучающихся. Обычно LMS – это веб-приложение, которое позволяет обучающимся пройти аутентификацию, зарегистрироваться на курс, изучить курс, получить оценки" (определение лаборатории Learning Systems Architecture Lab – лаборатория архитектуры обучающих систем). В этом документе термин будет использоваться в контексте существующих систем, использующих образовательные стандарты.

Learning Record Store (LRS, хранилище учебных достижений): Система, в которой хранится учебная информация. До появления xAPI большинство LRS были LMS; однако этот документ использует термин LRS, чтобы показать, что для реализации xAPI не требуется полной LMS. Реализация xAPI зависит от LRS.

MUST / SHOULD / MAY (ДОЛЖЕН / СЛЕДУЕТ / МОЖЕТ): Три уровня, определяющие степень соответствия спецификации xAPI. Система, которая не выполняет требование, помеченное как ДОЛЖЕН (или НЕ ДОЛЖЕН), не соответствует спецификации. Отказ от требования СЛЕДУЕТ не приводит к нарушению спецификации, но противоречит лучшей практике использования. МОЖЕТ указывает на опциональное требование, решение об исполнении которого принимает разработчик системы без риска нарушить спецификацию xAPI.

Profile (профиль): Конструкция, где хранится информация об учащемся или задаче, как правило, в парах имя/документ, которые имеют значение для некоторого модуля системы.

Registration (регистрация): Запрос учащегося на работу с определённым экземпляром класса Activity (задача).

Representational State Transfer (REST, передача репрезентативного состояния): Архитектура для проектирования сетевых веб-сервисов. Она основана на протоколе HTTP и использует лучшие веб-практики на данный момент.

Service (сервис): Программный компонент, отвечающий за один или более аспектов распределённого процесса обучения. LMS традиционно сочетает в себе много сервисов для создания учебного процесса в целом.

Statement (утверждение): Простая конструкция, состоящая из цепочки актёр (учащийся)-глагол-объект с указанием результата и контекста для отслеживания некоторого аспекта учебного процесса. Множество из нескольких конструкций типа Statement (утверждение) может быть использовано для отслеживания всех деталей учебного процесса.

Tin Can API (TCAPI): Предыдущее название программного интерфейса, определённого в этом документе, часто используемое в неофициальных ссылках на Experience API.

Verb (глагол): Определяет действие, выполненное экземпляром класса Actor (актёр) с экземпляром класса Activity (задача) в конструкции типа Statement (утверждение).

4.0 Statement

Описание

Statement (утверждение) – ядро спецификации xAPI. Все учебные события сохраняются в форме утверждений. Любое утверждение в xAPI родственно предложению «Я сделал это».

4.1 Свойства утверждений

Детали

Каждое свойство утверждения описано в таблице, приведённой ниже.

Свойство	Тип	Описание	Примечание
id	UUID	UUID назначается LRS, если не установлен провайдером задач.	Рекомендуемое
actor	Object	Тот, о котором сообщается в утверждении, лицо или группа лиц. Определяет «Я» в предложении «Я сделал это».	Обязательное
verb	Object	Действие учащегося или группы. Определяет «сделал» в предложении «Я сделал это».	Обязательное
object	Object	Задача, агент или другое утверждение, которое является объектом данного утверждения. Определяет «это» в предложении «Я сделал это». Заметьте, что объекты, которые распространяются в качестве значения этого поля, должны иметь свойство objectType. Если оно не указано, то принимается, что объект описывает задачу.	Обязательное
result	Object	Объект, который содержит детали измеренного результата обучения, относящиеся к указанному действию.	Опциональное
context	Object	Контекст, который придаёт смысл утверждению. Например: команда, в которой работает актёр; высота, на которой было выполнено	Опциональное

		упражнение на пилотажном тренажёре.	
timestamp	Date/Time	Дата (в формате ISO 8601) события, которое описывается в утверждении. Если она не указана, то LRS будет использовать дату сохранения записи.	Опциональное
stored	Date/Time	Дата (в формате ISO 8601) сохранения записи. Устанавливается LRS.	Устанавливается LRS
authority	Object	Агент, который устанавливает истинность утверждения. Подтверждается LRS посредством процедуры аутентификации. Заполняется LRS, если информация отсутствует.	Опциональное
version	Version	Версия xAPI в формате Semantic Versioning 1.0.0.	Не рекомендуется
attachments	Массив вложенных объектов типа Object	Заголовки объектов, вложенных в утверждение.	Опциональное

Помимо значений свойств, устанавливаемых во время LRS-обработки (id, authority, stored, timestamp, version), утверждения остаются неизменными. Отметим, что содержание учебных задач, на которые ссылаются в утверждениях, не является частью самого утверждения. Поэтому, несмотря на то, что утверждения являются неизменными, содержание задач может меняться. Это означает, что глубокая сериализация объекта типа Statement в формате JSON изменится, если поменяется содержание объекта типа Activity (см. параметр format в описании API утверждения).

Требования

- Утверждение ДОЛЖНО использовать каждое свойство не более одного раза.
- Утверждение ДОЛЖНО иметь свойства actor, verb и object.
- Утверждение МОЖЕТ употреблять свойства в любом порядке.

Пример

Пример простейшего утверждения, которое имеет все обязательные и рекомендуемые свойства:

```
{
    "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"
    }
}
```

Другие примеры утверждений приведены в Приложении A.

4.1.1 ID

Описание

UUID (см. RFC 4122), который должен быть представлен в виде строки.

Требования

- LRS ДОЛЖНО генерировать ID, если полученное утверждение его не содержит.
- Провайдеру задач СЛЕДУЕТ генерировать ID самостоятельно.

4.1.2 Actor

Описание

Обязательное свойство, которое может иметь тип Agent или Group.

4.1.2.1 Actor имеет тип Agent

Описание

Объект Agent символизирует человека или систему.

Детали

- Объект Agent ДОЛЖЕН идентифицироваться с помощью одного из четырёх типов обратных функциональных идентификаторов (см. раздел 4.1.2.3 Обратный функциональный идентификатор);
- Объект Agent НЕ ДОЛЖЕН иметь более одного обратного функционального идентификатора;
- Для объекта Agent НЕ СЛЕДУЕТ использовать обратный функциональный идентификатор, который также используется в качестве идентификатора группы.

Ниже в таблице перечислены свойства объектов типа Agent.

Свойство	Тип	Описание	Примечание
objectType	String	"Agent". Это свойство опционально, кроме случаев, когда Agent используется как объект некоторого утверждения.	Опциональное
name	String	Полное имя объекта Agent.	Опциональное
см. раздел 4.1.2.3 Обратный функциональный идентификатор		Некоторый обратный функциональный идентификатор, уникальный для объекта Agent.	Обязательное

4.1.2.2 Actor имеет тип Group

Описание

Объект Group представляет группу объектов типа Agent и может быть использован в тех же случаях, что объект типа Agent. Есть две разновидности объекта Group: анонимные и зарегистрированные группы.

Детали

Анонимная группа используется для описания совокупности агентов, которая не имеет идентификатора, например, случайная группа людей.

В таблице ниже перечислены все свойства анонимной группы.

Свойство	**Тип**	**Описание**	**Примечание**
objectType	String	"Group".	Обязательное
name	String	Имя группы.	Опциональное

member	Массив объектов типа Agent	Члены группы.	Обязательное

Зарегистрированная группа используется для уникальной идентификации группы агентов.

В таблице ниже перечислены все свойства зарегистрированной группы.

Свойство	**Тип**	**Описание**	**Примечание**
objectType	String	"Group".	Обязательное
name	String	Имя группы.	Опциональное
member	Массив объектов типа Agent	Члены группы.	Опциональное
см. раздел 4.1.2.3 Обратный функциональный идентификатор		Некоторый обратный функциональный идентификатор, уникальный для объекта Group.	Обязательное

Требования

- Система, обрабатывающая утверждение, ДОЛЖНА рассматривать каждую анонимную группу как новую, даже если она имеет одинаковый с другой группой набор членов.
- Система, обрабатывающая утверждение, НЕ ДОЛЖНА считать, что свойство member содержит точный список агентов анонимной или зарегистрированной группы.

Требования к анонимной группе

- Анонимная группа ДОЛЖНА включать в себя свойство member, содержащее список составляющих агентов.
- Анонимная группа НЕ ДОЛЖНА содержать в свойстве member объекты типа Group.
- Анонимная группа НЕ ДОЛЖНА включать никакие обратные функциональные идентификаторы.

Требования к зарегистрированной группе

- Зарегистрированная группа ДОЛЖНА иметь ровно один обратный функциональный идентификатор.
- Зарегистрированная группа НЕ ДОЛЖНА содержать в свойстве member объекты типа Group.
- Зарегистрированной группе НЕ СЛЕДУЕТ использовать обратные функциональные идентификаторы, которые также используются в качестве идентификаторов агентов.
- Зарегистрированная группа МОЖЕТ включать свойство member, содержащее список составляющих агентов.

4.1.2.3 Обратный функциональный идентификатор

Описание

Обратный функциональный идентификатор является свойством агента или зарегистрированной группы, по которому всегда можно однозначно определить соответствующего агента или группу.

Пояснение

Учебный процесс становятся бессмысленным, если он не может быть соотнесён с конкретным лицом и/или группой. В утверждении xAPI это достигается посредством обратных функциональных идентификаторов, которые являются свободной интерпретацией широко распространенных FOAF-правил (см. http://xmlns.com/foaf/spec/#term_Agent).

Детали

В таблице ниже приведены все свойства, которые могут быть использованы в качестве обратных функциональных идентификаторов.

Свойство	Тип	Описание
mbox	mailto IRI	Требуемый формат: "mailto:email address". Для этого свойства и для свойства mbox_sha1sum должен использоваться только тот адрес электронной почты, который был назначен данному агенту.
mbox_sha1sum	String	Хеш-код свойства mbox по алгоритму SHA1. LRS МОЖЕТ включать агентов с совпадающим хеш-кодом, когда запрос построен на свойстве mbox.
openid	URI	openID, который однозначно идентифицирует агента.

account	Object	Учётная запись пользователя существующей системы, например LMS или интранет.

4.1.2.4 Объект Account

Описание

Учётная запись пользователя в существующей системе, например в закрытой системе (LMS или интранет) или публичной системе (социальной сети).

Детали

- Если система, которая предоставляет объект Account, использует OpenID, то провайдеру задач СЛЕДУЕТ использовать свойство openid вместо account.
- Если провайдер задач обеспокоен раскрытием персональных данных об агенте или группе, то ему СЛЕДУЕТ использовать непрозрачное имя учётной записи (например, номер учётной записи), чтобы идентифицировать все утверждения о персоне, поддерживая анонимность.

Ниже в таблице перечислены все свойства объекта Account.

Свойство	Тип	Описание	Примечание
homePage	IRL	Домашняя страница системы, где зарегистрирована учётная запись. Свойство основано на accountService HomePage спецификации FOAF.	Обязательное
name	String	Уникальный идентификатор или имя, используемое для входа с учётной записью. Свойство основано на accountName спецификации FOAF.	Обязательное

Пример

Этот пример показывает описание агента, который идентифицируется с помощью непрозрачного имени:

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

4.1.3 Verb

Описание

Verb определяет действие между актёром и задачей.

Пояснение

Verb в утверждении xAPI описывает действие, выполняемое во время обучения. xAPI не определяет специальных значений для Verb (за одним исключением, а именно зарезервированное значение "http://adlnet.gov/expapi/verbs/voided", «аннулировано»). Вместо этого спецификация xAPI определяет, как создавать новые объкеты типа Verb, чтобы профессиональные сообщества могли установить их толкование и сделать их доступными для использования. Предопределённый список значений Verb будет ограничен по определению и, возможно, в будущем не сможет охватить все действия, выполняемые во время обучения.

Детали

Действия представлены в утверждении как объекты типа Verb, состоящие из IRI и набора отображаемых имён, которые соответствуют нескольким языкам или диалектам, что обеспечивает читаемость. В таблице ниже перечислены все свойства объекта Verb.

Свойство	Тип	Описание	Примечание
id	IRI	Соответствует определению действия. Каждое определение соответствует значению действия и не является отдельным словом. IRI должно быть читаемым и отображать смысл действия.	Обязательное
display	Словарь языков	Читаемое представление действия на одном или нескольких языках. Оно не влияет на смысл утверждения, а служит для того, чтобы предоставить удобочитаемое отображение значения уже определённого действия.	Рекомендуемое

Требования

- Свойство display ДОЛЖНО использоваться для отображения значения, которое определяется IRI объекта Verb.

- Система, которая читает утверждение, ДОЛЖНА использовать IRI объекта Verb для отображения смысла.
- Свойство display НЕ ДОЛЖНО использоваться для изменения смысла действия.
- Система, которая читает утверждение, НЕ ДОЛЖНА использовать свойство display для определения смысла утверждения.
- Система, которая читает утверждение, НЕ ДОЛЖНА использовать свойство display для любых целей, кроме вывода на экран. Использование свойства display для агрегации или категоризации утверждений является примером нарушения этого требования.
- Свойство display СЛЕДУЕТ использовать во всех утверждениях.
- Для свойства id СЛЕДУЕТ использовать IRI, понятные человеку и передающие смысл действия.

Пример

```
{
    "verb" : {

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

Данный пример приведён только для иллюстрации. Не предполагается, что объект Verb со значением, которое определено свойством id из примера, будет использоваться. Это относится ко всем примерам объектов типа Verb, которые приведены в данной спецификации, за исключением зарезервированного действия "http://adlnet.gov/expapi/verbs/voided".

4.1.3.1 Семантика и язык в объекте Verb

Детали

Семантика

IRI в объекте Verb идентифицирует семантику слова, а не само слово.

Например, английское слово «fired» может менять значение в зависимости от контекста: «fired a weapon» (стрелял из оружия), «fired a kiln» (обжигал в печи)

или «fired an employee» (уволил сотрудника). В этом случае IRI ДОЛЖЕН указать на один из этих вариантов, но не слово «fired».

В свойстве display могут использоваться глаголы в различных временных формах. Для IRI предполагается использование глаголов в форме прошедшего времени, однако в display допускается использования глаголов в других временных формах, если это не приведёт к потере смысла.

Язык

Объект Verb в Experience API определяется его IRI и содержит в себе некоторый смысл, который не зависит от конкретного языка.

Например, объект Verb с IRI, равным http://example.org/firearms#fire, может означать «стрелять из ружья» ("firing a gun"), а с "IRI", равным http://example.com/فعل/خواندن, может означать «читать книгу» ("reading a book").

4.1.3.2 Использования в профессиональных сообществах

Описание

Профессиональные сообщества будут создавать новые объекты типа Verb для того, чтобы удовлетворить потребности своих членов.

Поэтому ожидается, что профессиональные сообщества xAPI будут создавать профили, списки и хранилища объектов типа Verb. ADL создаёт сопутствующий документ, содержащий объекты типа Verb для нужд сообщества ADL.

Во исполнение изложенных ниже требований существует коллекция IRI рекомендуемых объектов типа Verb. Возможно, что провайдеры задач пожелают использовать другие глаголы для описания указанных действий.

Требования к сообществам

- Любой, кто создаёт новый объект Verb, ДОЛЖЕН владеть IRI или иметь разрешение от владельца для использования его в объекте Verb xAPI;
- Любому, кто создаёт новый объект Verb, СЛЕДУЕТ сделать удобочитаемое описание этого объекта и его использования, доступное по его IRI.

Требования к провайдерам задач

- Провайдерам задач СЛЕДУЕТ использовать существующие объекты типа Verb, если это возможно.
- Провайдеры задач МОГУТ создавать и использовать объекты типа Verb, если нет подходящих.

4.1.4 Object

Описание

Object – объект утверждения, в качестве которого могут выступать задача, агент или группа, подутверждение или ссылка на утверждение. Объект определяет «это» в предложении «Я сделал это».

Некоторые примеры:

- Объектом является задача: «Джефф написал эссе о прогулке».
- Объектом является агент: «Нелли взяла интервью у Джеффа».
- Объектом является подутверждение или ссылка на утверждение (имеют различные реализации, но одинаковое прочтение): «Нелли прокомментировала утверждение „Джефф написал эссе о прогулке“».

Детали

Объектам, которые выступают в качестве объекта утверждения, СЛЕДУЕТ содержать поле objectType. Если это поле отсутствует, то предполагается, что объект имеет тип Activity (задача). Другими допустимыми значениями являются: Agent, Group, SubStatement, StatementRef. Набор свойств объекта утверждения зависит от его типа.

4.1.4.1 ObjectType имеет значение Activity

Детали

Объектом утверждения является какая-либо деятельность (задача). В нижеследующей таблице перечислены свойства объекта утверждения для этого случая.

Свойство	Тип	Описание	Примечание
objectType	String	Если указано, то ДОЛЖНО иметь значение Activity.	Опциональное

id	IRI	Идентификатор деятельности.	Обязательное
definition	Object	Метаданные (см. ниже).	Опциональное

Если один и тот же id будет использоваться для двух различных видов деятельности, то это поставит под вопрос корректность построенных на них утверждений. Это означает, что LRS должны рассматривать два вида деятельности с одинаковыми id как один вид, даже если предполагалось обратное. Таким образом, в случае возникновения конфликта между системами не существует способа восстановить первоначальный замысел.

В таблице ниже перечислены метаданные объекта типа Activity.

Свойство	**Тип**	**Описание**	**Примечание**
name	Словарь языков (см. 5.2 Словарь языков)	Удобочитаемое название деятельности (задачи) с указанием языка, на котором оно записано.	Рекомендуемое
description	Словарь языков (см. 5.2 Словарь языков)	Описание деятельности (задачи) с указанием языка, на котором оно записано.	Рекомендуемое
type	IRI	Вид деятельности.	Рекомендуемое
moreInfo	IRL	СЛЕДУЕТ указывать путь к документу с описанием конкретной деятельности, который также МОЖЕТ содержать описания того, как начать эту деятельность.	Опциональное
Свойства взаимодействия (см. Интерактивные задачи).			
extensions	Object	Карта других требуемых свойств (см. Расширения)	Опциональное

Примечание: фрагмент IRI (так же называемый относительным IRL) не является валидным IRI. Как и в случае с объектом типа Verb, провайдерам задач рекомендуется по мере возможностей использовать утвердившиеся, широко распространённые объекты типа Activity.

Требования к полю id объекта Activity

- Поле id ДОЛЖНО быть уникальным.
- Поле id ДОЛЖНО всегда указывать на один и тот же объект типа Activity.
- Для поля id СЛЕДУЕТ использовать домен, авторизованный для этих целей.

- Значение поля id СЛЕДУЕТ определять по схеме, обеспечивающей уникальность всех объектов типа Activity в пределах одного домена.
- Поле id МОЖЕТ указывать на метаданные или IRL объекта типа Activity.

Требования к LRS

- LRS НЕ ДОЛЖНО корректировать своё поведение в случае, когда два автора или две организации используют одинаковый id для объектов типа Activity.
- LRS НЕ ДОЛЖНО воспринимать ссылки на один id как ссылки на разные объекты типа Activity.
- При получении утверждения, содержащего объект типа Activity со старым значением поля id и новыми (отличными от хранимых) метаданными, LRS СЛЕДУЕТ решить, имеет ли провайдер задач право изменить объект. Если имеет – LRS СЛЕДУЕТ изменить хранимые метаданные.
- LRS МОЖЕТ вносить небольшие поправки в метаданные объекта типа Activity. Например, можно исправить орфографические ошибки. Но недопустимо менять набор правильных ответов.

Требования к провайдеру задач

- Провайдер задач ДОЛЖЕН гарантировать уникальность id объекта типа Activity.
- Провайдер задач ДОЛЖЕН для одного конкретного id объекта типа Activity генерировать только совместимые между собой утверждения и состояния.
- Провайдер задач НЕ ДОЛЖЕН допускать нарушений совместимости, связанных с обновлением объекта типа Activity (например, при редактировании или смене платформы).

Требования к метаданным

- Если поле id объекта типа Activity представляет собой IRL, то LRS СЛЕДУЕТ сделать GET-запрос по этому IRL, включив HTTP-заголовок "Accept: application/json, */*". Это СЛЕДУЕТ сделать как можно скорее после анализа поля id.
- При загрузке по IRL валидного определения объекта типа Activity в формате JSON LRS СЛЕДУЕТ зарегистрировать у себя это определение для данного объекта, даже если оно не содержит в себе использованных зарезервированных имён или определений.
- Любой объект типа Activity с IRL-идентификатором МОЖЕТ иметь метаданные в формате JSON, который используется в утверждениях, с заголовком "Content-Type: application/json".

- При загрузке любого документа, из которого LRS может извлечь метаданные объекта типа Activity по IRL из поля id, LRS МОЖЕТ использовать эти метаданные для внутреннего представления объекта.

Интерактивные задачи

Пояснение

Традиционное электронное обучение включает элементы для взаимодействия и оценивания результатов обучения. Чтобы сделать такие элементы частью Experience API, данная спецификация включает встроенные определения взаимодействий, взятые из модели данных SCORM 2004, 4-е издание. Цель этих определений: предоставление простого и удобного инструмента для записи данных о взаимодействии. Эти определения просты в использовании, но их возможности ограничены. Предполагается, что профессиональные сообщества, требующие более сложных данных о взаимодействии, будут использовать расширения типа и метаданных объектов типа Activity.

Детали

Свойства интерактивной задачи перечислены в таблице ниже.

Свойство	**Тип**	**Описание**	**Примечание**
interactionType	String	Соответствует "cmi.interactions.n.type" в книге Run-Time Environment (RTE) спецификации SCORM 2004, 4-е издание.	Опциональное
correctResponses Pattern	Массив элементов типа String	Соответствует "cmi.interactions.n.correct_ responses.n.pattern" в книге Run-Time Environment (RTE) спецификации SCORM 2004, 4-е издание, где последнее n – индекс массива.	Опциональное
choices | scale | source | target | steps	Массив интерактив-ных элементов	В зависимости от типа взаимодействия, указанного в interactionType (см. ниже).	Опциональное

Требования

- Интерактивная задача ДОЛЖНА иметь корректное значение interactionType.
- Интерактивной задаче СЛЕДУЕТ иметь значение "http://adlnet.gov/expapi/activities/cmi.interaction" в поле type объекта типа Activity.
- LRS, получая валидный interactionType, МОЖЕТ проверить корректность остальных свойств в соответствии с таблицей ниже и МОЖЕТ вернуть HTTP 400 "Bad Request", если есть ошибки.

Интерактивные элементы

Детали

Интерактивные элементы определены следующим образом:

Свойство	Тип	Описание	Примечание
id	String	Соответствует "cmi.interactions.n.id" в книге Run-Time Environment (RTE) спецификации SCORM 2004, 4-е издание.	Обязательное
description	Словарь языков (см. 5.2 Словарь языков)	Описание интерактивного элемента (например, текст к заданию с возможностью множественного выбора) с указанием языка, на котором оно записано.	Опциональное

Следующая таблица отображает поддерживаемые интерактивные элементы в модели компьютеризированного обучения (Computer Managed Instruction, CMI) для интерактивных задач с заданным значением поля interactionType.

Значение поля interactionType	Поддерживаемые списки элементов
choice, sequencing	choices
likert	Scale
matching	source, target
performance	Steps
true-false, fill-in, numeric, other	[элементы не определены]

Требования

- В массиве интерактивных элементов все id ДОЛЖНЫ быть уникальными.
- В значение поля id интерактивных элементов НЕ СЛЕДУЕТ включать пробелы.

Пример

В Приложении C приведены определения интерактивных задач для каждого типа cmi.interaction.

4.1.4.2 ObjectType имеет значение Agent или Group

Требования

- Утверждения, в которых используются Agent или Group в качестве объекта, ДОЛЖНЫ корректно определить свойство "objectType".

Описания объектов типа Agent и Group приведены в разделе 4.1.2 Actor.

4.1.4.3 ObjectType имеет значение Statement

Пояснение

Есть два способа использовать утверждения в качестве объекта другого утверждения. Во-первых, объектом можно сделать уже существующее утверждение, используя ссылку на утверждение. Типичный случай использования ссылки в утверждении: оценивание или комментирование некоторого учебного процесса, который можно рассматривать как независимый. При аннулировании утверждений так же используются ссылки. Во-вторых, объектом может быть новое утверждение, так называемое подутверждение. Типичное применение подутверждений: навыки, которые можно описать в виде собственных утверждений. Каждый вариант использования утверждений в качестве объектов определен ниже.

Ссылка на утверждение

Описание

Ссылка указывает на существующее утверждение.

Требования

- Ссылка ДОЛЖНА содержать поле "objectType", в котором указано значение "StatementRef".
- Ссылка ДОЛЖНА содержать поле "id", равное "UUID" утверждения. Проверять наличие указанного утверждения на LRS не требуется.

В таблице ниже перечислены свойства объекта, описывающего ссылку на утверждение.

Свойство	Тип	Описание	Примечание
objectType	String	ДОЛЖНО иметь значение "StatementRef".	Обязательное
id	UUID	UUID утверждения.	Обязательное

Пример

Предположим, что некоторое утверждение, сохранённое в LRS, имеет идентификатор 8f87ccde-bb56-4c2e-ab83-44982ef22df0. Следующий пример показывает, как можно прокомментировать данное утверждение в новом утверждении:

```
{
    "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!"
    }
}
```

Подутверждения

Описание

Подутверждение – это новое утверждение, которое является частью родительского утверждения.

Требования

- Свойство objectType подутверждения ДОЛЖНО иметь значение "SubStatement".

- Подутверждение ДОЛЖНО быть валидным утверждением в дополнение к другим требованиям, которые распространяются на подутверждения.
- Подутверждение НЕ ДОЛЖНО иметь свойства id, stored, version, authority.
- Подутверждение НЕ ДОЛЖНО иметь вложенных подутверждений.

Пример

Интересным вариантом использования подутверждений является составление утверждений о намерениях. Например, утверждение с подутверждением может иметь форму "<Я> <планировал> (<Я> <сделал> <это>)", которая содержит намерение совершить некоторое действие. В примере ниже сформулировано следующее утверждение: «Я планировал посетить 'Некоторый удивительный сайт'».

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

```
    }
}
```

4.1.5 Result

Описание

Необязательное поле, которое описывает измеренный результат обучения, относящийся к утверждению, с которым это поле связано.

Детали

Следующая таблица содержит свойства объекта типа Result.

Свойство	Тип	Описание	Примечание
score	Object	Баллы, полученные агентом в связи с успешностью или качеством обучения (см. Score).	Опциональное
success	Boolean	Отвечает на вопрос: была ли успешной попытка решить задачу?	Опциональное
completion	Boolean	Отвечает на вопрос: была ли задача завершена?	Опциональное
response	String	Ответ в формате задачи.	Опциональное
duration	В формате ISO 8601 с точностью 0.01 секунды	Продолжительность решения задачи.	Опциональное
extensions	Object	Набор других свойств, которые требуются для описания результата (см. Расширения).	Опциональное

4.1.5.1 Score

Описание

Опциональное поле, содержащее баллы, которые получены агентом за решение задачи.

Детали

Таблица ниже определяет объект типа Score.

Свойство	Тип	Описание	Примечание
scaled	Десятичное число от -1 до 1 включительно	Соответствует "cmi.score.scaled" в спецификации SCORM 2004, 4-е издание.	Рекомендуемое
raw	Десятичное число от min до max включительно (если они указаны, иначе без ограничений)	Соответствует "cmi.score.raw" в спецификации SCORM 2004, 4-е издание.	Опциональное
min	Десятичное число меньше max (если указано)	Соответствует "cmi.score.min" в спецификации SCORM 2004, 4-е издание.	Опциональное
max	Десятичное число больше min (если указано)	Соответствует "cmi.score.max" в спецификации SCORM 2004, 4-е издание.	Опциональное

Требования

- В объект типа Score СЛЕДУЕТ включать поле scaled, если баллы, полученные за выполнение задачи, известны.
- Объект типа Score НЕ СЛЕДУЕТ использовать для учёта текущей или итоговой суммы баллов. Для этих целей рекомендуется использовать расширения профиля пользователя.

4.1.6 Context

Описание

Необязательное поле, которое предназначено для дополнительной контекстно-зависимой информации утверждения. Все свойства объекта типа Context являются опциональными.

Пояснение

Поле context утверждения предоставляет место для описания контекстно-зависимой информации. Например: данные преподавателя, индивидуальное или групповое обучение, роль задачи в учебном процессе.

Детали

Следующая таблица содержит свойства объекта типа Context.

Свойство	Тип	Описание	Примечание
registration	UUID	Учётная запись, с которой связано утверждение.	Опциональное
instructor	Agent или Group	Преподаватель, которого затрагивает утверждение, если он не включен в утверждение в качестве актёра.	Опциональное
team	Group	Команда, которую затрагивает утверждение, если эта команда не является актёром в утверждении.	Опциональное
Context Activities	Объект типа context Activities	Соответствие типов контекста учебным задачам, с которыми связано утверждение. Валидными типами контекста являются: "parent" (родитель), "grouping" (группировка), "category" (категория) и "other" (другое).	Опциональное
revision	String	Исправление учебной задачи, связанной с утверждением. Используется свободный формат.	Опциональное
platform	String	Платформа, на которой реализована учебная задача.	Опциональное
language	String (RFC 5646)	Код языка, с использованием которого проходило выполнение учебной задачи, описанной в утверждении, если это имеет смысл и этот язык известен.	Опциональное
statement	Ссылка на объект типа Statement	Другое утверждение, которое рассматривается как контекст текущего утверждения.	Опциональное
extensions	Object	Соответствия между другими областями контекста, имеющими отношение к утверждению. Например, в пилотажном тренажёре высота, скорость, ветер, положение, GPS-координаты могут иметь отношение к утверждению.	Опциональное

Требования

- Свойство revision ДОЛЖНО использоваться только в случае, когда объектом утверждения является задача.
- Свойство platform ДОЛЖНО использоваться только в случае, когда объектом утверждения является задача.
- Свойство language НЕ ДОЛЖНО использоваться, когда значение неизвестно или свойство в контексте задачи неприменимо.
- Свойство revision СЛЕДУЕТ использовать для отслеживания исправлений незначительных ошибок, например орфографических ошибок.
- Свойство revision НЕ СЛЕДУЕТ использовать, когда имеются значительные изменения в целях обучения, методах обучения или оценивания (в этом случае используйте новый идентификатор задачи).

Примечание: Свойство revision не влияет на процессы в рамках xAPI. Значение этого свойства сохраняется для использования при построении отчётов.

4.1.6.1 Свойство registration

Описание

Отдельный ученик, осуществляющий определённую учебную деятельность.

Детали

Когда LRS является частью LMS, то процедуру регистрации осуществляет LMS. Спецификация Experience API трактует процедуру регистрации более широко. Регистрация может рассматриваться как некоторая попытка или сессия и может охватывать выполнение нескольких задач. Не ожидается, что выполнение задачи отменяет регистрацию. Регистрация не ограничивается одним агентом.

4.1.6.2 Свойство contextActivities

Описание

Соответствие типов контекста экземплярам учебной деятельности, с которыми связано утверждение.

Пояснение

Многие утверждения затрагивают не одну учебную задачу, которая находится в центре внимания, но и другие контекстуально-сопутствующие учебные задачи.

Свойство contextActivities позволяет все сопутствующие задачи представить в структурированном виде.

Детали

В утверждении МОЖЕТ быть использован любой из стандартных типов контекста, в том числе все или ни одного. Существует четыре типа контекста:

Parent (родитель) – любая задача с прямым отношением к задаче, которая является объектом утверждения. В почти всех случаях либо есть только один родитель, либо не существует ни одного. Например: утверждение относительно вопроса в коротком тесте будет иметь тест в качестве родительской задачи.

Grouping (группировка) – задача с косвенным отношением к задаче, которая является объектом утверждения. Например: учебный курс, который является частью образовательной программы. Курс изучают несколько групп студентов. Курс выступает в качестве родителя по отношению к группе, образовательная программа и группа связаны отношением группировки.

Category (категория) – задача, которая используется для категоризации утверждения. "Tag" (метка) является синонимом. Категорию СЛЕДУЕТ использовать для профилирования xAPI-процессов, а также других категоризаций. Например: Анна пытается сдать экзамен по биологии, утверждение отслеживается с помощью профиля CMI-5. Задачей данного утверждения является экзамен, а категорией – профиль CMI-5.

Other (другое) – задача, принадлежащая контексту, которая не соответствует ни одному из приведённых выше типов. Например: Анна изучает учебник для экзамена биологии. Задачей утверждения является учебник, а экзамен – контекстная задача, которая имеет тип "other".

Простые объекты Activity могут выступать в качестве значения свойства contextActivities. Это сделано для того, чтобы утверждения из версии 0.95 были совместимы с версией 1.0.0.

Примечание: Типы отношений, приведённые в этом разделе, не предназначены для описания всех отношений, в которые могут вступать объекты утверждений. Вместо этого они предназначены для описания отношений, подходящих рассматриваемому утверждению (хотя природа объекта часто будет иметь большое значение в определении отношений). Например, уместно в утверждении о тесте установить курс как родительский элемент теста, но не определять все

возможные образовательные программы, которые содержат курс, в качестве различных группировок теста.

Требования

- Любой ключ в объекте contextActivities ДОЛЖЕН иметь одно из следующих значений: "parent", "grouping", "category", "other".
- Любое значение объекта contextActivities ДОЛЖНО быть либо простым объектом типа Activity, либо массивом объектов типа Activity.
- LRS ДОЛЖНО сохранять любое значение объекта contextActivities как массив, даже если получен один объект Activity.
- LRS ДОЛЖНО возвращать простой объект Activity как массив с длиной, равной единице.
- Клиенту СЛЕДУЕТ обеспечить то, что значение объекта contextActivities является массивом объектов типа Activity, а не одним объектом Activity.

Пример

Рассмотрим следующую иерархическую структуру: "Вопросы с 1 по 6" являются частью теста "Тест 1", который, в свою очередь, принадлежит курсу "Алгебра 1". Шесть вопросов зарегистрированы как часть теста посредством объявления "Тест 1" родителем этих вопросов. Также они сгруппированы с помощью других утверждений о курсе "Algebra 1" для полного описания иерархии. Это имеет практическое значение, когда объектом утверждения является агент, а не задача: "Andrew mentored Ben with context Algebra I" (Эндрю был ментором Бена в рамках курса Алгебра I).

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

4.1.7 Timestamp

Описание

Время, в которое произошло действие.

Детали

В утверждении timestamp может отличаться от значения поля stored (время, которое сохранено утверждение). А именно, могут возникать задержки между выполнением действия и сохранением утверждения в LRS.

Когда действие происходит в течение времени, timestamp может представлять начало или конец действия, а также любой другой момент во время выполнения действия. Ожидается, что профессиональные сообщества определят момент времени, который будет записан в поле timestamp, в зависимости от типа действия. Например: если записывается информация об обеде в ресторане, то было бы правильно записать в timestamp время начала обеда; когда сохраняется информация о завершении обучения по образовательной программе, то было бы правильно записать в timestamp время окончания обучения. Эти примеры приведены только в целях интерпретации сказанного и не являются предписывающими.

Требования

- Поле timestamp ДОЛЖНО быть отформатировано в соответствии с ISO 8601.
- В поле timestamp СЛЕДУЕТ включать часовой пояс.
- Значению timestamp СЛЕДУЕТ присваивать текущее или прошлое время вне подутверждения.
- Поле timestamp МОЖЕТ указывать на любой момент интервала времени, в течение которого происходило действие.
- Значение timestamp МОЖЕТ быть округлено до миллисекунд (миллисекунды ДОЛЖНЫ быть сохранены).
- Значение timestamp МОЖЕТ указывать на момент времени в будущем внутри подутверждения, чтобы обозначить запланированный срок завершения.

4.1.8 Stored

Описание

Момент времени, в который утверждение было сохранено в LRS.

В свойстве stored указывается точное время сохранения утверждения. Для указания момента времени, в который происходит описанное в утверждении действие, используйте свойство timestamp.

Требования

- Свойство stored ДОЛЖНО быть отформатировано в соответствии с требованиями ISO 8601.
- В значение свойства stored СЛЕДУЕТ включать часовой пояс.
- В значение свойства stored СЛЕДУЕТ записывать текущее или прошедшее время.
- Значение свойства stored МОЖЕТ быть округлено до миллисекунд (миллисекунды ДОЛЖНЫ быть сохранены).

4.1.9 Authority

Описание

Свойство authority содержит информацию о том, кто подтвердил истинность утверждения.

Детали

Подтвердить истинность утверждения может пользователь, некоторая система или приложение.

Требования

- Значением свойства authority ДОЛЖЕН быть объект типа Agent, кроме случая, когда авторизация осуществляется по трёхзвенной схеме протокола OAuth, тогда значением ДОЛЖНА быть группа, содержащая два объекта типа Agent, которые символизируют приложение и пользователя.
- LRS ДОЛЖНО сохранять пользователя в качестве объекта типа Agent, если пользователь подключается напрямую (используя базовую аутентификацию HTTP) или входит в состав группы.
- LRS ДОЛЖНО гарантировать наличие свойства authority у всех сохранённых утверждений.
- LRS СЛЕДУЕТ перезаписывать свойство authority у всех сохранённых утверждений на основе токенов, использованных для отправки этих утверждений.
- LRS МОЖЕТ оставить свойство authority без изменений, но так СЛЕДУЕТ делать только в тех случаях, когда были созданы сильные доверительные отношения, и с особой осторожностью.
- LRS МОЖЕТ идентифицировать пользователя по любым легальным свойствам, если пользователь соединяется непосредственно (используя

базовую аутентификацию HTTP) или по трёхзвенной схеме протокола OAuth.

Использование токена OAuth в качестве значения свойства authority

Описание

Этот процесс описывает использование протокола OAuth. Поддерживаются двухзвенные и трёхзвенные схемы протокола OAuth.

Детали

Этот процесс предполагает, что утверждение сохранено с использованием валидного OAuth-соединения, а хранилище LRS создаёт или модифицирует свойство authority утверждения.

В трехзвенной схеме протокола OAuth процесс аутентификации включает и OAuth-приложение, и пользователя OAuth-сервера (провайдера ресурсов). Например, запросы авторизованного Twitter-плагина под учётной записью Facebook будут включать удостоверение, которое содержит информацию как о Twitter-плагине, так и о пользователе Facebook.

Требования

- Свойство authority ДОЛЖНО содержать объект типа Agent, который представляет OAuth-приложение либо непосредственно, либо как часть группы в случае трехзвенной схемы протокола OAuth.
- Объект Agent, описывающий OAuth-приложение, ДОЛЖЕН идентифицироваться учётной записью.
- Объект Agent, описывающий OAuth-приложение, ДОЛЖЕН использовать ключ приложения как имя учётной записи.
- Если объект Agent описывает зарегистрированное OAuth-приложение, то адресат токена запроса (token request) ДОЛЖЕН использоваться как значение поля homePage учётной записи.
- Если объект Agent описывает незарегистрированное OAuth-приложение, то адресат временного удостоверения (temporary credentials) ДОЛЖЕН использоваться как значение поля homePage учётной записи.
- Хранилище LRS НЕ ДОЛЖНО доверять приложениям в случае, когда имя учётной записи из того же источника, что и незарегистрированное приложение (несколько незарегистрированных приложений могут выбрать одинаковый ключ, поэтому нет надежного способа проверки комбинации временного удостоверения и имени учётной записи).

- Каждому незарегистрированному OAuth-приложению СЛЕДУЕТ использовать уникальный ключ.

Пример

Сопоставление OAuth-потребителя и пользователя.

```
"authority": {
    "objectType" : "Group",
    "member": [
        {
            "account": {
                "homePage":"http://example.com/xAPI/OAuth/Token",
                "name":"oauth_consumer_x75db"
            }
        },
        {
            "mbox":"mailto:bob@example.com"
        }
    ]
}
```

4.1.10 Version

Описание

Информация о версии помогает системам, которые обрабатывают данные из LRS, сориентироваться. Так как модель данных утверждения гарантирует совместимость всех версий 1.0.x, то для поддержки обмена данными между хранилищами LRS предоставляется возможность выбора версии утверждения.

Требования

- Поле version ДОЛЖНО быть отформатировано в соответствии с требованиями к описанию версии API, изложенными в разделе 6.2 Нумерация версий API.

Требования к LRS

- Хранилище LRS ДОЛЖНО принять все валидные утверждения, версия которых начинается с "1.0."
- Хранилище LRS ДОЛЖНО отклонять все утверждения, версия которых не начинается с "1.0."

- Утверждения, возвращённые LRS, ДОЛЖНЫ сохранить версию, с которой они были приняты. Если версия не указана, то клиент ДОЛЖЕН предполагать, что утверждение соответствует версии 1.0.0.

Требования к клиенту

- Если клиент указывает версию утверждения, он ДОЛЖЕН указывать значение 1.0.0.
- Клиентам НЕ СЛЕДУЕТ назначать версию утверждениям.

4.1.11 Вложения

Описание

Цифровой артефакт, который свидетельствует о приобретённом опыте.

Пояснение

В некоторых случаях вложение может быть важной частью записи об учебных достижениях. Вспомните о моделировании управления воздушным движением, эссе, видео и т.д. Другой пример такого вложения: изображение сертификата, полученного в результате обучения. Полезно иметь способы сохранения этих вложений в LRS и извлечения их оттуда. В случае необходимости прикрепить вложение к подутверждению, мы настоятельно рекомендуем использовать для этого поле attachment утверждения таким же образом, как для самого утверждения.

Детали

В таблице ниже перечислены все свойства объекта типа Attachment.

Свойство	Тип	Описание	Примечание
usageType	IRI	Определяет назначение вложения. Например, один из возможных вариантов использования вложений: сертификат о завершении. Для этого варианта должен быть разработан специальный IRI.	Обязательное
display	Словарь языков	Отображаемое имя вложения.	Обязательное
description	Словарь языков	Описание вложения.	Опциональное

contentType	Internet Media Type	Тип данных вложения.	Обязательное
length	Integer	Размер вложения в октетах.	Обязательное
sha2	String	SHA-2 (SHA-256, SHA-384, SHA-512) хеш-код данных вложения. SHA-224 НЕ СЛЕДУЕТ использовать: рекомендуемый минимальный размер ключа составляет 256 бит.	Обязательное
fileUrl	IRL	IRL, по которому вложение может получено или было получено ранее.	Опциональное

Процедура обмена вложениями

1) Утверждение с вложением оформляется в соответствии с форматом передачи, описанным ниже.
2) Утверждение посылают в систему назначения, указав значение "multipart/mixed" в поле заголовка Content-Type. Вложения размещаются в конце сообщения.
3) Система принимает или отклоняет переданное утверждение на основе информации, содержащейся в первой части сообщения.
4) Если система принимает вложение, она ДОЛЖНА сверить код SHA-2 полученных данных с кодом, указанным в заголовке утверждения.

Требования к вложениям в пакетах утверждений

Пакеты утверждений, выборки утверждений и отдельные утверждения, содержащее вложения ДОЛЖНЫ удовлетворять одному из следующих требований:

- ДОЛЖНЫ иметь тип "application/json" и содержать поле fileUrl для каждого вложения, КРОМЕ выборки, когда фильтр вложений attachments отключен.
- ДОЛЖНЫ соответствовать определению "multipart/mixed" в RFC 1341 и следующим правилам:
    - Первая часть документа ДОЛЖНА иметь тип "application/json" и содержать само утверждение.
    - Каждая дополнительная часть документа должна содержать данные вложений. Для этого надо использовать метод PUT или POST вместо механизмов утверждения.
    - Заголовок каждой дополнительной части после первой ДОЛЖЕН содержать поле "X-Experience-API-Hash".

- Это поле ДОЛЖНО совпадать с полем sha2 заголовка вложения, которое соответствует вложению, содержащемуся в этой части.
- Заголовок каждой дополнительной части после первой ДОЛЖЕН содержать поле Content-Transfer-Encoding, которое имеет значение "binary".
- СЛЕДУЕТ прикреплять только одну копию вложения, когда одно и то же вложение используется в нескольких утверждениях, входящих в одно сообщение.
- В заголовок каждой части СЛЕДУЕТ включать поле Content-type, в первой части это поле ДОЛЖНО иметь значение "application/json".

Требования к LRS

- LRS ДОЛЖНО включать вложения в соответствии с форматом передачи, описанным выше, по запросу клиента (см. раздел 7.2 Statement API).
- LRS НЕ ДОЛЖНО запрашивать утверждения из другого хранилища без вложений.
- LRS НЕ ДОЛЖНО передавать утверждения в другое хранилище без вложений, если они есть.
- Получая PUT или POST-запрос с типом документа "application/json", LRS ДОЛЖНО принять пакеты утверждений, которые не содержат вложений.
- Получая PUT или POST-запрос с типом документа "application/json", LRS ДОЛЖНО принять пакеты утверждений, которые содержат только вложения с полем fileUrl.
- Получая PUT или POST-запрос с типом документа "multipart/mixed", LRS ДОЛЖНО принять пакеты утверждений, которые содержат вложения в соответствии с форматом передачи, описанным выше.
- Получая PUT или POST-запрос с типом документа "multipart/mixed", LRS ДОЛЖНО отклонить пакеты утверждений с вложениями, которые либо не содержат поле fileUrl, либо содержат неверный хеш-код.
- Получая PUT или POST с типом документа "multipart/mixed", LRS СЛЕДУЕТ принять двоичные данные вложений.
- LRS МОЖЕТ не принимать утверждения или партии утверждений, размер которых превышает значения, указанные в настройках.

Примечание: Нет никаких указаний на то, чтобы пакеты утверждений, использующие формат "mime/multipart", содержали вложения.

Требования к клиенту

- Клиент МОЖЕТ отправить утверждения с вложениями, как описано выше.

- Клиент МОЖЕТ отправить несколько утверждений, которые могут иметь вложения, используя метод "POST".
- Клиент МОЖЕТ отправить пакеты с типом "application/json", игнорируя все требования формата "multipart/mixed", если описание каждого вложения имеет поле fileUrl.

Пример

Ниже приведён пример очень простого утверждения с приложением. Пожалуйста, учтите следующее:

- В примере ограничитель тела сообщения был выбран так, чтобы продемонстрировать валидные классы символов;
- Выбранный ограничитель не появляется ни в одной из частей сообщения;
- Для удобочитаемости вложение в примере представлено в формате "text/plain". Даже если оно было бы в двоичном формате, например "image/jpeg", то никакое кодирование не требуется: в сообщение были бы включены исходные данные;
- В соответствии с RFC 1341 ограничитель начинается с последовательности <CRLF>, за которой следует строка "--", затем строка, объявленная в заголовке.

Не забудьте про последовательность <CRLF>, когда будете собирать или анализировать сообщение.

Заголовки:

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

Содержимое:

```
--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:495395e777cd98da653df9615d09c0fd6bb2f8d4788394cd53c56a3bfdcd848
a

here is a simple attachment
--abcABC0123'()+_,-./:=?--
```

4.1.12 Ограничения целостности данных

Детали

Все свойства в утверждении соответствуют определённым типам, и эти типы определяют поведение системы, обрабатывающей утверждения. Для наглядности здесь приведены некоторые ключевые требования и отмечено, каким образом должны вести себя совместимые системы.

Требования к клиентам

Нижеперечисленные требования повторяют наиболее важные из описанных выше, чтобы подчеркнуть, разъяснить и предоставить руководство по их реализации. Полная валидация IRI – весьма трудоёмкая задача, поэтому значительная часть ответственности за переносимость данных возложена на клиента.

- Значения типа IRI ДОЛЖНЫ быть валидными IRI.
- Ключи языков в словаре ДОЛЖНЫ иметь валидные по RFC 5646 значения.
- Для построения IRI СЛЕДУЕТ использовать библиотеку, а не конкатенацию строк.

Требования к LRS

- LRS ДОЛЖНО отклонять утверждения:
  - с любыми null-значениями (кроме внутренних расширений);
  - со строками, когда требуются числа, даже если строки содержат соответствующие значения;
  - со строками, когда требуются логические значения, даже если строки содержат соответствующие значения;
  - с ключами или значениями, не соответствующими требуемому формату, включая пустые строки, если требуется строка определённого формата (например: IRI, mailto IRI или UUID);
  - когда указанный ключ не совпадает ни с одним из допустимых по стандарту;
  - когда указанное значение перечислимого типа не совпадает точно ни с одним из допустимых по стандарту для этого типа.
- LRS ДОЛЖНО отклонять утверждения, содержащие IRL или IRI без схемы.
- LRS ДОЛЖНО как минимум проверять соответствие длины лексем ключей в словаре языков стандарту RFC 5646.
- LRS ДОЛЖНО обрабатывать и сохранять числа с точностью не менее 32-бит в формате с плавающей точкой по стандарту IEEE 754.
- LRS ДОЛЖНО контролировать значения параметров по тем же стандартам, которые используются для полей утверждений соответствующих типов. Примечание: строки не берутся в кавычки, поскольку находятся внутри JSON.
- LRS МОЖЕТ использовать максимально эффективную проверку для IRL и IRI, чтобы обеспечить отклонение при несоответствии формату.
- LRS МОЖЕТ использовать максимально эффективную проверку для ключей словаря языков, чтобы обеспечить отклонение при несоответствии формату.

4.2 Извлечение утверждений

Описание

Выборка утверждений может быть выполнена с помощью запроса к узлу "statements" (см. раздел 7.2 Statement API).

Детали

Следующая таблица содержит структуру данных, отображающих результат запроса к Statement API.

Свойство	Тип	Описание	Примечание
statements	Массив объектов типа Statement	Список утверждений. Если список неполный и имеются ещё данные, то они будут располагаться в свойстве statements контейнера, доступного по IRL из свойства more текущего объекта.	Обязательное
more	IRL	Относительный IRL, который может использоваться для получения оставшихся данных. Включает полный путь и, возможно, параметры запроса, но не включает схему, сервер и порт. Пустая строка, если больше нет данных для извлечения. Данный IRL должен оставаться доступным не менее 24 часов после того, как был возвращён LRS. Во избежание необходимости сохранять эти IRL и данные из соответствующих запросов, LRS могут включать всю необходимую информацию прямо в IRL, но следует избегать чересчур длинных IRL. Конечному пользователю не следует пытаться вникнуть в смысл содержащейся в IRL информации.	Обязательное, если список утверждений неполный, иначе опциональное.

Требования

- IRL, содержащийся в свойстве more, ДОЛЖЕН оставаться доступным не менее 24 часов после получения из LRS.
- LRS МОЖЕТ включать в IRL всю необходимую информацию для получения оставшихся результатов запроса без необходимости хранить IRL и соответствующие данные запроса.
- LRS НЕ СЛЕДУЕТ генерировать чересчур длинные IRL с большим количеством деталей.
- Конечному пользователю НЕ СЛЕДУЕТ пытаться вникнуть в смысл информации, содержащейся в IRL из свойства more.

4.3 Аннулирование утверждений

Пояснение

Гарантия того, что LRS содержит точный и полный набор данных, обеспечивается тем, что утверждения не могут быть изменены или удалены. Неизменность утверждений – ключевой фактор, обеспечивающий возможность построения распределённой системы.

Однако не все утверждения сохраняют актуальность после того, как были объявлены. Ошибки могут стать причиной того, что ранее заявленное утверждение будет помечено, как недействительное. Этот процесс называется «аннулированием утверждения,» и для этого используется зарезервированное действие "http://adlnet.gov/expapi/verbs/voided". Никакое утверждение, выполняющее действие "voided", не может быть аннулировано само.

Требования

- При объявлении утверждения, аннулирующего другое, его объект ДОЛЖЕН иметь значение поля objectType, равное "StatementRef".
- При объявлении утверждения, аннулирующего другое, его объект ДОЛЖЕН содержать идентификатор аннулируемого утверждения в своём поле id.
- При получении утверждения с аннулированием LRS СЛЕДУЕТ отклонить запрос, содержащий аннулирующее утверждение, со статусом HTTP 403 "Forbidden" (запрещено), если запрос поступил не из источника, авторизованного на выполнение аннулирования.
- При получении утверждения с аннулированием LRS СЛЕДУЕТ вернуть ошибку, если аннулируемое утверждение не найдено.

- При получении утверждения с аннулированием LRS МОЖЕТ отменить любые изменения в описании задачи или агента, связанные с аннулированным утверждением.
- Провайдеру задач, желающему восстановить ранее аннулированное утверждение, СЛЕДУЕТ создать его вновь под новым идентификатором.
- Системе отчётов НЕ СЛЕДУЕТ по умолчанию отображать аннулированные или аннулирующие утверждения.

Примечание: О создании ссылок на другие утверждения смотрите «Ссылка на утверждение» в разделе 4.1.4.3 ObjectType имеет значение "Statement". О поведении аннулированных утверждений при обращении к ним смотрите StatementRef в разделе 7.2 Statement API.

Пример

В данном примере приведено утверждение, аннулирующее предыдущее с id "e05aa883-acaf-40ad-bf54-02c8ce485fb0".

```
{
    "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"
    }
}
```

4.4 Подписанные утверждения

Описание

Утверждение может включать электронную цифровую подпись, чтобы гарантировать стойкое и длительное подтверждение его подлинности и сохранности.

Пояснение

Некоторые утверждения могут иметь административное или правовое значение – им необходима надёжная и длительная гарантия их подлинности и сохранности. Может возникнуть необходимость подтвердить подлинность этих утверждений, не полагаясь на систему, посредством которой они были сохранены, а, возможно, и при отсутствии доступа к этой системе. Электронная цифровая подпись предполагает наличие третьей стороны для валидации таких утверждений.

Детали

Подписанные утверждения включают подпись в формате JSON (JSON web signature, JWS) в качестве вложения. Это позволяет передавать подпись совместно с оригинальными данными утверждения. В целях совместимости в JWS используется алгоритм «RSA + SHA», а для идентифицируемости подписывающей стороны СЛЕДУЕТ использовать сертификат X.509.

Требования

- Подписанные утверждения ДОЛЖНЫ включать JWS, как описано в документе http://tools.ietf.org/html/draft-ietf-jose-json-web-signature, в виде вложения со значениями usageType "http://adlnet.gov/expapi/attachments/signature" и contentType "application/octet-stream".
- JWS-подпись ДОЛЖНА содержать сериализацию валидного JSON-представления утверждения, полученную до добавления подписи.
- JWS-подпись ДОЛЖНА использовать алгоритм "RS256", "RS384" или "RS512".
- JWS-подпись СЛЕДУЕТ создавать с использованием закрытого ключа, привязанного к X.509-сертификату.
- Если для подписи утверждения использовался сертификат X.509, то в заголовок JWS СЛЕДУЕТ включать свойство x5c, содержащее соответствующую цепочку сертификатов.
- LRS ДОЛЖНО отклонять запросы на сохранение утверждений с некорректной подписью, используя код HTTP 400.
- LRS СЛЕДУЕТ вставлять сообщение в ответ на отклонённое утверждение.
- Чтобы убедиться в корректности подписей, LRS ДОЛЖНО проделать следующее:
    - Декодировать JWS-подпись и загрузить подписанную копию утверждения из тела JWS-подписи.
    - Удостовериться, что исходное утверждение логически эквивалентно содержимому подписанной копии.

        - При проведении сравнения различия, которые могут быть внесены при обработке утверждений на стороне LRS (поля id, authority, stored, timestamp и version), ДОЛЖНЫ игнорироваться.
    - Если заголовок JWS содержит X.509 сертификат, то проверить подпись по сертификату, как указано в JWS.
- Клиенты НЕ ДОЛЖНЫ предполагать, что подпись корректна, исходя только из того, что LRS её приняло.

Примечание: Этап валидации по включённому X.509 сертификату является средством для обнаружения ошибок в подписи, а не мерой защиты. Действия, необходимые для аутентификации подписанного утверждения, различаются в зависимости от требуемого уровня достоверности и выходят за рамки спецификации.

Пример

Смотрите Приложение E: Пример подписанного утверждения.

5.0 Смешанные типы

5.1 Документ

Описание

Для провайдеров задач Experience API обеспечивает возможность сохранения произвольных данных в виде документов, которые могут быть связаны с задачей, агентом или их комбинацией.

Детали

Обратите внимание, что в следующей таблице приведены общие свойства документа не как объекта JSON. Идентификатор записан в IRL, updated – информация HTTP -заголовка, contents – сам HTTP-документ.

Свойство	Тип	Описание
id	String	Устанавливается провайдером задач, уникальный в пределах агента или задачи.
updated	Timestamp	Дата последнего обновления документа.
contents	Произвольные двоичные данные	Содержание документа.

5.2 Словарь языков

Описание

Словарь языков – это таблица, в каждой записи которой ключом является идентификатор языка, определённый в RFC 5646, а значением – строка на этом языке. Эта соответствие должно содержать как можно больше языков и значений.

5.3 Расширения

Описание

Расширения используются как часть определения задачи, контекста утверждения или результата утверждения. В каждом случае расширение позволяет естественным способом адаптировать эти элементы к конкретному варианту использования. Расширения могут иметь значения только для одного конкретного приложения или стать правилом, используемым всем профессиональным сообществом.

Детали

Расширения определяются парой (ключ, значение) и логически связаны с частью утверждения, где они присутствуют. В качестве значения расширения могут выступать любые данные с произвольной структурой. Расширения в контексте утверждения обеспечивают контекст учебным достижениям, в то время как в результатах утверждения содержат элементы, связанные с некоторым результатом обучения. В учебных задачах расширения предоставляют дополнительную информацию, которая помогает определить задачу в рамках некоторого пользовательского приложения или сообщества. Смысл и структура расширения, соответствующего ключу IRI, определяются лицом, которое администрирует IRI.

Требования

- Ключом расширения ДОЛЖЕН быть IRI.
- LRS НЕ ДОЛЖНО отклонять утверждения на основе расширений.
- Клиенту СЛЕДУЕТ всегда стремиться отобразить как можно больше информации во встроенных элементах, для того чтобы использовать интероперабельность инструментов, совместимых со спецификацией Experience API.
- Всем IRI расширений СЛЕДУЕТ иметь администратора.
- Администратору IRL ключа расширения СЛЕДУЕТ сделать человекочитаемое описание смысла расширения, доступное по IRL.

Примечание: Утверждение, определённое полностью его расширениями, становится бессмысленным, поскольку никакая другая система не может понять его.

5.4 Метаданные идентификатора

Описание

Дополнительная информация может быть предоставлена в рамках утверждения об идентификаторе. Это позволяет получить метаданным о IRI без необходимости его анализа.

Детали

Есть несколько типов идентификаторов IRI, используемых в этой спецификации:

- идентификатор действия;
- идентификатор задачи;
- тип задачи;

- ключ расширения;
- назначение вложения.

По вопросам построения метаданных об идентификаторах задач смотрите информацию в разделе 4.1.4.1, посвящённом объекту с определением задачи.

Метаданные всех остальных идентификаторов строятся по следующему формату:

Свойство	**Тип**	**Описание**	**Примечание**
name	Словарь языков	Удобочитаемое имя	Опциональное
description	Словарь языков	Описание	Опциональное

Если эти метаданные построены, как описано выше, то это – канонический источник информации об идентификаторе, который они описывают. Так же, как с действиями, мы рекомендуем, чтобы провайдеры задач искали и использовали уже опубликованные, широко распространённые идентификаторы для всех типов идентификаторов IRI, кроме идентификатора задачи.

Требования

- Метаданные МОГУТ быть снабжены идентификатором.
- Если метаданные присутствуют, то в них СЛЕДУЕТ включить и имя, и описание.
- Для любого из перечисленных выше идентификаторов, если IRI – IRL, созданный для использования с этой спецификацией, то администратору этого IRL СЛЕДУЕТ сделать эти метаданные доступными в формате JSON по этому IRL, когда в запросе указан тип Content-Type "application/json".
- Когда идентификатор уже существует, провайдеру задач СЛЕДУЕТ использовать соответствующий существующий идентификатор.
- Провайдер задач МОЖЕТ создать и использовать свои собственные действия, когда подходящего идентификатора ещё не существует.
- Другие источники информации МОГУТ использоваться, чтобы восполнить потерянные детали, например переводы, или полностью заменить метаданные, если они не были предоставлены или к ним нет доступа. Сюда МОГУТ входить метаданные в других форматах, сохранённые в IRL идентификатора, особенно если этот идентификатор не был предназначен для использования с этой спецификацией.

6.0 Сообщения во время выполнения

Разделы 6 и 7 описывают техническую сторону Experience API, имея дело с тем, как утверждения передаются между провайдером задач и LRS. Ряд библиотек находятся в стадии разработки для ряда технологий (в том числе JavaScript), с помощью которых будет реализована эта часть спецификации. Поэтому от разработчиков содержимого не требуется полного понимания каждой детали этой части спецификации.

6.1 Кодирование

Требования

- Все строки ДОЛЖНЫ кодироваться и интерпретироваться как UTF-8.

6.2 Нумерация версий API

Пояснение

Будущие версии этой спецификации могут содержать изменения, такие как новые свойства утверждений.

Системы, извлекающие утверждения, могут получать ответы, которые будут содержать утверждения различных версий. Заголовок version позволяет обработать эту ситуацию правильно и установить, существуют ли частичные или смешанные реализации версий в LRS.

Использование семантической нумерации версий позволит клиентам и LRS-хранилищам анализировать совместимость версий.

Детали

Начиная с 1.0.0, в xAPI версии будут нумероваться согласно Semantic Versioning 1.0.0. Каждый запрос от клиента и каждый ответ от LRS включают заголовок HTTP с именем "X-Experience-API-Version" и номером версии как значение. Например, "X-Experience-API-Version: 1.0.1".

Требования к LRS

- LRS ДОЛЖНО включать заголовок "X-Experience-API-Version" в каждый ответ.
- LRS ДОЛЖНО присваивать этому заголовку значение "1.0.1".

- LRS ДОЛЖНО принимать запросы с версией в заголовке "1.0" как "1.0.0".
- LRS ДОЛЖНО отклонять запросы с версией в заголовке до "1.0.0", кроме случаев, когда запросы перенаправляются на полностью совместимую реализацию предыдущей версии, указанной в заголовке.
- LRS ДОЛЖНО отклонять запросы с версией в заголовке "1.1.0" или выше.
- LRS ДОЛЖНО отклонять запросы, отвечая ошибкой HTTP 400 и кратким описанием проблемы.

Требования к клиенту

- Клиент ДОЛЖЕН включать заголовок "X-Experience-API-Version" в каждый запрос.
- Клиент ДОЛЖЕН присваивать этому заголовку значение "1.0.1".
- Клиенту СЛЕДУЕТ допускать ответы с версией "1.0.0" или старше.
- Клиенту СЛЕДУЕТ допускать получение структур данных с дополнительными свойствами.
- Клиенту СЛЕДУЕТ игнорировать любые свойства, не определённые в версии 1.0.0 спецификации.

Требования к преобразованиям между версиями

- Системы НЕ ДОЛЖНЫ преобразовывать утверждения более новых версий в формат предыдущих версий, например, чтобы преодолеть различия в версиях.
- Системы МОГУТ преобразовать утверждения более старых версий в более новую версию только методами, описанными в Приложении D: Преобразование утверждений к версии 1.0.0.

6.3 Параллелизм

Описание

Управление параллелизмом гарантирует, что приложение, используя метод PUT, не сохранит в LRS изменения, основанные на старых данных.

Детали

xAPI будет использовать теги сущностей (ETags) в HTTP 1.1 для управления параллелизмом в различных частях API, где метод PUT может перезаписать существующие данные:

- State API,
- Agent Profile API,
- Activity Profile API.

State API допускает запросы методом PUT без заголовков параллелизма, так как конфликты состояний маловероятны. Требования ниже применяются только к Agent Profile API и Activity Profile API.

Требования к клиенту

- Клиент, используя либо Agent Profile API или Activity Profile API ДОЛЖЕН включить в запрос заголовок If-Match или заголовок If-None-Match.

Требования к LRS

- В ответе на GET-запрос LRS ДОЛЖНО добавить заголовок ETag. Формат заголовка должен быть таким, чтобы приложения, которые используют API, но не могут прочитать заголовок ETag, смогли вычислить его сами.
- В ответе на GET-запрос LRS ДОЛЖНО вычислить значение этого заголовка – шестнадцатеричная строка, содержащая дайджест содержимого по алгоритму SHA-1.
- В ответе на GET-запрос LRS ДОЛЖНО вложить заголовок в кавычки.
- В ответе на PUT-запрос, если он содержит ETag, LRS ДОЛЖНО обрабатывать заголовок If-Match, как описано в RFC2616 HTTP 1.1, с целью поиска изменений, внесённых после получения данных клиентом, сделавшим запись.
- В ответе на PUT-запрос, если он содержит "*", LRS ДОЛЖНО обрабатывать заголовок If-None-Match, как описано в RFC2616 HTTP 1.1, с целью поиска ресурса, о котором не знает приложение.

Если предусловие заголовка в любом из PUT-запросов, перечисленных выше, не выполняется, то LRS:

- ДОЛЖНО вернуть HTTP-статус 412 "Precondition Failed".
- НЕ ДОЛЖНО выполнять изменений ресурса.

Если PUT-запрос получен без какого-либо заголовка для ресурса, который уже существует, то LRS:

- ДОЛЖНО вернуть HTTP-статус 409 "Conflict".

- ДОЛЖНО вернуть текст, который объясняет, что клиенту СЛЕДУЕТ делать в этой ситуации:
    - проверить текущее состояние ресурса,
    - установить заголовок If-Match с текущим ETag для разрешения конфликта.
- НЕ ДОЛЖНО выполнять изменений ресурса.

6.4 Безопасность

Пояснение

Чтобы сбалансировать интероперабельность и требования по безопасности в различных средах, было определено несколько вариантов аутентификации.

Детали

Матрица, которая приведена ниже, описывает возможные сценарии аутентификации.

Зарегистрированное приложение – это приложение, которое будет проходить аутентификацию в LRS в качестве OAuth-приложения (OAuth consumer) и было зарегистрировано совместно с LRS.

Зарегистрированный пользователь – учётная запись пользователя в LRS или в системе, которой LRS доверяет процедуру идентификации пользователей.

	Зарегистрированный пользователь	**Анонимный пользователь**
Зарегистрированное приложение	Стандартные процедуры протокола OAuth.	LRS доверяет приложению доступ к xAPI без проверки пользователя. Процедуры протокола OAuth не вызываются.
Незарегистрированное приложение	Приложение агента не идентифицировано как зарегистрированный агент, и LRS не может идентифицировать его.	
Без приложения	Вместо Oauth используется базовая HTTP-аутентификация, так как никакое приложение не указано.	

Без аутентификации	МОЖЕТ поддерживаться LRS, возможно, в целях тестирования.

Требования

LRS ДОЛЖНО поддержать аутентификацию, используя хотя бы один из следующих методов:

- OAuth 1.0 (RFC 5849) с методами цифровой подписи "HMAC-SHA1", "RSA-SHA1", "PLAINTEXT".
- Базовая аутентификация HTTP.
- Общие карты доступа (детали реализации в более поздней версии)
- LRS ДОЛЖНО принимать или делегировать решение о валидности утверждений и определять, какие операции могут быть выполнены на основе используемых удостоверений.

6.4.1 Описание сценариев

Требования

- LRS ДОЛЖНО записывать имя приложения и его уникальный ключ (идентификатор).
- LRS ДОЛЖНО поддерживать механизм для выполнения регистрации или передать полномочия другой системе, которая обеспечивает такой механизм.
- LRS ДОЛЖНО допускать полную поддержку xAPI:
  - с любым методом, описанным ниже,
  - по любому сценарию, описанному ниже.
- LRS МОЖЕТ (по соображениям безопасности):
  - поддерживать только подмножество методов, описанных ниже,
  - ограничить зарегистрированных пользователей и приложения.
- LRS СЛЕДУЕТ как минимум поддерживать протокол OAuth с алгоритмами цифровой подписи "HMAC-SHA1" и "RSA-SHA1".

Зарегистрированное приложение и пользователь. Описание процесса и требования

- Используйте узлы из раздела 6.4.2 Уровни OAuth-авторизации для реализации стандартной процедуры протокола OAuth (в данной спецификации детали не приведены).
- Если эта форма аутентификации используется для записи утверждения и не указано свойство authority, LRS должно записать в качестве значения этого

свойства группу, которая состоит из объекта Agent, представляющего зарегистрированное приложение, и объекта Agent, представляющего зарегистрированного пользователя.

Зарегистрированное приложение и анонимный пользователь. Описание процесса и требования

- LRS принимает запросы, подписанные с помощью OAuth, с удостоверениями зарегистрированных приложений, с пустым токеном и секретным токеном.
- Если эта форма аутентификации используется для записи утверждения и не указано свойство authority, LRS должно записать в качестве значения этого свойства объект Agent, представляющий зарегистрированное приложение.

Незарегистрированное приложение и зарегистрированный пользователь. Описание процесса и требования

- Возьмите пустой ключ приложения.
- Отправьте запрос «Temporary Credential».
- Определите поле consumer_name и другие стандартные параметры; пользователь увидит значение consumer_name и предупреждение о том, что подлинность приложения не может быть установлена.
- LRS ДОЛЖНО записать в качестве значения свойства authority группу, которая состоит из приложения и аутентифицируемого пользователя, поскольку OAuth определяет приложение.

Нет приложения и зарегистрированный пользователь. Описание процесса и требования

- Используйте комбинацию имени пользователя и пароля, которая соответствуют учётной записи в LRS.
- Значением свойства "authority" становится объект Agent, который идентифицируется логином, если не …
    - определено другое значение свойства authority и …
    - LRS доверяет зарегистрированному пользователю определять значение этого свойства.

Без авторизации. Описание процесса и требования

- Запросы должны включать заголовки базовой аутентификации HTTP без указания имени и пароля, чтобы отличить явно незаверенный запрос от запроса, которому следует предоставить базовую аутентификацию HTTP.

6.4.2 Уровни OAuth-авторизации

Описание

Эти рекомендации определяют уровни доступа (scope), которые должны позволить LRS и взаимодействующему приложению, используя xAPI, договориться о наборе разрешений, требуемых приложению, минимизируя опасность неправильного использования. Ограничения каждого уровня доступа накладываются на ограничения безопасности, указанные в учётной записи пользователя, которая связана с запросом.

Детали

В следующей таблице перечислены уровни xAPI:

Уровень	**Разрешения**
statements/write	Запись любого утверждения
statements/read/ mine	Чтение утверждений, написанных «мной», т.е. значение свойства authority совпадает с тем, которое LRS назначило бы, когда записывало утверждение с текущим токеном.
statements/read	Чтение любого утверждения
state	Чтение и запись данных о состоянии, ограниченные задачами и актёрами, связанными с текущим токеном до той степени, до которой возможно определить эти связи.
define	Определение или переопределение задач и актёров. При сохранении утверждения, когда это не разрешено, идентификаторы будут сохранены, и LRS сможет сохранить само утверждение в целях аудита, но не может обновлять своё внутреннее представление, менять актёров или задачи.
profile	Чтение и запись данных профиля, ограниченные задачами и актёрами, связанными с текущим токеном до той степени, до которой возможно определить эти связи.
all/read	Неограниченный доступ на чтение.
all	Неограниченный доступ.

Расширенные параметры OAuth

Обратите внимание на то, что параметры consumer_name и scope не являются частью OAuth 1.0, и поэтому, если они используются, то должны быть переданы как строка запроса или параметры формы, но не в заголовке OAuth.

Узлы OAuth

Имя	Узел	Пример
Запрос временного удостоверения (Temporary Credential Request)	OAuth/initiate	http://example.com/xAPI/OAuth/initiate
Авторизация владельца ресурса (Resource Owner Authorization)	OAuth/authorize	http://example.com/xAPI/OAuth/authorize
Запрос токена для доступа (Token Request)	OAuth/token	http://example.com/xAPI/OAuth/token

Пример

Список уровней доступа определяет набор разрешений, которые требуются. Например, преподаватель мог бы предоставить уровень "statements/read" сервису для построения отчётов, но LRS ограничит этот инструмент утверждениями, которые может читать сам преподаватель, если будет запрашивать их непосредственно у LRS.

Требования

- LRS ДОЛЖНО принимать параметр scope, как определено в OAuth 2.0.
- LRS ДОЛЖНО назначать уровни "statements/write" и "statements/read/mine", если уровень не указан.
- LRS ДОЛЖНО поддерживать как минимум уровень "all".
- LRS МОЖЕТ поддерживать другие уровни доступа.
- Клиенту СЛЕДУЕТ запрашивать только минимально необходимые уровни доступа, чтобы увеличить вероятность получения доступа.

7.0 Передача данных (REST)

Описание

Эта секция описывает четыре программных интерфейса xAPI: Statement API, State API, Agent Profile API и Activity Profile API. Эти четыре интерфейса Experience API обрабатываются с помощью методов RESTful HTTP. Statement API может использоваться отдельно, чтобы отслеживать записи о приобретённом опыте.

Примечание: Во всех примерах узлов данной спецификации "http://example.com/xAPI/" является базовым IRI для LRS. Все остальные IRI, построенные с использованием базового IRI, представляют конкретные используемые узлы.

Требования

- LRS ДОЛЖНО отклонить со статусом "HTTP 400 Bad Request" любой запрос к любому из API, содержащий параметры, которые не описаны в спецификации и которые LRS не может распознать. Примечание: LRS-хранилище может распознавать и обрабатывать параметры, которые не описаны в этой спецификации.
- LRS ДОЛЖНО отклонить со статусом "HTTP 400 Bad Request" любой запрос к любому из API, содержащий параметры, которые совпадают с параметрами описанными в спецификации во всём, кроме регистра.
- LRS ДОЛЖНО отклонить пакет утверждений, если отклонено хотя бы одно утверждение.

7.1 Коды ошибок

Описание

Список ниже предлагает некоторое общее руководство по кодам ошибок HTTP, которые могут быть возвращены из различных методов API.

Детали

- 400 Bad Request – Указывает на ошибку, вызванную неправильным или недостающим аргументом. Термин "неправильный аргумент" включает неправильно сформированный JSON и ошибочную структуру объекта.
- 401 Unauthorized – Указывает, что требуется аутентификация, а если аутентификация была отправлена в запросе, то сообщает, что удостоверение отклонено.

- 403 Forbidden – Указывает, что запрос не разрешён для данного удостоверения. Обратите внимание на то, что это отличается от ситуации, когда отклоняется удостоверение. В этом случае удостоверение было признано действительным, но аутентифицированному клиенту не разрешают выполнить данное действие.
- 404 Not Found – Указывает, что требуемый ресурс не был найден. Может быть возвращён любым методом, который возвращает однозначно идентифицируемый ресурс, например, в State API, Agent Profile API и Activity Profile API вызов процедуры позиционирования на определённый документ, метод извлечения одного утверждения.
- 409 Conflict – Указывает на ошибку из-за конфликта с текущим состоянием ресурса в случае вызовов State API, Agent Profile API и Activity Profile API, или вызовах PUT и POST для утверждений. Дополнительную информацию смотрите в разделе 6.3 Параллелизм.
- 412 Precondition Failed – Указывает на ошибку из-за невыполнения предварительного условия, отправленного с запросом, в случае вызовов State API, Agent Profile API и Activity Profile API. Дополнительную информацию смотрите в разделе 6.3 Параллелизм.
- 413 Request Entity Too Large – Указывает, что LRS отклонило утверждение или документ, потому что его размер больше, чем максимум, установленный LRS. LRS МОЖЕТ выбрать любой предел и может изменить этот предел из любых соображений, но ДОЛЖНО быть в состоянии принять утверждение любого размера при выставлении соответствующих настроек.
- 500 Internal Server Error – Указывают на общее состояние ошибки, как правило, неожиданное исключение в процессе обработки информации на сервере.

Требования

- LRS ДОЛЖНО возвратить код ошибки, наиболее соответствующий причине ошибки из списка выше.
- LRS СЛЕДУЕТ возвратить сообщение в ответе, объясняя причину ошибки.

7.2 Statement API

Описание

Базовый механизм взаимодействия Experience API.

7.2.1 PUT Statements

Детали

Пример обращения: http://example.com/xAPI/statements

Сохраняет объект Statement с заданным id.

Возвращает: 204 No Content

Параметр	Тип	По умолчанию	Описание	Примечание
statementId	String		Id объекта Statement, который должен быть записан в хранилище.	Обязательный

Требования

- LRS НЕ ДОЛЖНО изменять своё состояние при получении объекта Statement со statementID, который уже есть в LRS. Вне зависимости от того, какой ответ даёт LRS: 409 Conflict или 204 No Content, – LRS НЕ ДОЛЖНО модифицировать объект Statement или любой другой объект.
- Если LRS получает объект Statement с id, который уже используется, то LRS СЛЕДУЕТ проверить, что полученный объект соответствует существующему, и вернуть 409 Conflict, если они не совпадают.
- LRS МОЖЕТ ответить раньше, чем утверждения, которые были сохранены, будут доступны для поиска.

7.2.2 POST Statements

Детали

Пример обращения: http://example.com/xAPI/statements

Сохраняет объект Statement или набор объектов. Так как PUT-метод предназначается для сохранения объекта Statement с определённым идентификатором, POST должен использоваться вместо PUT, чтобы сохранить группу объектов типа Statement или сохранить один объект Statement без предварительного указания идентификатора. Альтернатива для систем, которые производят большое количество объектов типа Statement, заключается в предоставлении аналогичного API на стороне провайдера задач, чтобы LRS имело

возможность периодически проводить обновление или добавление объектов типа Statement списком. Например, ежедневный сбор утверждений межвузовской LRS с университетских LMS. Это, вероятно, единственная реальная возможность для систем, которые генерируют много данных.

Возвращает: 200 OK, идентификаторы сохранённых объектов типа Statement (UUID).

Требования

- LRS НЕ ДОЛЖНО изменять свое состояние при получении объекта Statement с statementID, который уже есть в LRS. Вне зависимости от того, какой ответ даёт LRS: 409 Conflict или 200 OK, – LRS НЕ ДОЛЖНО модифицировать объект Statement или любой другой объект.
- Если LRS получает объект Statement с id, который уже используется, то LRS СЛЕДУЕТ проверить, что полученный объект соответствует существующему, и вернуть 409 Conflict, если они не совпадают.
- LRS МОЖЕТ ответить раньше, чем утверждения, которые были сохранены, будут доступны для поиска.
- GET Statements МОЖЕТ быть вызван методом POST с помощью полей формы, если необходимо преодолеть ограничение на длину строки запроса.
- LRS ДОЛЖНО различать запросы POST на добавление новых объектов и просмотр сохранённых объектов на основе анализа параметров запроса.

7.2.3 GET Statements

Детали

Пример обращения: http://example.com/xAPI/statements

Этот метод можно вызвать, чтобы извлечь один или множество объектов типа Statement. Если задан параметр statementId или voidedStatementId, то возвращается единственный объект типа Statement.

Иначе возвращает: Объект StatementResult, список объектов типа Statement в обратном хронологическом порядке, основанном на времени сохранения, согласно разрешениям и ограничению длины списка. Если дополнительные результаты будут доступны, то их IRL будет включён в объект StatementResult, для того чтобы существовала возможность их извлечения.

Возвращает: 200 OK, объект Statement или StatementResult (см. раздел 4.2)

Параметр	Тип	По умол-чанию	Описание
statementId	String		Id объекта Statement, который должен быть извлечён из хранилища.
voided StatementId	String		Id аннулированного объекта Statement (см. Аннулирование утверждений).
agent	Объект Agent или зарегистри-рованная группа объектов (JSON)		Фильтр, возвращает только те объекты типа Statement, для которых указанный объект Agent или Group является актёром или объектом утверждения. • Объект Agent или Identified Group равны, когда один и тот же обратный функциональный идентификатор используется в каждом объекте, и обратные функциональные идентификаторы имеют одинаковые значения. • В целях этого фильтра группы, у которых есть участники, соответствующие по обратному функциональному идентификатору указанному объекту Agent, считают совпадающими. См. определение объектов Agent, Group.
verb	Идентифи-катор объекта Verb (IRI)		Фильтр, возвращает только те объекты типа Statement, которые имеют указанный идентификатор объекта Verb.
activity	Идентифи-катор объекта Activity (IRI)		Фильтр, возвращает только те объекты типа Statement, которые имеют указанный идентификатор объекта Activity.
registration	UUID		Фильтр, возвращает только те объекты типа Statement, которые имеют указанный регистрационный идентификатор. Обратите внимание на то, что, хотя часто уникальный регистрационный идентификатор будет использоваться для одного актёра, назначенного на одну задачу, могут быть исключения. Если требуется утверждение только для определённого актёра или задачи, то они также должны быть указаны в запросе.

related_ activities	Boolean	False	Применяет activity-фильтр более широко. Включает утверждения, для которых объект, любая из задач контекста или любое из этих свойств в подутверждении соответствуют параметру activity.
related_ agents	Boolean	False	Применяет agent-фильтр более широко. Включает утверждения, для которых актёр, объект, заверитель, преподаватель, команда или любое из этих свойств в подутверждении соответствуют параметру agent.
since	Timestamp		Только утверждения, сохраненные после указанной даты, будут возвращены (исключая указанную дату).
until	Timestamp		Только утверждения, сохраненные до указанной даты, будут возвращены.
limit	Nonnegative Integer	0	Максимальное количество утверждений, которое будет возвращено. Значение "0" указывает на то, что максимум определяется возможностями сервера.
format	String: ("ids", "exact", "canonical")	exact	Если выбран формат "ids", для объектов Agent, Activity и Group возвращаются только данные, необходимые для идентификации. У анонимных групп возвращаются участники с аналогичными данными. Если выбран формат "exact", утверждение возвращается так, как было сохранено. Если выбран формат "canonical", только один язык следует возвращать из каждого словаря. Для того чтобы выбрать наиболее релевантный язык, LRS будет использовать заголовок Accept-Language, как определено в RFC 2616 (HTTP 1.1). Эта логика будет применена к каждому словарю в отдельности, а не к ресурсу (списку утверждений) в целом.
attachments	Boolean	False	Если "true", LRS использует multipart-формат ответа и включает все вложения, как описано выше. Если "false", LRS отправляет ответ с заголовком "Content-Type application/json" и не может использовать вложения.

ascending	Boolean	False	Если "true", результаты возвращаются в порядке возрастания времени сохранения.

Примечание

Все параметры являются опциональными.

Требования

- LRS ДОЛЖНО отклонить с ошибкой HTTP 400 любой запрос, содержащий одновременно параметры statementId и voidedStatementId.
- LRS ДОЛЖНО отклонить с ошибкой HTTP 400 любой запрос, содержащий параметр statementId или voidedStatementId, а также любой другой параметр, кроме attachments или format.
- LRS ДОЛЖНО включать заголовок "X-Experience-API-Consistent-Through", в ISO 8601 (объединённый формат даты и времени), во все ответы на запросы утверждений со значением предполагаемой даты сохранения утверждений, если оно не сохранено с однозначно определённой датой. В этом времени СЛЕДУЕТ учитывать любой фактор, вызывающий временную задержку, например, чрезмерная нагрузка на систему.
- Если параметр attachments GET-запроса используется и имеет значение "true", то LRS ДОЛЖНО использовать формат ответа "multipart" и включить в ответ все вложения, как описано в 4.1.11.
- Если параметр attachments GET-запроса используется и имеет значение "false", то LRS НЕ ДОЛЖНО включать в ответ вложения и ДОЛЖНО установить тип содержания "application/json".

Правила фильтрации объектов типа StatementRef

Для всех параметров запроса, кроме since, until и limit, объекты типа Statement, которые указывают на другой объект Statement посредством свойства StatementRef, будут попадать под действие условий фильтра, если целевой объект попадает под условия этого фильтра. Правило действует рекурсивно, поэтому «Утверждение a», указывающее на «Утверждение b», которое в свою очередь указывает на «Утверждение c», будет включено в результат запроса, если «Утверждение c» удовлетворяет условиям фильтра. Несмотря на это, условия, выраженные параметрами since, until и limit, будут применяться непосредственно к утверждению, содержащему StatementRef.

Например, рассмотрим утверждение «Ben passed explosives training» и утверждение «Andrew confirmed», которое ссылается на первое. Во втором утверждение нет фразы "Ben" или "explosives training", однако когда выполняются

запросы с Actor-фильтром "Ben" или Activity-фильтром "explosives training", то оба утверждения будут включены в результат, если они удовлетворяют условиям, которые выражены параметрами since, until и limit.

Правило, описанное в этом разделе, не применяется, когда утверждения извлекается с помощью параметров statementId или voidedStatementId.

Примечание: Ссылка на объект Statement, расположенная в поле statement контекста утверждения, не влияет на результаты фильтрации.

7.2.4 Аннулированные утверждения

Требования

- LRS НЕ ДОЛЖНО возвращать утверждение, которое было аннулировано, если в запросе не указано значение voidedStatementId.
- LRS ДОЛЖНО возвращать любые утверждения, указывающие на аннулированные утверждения, когда они извлекаются, используя параметры since, until и limit, до тех пор, пока они сами не будут аннулированы (см. раздел, посвященный правилам фильтрации объектов типа StatementRef). Действие этого требования распространяется на аннулирующие утверждения, которые сами не могут быть аннулированы. Инструменты для построения отчётов могут идентифицировать наличие и statementId аннулированных утверждений по ссылке, указанной в аннулирующем утверждении.
- Инструментам для построения отчётов, желающим извлечь аннулированные утверждения, СЛЕДУЕТ запрашивать их индивидуально, используя voidedStatementId.

7.3 Document APIs

Три программных интерфейса обеспечивают хранение документов для провайдеров учебных задач и агентов. Детали каждого API можно найти в следующих разделах, а информация в этом разделе относится ко всем трём API.

Детали

API	Метод	Узел	Пример
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	agents/profile	http://example.com/xAPI/agents/profile

Требования

- Провайдер задач МОЖЕТ отправлять документы в любой из интерфейсов для задач и агентов, о которых LRS ещё не знает.
- LRS НЕ ДОЛЖНО отклонять документ из-за отсутствия предварительных знаний о задаче или агенте.

JSON-процедура и требования

Если провайдер задач хранит переменные как объекты JSON в документе с типом контента "application/json", то он может манипулировать ими, используя метод POST.

Следующий процесс идёт посредством метода POST и его требований. Например, документ содержит:

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

Когда LRS получает запрос POST с типом контента "application/json" в отношении существующего в хранилище документа такого же типа, оно ДОЛЖНО объединить присланный документ с существующим документом. В этом контексте слияние документов определяется так:

- десериализуйте объекты каждого документа;
- для каждого свойства, непосредственно определённого в присланном объекте, создайте соответствующее свойство у существующего объекта и сделайте его равным по значению свойству в присланном объекте;
- сохраните любую валидную JSON-сериализацию существующего объекта как документ, на который ссылался запрос.

Обратите внимание, что объединяются только свойства верхнего уровня, даже если свойство верхнего уровня является объектом. Все содержание каждого исходного свойства заменяется на все содержимое каждого присланного свойства.

Например, этот документ отправлен запросом POST и имеет идентификатор существующего в хранилище документа, описанного выше:

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

В результате в LRS будет сохранен документ:

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

Если оригинал документа существует, оригинал документа или присланный документ не имеют Content-Type: "application/json", или какой-либо документ не может быть разбит на JSON-объекты, LRS ДОЛЖНО ответить HTTP-сообщением с кодом 400 Bad Request, и НЕ ДОЛЖНО обновлять существующий документ.

Если оригинал документа не существует, то LRS ДОЛЖНО рассматривать запрос как PUT-запрос на сохранение отправленного документа.

Если слияние документов прошло успешно, то LRS ДОЛЖНО ответить HTTP-сообщением с кодом 204 No Content.

Если провайдеру задач требуется удалить свойство, ему СЛЕДУЕТ использовать PUT-запрос для замены всего документа, как описано ниже.

7.4 State API

Описание

Как правило, этот интерфейс используется для организации временного хранилища провайдеров задач, которые не имеют своих хранилищ или должны сохранять состояние приложения при переходе с одного на другое устройство.

Детали

Семантика вызова строится вокруг параметра stateId. Если он включен в вызов, то методы GET и DELETE будут воздействовать на одно заданное состояние документа, помеченное как "stateId". В противном случае GET вернёт доступные идентификаторы, а DELETE удалит все состояния в контексте значений других параметров.

Single Document (PUT | POST | GET | DELETE)

Пример обращения: http://example.com/xAPI/activities/state

Сохраняет, извлекает или удаляет документ, определённый заданным stateId, который существует в контексте указанной задачи, агента или регистрации (если указано).

Возвращает (PUT | POST | DELETE): 204 No Content
Возвращает (GET): 200 OK, состояние контента

Параметр	Тип	Описание	Примечание
activityId	String	Задача, связанная с указанным состоянием.	Обязательный
agent	JSON	Агент, связанный с указанным состоянием.	Обязательный
registration	UUID	Регистрация, связанная с указанным состоянием.	Опциональный
stateId	String	Идентификатор состояния в заданном контексте.	Обязательный

Multiple Document GET

Пример обращения: http://example.com/xAPI/activities/state

Выбирает идентификаторы всех данных о состоянии для контекста (Задача + Агент [+ регистрация, если указана]). Если параметр since указан, то запрос ограничивается записями, которые были созданы или обновлены после указанной даты (исключая указанную дату).

Возвращает: 200 OK, массив идентификаторов

Параметр	Тип	Описание	Примечание
activityId	String	Идентификатор задачи, связанной с указанными состояниями.	Обязательный
agent	JSON	Агент, связанный с указанными состояниями.	Обязательный
registration	UUID	Идентификатор регистрации, связанной с указанными состояниями.	Опциональный
since	Timestamp	Возвращает идентификаторы состояний, сохранённых после указанной даты (исключая указанную дату).	Опциональный

Multiple Document DELETE

Пример обращения: http://example.com/xAPI/activities/state

Удаляет данные состояний в контексте (Задача + Агент [+ регистрация, если указана]).

Возвращает: 204 No Content

Параметр	Тип	Описание	Примечание
activityId	String	Идентификатор задачи, связанной с указанными состояниями.	Обязательный
agent	JSON	Агент, связанный с указанным состоянием.	Обязательный
registration	UUID	Идентификатор регистрации, связанной с указанным состоянием.	Опциональный

7.5 Activity Profile API

Описание

Activity Profile API подобен State API, этот интерфейс позволяет сохранять пару (ключ, документ), связанную с некоторой задачей.

Детали

Семантика вызова строится вокруг параметра profileId. Если он указан, то метод GET будет воздействовать на один определённый документ, идентифицируемый "profileId". Иначе метод GET вернёт доступные идентификаторы.

Activity Profile API также включает метод для извлечения из LRS полного описания задачи.

Full Activity Object GET

Пример обращения: http://example.com/xAPI/activities

Загружает указанный объект типа Activity полностью.

Возвращает: 200 OK, содержание

Параметр	Тип	Описание	Примечание
activityId	String	Идентификатор загружаемой задачи.	Обязательный

Single Document PUT | POST | GET | DELETE

Пример обращения: http://example.com/xAPI/activities/profile

Сохраняет, извлекает или удаляет указанный профиль в контексте указанной задачи.

Возвращает (PUT | POST | DELETE): 204 No Content
Возвращает (GET): 200 OK, профиль

Параметр	Тип	Описание	Примечание
activityId	String	Идентификатор задачи, связанной с профилем.	Обязательный
profileId	String	Идентификатор профиля.	Обязательный

Multiple Document GET

Пример обращения: http://example.com/xAPI/activities/profile

Загружает идентификаторы всех профилей для некоторой задачи. Если параметр since указан, то запрос ограничивается записями, которые были созданы или обновлены после указанной даты (исключая указанную дату).
Возвращает: 200 OK, список идентификаторов

Параметр	Тип	Описание	Примечание
activityId	String	Идентификатор задачи, связанной с профилями.	Обязательный
since	Timestamp	Возвращает идентификаторы профилей, сохранённых после указанной даты (исключая указанную дату).	Опциональный

7.6 Agent Profile API

Описание

Agent Profile API подобен State API, но позволяет сохранять пару (key, document), которая связана с некоторым агентом.

Детали

Семантика вызова строится вокруг параметра profileId. Если он указан, то метод GET будет воздействовать на один определённый документ, идентифицируемый "profileId". Иначе метод GET вернёт доступные идентификаторы.

Agent Profile API также включает метод для извлечения из LRS специального объекта с собранной информацией об агенте, полученной из внешних источников, таких, как каталог.

Combined Information GET

Детали

Пример обращения: http://example.com/xAPI/agents

Возвращает специальный объект типа Person для указанного агента. Объект Person подобен объекту Agent, но в отличие от последнего в качестве значений полей выступают массивы, поэтому разрешено использовать многокомпонентные идентифицирующие свойства. Обратите внимание, что аргумент – обычный объект типа Agent с одним идентификатором и без массивов в качестве значений. Это отличается от FOAF-концепции персоны, здесь персона используется для обозначения человеко-центрированного взгляда на данные агентов в LRS, объекты типа Agent ссылаются только на одного человека (некоторый человек в одном контексте).

Требования

- Для способности LRS возвращать несколько идентифицирующих свойства для объекта Person СЛЕДУЕТ проверять наличие удостоверений, явно дающих разрешения.
- LRS СЛЕДУЕТ отклонять запросы, которые имеют недостаточно привилегий, сообщением с кодом 403 "Forbidden".
- Если у LRS нет дополнительной информации об агенте, то LRS всё же ДОЛЖНО по запросу передать объект Person, однако этот объект будет включать информацию только из объекта Agent, связанного с запрашиваемым агентом.

Свойства объекта Person

Детали

Параметр	Тип	Описание	Примечание
objectType	String	"Person"	Обязательный
name	Массив значений типа String.	Список имён агентов для поиска.	Опциональный
mbox	Массив IRI в форме "mailto:email address".	Список адресов электронной почты агентов для поиска.	Опциональный
mbox_sha1sum	Массив значений типа String.	Список хеш-кодов почтовых IRI по алгоритму SHA1.	Опциональный
openid*	Массив значений типа String.	Список openids, которые однозначно определяют агента при поиске.	Опциональный
account*	Массив объектов типа Account.	Список учётных записей для поиска. Надо предоставить объект, содержащий полное описание учётной записи (свойства homePage и name, подробнее в разделе 4.1.2.4).	Опциональный

См. также раздел 4.1.2.1 Agent.

Возвращает: 200 OK, расширенный объект типа Agent

Параметр	Тип	Описание	Примечание
agent	Object (JSON)	Информация об агенте, о котором необходимо получить представление в расширенной форме.	Обязательный

Требования

Все свойства-массивы должны быть составлены из элементов с тем же определением, что и одноимённые свойства объекта Agent.

Single Agent or Profile PUT | POST | GET | DELETE

Пример обращения: http://example.com/xAPI/agents/profile

Сохраняет, извлекает или удаляет указанный профиль в контексте указанного агента.

Возвращает (PUT | POST | DELETE): 204 No Content
Возвращает (GET): 200 OK, профиль

Параметр	Тип	Описание	Примечание
agent	Object (JSON)	Агент, связанный с профилем.	Обязательный
profileId	String	Идентификатор профиля.	Обязательный

Multiple Agent or Profile GET

Пример обращения: http://example.com/xAPI/agents/profile

Загружает идентификаторы всех профилей для некоторого агента. Если параметр since указан, то запрос ограничивается записями, которые были созданы или обновлены после указанной даты (исключая указанную дату).

Возвращает: 200 OK, список идентификаторов

Параметр	Тип	Описание	Примечание
agent	Object (JSON)	Агент, связанный с профилем.	Обязательный
since	Timestamp	Возвращает идентификаторы профилей, сохранённых после указанной даты (исключая указанную дату).	Опциональный

7.7 О ресурсе

Описание

Возвращает JSON объект, содержащий информацию о данном LRS, в том числе поддерживаемую версию xAPI.

Пояснение

В первую очередь этот ресурс существует, чтобы позволить клиенту, поддерживающему несколько версий xAPI, решить, какую версию использовать

при общении с данным LRS. Расширения включены, чтобы могли появиться другие варианты использования.

Детали

Information GET

Пример обращения: http://example.com/xAPI/about

Возвращает: 200 OK, простой JSON-документ "about".

Параметр	Тип	Описание	Примечание
version	Массив строк с версиями	Версии xAPI, которые поддерживает данное LRS.	Обязательный
extensions	Object	Словарь других свойств, если необходимо.	Опциональный

Требования

- LRS ДОЛЖНО вернуть JSON-документ, описанный выше, со свойством version, в котором указываются все основные поддерживаемые версии с учётом последних дополнений и исправлений.
    - Для версии 1.0.0 данной спецификации это означает, что "1.0.0" ДОЛЖНО быть включено; "0.9" и "0.95" МОГУТ быть включены, т.к. "0.9" и "0.95" считаются основными версиями.
- LRS СЛЕДУЕТ разрешить анонимный доступ к этому ресурсу.
- LRS НЕ ДОЛЖНО отклонять запросы к этому ресурсу на основе версии заголовка, как требуется в разделе 6.2 Нумерация версий API.

7.8 Кросс-доменные запросы

Описание

Одной из целей xAPI является кросс-доменное отслеживания результатов, и хотя xAPI стремится обеспечивать отслеживание результатов из приложений, браузеры по-прежнему нуждаются в поддержке. Internet Explorer 8 и 9 не реализует Cross Origin Resource Sharing, а использует их собственный механизм кросс-доменных запросов, который не может использовать весь xAPI, как описано выше, потому что поддерживает только "GET" и "POST", не позволяя устанавливать HTTP-заголовки.

Детали/Требования

Ниже описан альтернативный синтаксис для приложений, который нужно использовать только тогда, когда нет возможности использовать обычный синтаксис для некоторых вызовов из-за упомянутых выше ограничений.

Метод: Все запросы xAPI должны быть выполнены методом POST. Подразумеваемый xAPI-метод должен быть указан как единственный параметр запроса (пример: http://example.com/xAPI/statements?method=PUT).

Заголовки: Любые необходимые параметры, которые могут появляться в заголовке HTTP, должны вместо этого быть включены как параметр формы с тем же именем.

Содержание: Если xAPI-вызов включает отправку содержания, то содержание должно быть закодировано и включено как параметр "content" формы. LRS будет интерпретировать значение этого параметра как последовательность UTF-8. Сохранение двоичных данных не поддерживается в этом синтаксисе.

Вложения: Отправка вложения требует, чтобы был отправлен запрос multipart/mixed, поэтому возможность отправки вложений не было предусмотрено в с этом синтаксисе (см. 4.1.11. Вложения).

- LRS ДОЛЖНО поддерживать описанный выше синтаксис.

Следует отметить, что версии Internet Explorer ниже 10 не поддерживают кросс-доменные запросы между HTTP и HTTPS. Это означает, что для IE9 и ниже, если LRS находится на домене HTTPS, то клиент также должен быть на домене HTTPS. Если LRS на HTTP, то клиент тоже должен быть на HTTP.

Могут быть случаи, когда в отношении клиента провайдера задач выдвигается специальное требование: поддерживать IE8 и IE9, а код клиента размещён в разных схемах (HTTP или HTTPS). В этих случаях требуется прокси для обмена сообщениями с LRS. Есть два простых решения: 1) настроить прокси на преобразование схемы клиенты в схему LRS; 2) разместить на прокси-сервере промежуточный LRS-сервер, который будет передавать данные в основное хранилище.

- LRS МОЖЕТ решить предоставлять как HTTP, так и HTTPS узлы для поддержки этого варианта использования.
- LRS и клиенту СЛЕДУЕТ проанализировать угрозы прежде, чем принять решение использовать эту схему.

7.9 Валидация

Описание

Роль LRS в пределах xAPI – сохранять и извлекать утверждения. Пока у LRS есть информация, необходимая для выполнения этих задач, ожидается, что LRS их выполнит. Валидация утверждений в Experience API сосредоточена исключительно на синтаксисе, а не на семантике. Реализация правил валидации объектов Verb, Activity и расширений – ответственность провайдера задач, отправляющего утверждение.

Требования

- LRS СЛЕДУЕТ применять правила, касающиеся структуры.
- LRS НЕ СЛЕДУЕТ применять правила, касающиеся семантики.

7.10 HTTP-заголовок

Описание

LRS будет отвечать на HEAD-запросы, возвращая только метаинформацию и используя HTTP-заголовки, а не тело документа.

Пояснение

Клиентам, которые общаются с LRS, может потребоваться установить существование конкретного утверждения, определить дату изменения документов, таких как State, Activity profile или Agent profile. Особенно для больших документов важно иметь возможность без передачи всего документа запросить в LRS и получить только интересующую информацию.

Требования к LRS

- LRS ДОЛЖНО ответить на любой HTTP HEAD-запрос, как если бы это был идентичный GET-запрос, за исключением:
    - Тело сообщения ДОЛЖНО быть пустым.
    - Заголовок Content-Length МОЖЕТ быть пропущен, для того чтобы избежать неоправданного использования ресурсов LRS.

Приложение A: Примеры утверждений

Пример простого утверждения (разрывы строк и отступы – только для наглядности):

```
{
    "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",
            "ru-RU":"создал"
        }
    },
    "object":{

"id":"http://example.adlnet.gov/xapi/example/simplestatement",
        "definition":{
            "name":{
                "en-US":"simple statement",
                "ru-RU":"простое утверждение"
            },
            "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.",
                "ru-RU":"Простое утверждение Experience API.
Обратите внимание, что LRS не требуется
                заранее иметь данные об объектах Actor (актёр,
учащийся), Verb (глагол, действие) и Object
                 (задача или другой объект утверждения)."
            }
        }
    }
}
```

Типичный результат действия «попытался»:

```
{
    "actor":{
        "objectType": "Agent",
        "name":"Example Learner",
        "mbox":"mailto:example.learner@adlnet.gov"
    },
```

```
    "verb":{
        "id":"http://adlnet.gov/expapi/verbs/attempted",
        "display":{
            "en-US":"attempted",
            "en-US":"попытался"
        }
    },
    "object":{
        "id":"http://example.adlnet.gov/xapi/example/simpleCBT",
        "definition":{
            "name":{
                "en-US":"simple CBT course",
                "ru-RU":"простой CBT-курс"
            },
            "description":{
                "en-US":"A fictitious example CBT course.",
                "ru-RU":"Вымышленный пример CBT-курса."
            }
        }
    },
    "result":{
        "score":{
            "scaled":0.95
        },
        "success":true,
        "completion":true
    }
}
```

Длинный пример содержит большинство возможных свойств. В нём приведено описание утверждения в таком виде, в котором его возвращает LRS, устанавливая свойства authority и stored.

```
{
    "id": "6690e6c9-3ef0-4ed3-8b37-7f3964730bee",
    "actor": {
        "name": "Team PB",
        "mbox": "mailto:teampb@example.com",
        "member": [
            {
                "name": "Andrew Downes",
                "account": {
                    "homePage": "http://www.example.com",
                    "name": "13936749"
                },
                "objectType": "Agent"
            },
            {
                "name": "Toby Nichols",
                "openid": "http://toby.openid.example.org/",
```

```
                "objectType": "Agent"
            },
            {
                "name": "Ena Hills",
                "mbox_sha1sum":
"ebd31e95054c018b10727ccffd2ef2ec3a016ee9",
                "objectType": "Agent"
            }
        ],
        "objectType": "Group"
    },
    "verb": {
        "id": "http://adlnet.gov/expapi/verbs/attended",
        "display": {
            "en-GB": "attended",
            "en-US": "attended",
            "ru-RU": "посетить"
        }
    },
    "result": {
        "extensions": {

"http://example.com/profiles/meetings/resultextensions/minuteslocati
on": "X:\\meetings\\minutes\\examplemeeting.one"
        },
        "success": true,
        "completion": true,
        "response": "We agreed on some example actions.",
        "duration": "PT1H0M0S"
    },
    "context": {
        "registration": "ec531277-b57b-4c15-8d91-d292c5b2b8f7",
        "contextActivities": {
            "parent": [
                {
                    "id":
"http://www.example.com/meetings/series/267",
                    "objectType": "Activity"
                }
            ],
            "category": [
                {
                    "id":
"http://www.example.com/meetings/categories/teammeeting",
                    "objectType": "Activity",
                    "definition": {
                        "name": {
                            "en": "team meeting",
                            "ru": "командный сбор"
                        },
                        "description": {
```

```
                            "en": "A category of meeting used for
regular team meetings.",
                            "ru": "Вид собрания для регулярного
сбора команды."
                        },
                        "type":
"http://example.com/expapi/activities/meetingcategory"
                    }
                }
            ],
            "other": [
                {
                    "id":
"http://www.example.com/meetings/occurances/34257",
                    "objectType": "Activity"
                },
                {
                    "id":
"http://www.example.com/meetings/occurances/3425567",
                    "objectType": "Activity"
                }
            ]
        },
        "instructor" :
        {
            "name": "Andrew Downes",
            "account": {
                "homePage": "http://www.example.com",
                "name": "13936749"
            },
            "objectType": "Agent"
        },
        "team":
        {
            "name": "Team PB",
            "mbox": "mailto:teampb@example.com",
            "objectType": "Group"
        },
        "platform" : "Example virtual meeting software",
        "language" : "tlh",
        "statement" : {
            "objectType":"StatementRef",
            "id" :"6690e6c9-3ef0-4ed3-8b37-7f3964730bee"
        }

    },
    "timestamp": "2013-05-18T05:32:34.804Z",
    "stored": "2013-05-18T05:32:34.804Z",
    "authority": {
        "account": {
            "homePage": "http://cloud.scorm.com/",
```

```
            "name": "anonymous"
        },
        "objectType": "Agent"
    },
    "version": "1.0.0",
    "object": {
        "id": "http://www.example.com/meetings/occurances/34534",
        "definition": {
            "extensions": {

"http://example.com/profiles/meetings/activitydefinitionextensions/r
oom":
                {"name": "Kilby", "id" :
"http://example.com/rooms/342"}
            },
            "name": {
                "en-GB": "example meeting",
                "en-US": "example meeting",
                "ru-RU": "пример собрания"
            },
            "description": {
                "en-GB": "An example meeting that happened on a
specific occasion with certain people present.",
                "en-US": "An example meeting that happened on a
specific occasion with certain people present.",
                "ru-RU": "Пример собрания, произошедшего по
специфической причине с обязательным
                присутствием людей."
            },
            "type": "http://adlnet.gov/expapi/activities/meeting",
            "moreInfo": "http://virtualmeeting.example.com/345256"
        },
        "objectType": "Activity"
    }
}
```

Приложение B: Примеры различных типов объектов утверждений

Activity (задача)

```
{
    "id": "http://www.example.co.uk/exampleactivity",
    "definition": {
        "name": {
            "en-GB": "example activity",
            "en-US": "example activity",
            "ru-RU": "примерная задача"
        },
        "description": {
            "en-GB": "An example of an activity",
            "en-US": "An example of an activity",
            "ru-RU": "Пример задачи"
        },
        "type": "http://www.example.co.uk/types/exampleactivitytype"
    },
    "objectType": "Activity"
}
```

Agent (человек или система)

```
{
    "name": "Andrew Downes",
    "mbox": "mailto:andrew@example.co.uk",
    "objectType": "Agent"
}
```

Group (несколько объектов типа Agent)

В данном примере описана зарегистрированная группа

```
{
    "name": "Example Group",
    "account" : {
        "homePage" : "http://example.com/homePage",
        "name" : "GroupAccount"
    },
    "objectType": "Group",
    "member": [
        {
            "name": "Andrew Downes",
            "mbox": "mailto:andrew@example.com",
            "objectType": "Agent"
        },
        {
            "name": "Aaron Silvers",
```

```
            "openid": "http://aaron.openid.example.org",
            "objectType": "Agent"
        }
    ]
}
```

Statement (утверждение)

В данном примере описано подутверждение с объектом StatementRef.

```
{
    "objectType": "SubStatement",
    "actor" : {
        "objectType": "Agent",
        "mbox":"mailto:agent@example.com"
    },
    "verb" : {
        "id":"http://example.com/confirmed",
        "display":{
            "en":"confirmed",
            "ru":"подтвердил"
        }
    },
    "object": {
        "objectType":"StatementRef",
        "id" :"9e13cefd-53d3-4eac-b5ed-2cf6693903bb"
    }
}
```

Приложение C: Примеры определений интерактивных задач (тип "cmi.interaction")

true-false (выбор из двух вариантов)

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

choice (множественный выбор)

```
"definition": {
    "description": {
        "en-US": "Which of these prototypes are available at the
beta site?" ,
        "ru-RU": "Какие из этих прототипов доступны на тестовом
сайте?"
    },
    "type": "http://adlnet.gov/expapi/activities/cmi.interaction",
    "interactionType": "choice",
    "correctResponsesPattern": [
        "golf[,]tetris"
    ],
    "choices": [
        {
            "id": "golf",
            "description": {
                "en-US": "Golf Example" ,
                "ru-RU": "Гольф"
            }
        },
        {
            "id": "facebook",
            "description": {
                "en-US": "Facebook App" ,
                "ru-RU": "Facebook"
            }
        },
        {
            "id": "tetris",
            "description": {
                "en-US": "Tetris Example" ,
```

```
                "ru-RU": "Тетрис"
            }
        },
        {
            "id": "scrabble",
            "description": {
                "en-US": "Scrabble Example" ,
                "ru-RU": "Эрудит"
            }
        }
    ]
}
```

fill-in (свободный ввод)

```
"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 (шкала Ликерта)

```
"definition": {
    "description": {
        "en-US": "How awesome is Experience API?",
        "ru-RU": "Как удивителен 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",
                "ru-RU": "Он вполне хорош"
            }
        },
        {
            "id": "likert_1",
            "description": {
                "en-US": "It's Pretty Cool",
```

```
                "ru-RU": "Он довольно крут"
            }
        },
        {
            "id": "likert_2",
            "description": {
                "en-US": "It's Damn Cool",
                "ru-RU": "Он чертовски крут"
            }
        },
        {
            "id": "likert_3",
            "description": {
                "en-US": "It's Gonna Change the World",
                "ru-RU": "Он собирается перевернуть весь мир"
            }
        }
    ]
}
```

matching (задание на установление соответствия)

```
"definition": {
    "description": {
        "en-US": "Match these people to their kickball team:"
        "ru-RU": "Сопоставьте игроков с командами по кикболу:"
    },
    "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 (многоэтапное задание)

```
"definition": {
    "description": {
        "en-US": "This interaction measures performance over a day
of RS sports:" ,
        "ru-RU": "Это взаимодействие совершается в течение дня в RS-
спорте:"
    },
    "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" ,
                "ru-RU": "Сетка матчей выиграна"
            }
        },
        {
            "id": "dg",
            "description": {
                "en-US": "Strokes over par in disc golf at
Liberty" ,
                "ru-RU": "Превышение числа бросков в диск-гольфе в
Либерти"
                }
            },
        {
            "id": "lunch",
            "description": {
                "en-US": "Lunch having been eaten" ,
                "ru-RU": "Употребление ланча"
            }
        }
    ]
}
```

sequencing (задание на восстановление последовательности)

```
"definition": {
    "description": {
        "en-US": "Order players by their pong ladder position:" ,
        "ru-RU": "Отсортируйте игроков по рейтингу:"
    },
    "type": "http://adlnet.gov/expapi/activities/cmi.interaction",
    "interactionType": "sequencing",
    "correctResponsesPattern": [
        "tim[,]mike[,]ells[,]ben"
    ],
    "choices": [
        {
            "id": "tim",
            "description": {
                "en-US": "Tim" ,
                "ru-RU": "Тим"
            }
        },
        {
            "id": "ben", "description": {
                "en-US": "Ben" ,
```

```
                "ru-RU": "Бен"
            }
        },
        {
            "id": "ells",
            "description": {
                "en-US": "Ells" ,
                "ru-RU": "Эллс"
            }
        },
        {
            "id": "mike",
            "description": {
                "en-US": "Mike" ,
                "ru-RU": "Майк"
            }
        }
    ]
}
```

numeric (числовой ввод)

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

other (прочее)

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

Приложение D: Приведение утверждений к формату, соответствующему версии 1.0.0

Пояснение

Данная спецификация имеет версию 1.0.0, таким образом, её реализациям не требуется поддерживать предыдущие версии. Однако предыдущие версии успели обрести популярность, и описанное здесь преобразование позволит сохранить данные, созданные в соответствии с предыдущими версиями спецификации, в системе, которая поддерживает новую версию спецификации.

Чтобы привести утверждение к формату 1.0.0 из формата 0.9, необходимо выполнить следующие действия:

- Конвертирование невозможно, если утверждение аннулировано, либо объект Verb или Activity имеет тип или свойства, которые не включены в спецификацию 0.9.
- В начало глагола приписать "http://adlnet.gov/expapi/verbs/".
- В начало id задачи, если оно не является полным абсолютным IRI, приписать "tag:adlnet.gov,2013:expapi:0.9:activities:".
- В расширениях в начало ключей, которые не является полными абсолютным IRI, приписать "tag:adlnet.gov,2013:expapi:0.9:extensions:".
- В начало типа задачи приписать "http://adlnet.gov/expapi/activities/".
- Для всех объектов типа Agent (пользователь):
  - Проверить наличие обратных функциональных идентификаторов в следующем порядке: mbox, mbox_sha1sum, openid, account. Сохранить первый из имеющихся, отбросив последующие.
  - Если представлено несколько значений выбранного обратного функционального идентификатора в массиве, оставить только первое, отбросив остальные.
  - Скопировать свойство name. Если представлено несколько значений в массиве, взять первое, отбросив остальные.
  - Удалить все прочие свойства.
- Удалить свойство voided, если оно имеется и имеет значение false. Помните: если оно имеет значение true, конвертировать утверждение нельзя!
- Добавить "version": "1.0.0".
- Если свойство authority не задано – задать его, сгенерировав объект типа Agent, идентифицированный аккаунтом (account) со свойствами homePage, равным домашней странице программы, осуществившей конвертацию, и accountName, равным "unknown".

- Если присутствует поле statement в контексте (свойство context) – удалить его.
- Сохранить прочие поля без изменений, включая stored. Поле stored должно меняться при сохранении в другой системе.

Чтобы привести утверждение к формату 1.0.0 из формата 0.95, необходимо выполнить следующие действия:

- Конвертирование невозможно, если утверждение аннулировано.
- Удалить свойство voided, если оно имеется и имеет значение false. Помните: если оно имеет значение true, конвертировать утверждение нельзя!
- Добавить "version": "1.0.0".
- Если свойство authority не задано – (задать его, сгенерировав объект типа Agent, идентифицированный аккаунтом (account) со свойствами homePage, равным домашней странице программы, осуществившей конвертацию, и accountName, равным "unknown".
- Если содержимое поля statement в контексте (свойство context) имеет тип, отличный от StatementRef – удалить его.
- Сохранить прочие поля без изменений, включая stored. Поле "stored" должно меняться при сохранении в другой системе.

Пример утверждения в формате 0.9:

```
{
    "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" ,
```

```
                "ru-RU": "Пример задачи" ,
            },
            "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" ,
                        "ru-RU": "Другая задача"
                    }
                }
            }
        }
    },
    "timestamp": "2012-06-01T19:09:13.245Z",
    "stored": "2012-06-29T15:41:39.165Z"
}
```

Утверждение, переведённое в формат 1.0.0:

```
{
    "version": "1.0.0",
    "id": "d1eec41f-1e93-4ed6-acbf-5c4bd0c24269",
    "actor": {
        "objectType": "Agent",
        "name": "Joe Schmoe",
        "mbox": "mailto:joe@example.com"
    },
    "verb": {
        "id": "http://adlnet.gov/expapi/verbs/completed",
```

```
        "display": {
            "en-US": "completed" ,
            "ru-RU": "выполнено" ,
        }
    },
    "object": {
        "objectType": "Activity",
        "id": "http://www.example.com/activities/001",
        "definition": {
            "name": {
                "en-US": "Example Activity" ,
                "ru-RU": "Пример задачи"
            },
            "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"
                            "ru-RU": "Другая задача" ,
                        }
                    }
                }
            ]
        }
    },
    "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"
        }
    }
}
```

Приложение E: Пример подписанного утверждения

Пример утверждения, подписанного, как описано в разделе 4.4 Подписанные утверждения.

Исходное утверждение для подписания. Разрывы строк в данном примере включены кодами CR+LF (0x0D + 0x0A).

```
{
    "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"
}
```

Пример закрытого ключа для сертификата X.509, который будет использован при подписании:

```
-----BEGIN RSA PRIVATE KEY-----
MIICXAIBAAKBgQDjxvZXF30WL4oKjZYXgR0ZyaX+u3y6+JqTqiNkFa/VTnet6Ly2
OT6ZmmcJEPnq3UnewpHoOQ+GfhhTkW13j06j5iNn4obcCVWTL9yXNvJH+Ko+xu4Y
l/ySPRrIPyTjtHdG0M2XzIlmmLqm+CAS+KCbJeH4tf543kIWC5pC5p3cVQIDAQAB
AoGAOejdvGq2XKuddu1kWXl0Aphn4YmdPpPyCNTaxplU6PBYMRjY0aNgLQE6bO2p
/HJiU4Y4PkgzkEgCu0xf/mOq5DnSkX32ICoQS6jChABAe20ErPfm5t8h9YKsTfn9
40lAouuwD9ePRteizd4YvHtiMMwmh5PtUoCbqLefawNApAECQQD1mdBW3zL0okUx
2pc4tttn2qArCG4CsEZMLlGRDd3FwPWJz3ZPNEEgZWXGSpA9F1QTZ6JYXIfejjRo
```

```
UuvRMWeBAkEA7WvzDBNcv4N+xeUKvH8ILti/BM58LraTtqJlzjQSovek0srxtmDg
5of+xrxN6IM4p7yvQa+7YOUOukrVXjG+1QJBAI2mBrjzxgm9xTa5odn97JD7UMFA
/WHjlMe/Nx/35V52qaav1sZbluw+TvKMcqApYj5G2SUpSNudHLDGkmd2nQECQFfc
lBRK8g7ZncekbGW3aRLVGVOxClnLLTzwOlamBKOUm8V6XxsMHQ6TE2D+fKJoNUY1
2HGpk+FWwy2D1hRGuoUCQAXfaLSxtaWdPtlZTPVueF7ZikQDsVg+vtTFgpuHloR2
6EVc1RbHHZm32yvGDY8IkcoMfJQqLONDdLfS/05yoNU=
-----END RSA PRIVATE KEY-----
```

Пример открытого сертификата X.509:

```
-----BEGIN CERTIFICATE-----
MIIDATCCAmqgAwIBAgIJAMB1csNuA6+kMA0GCSqGSIb3DQEBBQUAMHExCzAJBgNV
BAYTAlVTMRIwEAYDVQQIEwlUZW5uZXNzZWUxGDAWBgNVBAoTD0V4YW1wbGUgQ29t
cGFueTEQMA4GA1UEAxMHRXhhbXBsZTEiMCAGCSqGSIb3DQEJARYTZXhhbXBsZUBl
eGFtcGxlLmNvbTAeFw0xMzA0MDQxNTI4MzBaFw0xNDA0MDQxNTI4MzBaMIGWMQsw
CQYDVQQGEwJVUzESMBAGA1UECBMJVGVubmVzc2VlMREwDwYDVQQHEwhGcmFua2xp
bjEYMBYGA1UEChMPRXhhbXBsZSBDb21wYW55MRAwDgYDVQQLEwdFeGFtcGxlMRAw
DgYDVQQDEwdFeGFtcGxlMSIwIAYJKoZIhvcNAQkBFhNleGFtcGxlQGV4YW1wbGUu
Y29tMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDjxvZXF30WL4oKjZYXgR0Z
yaX+u3y6+JqTqiNkFa/VTnet6Ly2OT6ZmmcJEPnq3UnewpHoOQ+GfhhTkW13j06j
5iNn4obcCVWTL9yXNvJH+Ko+xu4Yl/ySPRrIPyTjtHdG0M2XzIlmmLqm+CAS+KCb
JeH4tf543kIWC5pC5p3cVQIDAQABo3sweTAJBgNVHRMEAjAAMCwGCWCGSAGG+EIB
DQQfFh1PcGVuU1NMIEdlbmVyYXRlZCBDZXJ0aWZpY2F0ZTAdBgNVHQ4EFgQUVs3v
5afEdOeoYeVajAQE4v0WS1QwHwYDVR0jBBgwFoAUyVIc3yvra4EBz20I4BF39IAi
xBkwDQYJKoZIhvcNAQEFBQADgYEAgS/FF5D0Hnj44rvT6kgn3kJAvv2lj/fyjztK
IrYS33ljXGn6gGyA4qtbXA23PrO4uc/wYCSDICDpPobh62xTCd9qObKhgwWOi05P
SBLqUu3mwfAe15LJBJBqPVZ4K0kppePBU8m6aIZoH57L/9t4OoaL8yKs/qjKFeI1
OFWZxvA=
-----END CERTIFICATE-----
```

Пример сертификата центра сертификации:

```
-----BEGIN CERTIFICATE-----
MIIDNzCCAqCgAwIBAgIJAMB1csNuA6+jMA0GCSqGSIb3DQEBBQUAMHExCzAJBgNV
BAYTAlVTMRIwEAYDVQQIEwlUZW5uZXNzZWUxGDAWBgNVBAoTD0V4YW1wbGUgQ29t
cGFueTEQMA4GA1UEAxMHRXhhbXBsZTEiMCAGCSqGSIb3DQEJARYTZXhhbXBsZUBl
eGFtcGxlLmNvbTAeFw0xMzA0MDQxNTI1NTNaFw0yMzA0MDIxNTI1NTNaMHExCzAJ
BgNVBAYTAlVTMRIwEAYDVQQIEwlUZW5uZXNzZWUxGDAWBgNVBAoTD0V4YW1wbGUg
Q29tcGFueTEQMA4GA1UEAxMHRXhhbXBsZTEiMCAGCSqGSIb3DQEJARYTZXhhbXBs
ZUBleGFtcGxlLmNvbTCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEA1sBnBWPZ
0f7WJUFTJy5+01SlS5Z6DDD6Uye9vK9AycgV5B3+WC8HC5u5h91MujAC1ARPVUOt
svPRs45qKNFIgIGRXKPAwZjawEI2sCJRSKV47i6B8bDv4WkuGvQaveZGI0qlmN5R
1Eim2gUItRj1hgcC9rQavjlnFKDY2rlXGukCAwEAAaOB1jCB0zAdBgNVHQ4EFgQU
yVIc3yvra4EBz20I4BF39IAixBkwgaMGA1UdIwSBmzCBmIAUyVIc3yvra4EBz20I
4BF39IAixBmhdaRzMHExCzAJBgNVBAYTAlVTMRIwEAYDVQQIEwlUZW5uZXNzZWUx
GDAWBgNVBAoTD0V4YW1wbGUgQ29tcGFueTEQMA4GA1UEAxMHRXhhbXBsZTEiMCAG
CSqGSIb3DQEJARYTZXhhbXBsZUBleGFtcGxlLmNvbYIJAMB1csNuA6+jMAwGA1Ud
EwQFMAMBAf8wDQYJKoZIhvcNAQEFBQADgYEADhwTebGk735yKhm8DqCxvNnEZ0Nx
sYEYOjgRG1yXTlW5pE691fSH5AZ+T6fpwpZcWY5QYkoN6DnwjOxGkSfQC3/yGmcU
DKBPwiZ5O2s9C+fE1kUEnrX2Xea4agVngMzR8DQ6oOauLWqehDB+g2ENWRLoVgS+
ma5/Ycs0GTyrECY=
-----END CERTIFICATE-----
```

JWS-заголовок. Обратите внимание, что, кроме выбранного алгоритма, он включает цепочку сертификатов.


```
{
    "alg": "RS256",
    "x5c": [

"MIIDATCCAmqgAwIBAgIJAMB1csNuA6+kMA0GCSqGSIb3DQEBBQUAMHExCzAJBgNVBAYTAlVTMR
IwEAYDVQQIEwlUZW5uZXNzZWUxGDAWBgNVBAoTD0V4YW1wbGUgQ29tcGFueTEQMA4GA1UEAxMHR
XhhbXBsZTEiMCAGCSqGSIb3DQEJARYTZXhhbXBsZUBleGFtcGxlLmNvbTAeFw0xMzA0MDQxNTI4
MzBaFw0xNDA0MDQxNTI4MzBaMIGWMQswCQYDVQQGEwJVUzESMBAGA1UECBMJVGVubmVzc2VlMRE
wDwYDVQQHEwhGcmFua2xpbjEYMBYGA1UEChMPRXhhbXBsZSBDb21wYW55MRAwDgYDVQQLEwdFeG
FtcGxlMRAwDgYDVQQDEwdFeGFtcGxlMSIwIAYJKoZIhvcNAQkBFhNleGFtcGxlQGV4YW1wbGUuY
29tMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDjxvZXF30WL4oKjZYXgR0ZyaX+u3y6+JqT
qiNkFa/VTnet6Ly2OT6ZmmcJEPnq3UnewpHoOQ+GfhhTkW13j06j5iNn4obcCVWTL9yXNvJH+Ko
+xu4Yl/ySPRrIPyTjtHdG0M2XzIlmmLqm+CAS+KCbJeH4tf543kIWC5pC5p3cVQIDAQABo3sweT
AJBgNVHRMEAjAAMCwGCWCGSAGG+EIBDQQfFh1PcGVuU1NMIEdlbmVyYXRlZCBDZXJ0aWZpY2F0Z
TAdBgNVHQ4EFgQUVs3v5afEdOeoYeVajAQE4v0WS1QwHwYDVR0jBBgwFoAUyVIc3yvra4EBz20I
4BF39IAixBkwDQYJKoZIhvcNAQEFBQADgYEAgS/FF5D0Hnj44rvT6kgn3kJAvv2lj/fyjztKIrY
S33ljXGn6gGyA4qtbXA23PrO4uc/wYCSDICDpPobh62xTCd9qObKhgwWOi05PSBLqUu3mwfAe15
LJBJBqPVZ4K0kppePBU8m6aIZoH57L/9t4OoaL8yKs/qjKFeI1OFWZxvA=",

"MIIDNzCCAqCgAwIBAgIJAMB1csNuA6+jMA0GCSqGSIb3DQEBBQUAMHExCzAJBgNVBAYTAlVTMR
IwEAYDVQQIEwlUZW5uZXNzZWUxGDAWBgNVBAoTD0V4YW1wbGUgQ29tcGFueTEQMA4GA1UEAxMHR
XhhbXBsZTEiMCAGCSqGSIb3DQEJARYTZXhhbXBsZUBleGFtcGxlLmNvbTAeFw0xMzA0MDQxNTI1
NTNaFw0yMzA0MDIxNTI1NTNaMHExCzAJBgNVBAYTAlVTMRIwEAYDVQQIEwlUZW5uZXNzZWUxGDA
WBgNVBAoTD0V4YW1wbGUgQ29tcGFueTEQMA4GA1UEAxMHRXhhbXBsZTEiMCAGCSqGSIb3DQEJAR
YTZXhhbXBsZUBleGFtcGxlLmNvbTCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEA1sBnBWPZ0
f7WJUFTJy5+01SlS5Z6DDD6Uye9vK9AycgV5B3+WC8HC5u5h91MujAC1ARPVUOtsvPRs45qKNFI
gIGRXKPAwZjawEI2sCJRSKV47i6B8bDv4WkuGvQaveZGI0qlmN5R1Eim2gUItRj1hgcC9rQavjl
nFKDY2rlXGukCAwEAAaOB1jCB0zAdBgNVHQ4EFgQUyVIc3yvra4EBz20I4BF39IAixBkwgaMGA1
UdIwSBmzCBmIAUyVIc3yvra4EBz20I4BF39IAixBmhdaRzMHExCzAJBgNVBAYTAlVTMRIwEAYDV
QQIEwlUZW5uZXNzZWUxGDAWBgNVBAoTD0V4YW1wbGUgQ29tcGFueTEQMA4GA1UEAxMHRXhhbXBs
ZTEiMCAGCSqGSIb3DQEJARYTZXhhbXBsZUBleGFtcGxlLmNvbYIJAMB1csNuA6+jMAwGA1UdEwQ
FMAMBAf8wDQYJKoZIhvcNAQEFBQADgYEADhwTebGk735yKhm8DqCxvNnEZ0NxsYEYOjgRG1yXTl
W5pE691fSH5AZ+T6fpwpZcWY5QYkoN6DnwjOxGkSfQC3/yGmcUDKBPwiZ5O2s9C+fE1kUEnrX2X
ea4agVngMzR8DQ6oOauLWqehDB+g2ENWRLoVgS+ma5/Ycs0GTyrECY="
    ]
}
```


JWS-подпись:


```
ew0KICAgICJhbGciOiAiUlMyNTYiLA0KICAgICJ4NWMiOiBbDQogICAgICAgICJNSUlEQVRDQ0F
tcWdBd0lCQWdJSkFNQjFjc051QTYra01BMEdDU3FHU0liM0RRRUJCUVVBTUhFeEN6QUpCZ05WQk
FZVEFsVlRNUkl3RUFZRFZRUUlFd2xVWlc1dVpYTnpaV1V4R0RBV0JnTlZCQW9URDBWNFlXMXdiR
1VnUTI5dGNHRnVlVEVRTUE0R0ExVUVBeE1IUlhoaGJYQnNaVEVpTUNBR0NTcUdTSWIzRFFFSkFS
WVRaWGhoYlhCc1pVQmxlR0Z0Y0d4bExtTnZiVEFlRncweE16QTBNRFF4TlRJNE16QmFGdzB4TkR
BME1EUXhOVEk0TXpCYU1JR1dNUXN3Q1FZRFZRUUdFd0pWVXpFU01CQUdBMVVFQ0JNSlZHVnVibV
Z6YzJWbE1SRXdEd1lEVlFRSEV3aEdjbUZ1YTJ4cGJqRVlNQllHQTFVRUNoTVBSWGhoYlhCc1pTQ
kRiMjF3WVc1NU1SQXdEZ1lEVlFRTEV3ZEZlR0Z0Y0d4bE1SQXdEZ1lEVlFRREV3ZEZlR0Z0Y0d4
bE1TSXdJQVlKS29aSWh2Y05BUWtCRmhObGVHRnRjR3hsUUdWNFlXMXdiR1V1WTI5dE1JR2ZNQTB
HQ1NxR1NJYjNEUUVCQVFVQUE0R05BRENCaVFLQmdRRGp4dlpYRjMwV0w0b0tqWllYZ1IwWnlhWC
t1M3k2K0pxVHFpTmtGYS9WVG5ldDZMeTJPVDZabW1jSkVQbnEzVW5ld3BIb09RK0dmaGhUa1cxM
2owNmo1aU5uNG9iY0NWV1RMOXlYTnZKSCtLbyt4dTRZbC95U1BSck1QeVRqdEhkRzBNMlh6SWxt
```

bUxxbStDQVMrS0NiSmVINHRmNTQza0lXQzVwQzVwM2NWUUlEQVFBQm8zc3dlVEFKQmdOVkhSTUV
BakFBTUN3R0NXQ0dTQUdHK0VJQkRRUWZGaDFQY0dWdVUxTk1JRWRsYm1WeVlYUmxaQ0JEWlhKMG
FXWnBZMkYwWlRBZEJnTlZIUTRFRmdRVVZzM3Y1YWZFZE9lb1llVmFqQVFFNHYwV1MxUXdId1lEV
lIwakJCZ3dGb0FVeVZJYzN5dnJhNEVCejIwSTRCRjM5SUFpeEJrd0RRWUpLb1pJaHZjTkFRUZC
UUFEZ1lFQWdTL0ZGNUQwSG5qNDRydlQ2a2duM2tKQXZ2MmxqL2Z5anp0S0lyWVMzM2xqWEduNmd
HeUE0cXRiWEEyM1ByTzR1Yy93WUNTRElDRHBQb2JoNjJ4VENkOXFPYktoZ3dXT2kwNVBTQkxxVX
UzbXdmQWUxNUxKQkpCcVBWWjRLMGtwcGVQQlU4bTZhSVpvSDU3TC85dDRPb2FMOHlLcy9xaktGZ
UkxT0ZXWnh2QT0iLA0KICAgICAgICAiTUlJRE56Q0NBcUNnQXdJQkFnSUpBTUIxY3NOdUE2K2pN
QTBHQ1NxR1NJYjNEUUVCQlFVQU1IRXhDekFKQmdOVkJBWVRBbFZUTVJJd0VBWURWUVFJRXdsVVp
XNXVaWE56WldVeEdEQVdCZ05WQkFvVEQwVjRZVzF3YkdVZ1EyOXRjR0Z1ZVRFUU1BNEdBMVVFQX
hNSFJYaGhiWEJzWlRFaU1DQUdDU3FHU0liM0RRRUpBUllUWlhoaGJYQnNaVUJsZUdGdGNHeGxMb
U52YlRBZUZ3MHhNekEwTURReE5USTFOVE5hRncweU16QTBNRE14TlRJMU5UTmFNSEV4Q3pBSkJn
TlZCQVlUQWxWVE1SSXdFQVlEVlFRSUV3bFVaVzV1WlhOelpXVXhHREFXQmdOVkJBb1REMFY0WVc
xd2JHVWdRMjl0Y0dGdWVURVFNQTRHQTFVRUF4TUhSWGhoYlhCc1pURWlNQ0FHQ1NxR1NJYjNEUU
VKQVJZVFpYaGhiWEJzWlVCbGVHRnRjR3hsTG1OdmJUQ0JuekFOQmdrcWhraUc5dzBCQVFFRkFBT
0JqUUF3Z1lrQ2dZRUExc0JuQldQWjBmN1dKVUZUSnk1KzAxU2xTNVo2RERENlV5ZTl2SzlBeWNn
VjVCMytXQzhIQzV1NWg5MU11akFDMUFSUFZVT3RzdlBSczQ1cUtORklnSUdSWEtQQXdaamF3RUk
yc0NKUlNLVjQ3aTZCOGJEdjRXa3VHdlFhdmVaR0kwcWxtTjVSMUVpbTJnVUl0UmoxaGdjQzlyUW
F2amxuRktEWTJybFhHdWtDQXdFQUFhT0IxakNCMHpBZEJnTlZIUTRFRmdRVXlWSWMzeXZyYTRFQ
noyMEk0QkYzOUlBaXhCa3dnYU1HQTFVZEl3U0JtekNCbUlBVXlWSWMzeXZyYTRFQnoyMEk0QkYz
OUlBaXhCbWhkYVJ6TUhFeEN6QUpCZ05WQkFZVEFsVlRNUkl3RUFZRFZRUUlFd2xVWlc1dVpYTnp
aV1V4R0RBV0JnTlZCQW9URDBWNFlXMXdiR1VnUTI5dGNHRnVlVEVRTUE0R0ExVUVBeE1IUlhoaG
JYQnNaVEVpTUNBR0NTcUdTSWIzRFFFSkFSWVRaWGhoYlhCc1pVQmxlR0Z0Y0d4bExtTnZiWUlKQ
U1CMWNzTnVBNitqTUF3R0ExVWRFd1FGTUFNQkFmOHdEUVlKS29aSWh2Y05BUUVGQlFBRGdZRUFE
aHdUZWJHazczNXlLaG04RHFDeHZObkVaME54c1lFWU9qZ1JHMXlYVGxXNXBFNjkxZlNINUFaK1Q
2ZnB3cFpjV1k1UVlrb042RG53ak94R2tTZlFDMy95R21jVURLQlB3aVo1TzJzOUMrZkUxa1VFbn
JYMlhlYTRhZ1ZuZ016UjhEUTZvT2F1TFdxZWhEQitnMkVOV1JMb1ZnUyttYTUvWWNzMEdUeXJFQ
1k9Ig0KICAgIF0NCn0.ew0KICAgICJ2ZXJzaW9uIjogIjEuMC4wIiwNCiAgICAiaWQiOiAiMzNj
ZmY0MTYtZTMzMS00YzlkLTk2OWUtNTM3M2ExNzU2MTIwIiwNCiAgICAiYWN0b3IiOiB7DQogICA
gICAgICJtYm94IjogIm1haWx0bzpleGFtcGxlQGV4YW1wbGUuY29tIiwNCiAgICAgICAgIm5hbW
UiOiAiRXhhbXBsZSBMZWFybmVyIiwNCiAgICAgICAgICAgIm9iamVjdFR5cGUiOiAiQWdlbnQiDQogI
CAgfSwNCiAgICAidmVyYiI6IHsNCiAgICAgICAgImlkIjogImh0dHA6Ly9hZGxuZXQuZ292L2V4
cGFwaS92ZXJicy9leHBlcmllbmNlZCIsDQogICAgICAgICJkaXNwbGF5Ijogew0KICAgICAgICA
gICAgImVuLVVTIjogImV4cGVyaWVuY2VkIg0KICAgICAgICB9DQogICAgfSwNCiAgICAib2JqZW
N0Ijogew0KICAgICAgICAiaWQiOiAiaHR0cHM6Ly93d3cueW91dHViZS5jb20vd2F0Y2g_dj14a
DRrSWlIM1NtOCIsDQogICAgICAgICJvYmplY3RUeXBlIjogIkFjdGl2aXR5IiwNCiAgICAgICAg
ImRlZmluaXRpb24iOiB7DQogICAgICAgICAgICAibmFtZSI6IHsNCiAgICAgICAgICAgICAgICA
iZW4tVVMiOiAiVGF4IFRpcHMgJiBJbmZvcm1hdGlvbiA6IEhvdyB0byBGaWxlIGEgVGF4IFJldH
VybiAiDQogICAgICAgICAgICB9LA0KICAgICAgICAgICAgImRlc2NyaXB0aW9uIjogew0KICAgI
CAgICAgICAgICJlbi1VUyI6ICJGaWxpbmcgYSB0YXggcmV0dXJuIHdpbGwgcmVxdWlyZSBm
aWxsaW5nIG91dCBlaXRoZXIgYSB0YXggcmV0dXJuIHdpbGwgcmVxdWlyZSBm
gICAgICAgIH0NCiAgICAgICAgIm5hbWUiOiAiVGF4IFJldHVybiBIZWxwIn0NCiAgICAgICAgI
EyOjAwOjAwWiINCn0.
FWuwaPhwUbkk7h9sKW5zSvjsYNugvxJ-
TrVaEgt_DCUT0bmKhQScRrjMB6P9O50uznPwT66oF1NnU_G0HVhRzS5voiXE-y7tT3z0M3-
8A6YK009Bk_digVUul-HA4Fpd5IjoBBGe3yzaQ2ZvzarvRuipvNEQCI0onpfuZZJQ0d8

Подписанное утверждение:

```
{
    "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":
"672fa5fa658017f1b72d65036f13379c6ab05d4ab3b6664908d8acf0b6a0c634"
        }
    ]
}
```

Вложение с подписью не отображено. О прикреплении вложений смотрите в разделе 4.1.11 Вложения.

Приложение F: Таблица всех узлов API

Узел (относительный IRI, которому предшествует базовый IRI LRS)	**Функционал**
statements	Запись и поиск объектов Statement (утверждений)
agents	Запись и поиск объектов Agent (агент)
agents/profile	Agent Profile API (вспомогательные документы об агентах)
activities	Запись и поиск объектов Activity (задача)
activities/profile	Activity Profile API (вспомогательные документы о задачах)
activities/state	State API
about	Информация об LRS
OAuth/initiate	Запрос временных полномочий
OAuth/authorize	Авторизация на ресурсе
OAuth/token	Запрос токена

Приложение G: Лицензия Apache 2.0

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A

PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

Лицензия Apache (неофициальный перевод)

Версия 2.0, январь 2004 г.

http://www.apache.org/licenses/
Условия использования, воспроизведения и распространения

1. Определения

«Лицензия» — это условия использования, воспроизведения и распространения в объёме, определяемом в разделах с 1 по 9 настоящего документа.

«Лицензиар» — это владелец авторского права или лицо, уполномоченное владельцем авторских прав, предоставляющие Лицензию.

«Юридическое лицо» — это объединение действующих лиц и остальных лиц, которые контролируют, контролируются или находятся под общим контролем с действующими лицами. Для целей данного определения, «контроль» означает (I) силу, прямую или косвенную, определяющую развитие или управление такого лица согласно договорённостям или иным образом, или (II) собственность пятидесяти процентов (50%) или более находящихся в обращении акций, или (III) доверительное управление этим лицом.

«Вы» — это физическое или юридическое лицо, использующее права, предоставленные Лицензией.

«Исходная форма» — это форма, предпочтительная для внесения изменений, включая исходный код, исходники документации, конфигурационные файлы и не ограничиваясь ими.

«Объектная форма» — это результат автоматического преобразования исходников, включающий исходный код, откомпилированный в объектный, сгенерированную документацию и другие виды медиа, но не ограничивающийся ими.

«Работа» — это объект авторского права в Исходной или Объектной форме, доступный на условиях Лицензии, о чём свидетельствует уведомление об авторстве, которое входит в Работу или прилагается к ней.

«Производные работы» — это любые работы в Исходной или Объектной форме, основанные на Работе или полученные из неё, для которых редакционные изменения, аннотации, развитие или иные изменения, имеют, в целом, независимое авторство. Для целей настоящей Лицензии, производные работы не

включают работы, которые отделимы от Работы и её производных, например, связываются по имени или ссылке с интерфейсом Работы.

«Вклад» — это любое авторское произведение, в том числе оригинальная версия Работы, все изменения и дополнения Работы и Производных работ, умышленно представленные Лицензиару для включения в Работу владельцем авторского права или лицом (юридическим лицом), уполномоченным представлять владельца авторских прав. Для целей данного определения, «представление» означает любую форму электронного, словесного или письменного сообщения, направленного Лицензиару или его представителям, включая, но не ограничиваясь электронными списками рассылки, системами управления исходным кодом и дефектами, управляемыми Лицензиаром или от его имени с целью обсуждения и улучшения Работы, за исключением сообщений, помеченных владельцем авторских прав, как «не вклад».

«Участник» — это Лицензиар, а также любое физическое или юридическое лицо, от имени которого Лицензиаром был получен вклад, впоследствии включенный в Работу.

2. Предоставление прав
В соответствии с условиями настоящей Лицензии, каждый Участник настоящим предоставляет Вам вечную, неэксклюзивную, бесплатную, безвозмездную, безотзывную лицензию прав на воспроизведение, изменение, публичный показ, публичное исполнение, сублицензирование и распространение Работы и Производных работ в Исходной и Объектной формах по всему миру.

3. Предоставление патентных прав
В соответствии с условиями настоящей Лицензии, каждый Участник настоящим предоставляет Вам вечную, неэксклюзивную, бесплатную, безвозмездную, безотзывную (кроме случаев, перечисленных в этом разделе) патентную лицензию производить, произвести, использовать, предлагать продать, продавать, импортировать и иным образом передавать Работу по всему миру. Эта Лицензия относится только к патентным правам, лицензированным Участником и с необходимостью нарушаемым отдельным Вкладом Участника или Вкладом Участника вкупе с Работой, в которую Вклад был сделан. Если Вы начинаете патентный спор в отношении любого лица (включая встречный иск), утверждая, что Работа или Вклад, включенный в работу, являются прямым или частичным нарушением патентных прав, то все патентные права, предоставленные Вам этой Лицензией, заканчиваются в день соответствующего судебного иска.

4. Распространение
Вы можете воспроизводить и распространять копии Работы или Производных работ на любом носителе, с изменениями или без, в Исходной или Объектной форме, при условии, что выполняются следующие условия:

(a) Вы должны предоставить всем другим получателям Работы и Производных работ, копию этой лицензии, и

(b) Вы должны снабдить все модифицированные файлы явными уведомлениями, что Вы изменили файлы, и

(c) Вы должны сохранить в Исходной форме любых Производных работ, которые вы распространяете, все авторские права, патенты, торговые марки, а также соответствующие атрибуции из Исходной формы Работы, за исключением тех, что не имеют отношения к какой-либо части Производной работы; и

(d) если Работа включает в себя текстовый файл NOTICE, как часть пакета, то любые Производные работы, распространяемые Вами, должны включать читаемую копию этого файла, за исключением тех замечаний, которые не имеют отношения к какой-либо части Производной работы, по крайней мере в одном из следующих мест: в текстовом файле NOTICE, который поставляется в составе Производной работы; в Исходной форме документации, если она поставляется вместе с Производной работой, в изображении, генерируемом Производной работой, где обычно появляются упоминания сторонних производителей. Содержимое файла NOTICE предоставляется для информационных целей и не изменяет Лицензию. Вы можете добавить свои собственные уведомления в Производные работы, которые Вы распространяете, рядом или в качестве добавления к тексту NOTICE, при условии, что такие дополнительные уведомления не могут быть истолкованы, как изменение лицензии. Вы можете добавить утверждение своего авторского права на Ваши изменения и предусмотреть дополнительные или иные лицензионные условия и условия использования, воспроизведения или распространения Ваших изменений или Производной работы в целом, при условии, что использование, воспроизведение и распространение Работы Вами соответствует условиям этой Лицензии.

5. Предоставление вкладов
Если Вы явно не указали иное, любые материалы, намеренно представленные Вами для включения в Работу Лицензиаром должны соответствовать положениям и условиям данной Лицензии без каких-либо дополнительных условий или ограничений. Вышесказанное никаким образом не заменяет и не изменяет условия любого отдельного лицензионного соглашения, заключённого Вами и Лицензиаром в отношении таких взносов.

6. Товарные знаки
Эта лицензия не даёт разрешения на использование торговых наименований, товарных знаков, знаков обслуживания или названий продуктов Лицензиара, за исключением случаев разумного и обычного использования при описании происхождения Работы и воспроизведении содержания файла NOTICE.

7. Отказ от гарантий
Если это не предусмотрено применимыми законами или не согласовано в письменной форме, Лицензиар предоставляет Работу (и каждый Участник предоставляет свои Вклады) «как есть», без гарантий и условий любого рода, явных или подразумеваемых, включая, без ограничений, любые условия или гарантии прав собственности, патентных прав, коммерческой ценности и пригодности для определённой цели. Вы несете полную ответственность за определение целесообразности использования или распространения Работы и несёте риски, связанные с осуществлением прав в соответствии с настоящей Лицензией.
8. Ограничение ответственности

Ни в каком случае и ни на каком правовом поле, будь то в результате гражданского правонарушения (включая халатность), по соглашению, или в других случаях, если только это не требуется действующим законодательством (например, в случае преднамеренных действий и грубой небрежности) или согласовано в письменной форме, никакой Участник не будет нести ответственность перед Вами за убытки, в том числе любые прямые, косвенные, специальные, случайные или последующие убытки любого характера, возникающие в результате этой Лицензии или в связи с использованием или невозможностью использования Работы (включая возмещение ущерба за потерю репутации, прекращение работы, компьютерный сбой или неисправность, любые другие коммерческие убытки или потери, но не ограничиваясь ими), даже если такой Участник был уведомлен о возможности таких убытков.

9. Принятие ответственности по гарантиям
При распространении работы Вы можете предложить и взимать плату за гарантии, поддержку, поручительство, компенсации и другие обязательства по ответственности или правам в соответствии с настоящей Лицензией. Тем не менее, при принятии таких обязательств, Вы действуете только от своего имени и под Вашу исключительную ответственность, а не от имени какого-либо другого Участника, и только тогда, когда Вы согласны компенсировать убытки, защищать и поддерживать каждого Участника от какой-либо ответственности или претензий, заявленных по причине Вашего принятия таких гарантий или дополнительной ответственности.

Конец определений и условий

Перевод с английского:
Александр Дмитриевич Копилов
Андрей Владимирович Лямин

Experience API Версия 1.0.1

Спецификация

В авторской редакции
Редакционно-издательский отдел Университета ИТМО
Зав. РИО Н.Ф. Гусарова
Подписано к печати
Заказ №
Тираж 100
Отпечатано на ризографе

УНИВЕРСИТЕТ ИТМО

Миссия университета – генерация передовых знаний, внедрение инновационных разработок и подготовка элитных кадров, способных действовать в условиях быстро меняющегося мира и обеспечивать опережающее развитие науки, технологий и других областей для содействия решению актуальных задач.

ЦЕНТР ДИСТАНЦИОННОГО ОБУЧЕНИЯ

Центр дистанционного обучения (ЦДО) был создан в 1999 году для планирования, координации и выполнения работ по созданию новых и развитию перспективных методов и технологий электронного обучения с целью повышения качества, доступности и эффективности университетского образования. В 2003 году в ЦДО была разработана система управления обучением AcademicNT, которая с 2006 года успешно используется в учебном процессе ряда университетов Санкт-Петербурга, а в 2008 году была награждена дипломом победителя всероссийского форума «Образовательная среда-2008». В стенах ЦДО разработано множество различных программных продуктов и решений таких, как система автоматизации работы приёмной комиссии, система для проведения интерактивных занятий, программный комплекс для удалённого управления оптико-электронными приборами, протокол управления удалённой лабораторией RLCP.

Сотрудники центра, в числе которых много преподавателей, аспирантов и студентов, занимаются научно-исследовательской и научно-методической работой, о чём свидетельствуют более 200 научных и методических публикаций, 17 успешно выполненных научно-исследовательских проектов. В настоящее время ЦДО участвует в создании национальной платформы открытого образования, осуществляет продвижение Университета ИТМО на международных образовательных площадках.

Редакционно-издательский отдел
Университета ИТМО
197101, Санкт-Петербург, Кронверкский пр., 49