LEARNING SOLUTIONS MAGAZINE

Your Source for Learning Technology, Strategy, and News

Five Things a Web Developer Needs to Know About the xAPI

by Andy Johnson, Jason Haag, Steve Foreman

October 15, 2014

If you are an instructional designer, you should read the *Learning Solutions* article published October 13, 2014, “Ten Steps to Plan and Communicate Your xAPI Design to a Web Developer.”

If you are a web developer, and have been asked to implement an Experience API (xAPI) solution that has been designed by an instructional designer in the training department, you may be wondering what the xAPI is and what it will take to implement. You may not be aware of the xAPI specification yet. This article will help you get started. It provides a conceptual framework for understanding what the xAPI is and what it does.

The xAPI is extremely useful in that it can be expressed as both human-readable and machine-readable in the same format because it uses natural language. It is claimed that the xAPI can track *anything*. *Anything* is really unsettling for a web developer, but don’t fret. Here are five of the basics to get you started in developing with xAPI.

1. Make sure you have the required technical knowledge and skills

As with any web-development effort, certain programming languages and technologies are needed to make things work. The following are key technical knowledge and skills that you will require:

- Application Programming Interfaces (APIs) (ease the ability to share data between applications)
- JavaScript Object Notation (JSON) (a popular, light-weight data format for APIs)
- REST (an architectural style applied to web services)

The data format xAPI uses is JavaScript Object Notation (JSON). While JSON has JavaScript in its name, there is no dependence on JavaScript for programming. If you are familiar with eXtensible Markup Language (XML), then JSON should come naturally to you. Like XML, JSON is syntax used for the storage and retrieval of structured information. JSON uses a hierarchical node structure, but unlike XML, the order of the siblings does not matter. Each JSON "node" is a string representation of a container of other "nodes," or a name/value pair, or an array of name/value pairs. There are no attributes like there are in XML. JSON is inherently more efficient than XML, requiring less machine time to process the data (on either end).

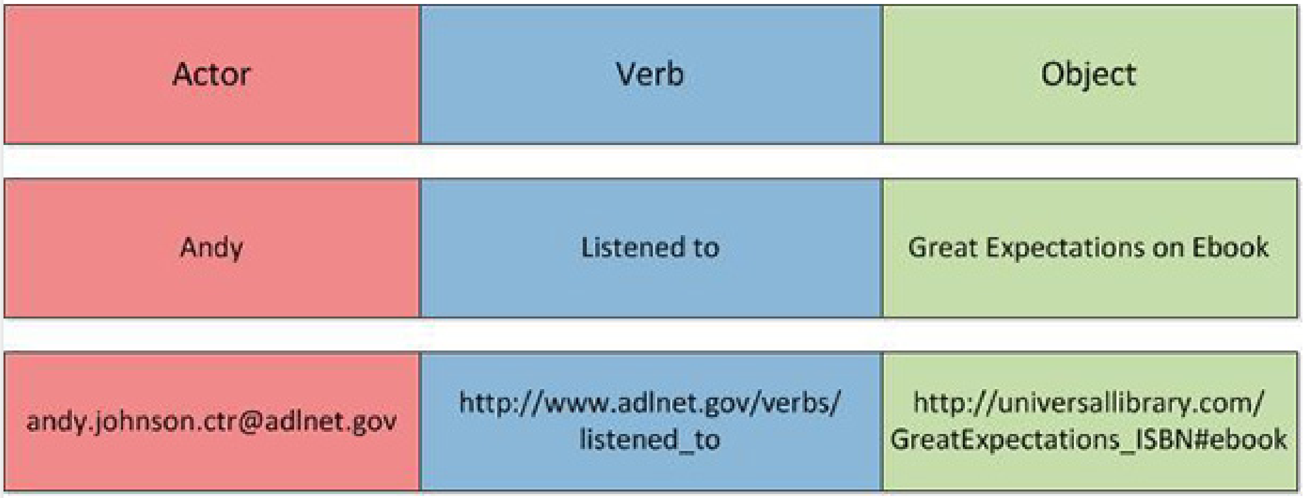
To align with current web best practices, a RESTful approach is used with all xAPI requests. HTTP methods combined with RESTful practices work through HTTP requests to store, modify, retrieve, and delete data. For the purposes of xAPI, you only need a conceptual understanding of REST as the technical details are built-in to the xAPI requirements and resources that use xAPI.

2. Learn the basic terminology

To understand the xAPI specification, you will need to read the documentation, but first, there are a few basics that you should understand:

Statement

Most xAPI data is communicated through **Statements**. A **Statement** is represented in a JSON format and includes, at a minimum, the *actor*, or user who is being tracked, the *verb*, or action they are taking, and the *object.* (Figure 1)



Actor	Verb	Object
Andy	Listened to	Great Expectations on Ebook
andy.johnson.ctr@adlnet.gov	http://www.adlnet.gov/verbs/ listened_to	http://universallibrary.com/ GreatExpectations_ISBN#ebook

Figure 1: **The basic elements and structure of an xAPI statement**

The object can be an activity, another actor, or a previous experience. There are other JSON components that are mandatory or optional, but the actor, verb, and object (usually in the form of an activity) are the most important.

Actors

An Actor is an individual or group, represented by a unique ID that can be associated with a persona. A persona is a particular representation, usually an account (email or otherwise), corresponding to a specific set of tasks used by the account. Some examples of unique IDs that could be used are email address or application accounts (e.g., Open ID, Twitter, Facebook, Google, etc.)

Verbs

The Verb in an xAPI statement describes the action performed during the learning experience. The xAPI does not specify any particular verbs. Instead, it defines how to create verbs so that communities of practice can establish verbs that are meaningful to their members and make them available for use by anyone. Verbs have unique identifiers that reference an embedded definition to assist in disambiguation. For example, using the verb, *fired*, is ambiguous. A definition is used to disambiguate the verb. You may define fired as "launched a projectile," or "terminated an employee." Verbs are the primary property used to sort statements, so using verbs correctly and consistently within a community of practice is very important.

Object

In object-oriented programming, a generic object acquires specifically defined properties as it becomes a more specific type. In grammar, the object of a sentence can take the form of a noun, pronoun, phrase, or clause. Objects in xAPI have both of these characteristics. An object used in an xAPI statement has an objectType, which determines specific properties for the object. There are three objectTypes: Activity, Actor, and Statement.

- **Activity ObjectType:** An object of the type *activity* is the most common type. Examples of activities in xAPI statements are "Jane attempted the cake recipe," "Jane passed the chemistry quiz," and "Jane equipped virtual sword" (i.e., in a virtual world or simulation). Activity objects are extremely granular and can identify something as large in scale as a specific virtual world or as small as a single virtual chair in that world. To put it another way, sometimes the activity object is something the learner uses (recipe) rather than something the user is part of (a cooking class).
- **Actor ObjectType:** An object of the type *actor* is useful for capturing person-to-person interactions. Examples would be "Andy sent Jason a message," "Jason invited Andy to join a business seminar," or "Intelligent Tutor recommends training to Andy." An actor object, like the defined actor, takes on a particular persona.
- **Statement ObjectType:** An object of the type *statement* is useful for applying actions to previously used xAPI statements. In the same way grammar uses clauses, sometimes a single verb and object isn't enough. Examples would be "Andy liked that Jason passed Biology 101," "Andy commented on Andy posted picture of a sunset," or "Andy shared that Andy got quiz question #1 wrong."

Agents

An actor that is an individual is called an agent. As there are specific endpoints and functions in xAPI that only work on individuals, this distinction is necessary.

Context properties

Context properties are some of the most important elements in xAPI. They allow the grounding of a statement by providing a context in which the experience took place. Pertinent information, such as the instructor of the content, the location or language the activity took place in, and related activities are all baseline parts of context. Furthermore, context can be extended to allow communities of practice to define their own fields. The context property can be used to define an "activity tree" that relates higher-level xAPI statements to more granular statements. This approach makes reporting and analytics easier to interpret.

IRIs

The uniqueness of objects and verbs are dictated by full *Internationalized Resource Identifiers (IRIs).* IRIs are similar to URIs and URLs, which may be more familiar to web developers, but an IRI can use international characters and is still unique. A best practice in xAPI is that the base portion of the IRI is a domain controlled by those creating verbs and objects. Figure 2 shows IRI structure and best practices:

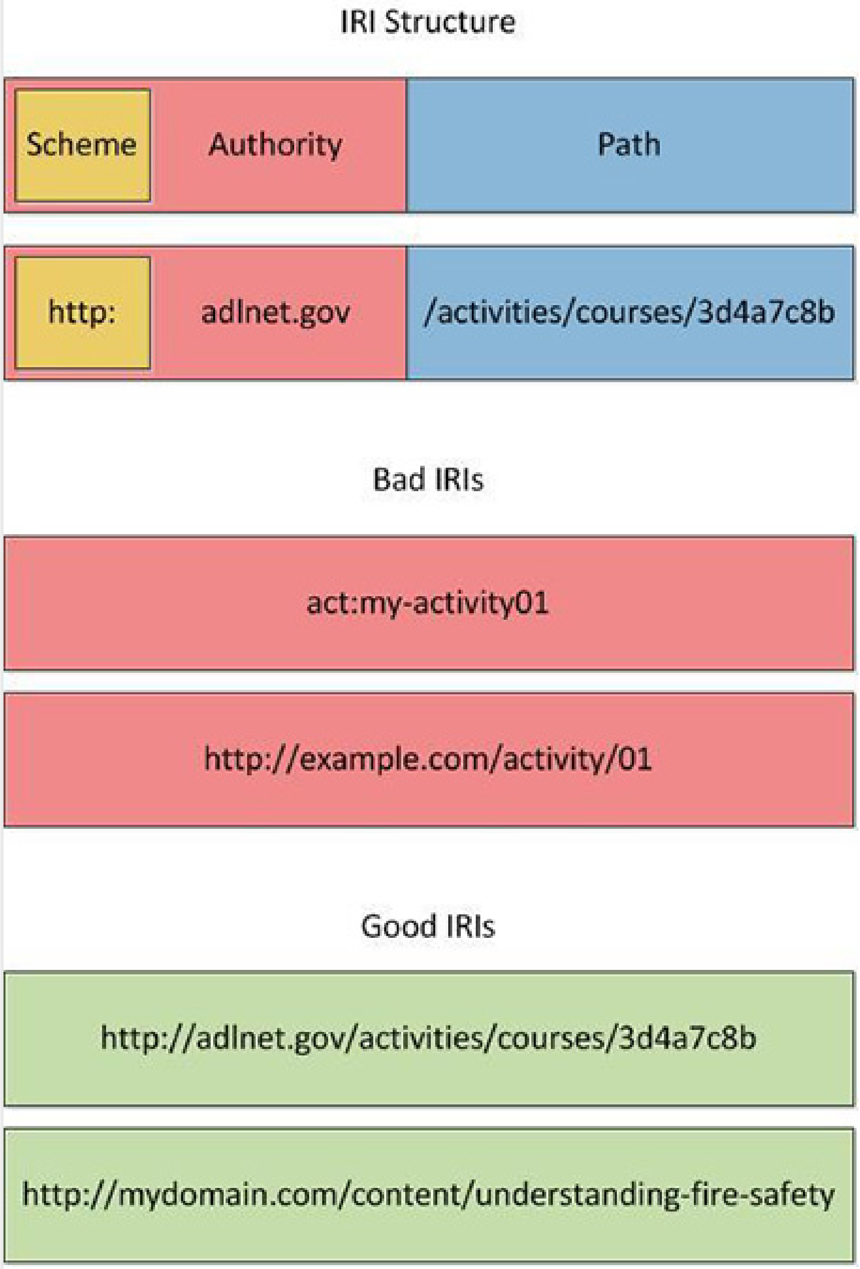

IRI Structure
Scheme
Authority
Path
http:
adlnet.gov
/activities/courses/3d4a7c8b
Bad IRIs
act:my-activity01
http://example.com/activity/01
Good IRIs
http://adlnet.gov/activities/courses/3d4a7c8b
http://mydomain.com/content/understanding-fire-safety

Figure 2: **IRI structure and best practices**

While a URL always begins with the authority, http://<domain name>/<path>, a URI or IRI can reference a file share or other type of address which may begin with a different type of authority.

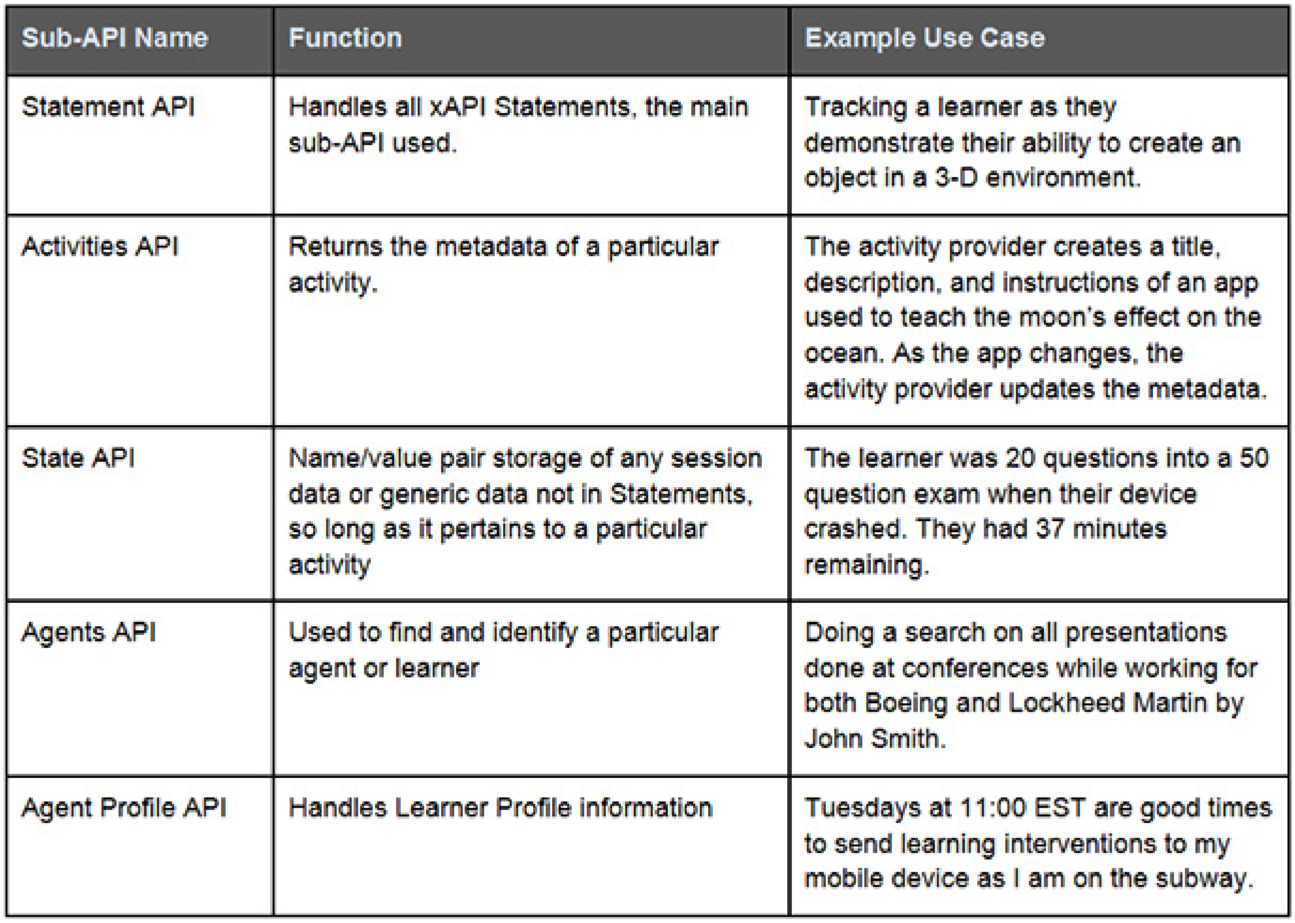
When you establish an IRI, make sure the authority is under your control so that changes to the authority, domain, or path do not adversely affect your IRI. Also, make sure you use a naming convention that is descriptive and unique.

In the graphic above, the act:my-activity01 is a bad IRI because the web developer doesn’t own the authority, *act*. The http://example.com/activity/01 is a bad IRI because the activity name, *01*, has no meaning, making it difficult to interpret and reuse.

LRS and endpoints

Statements are validated and stored by a *Learning Record Store (LRS).* An LRS can be part of a learning management system that supports xAPI, or a standalone system. The LRS is responsible for implementing multiple sub-APIs (more accurately, xAPI Resources), each of which have endpoints. An *endpoint* is simply a URL within the LRS that listens for HTTP requests. The xAPI is actually a set of sub-APIs and each sub-API has its own endpoint. The sub-APIs are described in the table below.

Table 1: **Sub-APIs, their functions, and examples**



Sub-API Name	Function	Example Use Case
Statement API	Handles all xAPI Statements, the main sub-API used.	Tracking a learner as they demonstrate their ability to create an object in a 3-D environment.
Activities API	Returns the metadata of a particular activity.	The activity provider creates a title, description, and instructions of an app used to teach the moon's effect on the ocean. As the app changes, the activity provider updates the metadata.
State API	Name/value pair storage of any session data or generic data not in Statements, so long as it pertains to a particular activity	The learner was 20 questions into a 50 question exam when their device crashed. They had 37 minutes remaining.
Agents API	Used to find and identify a particular agent or learner	Doing a search on all presentations done at conferences while working for both Boeing and Lockheed Martin by John Smith.
Agent Profile API	Handles Learner Profile information	Tuesdays at 11:00 EST are good times to send learning interventions to my mobile device as I am on the subway.

3. Go granular

Do you remember learning to avoid run-on sentences in grammar class? The same rule applies to xAPI. While verbose statements are allowed, xAPI statements are easier to consume in small amounts so that they can be queried and sorted quickly. Collecting a lot of information during an activity session and then sending one giant statement at the end will make reporting and analytics more challenging. Instead, you can use parts of the statement like the context property and timestamp to describe the bigger learning picture. An effective practice commonly used by developers is to send one statement to establish the overall activity, and a collection of more granular statements for each step within the overall activity.

Another important exercise is to determine the granularity of activities. Since an activity can be anything with an IRI, deciding on what constitutes a distinct activity is important for storage and retrieval of statements. For example, if all the steps in a process were embedded in a single activity statement, it would be more difficult to produce metrics on the individual steps. By using separate statements for the steps in the process, each step is assigned a unique identifier, allowing it to be queried, reported, and analyzed more easily.

4. Establish standards for organizational adoption of xAPI

The specification currently has few requirements when it comes to defining your own vocabulary for building xAPI statements. While this openness and flexibility enables powerful new opportunities, it also presents new challenges. As a developer, you must not only be familiar with the syntax of different xAPI verbs, activities, and vocabulary found within the specification, but also with their intended use and meaning within the intended learning environment. If this is your organization's first xAPI project, you should define and document the verbs you use in your statements. This documentation is the beginning of a controlled vocabulary that will help to ensure your organization uses verbs and statements consistently in future xAPI projects. If your organization has been working with xAPI for a while, make sure you follow the verb usage standards established in previous projects.

5. Expand your xAPI knowledge and skills

The open process by which the xAPI was developed continues to be one of its strong points. There are many resources available to help you expand you xAPI knowledge and skills. Take advantage of LRS documentation, open-source prototypes, and social-media sites with active xAPI discussions.

If you want to take a deeper dive into the xAPI, the following list of resources may be helpful:

- ADL's Introduction to the Experience API: What it is, why we did it.
- The actual Experience API specification: What everyone is so excited about. This is probably the most important resource that your developer can have.
- ADL SCORM xAPI wrapper: Simplifies communication from a SCORM package to an LRS (using xAPI, naturally).
- ADL's xAPI Developer Tool and Examples: This covers the ADL open-source learning record store, client examples, a JavaScript xAPI wrapper, and ADL xAPI verb script. Everything here can also be found at the ADL GitHub page.
- Resources from ADL's first xAPI BootCamp: These are a selection of materials from ADL's June 2014 xAPI event in Orlando, FL.
- Nine Awesome Reads About the Experience API, and One Cartoon: Exactly what it says on the tin.
- The Experience API—Liberating Learning Design: The eLearning Guild's xAPI Research Report

Conclusion

This article provides a starting point for understanding the xAPI. A working knowledge of JSON, REST, and APIs will get you started writing Statements. Each of the parts of an xAPI statement work together to describe a learning activity and its relationship with learners.

Implementing actors, verbs, and objects correctly, while applying the appropriate context and granularity, enables you to track, report, and analyze learning- and performance-software solutions. Establishing organizational standards and conventions will ensure that your organization gets the most out of its xAPI implementations. A great place to start is to explore and reuse open-source resources from web developers who are already in the field creating xAPI.

From the Editor: Want more? Explore the xAPI at DevLearn!

At DevLearn 2014 later this month there will be two xAPI highlights. The first is xAPI Hyperdrive, a new competition designed to showcase the best examples of innovation using the Experience API; it takes place Tuesday, October 28 at 3:30 PM, the afternoon before the conference begins.

During the conference itself, October 29 – 31, we offer a brand-new featured track—xAPI in Practice—that is included with conference registration. The xAPI in Practice program consists of a group of sessions that run throughout the conference, offering you the chance to explore real-world case studies showing how xAPI is being used in organizations today. Learn about the business value of xAPI and the opportunities it presents to improve your practice.

Topics Covered

Development Strategies, Emerging Topics, Learning Systems, Management, Training Strategies

Learning Solutions Magazine
A publication of The eLearning Guild
Learning Solutions Magazine is a service of Focuszone Media, Inc.

Site by: Cyclone Interactive