SHARED
LEARNING
COLLABORATIVE

Data Store Draft API – RFP Guidance

slcedu.org

Data Store Draft API – RFP Guidance

SLC Project Document
January 13, 2012

This document and the information contained herein is provided on an "AS IS" basis and SLC DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Change Log

Date	Version	Name	Change Summary
1/13/12	V1	EFloyd	Initial draft submission to SLC

Table of Contents

1. Introduction

This document is part of a series of documents that contain specifications for application software and system procurement where integration with SLI technologies is required. This *Draft Specification Document* provides a draft view of a future SLC released document and is intended to be referenced in vendor RFPs. As of this writing, the SLI standards are still in development. The technical information in this document should be considered preliminary.

This document provides specifications for alignment of applications with the Data Store Application Programming Interface (API).

1.1. Structure of Document

The draft specification is divided into five sections:

- **Overview –** Provides a broad description of the SLI technology upon which the requirements are based, including use case summaries.
- **Integration Approach** – Describes one or more approaches for integrating with a core SLI technology.
- **Configuration Options** – Discusses areas of potential configurability.
- **Standards and Technologies –**Identifies applicable standards and technologies and specifies their applicability to this standard. This section also identifies related projects, initiatives, and organizations.
- **Constraints** – Specifies constraints and exclusions that a proposed solution must satisfy.

2. Overview

This section provides an overview of the API.

2.1. What is the API?

The Application Programming Interface contains the building blocks that are necessary to create SLI applications. Application access to the SLI Data Store is strictly governed by this API.

2.2. The Objective

The overall objective of the API is to provide a stable, well-defined interface for software developers to build applications that use the SLI Data Store and adapt existing applications to leverage the SLI Data Store. The API is a real-time transactional interface intended for interactive applications. It is not intended to provide bulk data load or file extract and download services.

2.3. Use Case Summaries

Selected use case summaries are provided below in order to facilitate a general understanding of the API's intended uses.

Title	Summary
Ms. Harrison, 9th grade social studies teacher	Uses SLI dashboard to identify reading comprehension as a common underlying cause of a student's performance problems in multiple subjects. Uses SLI-compatible recommendation engine to select a unit of study and age-appropriate instructional materials focusing on ELA standards. Student assessment results are reflected on Teacher Dashboard and in student's learning map and are also fed back to the SLI's data store where they help inform future recommendations for students like this one. Teacher rates materials based on this experience.
Julia Sanford, District English Language Learner Coordinator	Uses ELL app to identify students ready to be designated as Fluent that have also been flagged as at risk of dropping out based on criteria identified by a data analytics tool. Researches similar student populations across multiple districts and finds a math intervention program that is centered on a math intervention app has been successful when given access to timely enrollment information and classroom-level student performance data. It also needs to return student usage and formative assessment data to the school so teachers can see everything on their dashboard. She locates a highly-rated app that also happens to be SLI-compatible, significantly reducing the IT Department's effort to get the app integrated with district systems.
Richard Jackson, State CIO	Leverages SLI to shift the focus of state and district development program from individual silo-specific projects to shared services. He was able to deliver more, higher-quality products to the field for the same budget.

3. Integration Approach

Application integration approach depends on the scope of the application integration effort being proposed. The following table provides some general guidelines.

Scope	Integration Approach
New application	Make SLI the primary data store. Design the application around the SLI Core Entity Model and the RESTful web services architecture.
Adding support to an existing application or service for using the SLI Data Store as a primary data store.	Map SLI Core Entity Model to/from the existing data store. Review SLI security requirements. Implement adapters to translate to/from the SLI model. Assess performance and address any issues.
Adding support to an existing application for using the SLI Data Store as an auxiliary data source.	Map SLI Core Entity Model to the application's requirements. Review SLI security requirements Implement adapters to translate from the SLI model. Provide configuration switches to select the SLI source.
Bulk synchronization of data with the SLI Data Store	Don't use the API. Use the bulk Data Integration facilities instead.

The full API specification, the Introductory API Usage document and examples of API client programmatic access code are available at slcedu.org.

3.1. Relationship to Other Standards and Technologies

API makes use of and facilitates the use of the following standards and technologies.

3.2. RESTful Web Services

The SLI API is RESTful. It is designed to have predictable, resource-oriented URLs, to use HTTP response codes to indicate API errors, and to use built-in HTTP features, like HTTP authentication and HTTP verbs, which can be understood by off-the-shelf HTTP clients. JSON and XML can be returned in all responses from the API, including errors.

The SLI API is designed to be a Level 3 RESTful API in the Richardson Maturity Model. It employs the HATEOAS (Hypertext As The Engine Of Application State) model, which is covered in more depth in the book, REST in Practice[1].

Applications that integrate with the SLI Data Store will be consumers of the API RESTful web services. The table below provides links to technology-specific documentation and example code for consuming RESTful web services.

[1] http://books.google.com/books?id=1D24-cGQRdsC&lpg=PP1&pg=PP1#v=onepage&q&f=false

Implementation Technology	Source Site
ASP.NET – this source starts with a good overview of RESTful services and, near the end, provides an example of consumer code.	http://msdn.microsoft.com/en-us/library/dd203052.aspx
JAVA – see Code Fragment 2, 3, and 4 for examples of consumer code.	http://www.oracle.com/technetwork/articles/javase/index-137171.html
PHP – this source provides PHP code samples for alternative approaches to accessing yahoo web services which could be adapted to the SLI API.	http://developer.yahoo.com/php/howto-reqRestPhp.html

Note that a RESTful API, by its nature, operates as a service that responds to client-initiated requests. Therefore, the SLI API operates strictly in "pull" mode. It does not directly support an equivalent to the SIF "push" mode[2].

Examples of SLI-specific access capabilities and URLs are provided in the following section.

3.3. Core Entity Model

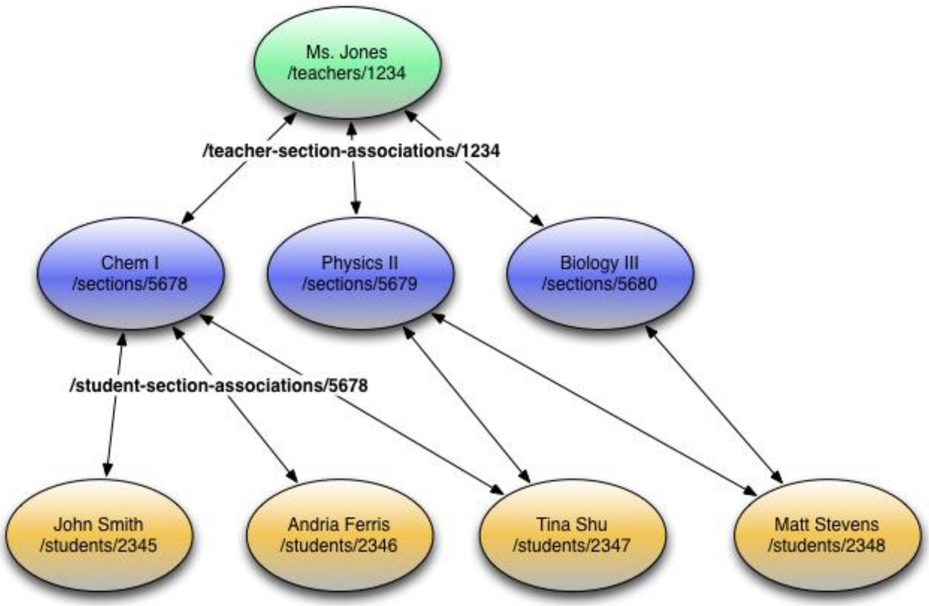
The SLI Core Entity Model (SLI CEM) is an abstract, technology-agnostic representation of the K-12 education information domain. The model includes entities that are easily recognized: school, student, course, section, among others. Those entities contain attributes that are also easily recognized, though a complete listing of entities and attributes are beyond the scope of this document.

SLI CEM contains entities along with the relationships that define how the entities interact with one another. Each entity includes a sufficient number of attributes to make the model applicable to real-world data. SLI CEM focuses on granular information rather than aggregate statistics. In addition, the model includes information that is necessary to produce aggregate and other types of statistics.

SLI CEM is expressly focused on representing the instructionally relevant classroom-level student and educator-focused data that educators can use to differentiate instruction, support individual student need and help to improve student outcomes. The SLI shares this priority use case with the Ed-Fi initiative. For this reason, the SLI CEM is based on the Ed-Fi Logical Data Model.

[2] http://www.sifsupport.com/wordpress/siflets/sif-push-and-pull-modes/

The following graphic provides an example of Teacher, Section (course), and Student entities and relationships.



In the SLI system, entities like “Ms. Jones, the Teacher” are directly referenced by a globally unique identifier. SLI creates and maintains a global unique identifier for every entity and association in the system. This is the primary method of identification by the API, and it ensures that identifiers assigned by the SLI system at creation time can be unique across very large data sets and won’t change over time. Additionally, the format for the unique identifiers follows IETF standards (RFC 4122), thus making the identifiers portable across systems. This unique identifier does not, however, replace state and district unique identifiers, which are maintained and may still be used to access entities through filtering and querying. However, it is the case that the SLI unique identifier of a containing entity or association (such as a parent EducationOrganization) is required as the base URL of a filter.

In order to ask the system which courses have been taught by Ms. Jones, we first discover this globally unique identifier. If the user making the request is the user represented in SLI as Ms. Jones, we can discover the unique identifier via the “self” resource. In this case, “self” would reference an SLI resource similar to “/teachers/1234”, where “1234” is the unique identifier for the “Ms. Jones” entity. In the responses generated by the API, resources such as “self” are returned as a set of links with corresponding URIs.

```
{
rel: self,
href: /teachers/1234
}
```

We can also use this unique identifier to provide unique URLs to retrieve Ms. Jones' relationship to all the courses she has taught, such as "/teacher-section-association/1234". Additionally, we can specify additional relationships by chaining them as path parts for the URL. So, the list of all student relationships for all of Ms. Jones' courses would be represented by "/teacher-section-association/1234/section-student-association". These additional resources would also appear as links in an API response.

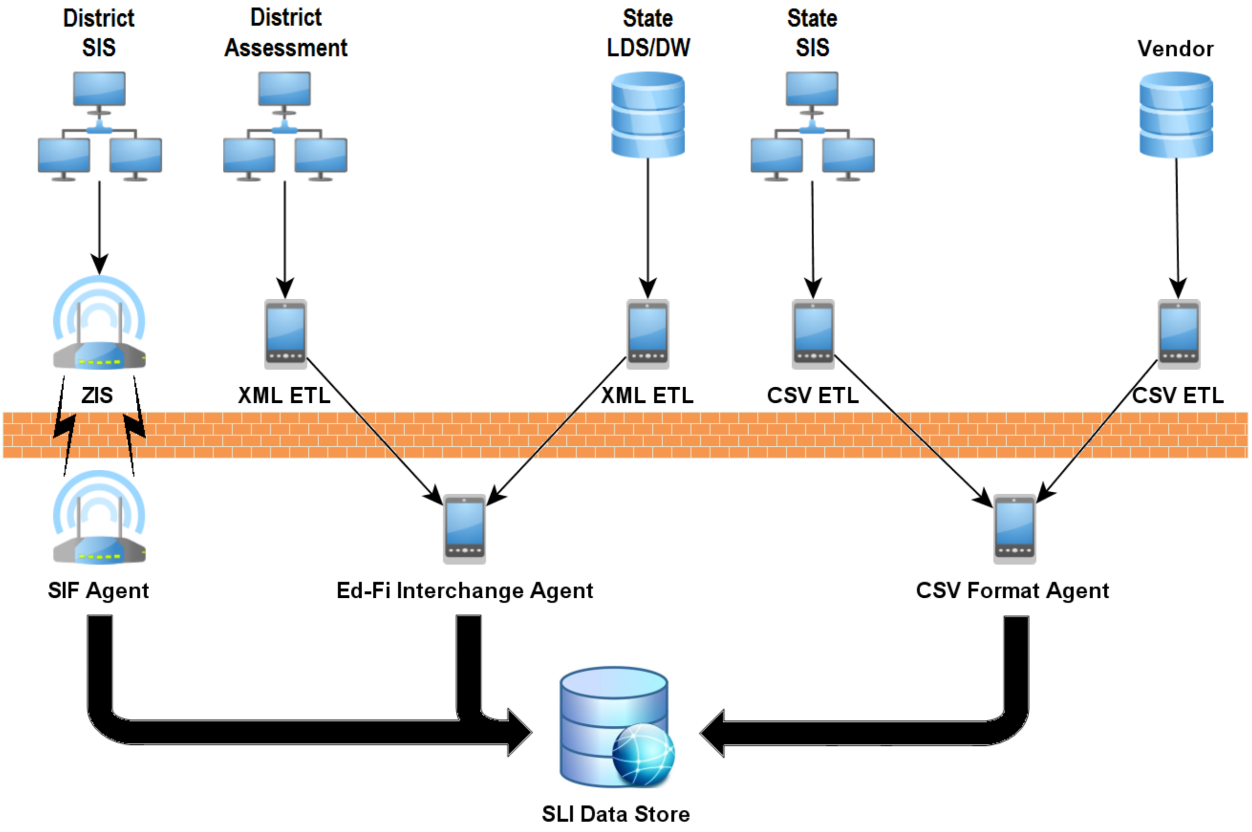
```
{
      rel: getCoursesTaught,
      href: /teacher-section-associations/1234
},
{
      rel: getStudentAssociations,
      href: /teacher-section-associations/1234/section-student-associations
},
{
      rel: getStudents,
      href: /teacher-section-associations/1234/section-student-associations/targets
}
```

The final link above for "getStudents" demonstrates the use of the keyword "targets" in the API. We append "targets" to the URL whenever we want to retrieve the actual entity records instead of the data for the relationships.

3.4. Data Integration

Data providers such as states, districts, agencies, and vendors will never interact with SLI CEM component directly. Instead, they are expected to export their data to one or more of the data exchange formats defined by SLI Bulk Data Ingestion and Validation layer. These formats include:

- Standardized XML (Ed-Fi Interchange Format)
- Comma Separated Values (CSV) Format
- Schools Interoperability Framework (SIF).

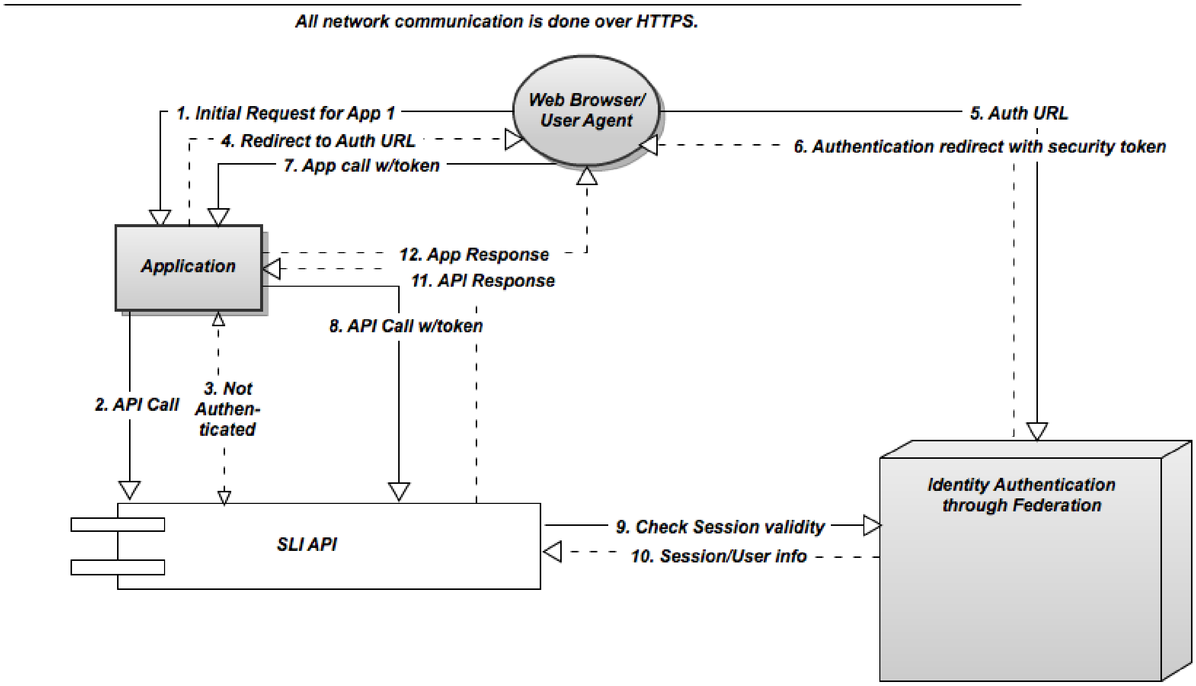


While both the API and Data Integration services provide a means to update and add information to the SLI Data Store, it is important that applications utilize these facilities for their intended purposes. Data Integration services provide facilities suitable for batch data interchange. These services are the subject of another RFP Guidance document. The API services described in this document are intended for real-time interactive applications.

3.5. *Identity Integration and Management*

Users are required to authenticate against their State (SEA) or Local (LEA) Education Agency Identity Provider, which provides the user-agent with an authentication token. Applications may then pass that authentication token, retrieved from a cookie or request header, to a Session Check API call. The Session Check API call either returns success, indicating the user has a valid session, or returns an Authentication URL, to which the application then redirects the user's browser so that the user may login using their configured Single Sign-On (SSO) capabilities. Directing the browser/user-agent to the authentication URL initiates the federated login process, which includes realm selection and authentication against the user's SEA or LEA directory. The authentication URL requires passing a referrer URL that is called once authentication is successful. This process is pictured below:

Application Authentication Diagram



In order to access data within the SLI, an application must be granted permission to access the data by an institution (SEA/LEA) Super-Administrator. A Super-Administrator, or their delegate, may grant access to their data for an application via the SLI admin portal. SLI recognizes the LEA as the ultimate arbiter of who is able to view or manipulate the LEA's data.

In addition to a blanket access to an LEA's data, an LEA Super-Administrator is able to configure permissions based on SLI entities and attributes (or collections of entities and attributes) in the model. As of this writing, constraints on LEA entity- and attribute-level permissions are not yet available. It is reasonable to expect, however, that they will include identification of a single "publisher" or "authoritative source" application for any given entity to ensure proper master data management. This is a post-alpha feature of SLI.

An application that is authorized to access data may only access data on behalf of a user. The combination of application permissions and user permissions determines what data the application may retrieve. By default, the SLI's applications (both Core and Third Party) have no access to any SEA/LEA data.

3.6. Integration Requirements by Application Category

API integration requirements vary by vendor category. The following vendor communities constitute the primary audience for this specification.

Vendor Category	Requirements
Common Core Search Engines	Navigation, Retrieval
Learning Management Systems	Navigation, Retrieval, Insert, Update
Adaptive Formative Assessment	Retrieval, Insert
Parent Portal and Outreach	Navigation, Retrieval
Adaptive Curriculum and Instructional	Navigation, Retrieval
Classroom Practice and Intervention	Navigation, Retrieval, Insert, Update
Adaptive Professional Development	Navigation, Retrieval, Insert, Update
Reporting Interfaces for Researchers	Navigation, Retrieval

4. Configuration

Areas of potential configurability include:

Area	Potential Configuration Items
Application Provision	SLC Registered Vendor Identifier, SLC Registered Application Identifier, Default Configuration values.
Installation and Configuration	Client Organization Identifier (DOE, District, School, Department, Service Provider), Initial SLI API Page, cross-reference of frequently-referenced local entity names to SLI global identifiers.

5. Standards and Technologies

The following standards and technologies are applicable to this specification:

Standard / Technology	Applicability
Representational State Transfer (REST) http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm	The SLI API is based on the REST architectural style
Hypertext Transfer Protocol (HTTP) 1.1 http://www.w3.org/Protocols/rfc2616/rfc2616.html	Primary technology for accessing RESTful web services
Hypertext As The Engine Of Application State (HATEOAS) http://roy.gbiv.com/untangled/2008/rest-apis-must-be-hypertext-driven	An elaboration by Roy Fielding on HATEOAS, an essential feature of REST
JavaScript Object Notation (JSON) http://www.json.org/	One of the supported structured message formats for request and response messages that implement the SLI API
Extensible Markup Language (XML) http://www.w3.org/TR/REC-xml/	One of the supported structured message format for request and response messages that implement the SLI API
RFC4122 Universally Unique Identifier (UUID) http://www.ietf.org/rfc/rfc4122.txt	Global unique identifiers used for entities and associations within the SLI Data Store

5.1. Related and Affiliated Efforts

Initiative / Project / Organization	Applicability
SLI Core Entity Model http://www.ed-fi.org/wp-content/uploads/2011/06/Public-Ed-Fi-Unifying-Data-Model-1.0-111111.pdf	The API provides an interface to the SLI Data Store which is organized according to the Core Entity Model.
SLI Data Integration slcedu.org	The API is one component of Data Integration. The other component is Bulk Data Ingestion and Validation
Identity Integration and Management (slcedu.org)	Ensures security and access control

Initiative / Project / Organization	Applicability
Portal Integration (slcedu.org)	SLI-developed shared applications that provide access to the Data Store
Learning Resource Metadata Initiative http://www.lrmi.net/	Tagging standards to facilitate content management and discovery
Learning Maps	Pathways through learning objective standards

6. Constraints

To be compliant with this specification, solutions will be subject to the following constraints.

1. The providing vendor and the application must be registered with SLC and must be configured with a unique SLC-provided Application Identifier.
2. The application must be authorized by the LEA Super Administrator to access local data stored in the SLI Data Store.
3. The application must provide a mechanism to pass a user ID and password to the authentication service link provided in an API authentication-redirect response.