Getting Started for ADF App Developers

Introduction

The Singapore Student Learning Space (SLS) is a learning portal to support new ways of learning for students.

In the SLS, students and teachers interact in the Virtual Learning Environment (VLE). The Application Development Framework (ADF) was conceptualised to allow the VLE to be extended with external, specialised tools. This is achieved by allowing the integration of external web applications with the SLS, with data exchange facilitated through API.

The ADF comprises of **Learning Tool Interoperability (LTI) v1.3** and **GraphQL API***. For more info on LTI, visit http://www.imsglobal.org/spec/lti/v1p3/

In this document, we share sample use cases, and the technical documentation for integration with the ADF. The ADF currently only supports the integration of web applications that are maintained independently by third-party developers.

Key Concepts

Each ADF app is:

- A web application with a web server to make API calls to SLS;
- Allowed to be installed into group(s) that is either a class taking a subject in a level (e.g. Secondary 2 Biology) or a group created in SLS without a subject/ level;
- Able to make API calls using a valid Access Token (e.g. get information about groups that have the app installed);
- Launched by users from SLS as a standalone app, within a lesson, or in the context of a VLE group; and
- Displayed either in a new tab or iframe within the SLS VLE application when launched.

Sample Use Case Walkthrough

For a sample use case of how an app is integrated into SLS through ADF, please refer to Sample Walkthrough.

ADF APIs

API Versioning

Semantic versioning is used to version the APIs. Breaking changes that are not backwards compatible will be released in major versions, while backwards compatible modifications would be released in minor versions.

API Specifications

For LTI integration, please refer to Adopting LTI v1.3.

What's Next

Please contact us to discuss potential integration opportunities. We provide a few tools/details to help with integration:

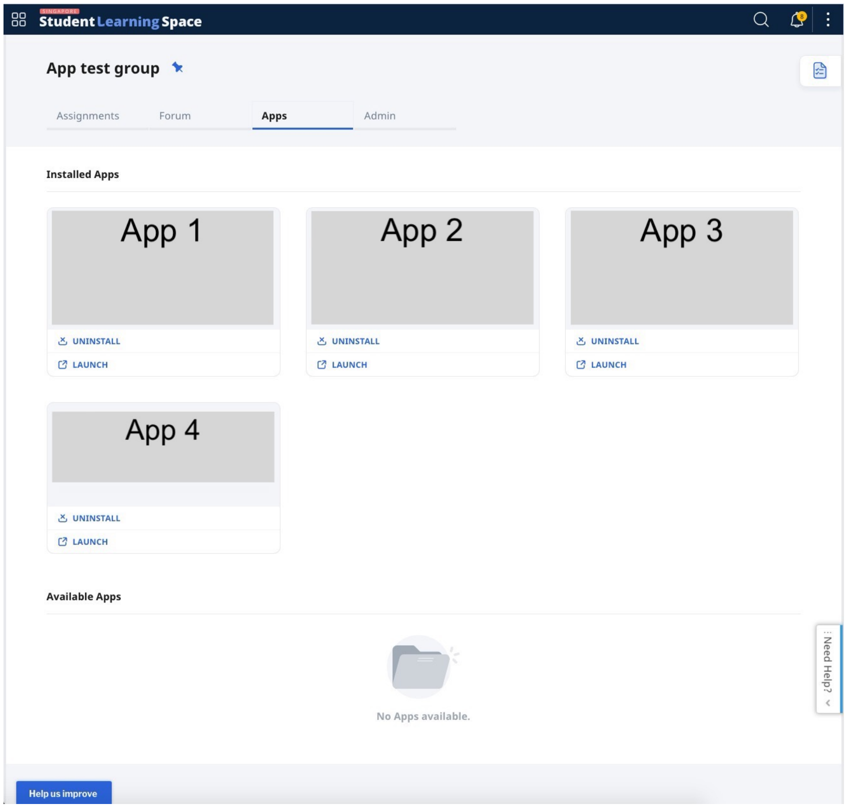
- Sample credentials for testing LTI integration such as clientId, OIDC authorisation URL, etc.; and
- Sample SLS accounts to view VLE as a student and teacher.

*Apps currently integrated with SLS via GraphQL will continue to be supported.

Copyright © 2022 Ministry of Education, Singapore. All rights reserved.

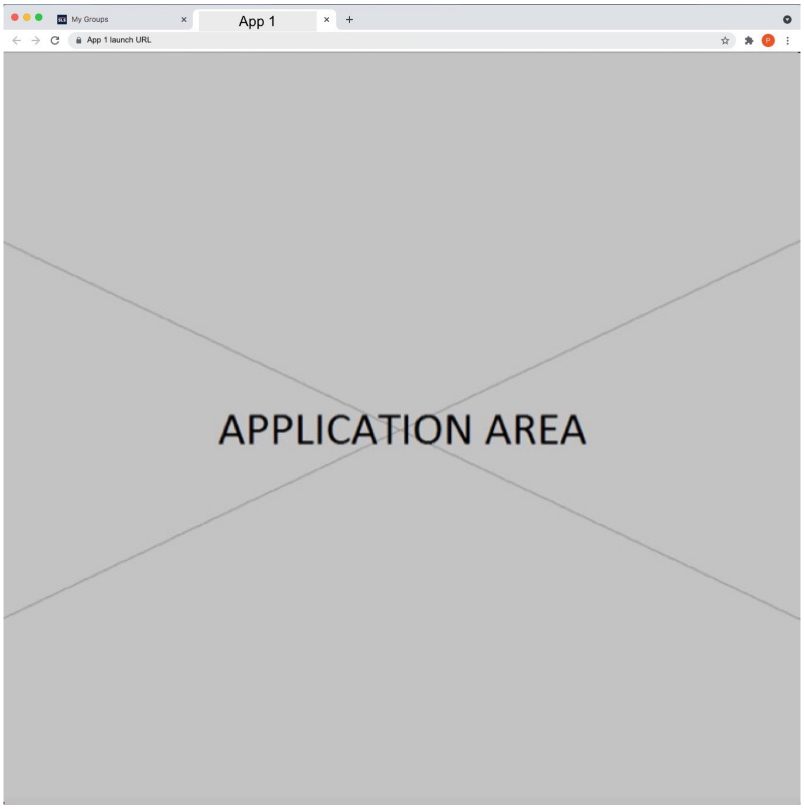
Sample Screens

Click the Apps tab to access the applications. A menu with the installed and available applications will show up.



Clicking on the launch icon will launch the application (in a new tab)

Copyright © 2022 Ministry of Education, Singapore. All rights reserved.



Copyright © 2022 Ministry of Education, Singapore. All rights reserved.

Adopting LTI v1.3

1. What is LTI

Learning Tools Interoperability is a learning standard from IMS. It defines how an Learning Management System (LMS, or Platform) to interact with third-party content provider (Tool).

In a nutshell, it is an e-learning standard.

LTI 1.3 Core Specs (http://www.imsglobal.org/spec/lti/v1p3/)

LTI Deep Linking 2.0 Specs (https://www.imsglobal.org/spec/lti-dl/v2p0)

IMS Security Framework 1.0 (https://www.imsglobal.org/spec/security/v1p0/)

2. How LTI works

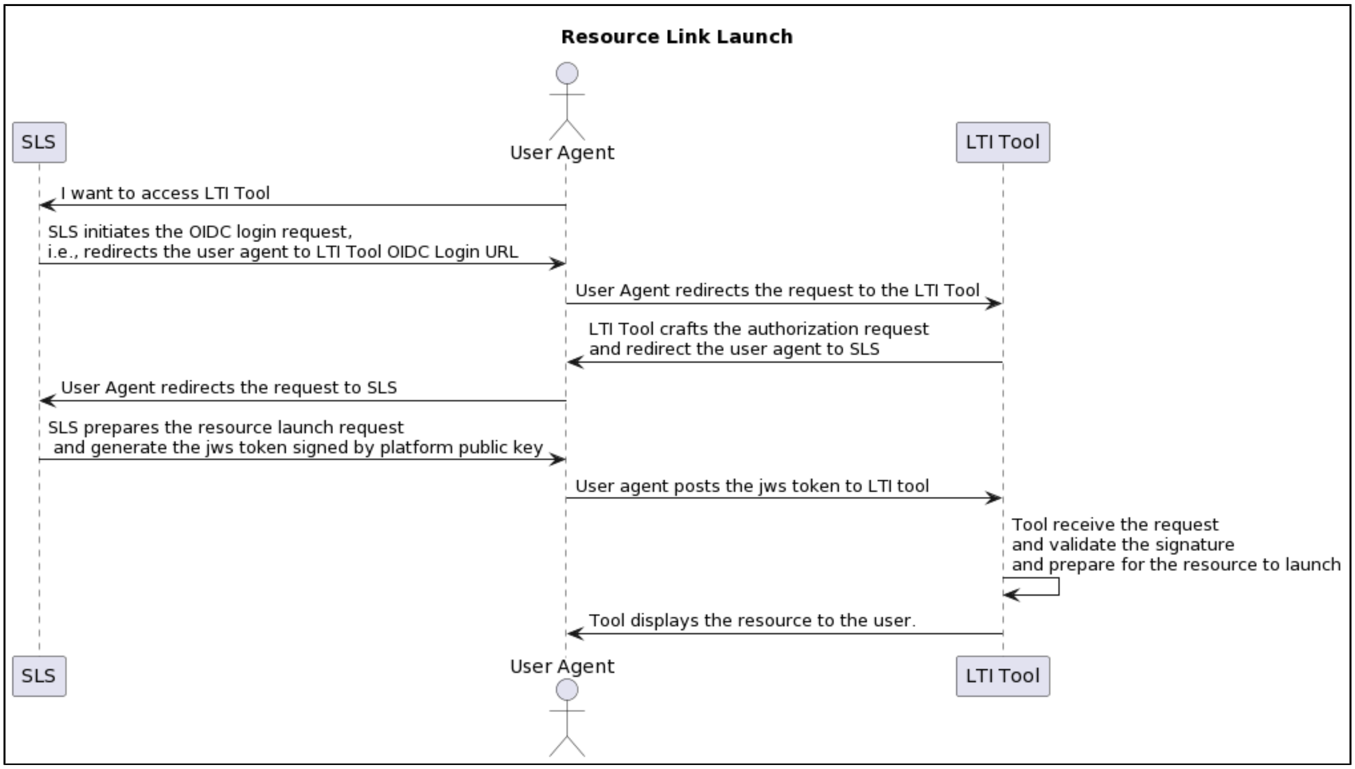
There are a lot of materials to read and watch. For the start, just check out some YouTube videos.

YouTube: LTI 1.3 Security and OpenID Connect (https://www.youtube.com/watch?v=d_Otmti7xKA)

YouTube: LTI 1.3 Resource Launch and OIDC (https://www.youtube.com/watch?v=g3y4vwtP6vQ)

YouTube: LTI 1.3 Deeplinking (https://www.youtube.com/watch?v=e5aano-h5Co)

3.1 Resource Link Launch



I want to access LTI Tool

SLS initiates the OIDC login request,
i.e., redirects the user agent to LTI Tool OIDC Login URL

User Agent redirects the request to the LTI Tool

LTI Tool crafts the authorization request
and redirect the user agent to SLS

User Agent redirects the request to SLS

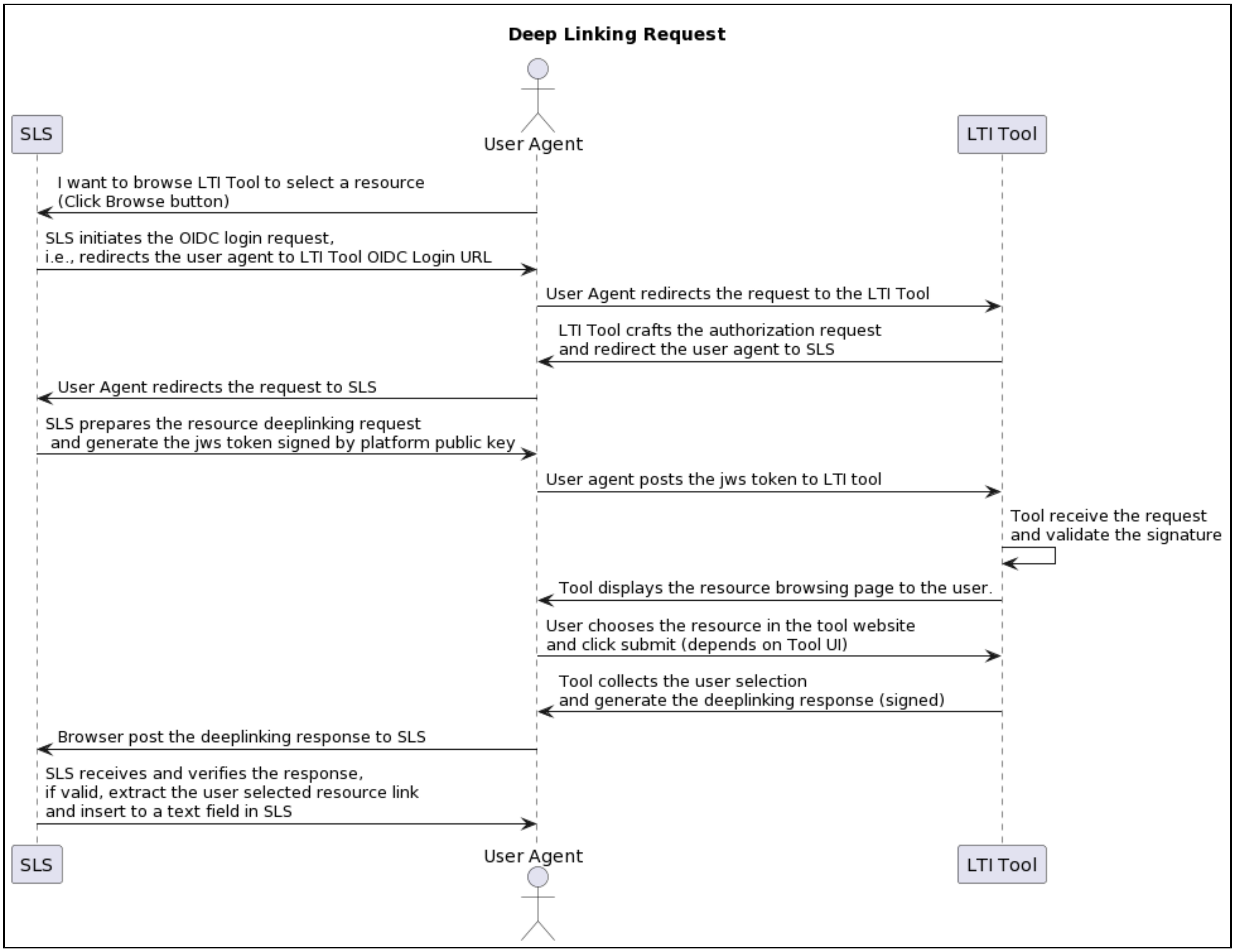
SLS prepares the resource launch request
and generate the jws token signed by platform public key

User agent posts the jws token to LTI tool

Tool receive the request
and validate the signature
and prepare for the resource to launch

Tool displays the resource to the user.

3.2 Deep Linking Request


Deep Linking Request
SLS
User Agent
LTI Tool
I want to browse LTI Tool to select a resource
(Click Browse button)
SLS initiates the OIDC login request,
i.e., redirects the user agent to LTI Tool OIDC Login URL
User Agent redirects the request to the LTI Tool
LTI Tool crafts the authorization request
and redirect the user agent to SLS
User Agent redirects the request to SLS
SLS prepares the resource deeplinking request
and generate the jws token signed by platform public key
User agent posts the jws token to LTI tool
Tool receive the request
and validate the signature
Tool displays the resource browsing page to the user.
User chooses the resource in the tool website
and click submit (depends on Tool UI)
Tool collects the user selection
and generate the deeplinking response (signed)
Browser post the deeplinking response to SLS
SLS receives and verifies the response,
if valid, extract the user selected resource link
and insert to a text field in SLS
SLS
User Agent
LTI Tool

Key Concepts:

> ***Context****. LTI uses the term* context *where you might expect to see the word "course". A context is roughly equivalent to a course, project, or other collection of* resources *with a common set of users and roles. LTI uses the word "context" instead of "course" because a course is only one kind of context (another type could be "group" or "section").*
>
> ***Resource****. Typically, within a context, users can integrate many LTI content items, or* resources*, sometimes arranging them into folders like "Week 1" or "Pre-Work". Conceptually, these platform integrations serve the same general purpose as any other type of item within the structure of a context's available content. In particular, commonly, users may scatter multiple* LTI links *through the content structure for a context that is linked to a particular resource. A platform MUST distinguish between each of these* LTI links *by assigning a* `resource_link_id` *to an* LTI Link*.*
>
> *While all of the LTI links integrated within a single context will share the same* `context_id`*, each link within the context will have a unique* `resource_link_id`*. This allows the hosting* tool *to differentiate the content or features it shows on a resource-by-resource basis (within a context) by, for example, providing configuration options such as a resource picker to the instructor or administrator after launching from a particular link.*

Copyright © 2022 Ministry of Education, Singapore. All rights reserved.

4. Place you can launch LTI app

There are three main use integration points for LTI Apps.

Use Case	Description
Install to Class Group	Once an App is approved by Adminstrator, teachers will be able to install the App into Class Group, so that students can access it through App Library in Class Group.
Embed into Lesson	Teachers can embed a resource url from the approved App directly into Lesson, so that students can access the resource through either Assignment or Self Study. If App supports the optional LTI Deep Linking 2.0 Specs, SLS also allows teachers to easily select the resource through Deeplinking.
Launch from App Library	If App supports the option for an independent url that can launch the App separately, this option can be set by the Administrator to allow an App to be globally searched and launched.

5. How to On-board a LTI App

5.1 What SLS needs from the App during registration

1. OIDC Login URL
2. Default Launch URL
3. Deep Linking URL (if applicable)
4. Public Key from the App
    - RSA, Key size: 2048
    - Base64 Encoded
    - Optional for basic resource launch

Sample Code

```
KeyPairGenerator keyPairGenerator = KeyPairGenerator.getInstance("RSA");
keyPairGenerator.initialize(2048);
KeyPair keyPair = SecurityUtils.generateRSAKeyPair();

PublicKey publicKey = keyPair.getPublic();
String publicKeyText = Base64.getEncoder().encodeToString(publicKey.getEncoded());
```

5.2 What SLS provides to the registered App

1. Client Id: this will be the audience claim in JWT sent from SLS
    a. App UUID
2. OIDC Authorization URL
    a. https://<SLS URL>/lti/authorize/<app uuid>
3. JWKS Endpoint: for app to retrieve SLS's public key
    a. https://<SLS URL>/lti/key-set/<app uuid>
4. Issuer: SLS has development, staging, pre-production, sandbox, production environment. Different environment will have different issuer.
    a. E.g. https://vle.sandbox.sls.ufinity.com (change according to environment)
5. User Identity: the following fields are provided.
    a. Unique Id
    b. Name
    c. Role (Learner or Instructor)
6. Deployment ID

6. Optional LTI Parameter(s)

1. Client ID : https://www.imsglobal.org/spec/lti/v1p3#client_id-login-parameter
    a. SLS will send the optional client id during the OIDC login request.

7. Common Integration Issues

✓ Because LTI integration involves multiple parties, it's very important to request App side to show some meaningful error information.

1. Configuration issues By App side (App Side)
    a. App side must configure the client id for SLS (the App UUID issued by SLS)
    b. App side must configure SLS's authorization URL
    c. App side must support JWKS to retrieve SLS' s public key (SLS will generate different public key for each app.)
    d. App side must ensure the issuer is configured
    e. App side must ensure all configurations are according to environment (Each environment will have different set of values)

Application Performance Guidelines

Suggested Service Level Agreement (SLA)

1. Concurrency: Expected concurrency based on App demographics; To be discussed with App Point-of-Contact (POC)
   a. For reference: SLS supports approximately 100,000 users under normal concurrently with the ability to ramp up to 300,000 users during Full Home-Based Learning (FHBL)
2. Transaction Timeout: 10 Seconds for 95% of the time with system availability at 99.9% per connection
3. Uptime: At least 90% uptime with scheduled downtime aligned with SLS maintenance schedule (updated on SLS website from June 2022)
   a. SLS Partnerships Team shall be notified of any scheduled downtime at least 4 weeks in advance. This applies to all environments, including non-production, to facilitate test plans on SLS side.
4. Service Support: 1st Level help desk support should be provided during school days from 9am to 9pm; POC should advise SLS Office on 1st and 2nd level service support details to allow SLS help desk to redirect appropriate calls to App
5. App should maintain separate communications channel to users to inform them of downtime and also keep SLS Office informed through the POC
6. Bug Fix is in alignment to SLS Business Impact Levels (BILs); refer to table:

Business Impact Level (BILs)	Response Time	Status Reporting	Resolution Time
BIL 1 (all days)	Within **ONE** hour	Every **TWO** hours	Within **FOUR** hours
BIL 2 (all days)	Within **TWO** hours	Every **EIGHT** hours	Within **TWENTY-FOUR** hours
BIL 3 (all days)	Within **TWO** hours	**Daily**	Within **SEVENTY-TWO** hours

Business Level Alignment with SLS Integration and Partnership Principles

1. SLS is not just a "gateway" for external Apps
   a. Once App is integrated, SLS will be seen as a gatekeeper/implicit endorser
2. Resources/Apps should not
   a. Duplicate current or upcoming SLS core features (the App can offer enhanced or customised features)
   b. Solicit further business in-App, especially to students
   c. Promote undesirable content or teaching and learning (T&L) behaviour; e.g. test/exam drill and practice, etc...
3. App should be financially supported for a minimum of 3 years (excluding and after development work has concluded)
4. Cost to Ministry of Education Division should be clearly highlighted through the POC

Copyright © 2022 Ministry of Education, Singapore. All rights reserved.

Change Log

v1.0.0-20180302

- Initial draft of documentation

v1.1.0-20180629

- Updated User object ID field to UUID

v1.2.0-20180904

- Changed SubjectGroup type to Group to support SLS Group

v1.3.0-20181216

- Added allGroups query to support retrieval of all groups installed with the app
- Added School object in User and Group types to return school information of a user and group respectively
- Updated descriptions and comments

v1.3.1-20190304

- Added openInNewTab field in Assignment type schema to support option to open App within iframe or new tab
- Updated in descriptions and comments

v1.4.1-20200309

- Added new EventType, i.e., LAUNCH_URL for Context to support embedding App in SLS Lesson
- Updated descriptions and comments

v1.5.1-20210805

- Deprecate Assignment, Task and Notification objects

v2.0.0-20220505

- Introduce LTI v1.3 specifications and sample use cases
- Removed GraphQL specifications
- Included App Performance Guidelines
- Included LTI v1.3 optional parameters
- Added App Library support

v2.0.1-20230105

- Updated Application Performance Guidelines SLA : Notify SLS for any downtime

Copyright © 2022 Ministry of Education, Singapore. All rights reserved.