Introduction

Global Platform Services provide a common set of Learning Management System (LMS) and Core services on which global School products can be built within Pearson. The GPS services have reusable learning management and content distribution capabilities that are self-serviced, well-documented, easily discoverable, and usable by multiple School product teams around the world. All GPS services are built to a common set of technical standards and governance principles.

The services have been designed and developed using a REST architecture. All of the REST API calls are documented on the API console.

This document is intended as a guide for those wishing to take advantage of the functionality built by the Shared Services team. It is a guide to getting started with each one of the services. To read about all the new GPS features created during this release, consult the Global Platform Services Release Notes v.1.6.

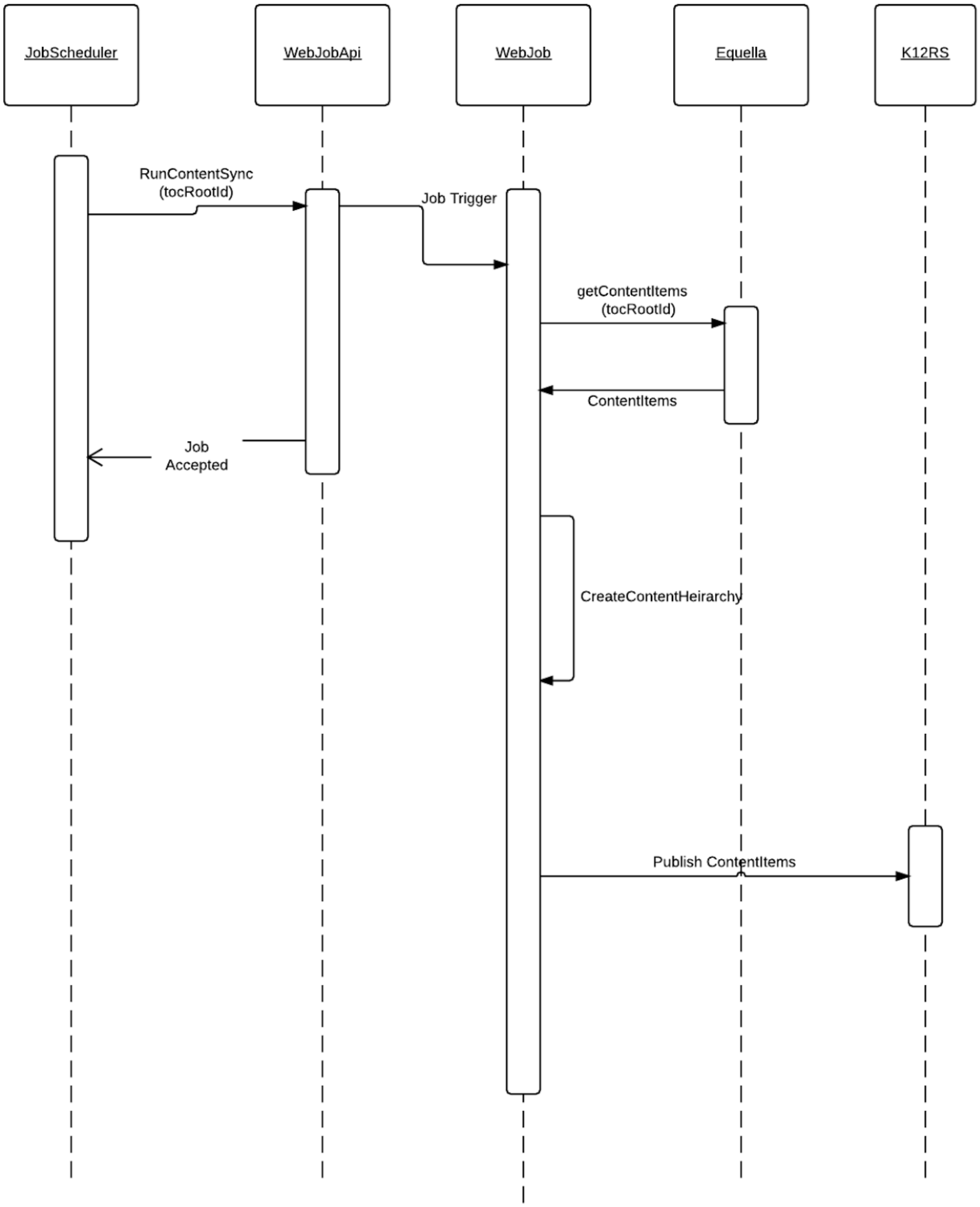
Initial Content Setup

Global Platform Services (GPS) acts as an intermediate connector between different Pearson products, specifically:

- Passaporte, which is as a user management system for sending messages to other Pearson products about users, roles, section assignments, and more.
- K12RS, which is a reporting service.
- EQUELLA, which is a content and resource service.

At first deployment of GPS, references for the content structure and items in EQUELLA are sent to K12RS via a job scheduler, web job API, and web job that will be run by the Global Platform Services team. A diagram of how this setup process works is shown below:

Equella ContentBulk Pull Service



SCORM Player

The SCORM player is able to play back and capture the learners' interactions with SCOs. This enables an LMS to track the learners' progress and use the data later for reporting and analytics. When used with the GPS content service, the GPS framework can automatically submit completed learner interactions and scores to the GPS assignment and grading services. The SCORM Player is implemented as an AngularJS directive.

The SCORM Player supports the following features:

- navigation through multiple SCO's through a table of contents that is parsed from the default organization structure in the imsmanifest
- implementation of a SCORM 2004 RTE to capture changes to the cmi model when the learner is interacting with a SCO
- integration with GPS services (if enabled)
- synchronization of scores in grading service. This creates a single location for retrieving grade data for assignments, both auto-graded and teacher graded. The scores are auto-populated and are translated to the same grading scale used by the assignment.

The project comes with a sample host application and SCORM packages to demonstrate the functionality.

AngularJS and jQuery are required to use the SCORM player.

Additionally, as part of the assessment service, consumers can use the API to return item-level student assessment results. API consumers can retrieve the SCORM interaction data for assessment SCOs in a well-structured format that is consistent with QTI Results Reporting. Consumers can generate assessment reports with that data. The service returns the following information:

- question label
- student response
- correct/incorrect
- score (called outcomes)
- total score for assessments

For more detailed information and instructions on the SCORM player, please refer to the readme file at:
https://devops-tools.pearson.com/stash/projects/LMSPOC/repos/scormplayer/browse/readme.md

Search

The Search API enables developers to call to query the LMS platform for available learning resources, in order to present those resources to the user. This API queries an EQUELLA repository, which administrators can register. It only returns live items. It supports freetext/keyword searches, metadata searches, facet counts, sorting, and pagination. The search response also supports TOC (table of contents) metadata fields and thumbnails that load as file attachments.

A developer of an LMS application is also able to issue a search that filters by content bundles, enabling the developer to display only content relevant to the user (by license, subscription, or other association). To accomplish this:

- content bundle IDs from the content bundle API are automatically populated into the EQUELLA metadata
- content bundle ID field supports multiple values
- content bundle ID field can be used to filter searches and to return facetcounts

At time of publication, the Search API includes the following:

- Gets all search domains
- Gets a search domain
- Creates a new search domain
- Updates an existing search domain
- Deletes an existing search domain
- Removes documents from the search domain
- Searches and returns data within a search domain
- Searches and returns facet counts within a search domain

Each function fits into one of these four categories: Post, Put, Get, and Delete.

Gets All Search Domains

This API gets information for all registered search domains. It returns metadata for each search domain. To use this API, select the response content type (at time of publication, application/json is the only option), then click on **Try it out!**

If there is a problem with what you entered, then one of these errors will be returned:

- **400**: Bad Request
- **403**: Forbidden
- **404**: Resource not found

If you are successful, then a response code of 200—along with a request URL, response body, and response headers—will be returned.

Gets a Search Domain

When you enter a search domain name, this API gives you information for that specific search domain. It returns metadata for that search domain.

To use this API, select the response content type (at time of publication, application/json is the only option), enter the domain name, then click on **Try it out!**

If there is a problem with what you entered, then one of these errors will be returned:

- **400**: Bad Request
- **403**: Forbidden
- **404**: Resource not found

If you are successful, then a response code of 200—along with a request URL, response body, and response headers—will be returned.

Creates a New Search Domain

This API creates a new search domain, when you fill in the search domain metadata.

To use this API, fill in the domain and searchDomain, select the parameter content type (at time of publication, application/json is the only option), and click on **Try it out!**

If there is a problem with what you entered, then one of these errors will be returned:

- **400**: Bad Request
- **403**: Forbidden
- **404**: Resource not found

If you are successful, then a response code of 204—along with a request URL, response body, and response headers—will be returned.

Updates an Existing Search Domain

This API updates the metadata for an existing search domain.

To use this API, enter the domain and searchDomain, select the parameter content type (at time of publication, application/json is the only option), then click on **Try it out!**

If there is a problem with what you entered, then one of these errors will be returned:

- **400**: Bad Request
- **403**: Forbidden
- **404**: Resource not found

If you are successful, then a response code of 204, a request URL, response body, and response headers will be returned.