SMARTERAPP SPECIFICATION

for

TEST INTEGRATION SYSTEM API

Authored by
Fairway Technologies, Inc.

Updated

29 August 2018

Revision History

Version	Revision Description	Author/Modifier	Date
1.0	Initial Release	Jeff Jones (Fairway)	August 29, 2018

Table of Contents

Purpose

This document is intended to help integrate TDS solutions more easily with the Test Integration System (TIS). Each of TIS's endpoints is detailed, describing its purpose and its request format.

TIS Overview

The TIS is responsible for:

- Receiving a test result from TDS (Test Delivery System)
- Sending it to THSS (Teacher Hand Scoring System) for hand scoring of items that require human scoring
- Receiving item scores back from THSS
- Inserting item scores into the file received from TDS
- Scoring the test
- Sending the scored test to downstream systems via SFTP

TIS Components

The TIS consists of the following 3 modules/parts:

TDS Receiver

This is a REST endpoint that receives test results in an XML format from the Test Delivery System. Each result received is inserted into a database where it is picked up and processed by the TIS Service.

TIS Service

This is a Windows service that continuously looks for new test results in the database that have not yet been processed. Once it finds these, it picks them up and processes them (either by sending to THSS, inserting scores from THSS, scoring the test or sending the test downstream).

TIS Scoring Daemon

This is a web application that talks to the THSS (Teacher HandScoring system) and is responsible for receiving item scores from THSS and sending it to the TIS Service for further processing.

TIS Operational Overview

A TRT is submitted by TDS. It may or may not have individual items scored in that TRT. The scenarios are:

- Tests completely scored
- Tests partially scored
- Multiple (usually two) assessments that need scoring

TDS sends TRT to TIS - some or all items have scores. TIS' 'TDS Receiver' webapp writes into DB as a row into xml repository table. Column called 'location' - when from TIS, it's set to 'source'. This represents the initial state of TRT processing, before TIS changes it.

The TIS windows service takes the 'source' TRT from the xmlrepository DB and validates it, which can change the status to 'reject' and log into several places, including the filesystem and the QC_ValidationException table.

If valid and all items are scored for a single-assessment exam, it creates a new row with location 'destination', setting the location of the old row from 'source' to 'archive' to store the original TRT. The destination TRT will have an aggregate score added that is calculated by TIS. For a combined assessment with multiple TRT's, TIS will wait for all the TRT's to arrive before creating the destination row.

If valid but some items need to be scored, it changes the location to 'destination'.

TIS adds elements to TRT associated to the exam/opportunity level as opposed to the item level.

Once all items are scored and saved into the TRT, TIS will add a new row to the DB with location 'destination', adding all the scores provided by THSS for each individual item, and the aggregate score for all items at the assessment level. Once a TRT is fully scored, the TRT is sent to RDW for storage and analysis.

Interaction with THSS

TIS leans on THSS to provide the scores for non-machine-scorable items. TIS looks through the destination TRT for unscored items and sends those items with the student's responses to THSS where they will sit until a teacher manually scores them. As a teacher scores responses, they are individually sent back to the TIS scoring daemon's API, and the scores are saved into the destination TRT.

REST endpoints and communication with external systems

The Test Integration System is built to communicate with all the peer and down-stream systems using a secured REST APIs (using OAuth). The token for secured communication would be supported/provided by the OpenAM system.

TIS Receiver - Data Exchange Protocol

Name	Description
Protocol	HTTPS RESTful API
Authentication and Authorization	OAuth 2.0
Base URL	http:// *[TIS_BASE_URL]*

Endpoint	HTTP Verb	Content-Type	Description	Body Content	Required Permissions	Responses
/api/testresult?statusCallback={statusCallbackUriEncoded}	POST	application/xml	Accept incoming TRT files: Receive a TDSReport from TDS. A callback URL is provided (url encoded) which will be saved with the file and used to send a acknowledgement when TIS has processed the file.	XML formatted TRT/TDSReport: The TRT format is documented at: http://www.smarterapp.org/documents/TestResultsTransmissionFormat.pdf		An HTTP status code and a message (optional) as the content if there's an error: 200: OK 400: request is not formatted correctly or does not contain the expected data. 500: an unhandled exception occurred while attempting to insert the file into the database
/api/testresult?statusCallback={statusCallbackUriEncoded}	POST	application/xml	Load a test package into TIS: This endpoint loads an administrative test package into TIS. This endpoint is accessed by the support tool loader, and does not need to be accessible to the rest of TDS.	XML formatted test package: The XSD for the test package formatted file is documented at: https://github.com/SmarterApp/TDS_SupportTool/blob/develop/client/src/main/resources/xsd/v4-test-package.xsd		An HTTP status code and a message (optional) as the content if there's an error: 201: CREATED 400: request is not formatted correctly or does not contain the expected data. 500: an unhandled exception occurred while attempting to insert the file into the database

Endpoint	HTTP Verb	Content-Type	Description	Body Content	Required Permissions	Responses
/api/assessments/[assessmentId]	DELETE	application/xml	Deletes and assessment from TIS: This endpoint removes an assessment from TIS. This endpoint is accessed by the support tool loader, and does not need to be accessible to the rest of TDS.	Assessment ID: Example: (SBAC_PT)MSB-Multiform-Mathematics-3		An HTTP status code and a message (optional) as the content if there's an error: 200: OK if the delete was successful 400: Bad Request if the key is null or empty 500: Internal Server Error if the delete operation failed.

TIS to TDS - Data Exchange Protocol

Name	Description
Protocol	HTTPS RESTful API
Authentication and Authorization	OAuth 2.0
Base URL	{statusCallbackUriEncoded} Note – this value comes from a previous TIS Receiver POST call

Endpoint	HTTP Verb	Content-Type	Description	Body Content	Required Permissions	Responses
/student/TIS/API/tisReply	POST	application/json	Reply from TIS to TDS: The reply that is in response to a submitted score request when TIS has complete processing the request.	Single JSON object: Example: {"oppKey" : "AA19A641-C782-4426-8586-E7D77EDF2D02", "success" : false, "error" : "too late, student left the building", “trt”: “<TRT XML CONTENTS>”} Notes: 'success' element is boolean. 'error' element is optional.		An HTTP status code and a message (optional) as the content if there's an error: 200: OK 400: request is not formatted correctly or does not contain the expected data.

TIS to THSS - Data Exchange Protocol

Name	Description
Protocol	HTTPS RESTful API
Authentication and Authorization	OAuth 2.0
Base URL	http:// *[THSS_BASE_URL]*

Endpoint	HTTP Verb	Content-Type	Description	Body Content	Required Permissions	Responses
/api/test/submit	POST	application/xml	ItemScoreRequest: This request is sent from TIS to THSS to request manual scoring on items. The student responses are included verbatim in the document so the grading can be completed. THSS will store these items and present them to the teacher for scoring.	Example: <ItemScoreRequest callbackUrl="http://localhost/ItemScoreClient/Scored.axd"> <ResponseInfo itemIdentifier="I-100-1001" itemFormat="GI"> <StudentResponse encrypted="true"><![CDATA[<Question></Question>]]></StudentResponse> <Rubric type="Uri" cancache="true" encrypted="true">c:\rubrics\rubric.xml</Rubric> <ContextToken><![CDATA[xxxxxxx]]></ContextToken> <IncomingBindings> <Bind name="chosenoption" type="string" value="A"/> </IncomingBindings> <OutgoingBindings> <Binding name="objectsetACount"/> </OutgoingBindings> </ResponseInfo> </ItemScoreRequest>		An HTTP status code and a message (optional) as the content if there's an error: 200: OK 400: request is not formatted correctly or does not contain the expected data.

THSS to TIS - Data Exchange Protocol

Name	Description
Protocol	HTTPS RESTful API
Authentication and Authorization	OAuth 2.0
Base URL	http:// *[TIS_Web BASE_URL](e.g. http://tis-web-deployment.sbtds.org:8080)*

Endpoint	HTTP Verb	Content-Type	Description	Body Content	Required Permissions	Responses
/ItemScoreClient/Scored.axd	POST	application/xml	ItemScoreResponse: When THSS gets a set of items scores from a user's grading session, it will send the scores back to TIS using the callback previously received in the ItemScoreRequest. TIS will record these scores into its DB.	Example: <ItemScoreResponse> <Score> <ScoreInfo scorePoint="0" maxScore="2" scoreDimension="overall" confLevel="23.5,45.6" scoreStatus="Scored"> <ScoreRationale> <Propositions> <Proposition name="Jar" description="The student correctly orders the liquids by density." state="NotAsserted" /> <Proposition name="Earth" description="The student correctly ranks the layers of Earth by density." state="NotAsserted" /> <Proposition name="Match" description="The student understands that Veg. Oil is the Atmosphere, Dish Soap is the Ocean, Milk is the Crust, and Honey is the Core." state="NotAsserted" /> </Propositions> <Bindings /> <Message> <![CDATA[Your response earned a score of 0. Full credit requires: BothOrder]]> </Message> </ScoreRationale> <SubScoreList> <!-- a list of children ScoreInfo instances --> </SubScoreList> </ScoreInfo> <ContextToken> <![CDATA[This token is what we received with the score request]]> </ContextToken> <ScoreLatency>10</ScoreLatency> </Score> </ItemScoreResponse>		An HTTP status code and a message (optional) as the content if there's an error: 200: OK 400: request is not formatted correctly or does not contain the expected data.

TIS to RDW - Data Exchange Protocol

Name	Description
Protocol	HTTPS RESTful API
Authentication and Authorization	OAuth 2.0
Base URL	These base URLs are configured in WebServiceSettings of TISService\App.config under DW1/DW2 WebService nodes

Endpoint	HTTP Verb	Content-Type	Description	Body Content	Required Permissions	Responses
These endpoints are configured in WebServiceSettings of TISService\App.config under DW1/DW2 WebService nodes	POST	application/xml	Submit scored TRT to Reporting Data Warehouse: When an opportunity has completed scoring, the resulting TRT with scores is sent to all configured Data Warehouse web services for long term storage and analysis.	XML formatted TRT/TDSReport: The TRT format is documented at: http://www.smarterapp.org/documents/TestResultsTransmissionFormat.pdf		An HTTP status code and a message (optional) as the content if there's an error: 200: OK 400: request is not formatted correctly or does not contain the expected data.