
DataShop Public API

A way to
programmatically get
and set data in
DataShop

The DataShop Team + others
4/26/2016

Contents

1	API	4
2	Note on format	4
	Method Title	4
3	What is an ID?	4
4	Representational state transfer (REST).....	4
4.1	HTTP Response Status Codes	5
5	Authentication	5
5.1	Access Credentials.....	5
5.2	Using your Access Credentials to Sign a Request	5
6	Getting Data	10
6.1	Get Dataset Metadata.....	10
6.2	Get Sample Metadata	13
6.3	Get Transactions	14
6.4	Get Student-Step Records.....	19
7	External Analyses	24
7.1	Access and ownership.....	24
7.2	Add External Analysis.....	24
7.3	Get External Analyses Metadata.....	26
7.4	Get External Analysis.....	26
7.5	Delete External Analysis.....	27
8	Custom Fields.....	29
8.1	Access and ownership.....	29
8.2	Data Types.....	29
8.3	Add Custom Field	29
8.4	Get Custom Field Metadata	31
8.5	Set Custom Field	31
8.6	Delete Custom Field	32
9	Learning Curve	33
9.1	Access.....	33
9.2	Get Learning Curve.....	33
10	KCM Import.....	35

10.1	Access.....	35
10.2	Import KCM	35
Appendix A.....		38
All Possible Result Codes.....		38
Appendix B.....		40
Revision History.....		40

1 API

This API is a list of methods for programmatically getting and modifying DataShop data. In broad strokes, the goal for the API is to provide enough functionality for getting transaction data with various parameters, and adding, modifying, and deleting custom-field data.

As of April 2016, we've implemented:

- Authentication
- Get Dataset Metadata
- Get Sample Metadata
- Get Transactions
- Get Student-Step Records
- Add and Get External Analyses
- Get External Analysis metadata
- Add and Delete Custom Fields
- Get Custom Fields metadata
- Set Custom Fields at the Transaction level
- Get Learning Curve categorizations
- Import KC Models

2 Note on format

Method Title

`https://url/[?optional_section]`

URL to perform the action.

3 What is an ID?

The DataShop API expects you to reference various objects by "ID", a unique identifier for each dataset, sample, custom field, external analysis, or transaction in the repository. The ID of any of these can be determined by performing a request to list the various items, which lists the IDs in the response. For example, a request for datasets will list the ID of each dataset in the "id" attribute of each dataset element. In the case of a custom fields and external analyses, the ID can also be discovered from the response of a successful request that adds a new custom field or external analysis.

4 Representational state transfer (REST)

The DataShop API design generally follows a "REST" approach to web services. In doing so, we've modeled DataShop as a collection of resources which can be retrieved and manipulated using HTTP. (See the "[RESTful Web Services](#)" section of Wikipedia's REST article.)

We do deviate from the REST approach slightly in one way: the DataShop web service ignores the type of HTTP method in each request. That means the service relies on unique URLs with verbs in them (e.g., "get" and "delete") instead of distinguishing requests from similar URLs by HTTP method (e.g., GET or DELETE).

4.1 HTTP Response Status Codes

DataShop Web Services will, in many cases, return an HTTP status code that you can use to determine how to analyze the body of the HTTP message. For example, if you know the HTTP response code was an error such as “401 Unauthorized”, then you know that the content of the body of the message will be XML describing the error.

For a list of web services result codes and the HTTP status codes that appear with them, see Appendix A.

5 Authentication

Authentication is a process for identifying and verifying who is sending a request. To use DataShop web services, you need to identify yourself as the sender of each request. This is accomplished by sending a digital signature that is derived from a pair of public/private access keys.

Note: To use DataShop web services, you must first request access. Visit <http://pslcdatashop.web.cmu.edu/WebServicesCredentials> to read the User Agreement and request access. You will need to first register with DataShop if you haven’t already.

Note: In addition to this documentation, consult the Java source of [the sample DataShop web services client](#) for code examples. See methods *signedRequest*, *encrypt*, and *httpTimestamp* in particular.

5.1 Access Credentials

Access credentials refers to your two access keys. You can retrieve or reset your access keys from DataShop’s [Web Services Credentials](#) page.

Access Key ID

Your Access Key ID identifies you as the party responsible for service requests. Include it with each request you send to us.

Secret Access Key

Your Access Key ID has a Secret Access Key associated with it. Use your Secret Access Key to calculate a signature to include in requests to DataShop web services. Your Secret Access Key is a secret, and should be known only by you and DataShop. You should never include your Secret Access Key in your requests to DataShop web services. You should never email your Secret Access Key to anyone. It is important to keep your Secret Access Key confidential to protect your account.

5.2 Using your Access Credentials to Sign a Request

To authenticate to DataShop, you will:

1. Create a request
2. Create a specific type of message signature
3. Send the request and signature to DataShop Web Services

See Figure 1 – Authentication process .

DataShop Web Services will then:

1. Retrieve your Secret Access Key
2. Create the same type of signature
3. Compare the two signatures

If the two signatures match, the request is considered authenticated; if they fail to match, then the request fails authentication.

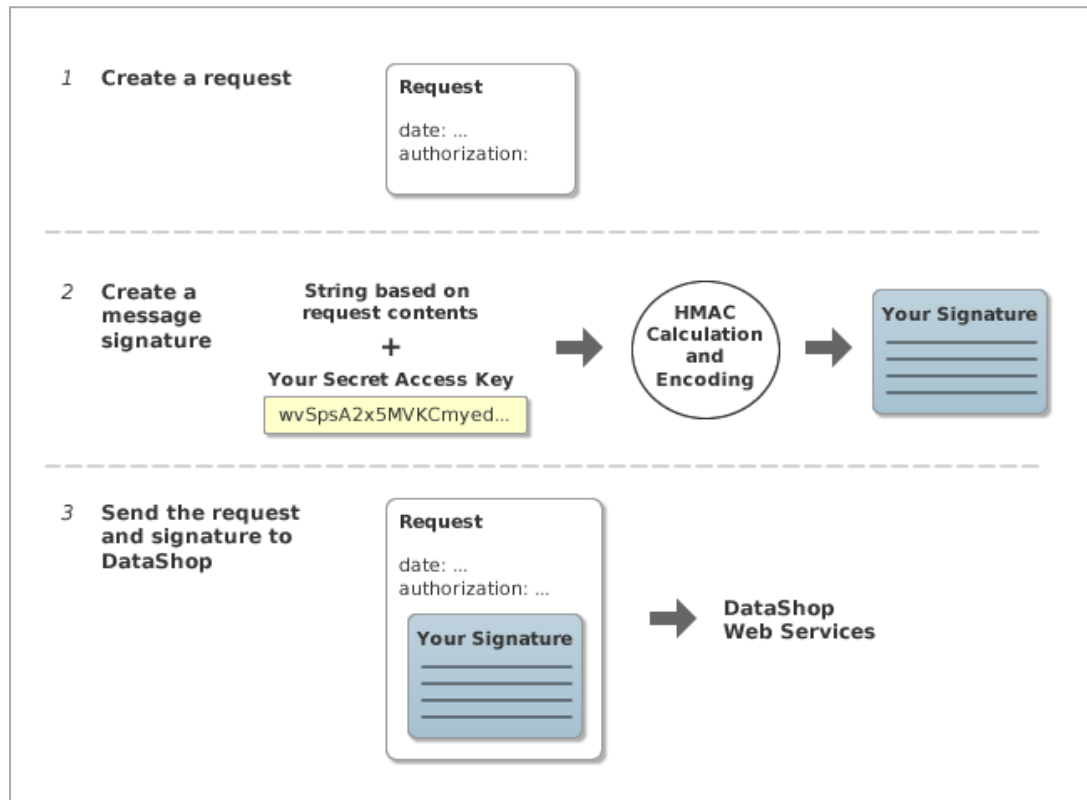


Figure 1 – Authentication process

Step 1: Create a request

In this step, you create an HTTP request in your program which has a standard request line, a number of request headers, and an optional message body (some DataShop Web Services requests require a body while others are only a URL).

To authenticate, your request must contain the following request headers:

date
authorization

Set “date” according to the *timestamp* format (see Table 1 - Contents of the string to sign).

You will set the value of “authorization” in Step 3, described below.

Step 2: Create a message signature

Each request you send must include an *HMAC-SHA* signature calculated with your Secret Access Key. [HMAC-SHA](#) is an industry-standard message authentication procedure that uses the [SHA-1](#) cryptographic hash function in combination with a secret key to create a message signature. DataShop uses it to verify the authenticity of your request.

The signature must be URL-encoded in UTF8. This signature will be included in the “authorization” header as described in Step 3.

The string you use to calculate the signature (the *string to sign*) is composed as follows:

```
method
contentMD5
contentType
timestamp
path
```

Each line in the string to sign is separated by a line break. See Table 1 - Contents of the string to sign for a description of each line.

Note: See the Java source of [the sample DataShop web services client](#) for a code example.

Table 1 - Contents of the string to sign

part of the string	description	example
method	the HTTP method used—GET, PUT, POST, or DELETE	GET
contentMD5	an MD5 hash of the message content if PUT or POST, an empty string if GET or DELETE	
contentType	MIME type of content if PUT or POST, an empty string if GET or DELETE	text/xml
timestamp	an HTTP date string (see <code>httpTimestamp</code> method below)	Tue, 20 Oct 2009 15:22:52 GMT
path	the portion of the request URL that is after “services” and before the first question mark (“?”).	/datasets/1/samples/3

```
/**
 * Format for HTTP date strings.
 * @return format for HTTP date strings
 */
private String httpTimestamp() {
    if (httpDateFmt == null) {
        httpDateFmt = new SimpleDateFormat("EEE, dd MMM yyyy HH:mm:ss zzz");
        httpDateFmt.setTimeZone(TimeZone.getTimeZone("GMT"));
    }
    return httpDateFmt.format(new Date());
}
```

}

Finally, you must **URL-encode** the signature you've created in UTF-8 character encoding. This is the `application/x-www-form-urlencoded` MIME format. Your programming language may have a utility method that will perform this encoding for you. For reference, see the Java class java.net.URLEncoder, which you can use to perform UTF-8 URL-encoding in Java.

Step 3: Send the request and signature to DataShop Web Services

After calculating the signature and URL-encoding it, you can now attach it to the request.

Set the "authorization" header as follows, where *publicApiToken* is your public token and *signature* is the HMAC-SHA signature you create in Step 2. :

`"DATASHOP " + publicApiToken + ":" + signature`

For example:

`DATASHOP OKLFETPSJJJJFSGR6D8E:wzXDj0%2BEY3iosiwcgH%2FsYktDZPM%3D%0D%0A`

Note the space after "DATASHOP".

Send the complete request to DataShop Web Services.

5.2.1 Example

You want to retrieve a description of the first sample ("All Data") for the dataset with id "1".

The request line looks like this:

GET `/services/datasets/1/samples/1` **HTTP/1.1**

The request body is empty.

You create a "date" header:

date: Tue, 20 Oct 2009 16:59:47 GMT

Then create the string to sign:

method: GET
contentMD5: <empty>
contentType: <empty>
timestamp: Tue, 20 Oct 2009 16:59:47 GMT
path: /datasets/1/samples/1

or

GET

Tue, 20 Oct 2009 16:59:47 GMT
/datasets/1/samples/1

Create the HMAC-SHA signature from the string to sign and URL-encode it in UTF-8:

wzXDj0%2BEY3iosiwcgH%2FsYktDZPM%3D%0D%0A

And with that, form the “authorization” header:

authorization: **DATASHOP**
OKLFETPSJZJJFSGR6D8E:wzXDj0%2BEY3iosiwcgH%2FsYktDZPM%3D%0D%0A

The full request will then look like this:

GET /services/datasets/1/samples/1 HTTP/1.1
date: Tue, 20 Oct 2009 16:59:47 GMT
authorization: **DATASHOP**
OKLFETPSJZJJFSGR6D8E:wzXDj0%2BEY3iosiwcgH%2FsYktDZPM%3D%0D%0A
accept: text/xml
user-agent: Java/1.6.0_13
host: https://pslcdatashop.web.cmu.edu
connection: keep-alive

6 Getting Data

6.1 Get Dataset Metadata

[http://pslcdatashop.web.cmu.edu/services/datasets/\[?id\]](http://pslcdatashop.web.cmu.edu/services/datasets/[?id])

Get list of datasets matching the parameters, or information about a single dataset (if an id is specified).

6.1.1 Request Parameters

access

all, viewable or editable. Default is **viewable**. All means return all the datasets, those that are public, view, edit, admin, or private (see descriptions below). Viewable means return datasets for projects on which you have “view”, “edit”, or “admin” privileges, in addition to public datasets. Editable means return only datasets for projects on which you have the “edit” or “admin” privilege.

Note: If you request a single dataset *and* specify a value for the “access” parameter, DataShop will respect the access parameter. This will result in an empty set in the case that the dataset requested does not match the access parameter specified (e.g., you specified “access=editable” but the dataset is only viewable).

verbose

true or false. Default is **false**. False returns the simpler, less verbose description for each dataset (see directly below) while true returns the more verbose description for each dataset (see further below). A verbose response is only possible for datasets you can view or edit (i.e., not private datasets).

6.1.2 Response Fields

access is your level of access to the dataset. DataShop models each user's dataset access as either "public", "view", "edit", "admin", or "private", which is the level of access they have to the dataset's containing project.

- **public:** you and any other registered DataShop user can view the data and download associated files
- **view:** you can view the data and download associated files.
- **edit:** in addition to the above, you can create samples, add or modify papers and files, add kc sets, and add kc models
- **admin:** in addition to the above, you can edit dataset and project metadata, add or modify datasets in the project, and manage project access.
- **private:** you cannot view the data or edit anything about the dataset

For a full description of the access levels, see <http://pslcdatashop.org/help?page=administration>

number_of_samples is the total number of samples including those that are accessible and those that are not shared.

number_of_accessible_samples is the number of samples that you can access, ignoring private samples owned by others.

number_of_unique_steps is the number of unique steps, where uniqueness is defined as a step within a specific problem hierarchy (the curriculum location where the problem appears). The same step attempted by two students equals only one unique step.

Note: The order of datasets in the response XML is indeterminate.

6.1.3 Example request:

GET <https://pslclatashop.web.cmu.edu/services/datasets/31>

6.1.4 Example response on success:

```
<?xml version="1.0" encoding="UTF-8"?>
<pslc_datashop_message result_code="0" result_message="Success.">
  <dataset id="31">
    <name>Pittsburgh Science of Learning Center Stoichiometry Study 1</name>
    <project>Stoichiometry Studies</project>
    <domain>Science</domain>
    <learnlab>Chemistry</learnlab>
    <pi>bleber</pi>
    <start_date>2005-09-02</start_date>
    <end_date>2006-06-07</end_date>
    <status>complete</status>
    <access>edit</access>
    <public>yes</public>
    <released>yes</released>

    <number_of_students>34</number_of_students>
    <number_of_unique_steps>16453</number_of_unique_steps>
    <number_of_steps>124882</number_of_steps>
    <number_of_transactions>245093</number_of_transactions>
    <number_of_samples>17</number_of_samples>
    <number_of_accessible_samples>3</number_of_accessible_samples>
    <number_of_kc_models>4</number_of_kc_models>
  </dataset>
</pslc_datashop_message>
```

6.1.5 Example response on error:

```
<?xml version="1.0" encoding="UTF-8"?>
<pslc_datashop_message
  result_code="-1"
  result_message="Error. Dataset 31 is not valid." />
```

6.1.6 Example request (verbose)

GET <https://pslclatashop.web.cmu.edu/services/datasets/31?verbose=true>

6.1.7 Example verbose XML response (more metadata):

```
<?xml version="1.0" encoding="UTF-8"?>
<pslc_datashop_message result_code="0" result_message="Success.">
  <dataset id="31">
    <name>Pittsburgh Science of Learning Center Stoichiometry Study 1</name>
    <project>Stoichiometry Studies</project>
    <domain>Science</domain>
    <learnlab>Chemistry</learnlab>
    <pi>bleber</pi>
    <start_date>2005-09-02</start_date>
    <end_date>2006-06-07</end_date>
    <status>complete</status>
    <access>edit</access>
    <public>yes</public>
    <released>yes</released>

    <curriculum>geometry 2005</curriculum>
    <tutor></tutor>
    <description></description>
    <has_study_data>yes</has_study_data>
    <hypothesis>Lorem ipsum</hypothesis>
    <school>Wilkinsburg High School</school>
    <additional_notes> School demographics for 2005-6 unless noted otherwise.
      Converter Version 4.15
      Loaded to production on 4/15/08 by Kyle.
    </additional_notes>

    <number_of_students>34</number_of_students>
    <number_of_unique_steps>16453</number_of_unique_steps>
    <number_of_steps>124882</number_of_steps>
    <number_of_transactions>245093</number_of_transactions>
    <number_of_samples>17</number_of_samples>
    <number_of_accessible_samples>3</number_of_accessible_samples>
    <number_of_kc_models>4</number_of_kc_models>

    <kc_model id="25">
      <name>Automatic-Model</name>
      <number_of_kcs>456</number_of_kcs>
      <observations_with_kcs>470117</observations_with_kcs>
      <logistic_regression_model_status>unable to run
    </logistic_regression_model_status>
      <cross_validation_status>unable to run</cross_validation_status>
    </kc_model>

    <kc_model id="32">
      <name>Single-KC</name>
      <number_of_kcs>1</number_of_kcs>
      <observations_with_kcs>126057</observations_with_kcs>
      <number_of_parameters>10</number_of_parameters>
      <logistic_regression_model_status>complete
    </logistic_regression_model_status>
      <aic>142117.54</aic>
```

```

        <bic>142468.24</bic>
<log_likelihood>-2759.55</log_likelihood>
<cross_validation_status>complete</cross_validation_status>
<cross_validation_rmse>0.3457</cross_validation_rmse>
<cross_validation_number_of_observations>126056</cross_validation_number_of_observat
ions>
<cross_validation_number_of_parameters>10</cross_validation_number_of_parameters>
    </kc_model>

    <kc_model id="14">
        <name>Unique-step</name>
        <number_of_kcs>9157</number_of_kcs>
        <observations_with_kcs>0</observations_with_kcs>
        <number_of_parameters>18314</number_of_parameters>
        <logistic_regression_model_status>not        scheduled        to        run
    </logistic_regression_model_status >
        <cross_validation_status>not scheduled to run</cross_validation_status>
    </kc_model>

</dataset>
</pslc_datashop_message>

```

6.2 Get Sample Metadata

[https://pslcdatashop.web.cmu.edu/services/datasets/\[id\]/samples/\[?id\]](https://pslcdatashop.web.cmu.edu/services/datasets/[id]/samples/[?id])

Get the list and descriptions of samples matching the parameters.

6.2.1 Request Parameters

access

viewable or editable. Default is **viewable**. Viewable means return only samples you can view.

Editable means return only samples you own and can therefore edit.

Note: If you request a single sample *and* specify a value for the “access” parameter, DataShop will respect the access parameter. This will result in an empty set in the case that the sample requested does not match the access parameter specified (e.g., you specified “access=editable” but the sample is only viewable).

verbose

true or false. Default is **false**. False returns the simpler, less verbose description for each sample (see directly below) while true returns the more verbose description for each sample (see further below), including a description of each filter in each sample.

6.2.2 Example request:

GET <https://pslcdatashop.web.cmu.edu/services/datasets/422/samples/37>

6.2.3 Example XML response for a sample query:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```

<pslc_datashop_message result_code="0" result_message="Success.">
  <sample id="37">
    <name>CWCTC-GeoArea06-07</name>
    <description>Just the CWCTC students on the area units (GEO-PA)</description>
    <owner>aleven</owner>
    <number_of_transactions>350384</number_of_transactions>
  </sample>
</pslc_datashop_message>

```

6.2.4 Example verbose XML response for a sample query:

```

<?xml version="1.0" encoding="UTF-8"?>
<pslc_datashop_message result_code="0" result_message="Success.">
  <sample id="37">
    <name>CWCTC-GeoArea06-07</name>
    <description>Just the CWCTC students on the area units (GEO-PA)</description>
    <owner>aleven</owner>
    <number_of_transactions>350384</number_of_transactions>
    <filter>
      <column>School Name</column>
      <operator>=</operator>
      <filter_text>cwctc</filter_text>
    </filter>
  </sample>
</pslc_datashop_message>

```

6.2.5 Example response on error:

```

<?xml version="1.0" encoding="UTF-8"?>
<pslc_datashop_message
  result_code="-4"
  result_message="Error. Sample 37 is not valid for dataset 422." />

```

6.3 Get Transactions

[https://pslcdatashop.web.cmu.edu/services/datasets/\[id\]/\[?samples/id\]/transactions](https://pslcdatashop.web.cmu.edu/services/datasets/[id]/[?samples/id]/transactions)

Returns zero or more transaction records in tab-delimited form, optionally zipped. If a sample is not specified, then the “All Data” sample is returned. Note that the specified dataset must be released for you to get its transactions.

See “Interacting with Cached Data” note below.

6.3.1 Request Parameters

zip

true or false. Default is **false**. If true, transaction data is zipped.

cols

Optional list of column names, comma-delimited. Default is to **include all columns**. See Table 2 - Transaction columns below for the list of column values to choose from. To specify whether or not to include custom-field columns, use the **cfs** parameter.

cfs

all, none, or a comma-delimited list of custom field IDs. Default is **none**. Optionally specify whether to include all custom fields, no custom fields, or a subset of custom fields. A custom field is a user-created column of data associated with transactions. See section 8 of this API for a description of how to create and modify custom fields, as well as how to retrieve a description of custom fields for a dataset.

headers

true or false. Default is **true**. Optionally specify whether a header row should be included in the output. If false, a header row is omitted.

limit

Maximum number of transactions to retrieve. Default is **100**, maximum is 5,000.

offset

From the beginning of the data, the number of transaction rows to skip. Default is **0**. For example, an offset of 0 would return rows starting with the 1st row, while an offset of 100 would return rows starting with the 101st row. Loop through transactions in batches by combining offset and limit. For example, to get rows in batches of 100, request **offset=0&limit=100** in the first iteration of the loop, which will return rows 1-100. Then in the next iteration of the loop, request **offset=100&limit=100**, which will return rows 101-200.

6.3.2 Example request (tab-delimited format, subset of columns):

GET

https://pslcdatashop.web.cmu.edu/services/datasets/114/samples/21/transactions?cols=row,anon_student_id,session_id,time,duration,student_response_type,tutor_response_type,problem_name,step_name,attempt_at_step,outcome,selection,input,feedback,kcs

6.3.3 Example response:

See Table 4 - Example tab-delimited transaction data.

6.3.4 Example response on error:

```
<?xml version="1.0" encoding="UTF-8"?>
<pslc_datashop_message
  result_code="-7"
  result_message="Error. Invalid column: feedbacks." />
```

Interacting with Cached Data

DataShop maintains a cached export file for each sample, which is used to provide data when you request it. Due to changes made in the web application or more data being logged, the cached file can fall out-of-date with what is available in the DataShop database. When this happens, a request for the sample will return an error stating that the cached file is not yet available. DataShop will then start caching the sample (if it hasn't already). You should try your request again after some time has passed. The amount of time needed for caching varies between a few minutes and a few hours, depending on the size of the sample.

Table 2 - Transaction columns

Column name in request	Column(s) contained in response	Data type	Notes on value
row	Row	numeric	
tx_id	Transaction Id	string	
anon_student_id	Anon Student Id	string	
session_id	Session Id	string	
time	Time	date	yyyy-MM-dd HH:mm:ss
time_zone	Time Zone	string	Possibly empty
duration	Duration (sec)	numeric	Null values are represented with a dot (".")
student_response_type	Student Response Type	string	Possibly empty
student_response_subtype	Student Response Subtype	string	Possibly empty
tutor_response_type	Tutor Response Type	string	Possibly empty
tutor_response_subtype	Tutor Response Subtype	string	Possibly empty
problem_hierarchy	Problem Hierarchy	string	
problem_name	Problem Name	string	
problem_view	Problem View	numeric	
problem_start_time	Problem Start Time	date	yyyy-MM-dd HH:mm:ss
step_name	Step Name	string	Possibly empty
attempt_at_step	Attempt At Step	numeric	Possibly empty if step name not provided in data.
outcome	Outcome	string	Possibly empty
selection	Selection	string	
action	Action	string	
input	Input	string	
feedback	Feedback Text, Feedback Classification	string, string	Both possibly empty
help_level	Help Level	string	Possibly empty
total_num_hints	Total # Hints	numeric	Possibly empty
condition	Condition Name, Condition Type These two columns will appear for each condition that applies to the student (there may be more than one).	string, string	Possibly empty
kcs	KC (<i>KC Model Name</i>), KC Category (<i>KC Model Name</i>) Includes all KCs for all models. One column per KC per model. Multiple columns per KC model in the case where more than one KC is associated with the transaction.	string	Possibly empty
school	School	string	Possibly empty
class	Class	string	Possibly empty

Note: For more documentation on these columns and the semantics of the format, see the [Import File Verification Tool documentation](#) and [Export By Transaction](#) as documented in the DataShop web application.

Table 3 – Custom field columns included when the “cfs” parameter is ‘all’ or a list of custom field IDs

Column name	Description	Data type	Notes on value
CF (<i>Custom Field Name</i>)	A single column per custom field requested.	<i>variable</i>	

Table 4 - Example tab-delimited transaction data

Row	Anon Student Id	Session Id	Time	Duration (sec)	Student Response Type	Tutor Response Type	Problem Name	Step Name
1	D8DD5	D8DD5-Jan07-15-12-09	2008-01-07 15:12:18.0		ATTEMPT	RESULT	CHARGE1A	(CHOOSE-ANSWER MC-1 1)
2	D8DD5	D8DD5-Jan07-15-12-09	2008-01-07 15:12:45.0	27	ATTEMPT	RESULT	CHARGE1A	(CHOOSE-ANSWER MC-2 2)

Attempt At Step	Outcome	Selection	Input	Feedback Text	Feedback Classification	KC (Default)	KC Category (Default)	KC (Unique-step)	KC Category (Unique-step)
1	CORRECT	MC-1	1			SELECT-MC-ANSWER		KC1797	
1	CORRECT	MC-2	2			SELECT-MC-ANSWER		KC1241	

6.4 Get Student-Step Records

[http://pslcdatashop.web.cmu.edu/services/datasets/\[id\]/\[?samples/id\]/steps](http://pslcdatashop.web.cmu.edu/services/datasets/[id]/[?samples/id]/steps)

Returns zero or more student-step records in tab-delimited format, optionally zipped. If a sample is not specified then the “All Data” sample is returned. Note that the specified dataset must be released for you to get its student-step records.

See “Interacting with Cached Data” note above.

6.4.1 Request Parameters

zip

true or false. Default is **false**. If true, the data is zipped.

cols

Optional list of column names, comma-delimited. Default is to **include all columns**. See Table 5 – Student-step columns below for the list of column values.

cfs [coming soon]

all, none, [ids]. Default is **all**. Optionally specify whether to include all, none, or some custom fields. If all, include all custom fields. If none, no custom fields are included. If a comma-delimited list of ids, then the specified custom fields are included, e.g. cfs=123,456.

kcms

all or none. Default is **all**. Optionally specify whether to include all KC models or none. If all, include all KC models. If none, no KC models are included. See Table 7 below for list of columns included with each KC model. (A future release of web services will support requesting a subset of KC models by specifying a comma-delimited list of KC model IDs).

headers

true or false. Default is **true**. Specify whether a header row should be included in the output. If false, a header row is omitted.

limit

Maximum number of student-step rows to retrieve. Default is **100**, maximum is 5,000.

offset

From the beginning of the data, the number of student-step rows to skip. Default is **0**. For example, an offset of 0 would return rows starting with the 1st row, while an offset of 100 would return rows starting with the 101st row. Loop through rows in batches by combining offset and limit. For example, to get rows in batches of 100, request **offset=0&limit=100** in the first iteration of the loop, which will return rows 1-100. Then in the next iteration of the loop, request **offset=100&limit=100**, which will return rows 101-200.

6.4.2 Example request (subset of columns):

GET

https://pslccdatashop.web.cmu.edu/services/datasets/114/samples/21/steps?cols=row,anon_student_id,condition,problem_hierarchy,problem_name,step_name,step_duration,first_attempt&kcms=all

6.4.3 Example response:

See Table 7 - Example tab-delimited step data.

6.4.4 Example response on error:

```
<?xml version="1.0" encoding="UTF-8"?>
<pslc_datashop_message
  result_code="-7"
  result_message="Error. Invalid column: time_zones." />
```

Table 5 – Student-step columns

Column name in request	Column(s) contained in response	Data type	Notes on value
row	Row	numeric	
anon_student_id	Anon Student Id	string	
problem_hierarchy	Problem Hierarchy Single column (different than Get Transactions).	string	
problem_name	Problem Name	string	
problem_view	Problem View	numeric	
step_name	Step Name	string	
step_start_time	Step Start Time	date	yyyy-MM-dd HH:mm:ss
first_transaction_time	First Transaction Time	date	"
correct_transaction_time	Correct Transaction Time	date	"
step_end_time	Step End Time	date	"
step_duration	Step Duration (sec)	numeric	Null values are represented with a dot (".")
correct_step_duration	Correct Step Duration (sec)	numeric	Null values are represented with a dot (".")
error_step_duration	Error Step Duration (sec)	numeric	Null values are represented with a dot (".")
first_attempt	First Attempt	string	Possibly empty
incorrects	Incorrects	numeric	Possibly empty
hints	Hints	numeric	Possibly empty
corrects	Corrects	numeric	Possibly empty
condition	Condition In the case of a student assigned to multiple conditions (factors in a factorial design), condition names are separated by a comma and space. This differs from the format in Get Transactions.	string	

Note: For more documentation on these columns and the semantics of the format, see [Student-Step Rollup](#) as documented in the DataShop web application..

Table 6 – KC model columns included when requested using the “kcms” parameter

Column name	Description	Data type	Notes on value
KC (KC Model Name)	In the case of multiple KCs assigned to a single step, KC names are separated by two tildes (“~~”).	string	Possibly empty
Opportunity (KC Model Name)	Opportunity number for the listed KC(s). In the case of multiple KCs assigned to a single step, opportunity number values are separated by two tildes (“~~”) and are given in the same order as the KC names.	numeric	Possibly empty
Predicted Error Rate (KC Model Name)	Predicted error rate for the listed KC(s). In the case of multiple KCs assigned to a single step, predicted error rate values are separated by two tildes (“~~”) and are given in the same order as the KC names.	numeric	Possibly empty

Note: For each KC model, the three columns listed in the above table will appear in the output.

Important: The format of the KC model columns returned by Get Student-Step Records is different from the Get Transactions web service. In the Get Transactions web service, multiple KCs associated with a transaction result in **multiple columns** (one column per KC). In the Get Student-Step Records web service, multiple KCs are contained in a **single value** and delimited with two tildes (“~~”). The same rule is applied to the Opportunity and Predicted Error Rate columns.

Table 7 - Example tab-delimited step data

Row	Anon Student Id	Condition	Problem Hierarchy	Problem Name	Step Name	Step Duration (sec)	First Attempt
1	D8DD5	a	Unit 1, Section 2	CHARGE1A	(CHOOSE-ANSWER MC-1 1)	5	correct
2	D8DD5	a	Unit 1, Section 2	CHARGE1A	(CHOOSE-ANSWER MC-2 2)	27	correct

KC (Default)	Opportunity (Default)	Predicted Error Rate (Default)	KC (Unique-step)	Opportunity (Unique-step)	Predicted Error Rate (Unique-step)
SELECT-MC-ANSWER~~APPLY-THEOREM	4~~3	0.553~~0.754	KC1797	2	0.876
SELECT-MC-ANSWER~~APPLY-THEOREM	5~~4	0.451~~0.750	KC1241	3	0.639

7 External Analyses

DataShop supports adding and deleting external analyses via web services or the web application. An external analysis is a file attached to a dataset that describes the result of an analysis on that data. In addition to the file itself, which can be of any type, the analysis also has a required title and optional fields for a description, statistical model used, and relevant KC model.

7.1 Access and ownership

A DataShop user may add an external analysis to a dataset in a project that they have “edit” or “admin” access to. The various access types are described in section 6.1.2.

To view a list of external analyses or to retrieve the content of a single analysis, the user must have “view”, “edit”, or “admin” access.

An external analysis has an owner, the user who created it. Only the owner of an external analysis or a DataShop administrator can delete the external analysis (or modify it in the web application). To delete an external analysis, the user must be both the owner of that analysis and have “edit” access.

7.2 Add External Analysis

[https://pslcdatashop.web.cmu.edu/services/datasets/\[id\]/analyses/add](https://pslcdatashop.web.cmu.edu/services/datasets/[id]/analyses/add)

Add and describe a new external analysis. An ID is assigned by DataShop after a successful add, and is returned in the response XML. This ID can be used in other operations that reference it. This service expects text content only; binary files can be added through the web application. For the purposes of displaying in the web application, DataShop will generate a file name for the text content added through this service.

7.2.1 Request POST Parameters

The body of the request

Required. The analysis text to be stored. Since the entire body of the request will be treated as the analysis, any parameters must be passed as part of the request URL.

title

Required. Name for the analysis. Must be no more than 255 characters.

description

Description of the analysis. Must be no more than 500 characters.

kc_model

Valid ID of a KC model for this dataset. KC model IDs can be obtained from a verbose request for dataset metadata.

statistical_model

Statistical model used in the generation of this analysis. Must be no more than 100 characters.

7.2.2 Example request:

PUT https://pslcdatashop.web.cmu.edu/services/datasets/123/analyses/add?title=Bayesian%20Knowledge%20Tracing%20model%20with%20cross-validation&kc_model=7

7.2.3 Example request using sample DataShop web services client:

```
C:\DS_webservices_java1.5>java -jar dist/datashop-webservices.jar  
"https://pslcdatashop.web.cmu.edu/services/datasets/76/analyses/add?title=Demo%20Title" file  
name_of_file
```

7.2.4 Example response on success:

```
<?xml version="1.0" encoding="UTF-8"?>  
<pslc_datashop_message  
  result_code="0"  
  result_message="Success."  
  analysis_id="3" />
```

7.2.5 Example response on error:

```
<?xml version="1.0" encoding="UTF-8"?>  
<pslc_datashop_message  
  result_code="-8"  
  result_message="Error. Required field(s) missing: title." />
```

7.3 Get External Analyses Metadata

[https://pslcdatashop.web.cmu.edu/services/datasets/\[id\]/analyses](https://pslcdatashop.web.cmu.edu/services/datasets/[id]/analyses)

Get a listing of external analyses with metadata for the specified dataset.

7.3.1 Request Parameters

None.

7.3.2 Example request to get all external analyses:

GET <https://pslcdatashop.web.cmu.edu/services/datasets/388/analyses>

7.3.3 Example response on success:

```
<?xml version="1.0" encoding="UTF-8"?>
<pslc_datashop_message result_code="0" result_message="Success.">
  <analysis id="1">
    <title>Rasch (1pl IRT) model with cross-validation</title>
    <description></description>
    <kc_model_name>Area</kc_model_name>
    <statistical_model>Rasch</statistical_model>
    <file_name>ds76__Rasch.txt</file_name>
    <owner>user@ANDREW.CMU.EDU</owner>
    <added>2012-06-06</added>
  </analysis>
</pslc_datashop_message>
```

7.3.4 Example response on error:

```
<?xml version="1.0" encoding="UTF-8"?>
<pslc_datashop_message
  result_code="-1"
  result_message="Error. Dataset ID 45 is not valid." />
```

7.4 Get External Analysis

[https://pslcdatashop.web.cmu.edu/services/datasets/\[id\]/analyses/\[id\]](https://pslcdatashop.web.cmu.edu/services/datasets/[id]/analyses/[id])

Get an external analysis, identified by its ID. Returns the contents of the file.

7.4.1 Parameters

None.

7.4.2 Example request to get an external analysis:

GET <https://pslcdatashop.web.cmu.edu/services/datasets/42/analyses/3>

7.4.3 Example response :

```
DATASETID: 76
DATE: 2012-06-04 10:42:46
KC MODEL: ANY
```

```

MODEL:      Rasch
MODEL SETUP: lme4, all defaults
COMPUTED BY: R

Data points:      5388
loglikelihood:    -2750.70090
AIC:              5507.40180
BIC:              5527.17759
RMSE:             0.39183
A':              0.77800
10-FOLD CROSS-VALIDATED RMSE USER-STRATIFIED:      0.41557
10-FOLD CROSS-VALIDATED RMSE STEP-STRATIFIED:      0.43114
10-FOLD CROSS-VALIDATED RMSE UN-STRATIFIED:        0.40701
10-FOLD CROSS-VALIDATED A' USER-STRATIFIED:        0.68308
10-FOLD CROSS-VALIDATED A' STEP-STRATIFIED:        0.58122
10-FOLD CROSS-VALIDATED A' UN-STRATIFIED:          0.72249
...

```

7.4.4 Example response on error:

```

<?xml version="1.0" encoding="UTF-8"?>
<pslc_datashop_message
  result_code="-9"
  result_message="Error. External analysis 3 is not valid for dataset 42." />

```

7.5 Delete External Analysis

[https://pslcdatashop.web.cmu.edu/services/datasets/\[id\]/analyses/\[id\]/delete](https://pslcdatashop.web.cmu.edu/services/datasets/[id]/analyses/[id]/delete)

Remove an external analysis. You must have “edit” or “admin” access to the dataset and be the owner of the external analysis.

No request parameters. No request text content; URL serves as the space for the parameters.

7.5.1 Example request to delete an external analysis:

DELETE **<https://pslcdatashop.web.cmu.edu/services/datasets/42/analyses/3/delete>**

7.5.2 Example response on success:

```

<?xml version="1.0" encoding="UTF-8"?>
<pslc_datashop_message
  result_code="0"
  result_message="Success. External analysis 3 successfully deleted." />

```

7.5.3 Example response on error:

```

<?xml version="1.0" encoding="UTF-8"?>
<pslc_datashop_message
  result_code="-12"
  result_message="Error. Insufficient privileges to delete external analysis 3. You
    are not the owner." />

```


8 Custom Fields

DataShop supports adding and modifying custom fields at the [transaction level](#). This section describes custom fields and how you can use them, and includes the API for creating, retrieving, and modifying custom fields through web services.

A custom field is a new column you define for annotating transaction data. Although the feature is new to web services, some datasets in DataShop already have custom fields. This is because some tutors have been instrumented to record custom fields while logging. At logging time, custom fields can be associated with a context, tool, or tutor message.

Some examples include a field that captures the time of each tutor response to the millisecond; a field noting the agent that took the action in a multi-agent system; and a field recording a categorization of the problem the student is working on.

8.1 Access and ownership

A custom field has an owner, the user who created it. Users who have *edit* or *admin* permission for a project can create custom fields for a dataset in it. Only the owner, project administrator, or a DataShop administrator can delete, modify the meta data, or set the values for the custom field. Only DataShop administrators can delete custom fields that were logged with the data.

8.2 Data Types

A custom field can be of one of the following data types:

Data type	Notes on value
number	Must be an integer or float. An integer is a number without a decimal place, while a float is a floating point number, which means it has a decimal place.
string	Must be no more than 65,000 characters.
date	Must be formatted as yyyy-MM-dd HH:mm:ss

8.3 Add Custom Field

[https://pslcdatashop.web.cmu.edu/services/datasets/\[id\]/customfields/add](https://pslcdatashop.web.cmu.edu/services/datasets/[id]/customfields/add)

Add and describe a new custom field. An ID is assigned by DataShop after a successful add, and is returned in the response XML. This ID can be used in other operations that reference it.

8.3.1 Request Parameters

None.

8.3.2 Request Post Parameters

All fields except description are required.

- **name:** descriptive name for the new custom field. Must be unique across all custom fields for the dataset. Must be no more than 255 characters.
- **description:** description for the new custom field. Must be no more than 500 characters.
- **level:** the level of aggregation that the custom field describes. The only accepted value for the current version of this web service is **transaction**. Future versions may support other levels such as **step** or **student**. Cannot be modified later.

8.3.3 Example request using sample DataShop web services client:

C:\DS_webservices_java1.5>java -jar dist/datashop-webservices.jar

"https://pslcdatashop.web.cmu.edu/services/datasets/123/customfields/add" file *name_of_file*

8.3.4 Example request:

POST https://pslcdatashop.web.cmu.edu/services/datasets/123/customfields/add

```
<?xml version="1.0" encoding="UTF-8"?>
<pslc_datashop_message>
  <custom_field>
    <name>pBoredom</name>
    <description>Probability the student is bored at this transaction,
    calculated using Ryan Baker's boredom detector.</description>
    <level>transaction</level>
  </custom_field>
</pslc_datashop_message>
```

8.3.5 Example response on success:

```
<?xml version="1.0" encoding="UTF-8"?>
<pslc_datashop_message
  result_code="0"
  result_message="Success." custom_field_id="211" />
```

8.3.6 Example response on error:

```
<?xml version="1.0" encoding="UTF-8"?>
<pslc_datashop_message
  result_code="-8"
  result_message="Error. Required field(s) missing: Name." />
```

See Appendix A for a full list of possible errors.

8.4 Get Custom Field Metadata

[https://pslcdatashop.web.cmu.edu/services/datasets/\[id\]/customfields/\[?id\]](https://pslcdatashop.web.cmu.edu/services/datasets/[id]/customfields/[?id])

Get a listing of custom fields with metadata for the specified dataset, or about just a single custom field.

8.4.1 Request Parameters

mine

true or false. Default is **false**. If true, only retrieve metadata for custom fields you own.

8.4.2 Example request to get all custom fields:

GET **<https://pslcdatashop.web.cmu.edu/services/datasets/388/customfields>**

8.4.3 Example request to get a single custom field:

GET **<https://pslcdatashop.web.cmu.edu/services/datasets/388/customfields/45>**

8.4.4 Example response on success:

```
<?xml version="1.0" encoding="UTF-8"?>
<pslc_datashop_message result_code="0" result_message="Success.">
  <custom_field id="45">
    <name>pBoredom</name>
    <description>Probability the student is bored at this transaction,
    calculated using Ryan Baker's boredom detector.</description>
    <type>number</type>
    <level>transaction</level>
    <owner>username</owner>
    <added>2013-04-22</added>
  </custom_field>
</pslc_datashop_message>
```

8.4.5 Example response on error:

```
<?xml version="1.0" encoding="UTF-8"?>
<pslc_datashop_message
  result_code="-11"
  result_message="Error. Custom field 45 not found." />
```

8.5 Set Custom Field

[https://pslcdatashop.web.cmu.edu/services/datasets/\[id\]/customfields/\[id\]/set](https://pslcdatashop.web.cmu.edu/services/datasets/[id]/customfields/[id]/set)

Annotate transaction data with custom fields. The custom field must first be created using the Add Custom Field method. Supports annotation for a single custom field across transactions, so to annotate a transaction with multiple custom fields, use multiple requests. This can only be done by the custom-field owner, project administrator, or a DataShop administrator on user-created custom fields.

8.5.1 Request Parameters

None.

8.5.2 Example request (tab-delimited):

Transaction Id	value
69902b300b768a522d19da399622a2ff	0.323
53dc58f4cdad5cee279ed6b13d664d69	HINT_MSG
afc5e2f613482506c0dcf2d38627e4f3	2014-11-08 18:56:55

8.5.3 Example request using sample DataShop web services client:

C:\DS_webservices_java1.5>java -jar dist/datashop-webservices.jar

"https://pslcdatashop.web.cmu.edu/services/datasets/123/customfields/211/set" file *name_of_file*

8.5.4 Example response on success:

```
<?xml version="1.0" encoding="UTF-8"?>
<pslc_datashop_message
  result_code="0"
  result_message="Success. Annotated 2 transactions." />
```

8.5.5 Example response on error:

```
<?xml version="1.0" encoding="UTF-8"?>
<pslc_datashop_message
  result_code="-10"
  result_message="Error. Invalid data." />
```

8.6 Delete Custom Field

[https://pslcdatashop.web.cmu.edu/services/datasets/\[id\]/customfields/\[id\]/delete](https://pslcdatashop.web.cmu.edu/services/datasets/[id]/customfields/[id]/delete)

Remove a custom field entirely, the metadata and all the values associated with transactions. This can only be done by the custom-field owner, project administrator (user-created custom fields), or a DataShop administrator.

No request parameters. No request text content; URL serves as the space for the parameters.

8.6.1 Example response on success:

```
<?xml version="1.0" encoding="UTF-8"?>
<pslc_datashop_message
  result_code="0"
  result_message="Success. Custom field successfully removed from 1,231 transactions." />
```

8.6.2 Example response on error:

```
<?xml version="1.0" encoding="UTF-8"?>
<pslc_datashop_message
  result_code="-11"
  result_message="Error. Custom field 2 not found." />
```


9 Learning Curve

DataShop supports learning curve categorization of Knowledge Components (KCs). This section describes the categorizations and how they can be retrieved through web services.

In the DataShop web application learning curves for a dataset and skill model are placed in one of four categories: Too Little Data, Low and Flat, No Learning and Still High. Learning curves that do not fall into any of the above "bad" or "at risk" categories are labeled "Good" as they appear to indicate substantial student learning.

The algorithm for categorizing KCs first discards points in each curve based on the *student threshold*. If a point has fewer than that number of students, it is ignored. Within the points of curve remaining:

- If the number of points is below the *opportunity threshold*, then that curve has **too little data**.
- If all points of the curve are beneath the *low error threshold*, then the curve is **low and flat**.
- If the slope of the predicted learning curve (as determined by the AFM algorithm) is below the *AFM slope threshold*, then the curve shows **no learning**.
- If the last point of the curve is above the *high error threshold*, then the curve is **still high**.

As with the web application, the user must specify the dataset in generating the web services learning curve classification report. The user can optionally specify a skill model; the default behavior will include all skill models for the dataset in the report. The above thresholds have default values or the user may specify them and override the default value. The output is in plain text, tab-delimited format. In addition to the learning curve category, the report also includes the KC intercept, KC slope, unique step count and step instance count.

9.1 Access

Any user who has view right to a dataset can use this web service.

9.2 Get Learning Curve

`https://pslcdatashop.web.cmu.edu/services/datasets/[id]/learningcurves/classify[?kc_model=modelName]`

Get a listing of learning curve categories for all skills for the specified dataset and skill model.

9.2.1 Request Parameters

kc_model

Not required. Default all KCs for the dataset. Case sensitive.

opportunity_threshold

Not required. Default 3. Any graph point that has lower than this opportunity count is eliminated.

student_threshold

Not required. Default 10. Any graph point that has lower than this student count is eliminated.

low_error_threshold

Not required. Default 20. Any graph point that has lower than this error rate is eliminated.

high_error_threshold

Not required. Default 40. Any graph point that has higher than this error rate is eliminated.

AFM_slope_threshold

Not required. Default 0.001. Any graph point that has lower than this slope is eliminated.

9.2.2 Example request with default:

GET

[https://pslcdatashop.web.cmu.edu/services/datasets/\[id\]/learningcurve/classify\[?kc_model=Area\]](https://pslcdatashop.web.cmu.edu/services/datasets/[id]/learningcurve/classify[?kc_model=Area])

9.2.3 Example request with specified parameter:

GET

[https://pslcdatashop.web.cmu.edu/services/datasets/\[id\]/learningcurve/classify\[?kc_model=Area&opportunity_threshold=1&student_threshold=2\]](https://pslcdatashop.web.cmu.edu/services/datasets/[id]/learningcurve/classify[?kc_model=Area&opportunity_threshold=1&student_threshold=2])

9.2.4 Example response on success:

Dataset: Geometry Area (1996-97)

AFM slope threshold: 0.001

Student threshold: 10

Opportunity threshold: 3

Low error threshold: 20.0

High error threshold: 40.0

KC Model	KC Name	Category	KC Intercept	KC Slope	# unique steps
	#opportunity	1 step instances	# step instances		
Area	Area formula	No learning	0.9598860130611057	0.0	11 50 1784
Area	Non-area formula	Good	0.9359422484296835	0.0018665230448641058	22 59 3083
Geometry	Geometry	No learning	0.9690713757580114	2.226319206785172E-4	22 59 4843

Summary

KC Model	%Good	%No learning	%Low and flat	%Still high	%Too little data	#
observation	AIC	Item CV				
Area	50	0	0	0	5104 5642,15	
Geometry	0	100	0	0	0 5104 0.408165	

9.2.5 Example response on error:

```
<?xml version="1.0" encoding="UTF-8"?>
<pslc_datashop_message result_code="-1" result_message="Error. Dataset 9 is not valid."/>

<?xml version="1.0" encoding="UTF-8"?>
<pslc_datashop_message result_code="-18" result_message="Error. Skill model area is not valid."/>
```

10 KCM Import

DataShop supports adding KC models to a dataset via web service. This process is called KCM import. This section describes the required parameters and data for KCM import, the acceptable format for import data and the expected results after KCM import.

Datashop web interface allows users to import a KCM to a dataset. KCM import makes the same functionality available as a web service. Similar to the web application, KCM import requires a dataset ID and a data file that contains the mappings of steps to skills. Steps are represented as step ID and they should be associated with the specified dataset. After the new KCM and the new skills and mappings of the skills to steps are saved to the database, aggregation for this KCM and statistical parameters (LL, AIC, BIC, CV, etc.) will be computed and saved.

10.1 Access

User who has editing right to a dataset can use this web service to import a KCM to the dataset.

10.2 Import KCM

[https://pslcdatashop.web.cmu.edu/services/datasets/\[id\]/importkcm/](https://pslcdatashop.web.cmu.edu/services/datasets/[id]/importkcm/)

Import a KCM to the dataset specified

10.2.1 Request Parameters

file: required, the file that stores mappings of step IDs and skills

10.2.2 Example request using DataShop web services client:

C:\DS_webservices_java1.5>java -jar dist/datashop-webservices.jar

"https://pslcdatashop.web.cmu.edu/services/datasets/123/importkcm/" file *name_of_file*

10.2.3 Example for tab-delimited data file:

POST <https://pslcdatashop.web.cmu.edu/services/datasets/123/importkcm/>

Step ID	KC (New_KCM)	KC (Another New_KCM)
617b8c4a416f7eb515d56bf7fa7eab80	skill1	skill4
0b062c22b215ab6e6d80e09ea09880b5	skill2	skill4
ebdf2271d8f64bebb4c9f0cda65ccab0	skill1	skill5
3a9a903dbb2845762e7735147a08ea05	skill1	skill5
b0b071e0a6dee842078d8ac31babfd71	skill2	skill6
0cdd9ea7c8bbe389199a76fbed089a36	skill3	skill7

10.2.4 Example response on success:

```
<?xml version="1.0" encoding="UTF-8"?>
<pslc_datashop_message
  result_code="0"
  result_message="Success. KCM(s): New KCM, Another New KCM saved successfully. Model
values are now being computed for the new KCM(s)." />
```

Note: After the new KCM(s) is(are) saved successfully, Datashop will start a process to reaggregate the dataset and compute AFM and Cross Validation for the new KCM(s). The amount of time needed for this process varies depending on the size of the dataset. The user can use the Datashop web application or Datashop web service (Dataset Metadata service) to check the final results at a later time.

10.2.5 Example response on error:

When dataset id is not found:

```
<?xml version="1.0" encoding="UTF-8"?>
<pslc_datashop_message
  result_code="-1"
  result_message="Error. Dataset [id] is not valid." />
```

When user doesn't have editing right to the dataset:

```
<?xml version="1.0" encoding="UTF-8"?>
<pslc_datashop_message
  result_code="-2"
  result_message="Error. Dataset [id] is not accessible." />
```

When input file format is not valid, such as file header doesn't have column "Step ID":

```
<?xml version="1.0" encoding="UTF-8"?>
<pslc_datashop_message
  result_code="-19"
  result_message="Error. Bad file format. The step identifier column "Step ID" was not
found." />
```

When input file format is not valid, such as no KCM found in headers:

```
<?xml version="1.0" encoding="UTF-8"?>
<pslc_datashop_message
  result_code="-19"
  result_message="Error. Bad file format. No KCM found to import. " />
```

When input file format is not valid, such as KCM name has invalid characters:

```
<?xml version="1.0" encoding="UTF-8"?>
<pslc_datashop_message
  result_code="-19"
  result_message="Error. Bad file format. Invalid character(s) found in model name. " />
```

Note: Valid characters for KC model names include space, dash, underscore, letters and numbers.

When input file format is not valid, such as KCM name is too long (max is 50 characters)

```
<?xml version="1.0" encoding="UTF-8"?>
<pslc_datashop_message
  result_code="-19"
  result_message="Error. Bad file format. Model name is ..., and it's too long: 55. " />
```

When input file format is not valid, such as KCM already exists in Datashop

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<pslc_datashop_message  
  result_code="-19"  
  result_message="Error. Bad file format. Model name is ..., and it exists already." />
```

When input file format is not valid, such as duplicate KCM names found in import file

```
<?xml version="1.0" encoding="UTF-8"?>  
<pslc_datashop_message  
  result_code="-19"  
  result_message="Error. Bad file format. Duplicate KCM names found in import file. " />
```

When input file contains invalid step ID, i.e. step ID doesn't exist for the specified dataset ID:

```
<?xml version="1.0" encoding="UTF-8"?>  
<pslc_datashop_message  
  result_code="-20"  
  result_message="Error. Invalid step ID found: ....." />
```

When a KCM import already running for this dataset

```
<?xml version="1.0" encoding="UTF-8"?>  
<pslc_datashop_message  
  result_code="-22"  
  result_message="KCM import already running (since 2015-04-06 8:31:43). Please try  
importing your model again after it has completed." />
```

Appendix A

All Possible Result Codes

Result Code	HTTP Status Code	Description	Services that could return this result code
0	200 OK	Success. [Optional message]	All
-1	404 Not Found	Error. Dataset [id] is not valid.	All
-2	401 Unauthorized	Error. Dataset [id] is not accessible.	All
-3	404 Not Found	Error. Sample [id] is not valid for dataset [id].	Get Samples, Get Transactions, Get Student-Steps
-4	401 Unauthorized	Error. Sample [id] is not accessible for dataset [id].	Get Samples, Get Transactions, Get Student-Steps
-5	400 Bad Request	Error. Invalid request parameter: [parameter].	All
-6	400 Bad Request	Error. Invalid value for parameter [param]: [value].	All
-7	401 Unauthorized	Error. Dataset [id] is not released.	Get Transactions, Get Student-Steps
-8	400 Bad Request	Error. Required field(s) missing: [fields].	Add External Analysis, Add Custom Field, Get Learning Curve
-9	404 Not Found	Error. [External analysis / Custom field] [id] is not valid for dataset [id].	Get/delete External Analysis, All Custom Field services
-10	400 Bad Request	Error. Invalid data.	Add External Analysis, Set Custom Field
-11	404 Not Found	Error. Custom field [id] not found.	Delete Custom Field
-12	401 Unauthorized	Error. Insufficient privileges to [add/delete/set] [an/a] [external analysis / custom field] [id]. [You are not the owner.]	Delete External Analysis, Delete Custom Field, Set Custom Field, Add External Analysis, Add Custom Field
-13	409 Conflict	Error. Custom field with name [name] already exists for this dataset.	Add Custom Field
-14	404 Not Found	Error. [Transactions/Student-steps] cache file for sample [N] [does not exist / is out-of-date]. Caching process is starting. Try again later.	Get Transactions, Get Student-Steps
-15	400 Bad Request	Error. Parameter [parameter] must be no more than [N] characters.	Add external analysis Generate Dataset Id
-16	400 Bad Request	Error. Invalid value for element	Add Custom Field

		'[element]': [value]	
-17	400 Bad Request	Invalid XML format.	Add Custom Field
-18	404 Not Found	Error. Skill model name [name] is not valid.	Get Learning Curve
-19	400 Bad Request	Error. Bad file format. The step identifier column "Step ID" was not found Or: No KCM found to import. Or: Invalid character(s) found in model name. Or: Model name is ..., and it's too long:... Or: Model name is ..., and it exists already. Or: Duplicate KCM names are found in import file.	KCM import
-20	400 Bad Request	Error. Invalid step ID	KCM import
-22	503 Service unavailable	Error: KC Model Import already running (since 2015-04-06 8:31:43). Please try importing your model again after it has completed.	KCM import
-99	404 Not Found	Error. No web service found matching the URL. For a list of valid URLs, see http://pslcdatashop.org/api/	All
-100	500 Internal Server Error	Unknown error.	All
-101	401 Unauthorized	Authorization failed. Check your credentials.	All
-102	406 Not Acceptable	This content is available only as [content type].	All
-103	405 Method Not Allowed	Operation not supported. (Note to reader: this is for unsupported PUT/POST/DELETE methods.)	All
-104	405 Method Not Allowed	[Method] requests not supported. (Note to reader: this is for methods head, options, etc.)	All

Appendix B

Revision History

Revision	Date Released
Revision 0.3	October 24, 2013
Added section “Custom Fields” and updated the result codes table.	
Revision 0.23	October 11, 2012
Added section “External Analyses” and updated the result codes table.	
Revision 0.22	December 17, 2010
Added new cross-validation elements and number of parameters to example XML for Get dataset metadata (verbose). Also changed element “lfa_status” to “logistic_regression_model_status” in the same section.	
Revision 0.21	June 29, 2010
Added “domain” element to <i>Get Dataset Metadata</i> examples as web services now include a “domain” element (if it’s set) in addition to “learnlab”.	