
DataShop Public API

A way to
programmatically get
and set data in
DataShop

The DataShop Team + others
1/15/2010

Contents

1	API	3
2	Note on format	3
	Method Title	3
3	What is an ID?	3
4	Representational state transfer (REST).....	3
4.1	HTTP Response Status Codes	3
5	Authentication	4
5.1	Access Credentials.....	4
5.2	Using your Access Credentials to Sign a Request	4
6	Getting Data.....	9
6.1	Get Dataset Metadata.....	9
6.2	Get Sample Metadata	12
6.3	Get Transactions	14
6.4	Get Student-Step Records.....	19
	Appendix A.....	24
	All Possible Result Codes.....	24

1 API

This API is a list of methods for programmatically getting and (in the future) modifying DataShop data. The initial target is to provide enough functionality for getting transaction data with various parameters, and adding, modifying, and deleting custom-field data.

As of January 2010, we've implemented:

- Authentication
- Get Dataset Metadata
- Get Sample Metadata
- Get Transactions
- Get Student-Step Records

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, 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.

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.

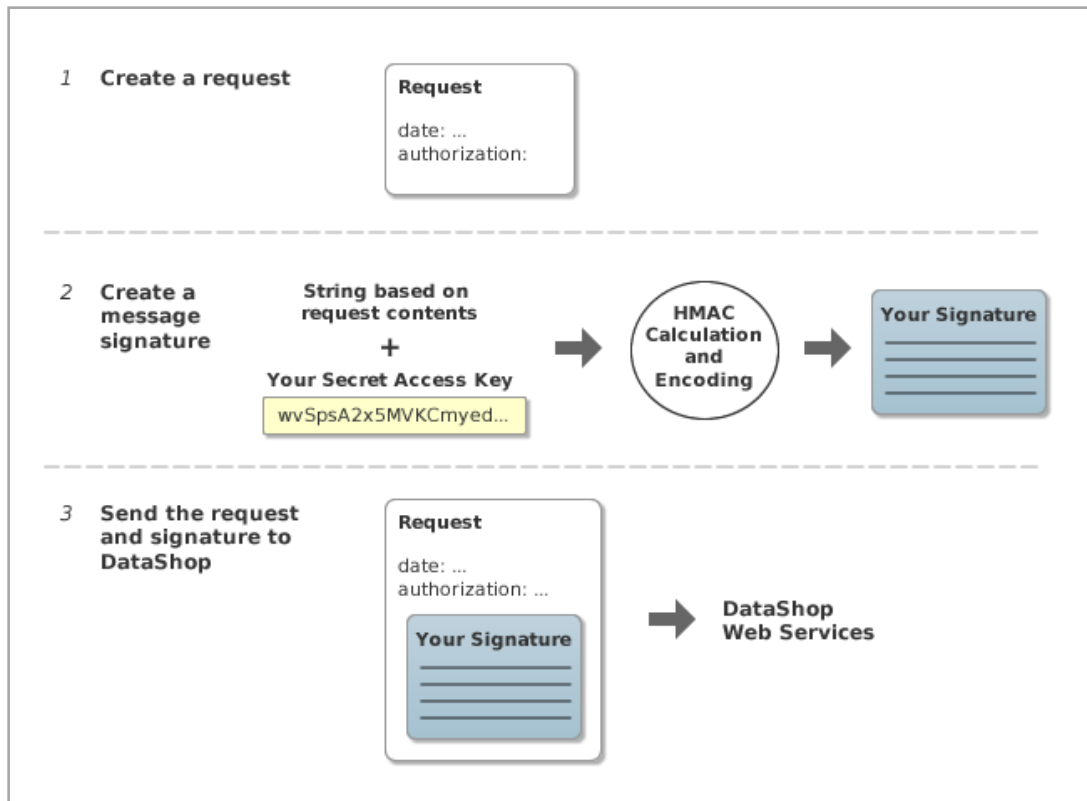


Figure 1 – Authentication process

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 OKLFETPSJZJJFSGR6D8E: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 or private. Viewable means return datasets for projects on which you have “view” or “edit” privileges, in addition to public datasets. Editable means return only datasets for projects on which you have the “edit” 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", or "private".

- **public:** you and any other registered DataShop user can view the data
- **view:** you can view the data, create samples, add kc models, and add kc sets.
- **edit:** you can both view the data and modify metadata, add or modify papers and files, and add or modify custom fields.
- **private:** you cannot view the data nor edit anything about the dataset

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://pslcdatashop.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>
    <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>

    <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://pslcdatashop.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>
    <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>

    <curriculum>geometry 2005</curriculum>
    <tutor></tutor>
```

```

</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>
  <lfa_status>unable to run</lfa_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>
  <lfa_status>complete</lfa_status>
  <aic>142117.54</aic>
  <bic>142468.24</bic>
</kc_model>
<kc_model id="14">
  <name>Unique-step</name>
  <number_of_kcs>9157</number_of_kcs>
  <observations_with_kcs>0</observations_with_kcs>
  <lfa_status>not scheduled to run</lfa_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.

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 or none. Default is **none**. Optionally specify whether to include all custom fields or none. (A future release of web services will support requesting a subset of custom fields by specifying a comma-delimited list of custom-field IDs).

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	
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	A variable number of “Level(<i>Level Type</i>)” columns, each describing a curriculum hierarchy level.	string	
problem_name	Problem Name	string	
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>) 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’

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 15:12:1 8.0		ATTEMPT	RESULT	CHARGE1A	(CHOOSE- ANSWER MC-1 1)
2	D8DD5	D8DD5- Jan07-15- 12-09	2008- 01-07 15:12:4 5.0	27	ATTEMPT	RESULT	CHARGE1A	(CHOOSE- ANSWER MC-2 2)

Attempt At Step	Outcome	Selection	Input	Feedback Text	Feedback Classification	KC(Default)	KC(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.

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://pslcdatashop.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 web services Get Student-Step Records is different from the format of these columns in the web application and from the Get Transactions web service. In the web application’s current step format, multiple KCs associated with a step results in **multiple rows**. In the web services version, multiple KCs are contained in a **single value** and delimited with two tildes (“~~”), resulting in a single row for the student-step. 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

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	403 Forbidden	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, Get Custom Field Data
-4	401 Unauthorized	Error. Sample [id] is not accessible for dataset [id].	Get Samples, Get Transactions, Get Student-Steps, Get Custom Field Data
-5	400 Bad Request	Error. Invalid request parameter: [parameter].	All
-6	400 Bad Request	Error. Invalid value for parameter [param]: [value].	All
-7	400 Bad Request	Error. Invalid column: [column].	Get Transactions, Get Student-Steps
-8	400 Bad Request	Error. Required field(s) missing: [fields].	Add Custom Field, Modify Custom Field Name, Modify Custom Field description, Set Custom Field
-9	404 Not Found	Error. Custom Field [id] is not valid for dataset [id].	All Custom Field services
-10	400 Bad Request	Error. Invalid data.	Set Custom Field
-11	404 Not Found	Error. No custom field data found.	Delete Custom Field
-12	401 Unauthorized	Error. Not owner.	Delete 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
-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