

Session Stream

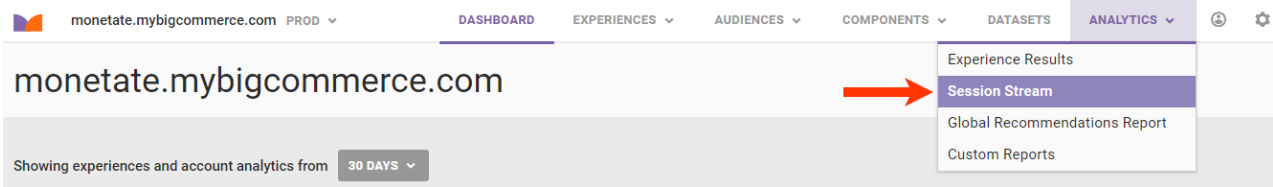
Session Stream provides open access to raw Monetate session data. It allows you to join Monetate data with your own business data for more advanced analysis and to inform future personalization initiatives across all channels. You can use this data in relational databases such as MySQL, data warehouses such as Amazon Redshift, or BI platforms such as Tableau and Domo.

Session Stream provides JSON data files compressed with Gzip in 15 minute increments. Each file contains raw data from every session that Monetate observed on your site during that time. Session Stream is available for all Monetate users.


Accessing Session Stream

Session Stream is provided via Secure File Transfer Protocol (SFTP).

Click **ANALYTICS** in the top navigation bar, and then select **Session Stream**.



On the **Sites** page, click **CREATE SFTP USER** to configure an SFTP username and authentication method. If you've already created an SFTP user, then you can use those credentials to access Session Stream.

 You must have the Administrator role in the Monetate platform to set up SFTP users.

monetate.mybigcommerce.com PROD

DASHBOARD EXPERIENCES AUDIENCES COMPONENTS DATASETS ANALYTICS

Sites Markets Stealth Groups API Keys

monetate.mybigcommerce.com PROD

monetate.mybigcommerce.com PROD

monetate.mybigcommerce.com

Production

Timezone

AMERICA/DOMINICA

SFTP Connection Settings for Accessing the Session Stream

Server URL: sftp://download.monetate.net

[Learn about the data you can get from the new Session Stream](#)

SFTP Connection Settings for Uploads

Server URL: sftp://sftp.monetate.net

Merchandise: /upload

Datasets: /datasets

SFTP Users for Accessing the Session Stream and Uploads

[CREATE SFTP USER](#)

USER NAME	DESCRIPTION	AUTH TYPE	LAST LOGIN
ftp65802828		Password	Not available

Use your SFTP user credentials to access the server URL listed in the SFTP Users for Accessing the Session Stream and Uploads table on the **Sites** tab. All Session Stream files are available in the `account_id/raw_data/session_stream/` directory. Files are organized hierarchically under the `session_stream` directory by four-digit year, two-digit month, then two-digit day. For example, the files from May 1, 2017, are in `session_stream/2017/05/01/`.

A Session Stream file is available for 820 days after it's generated.



Because each file contains raw data from every session that Monetate observed on your site, there can be thousands of files per day totaling several gigabytes depending on your site traffic.

Monetate names Session Stream files with a specific format. The following example filename is for a file from the 15-minute period starting May 1, 2017, at 3:45 AM: `20170501T034500.000Z_PT15M-1477542463784.json.gz`.

The name consists of the following parts:

- `20170501T034500.000Z` – The date and time in ISO-8601 basic format. The date is encoded as `YYYYMMDD` and the time as `hhmmss.sss`. `Z` is the time zone indicator for Zulu time (UTC±00:00). This is always `Z` in Session Stream.
- An underscore character (`_`)
- `PT15M` – The duration of the time interval in ISO-8601 duration format. This is always 15 minutes in Session Stream.
- A dash character (`-`)
- `1477542463784` – A unique string that prevents name collisions between multiple files within the same time interval.

Remember that open sessions can last up to 12 hours, and all file timestamps are in UTC time. If you download the files for the prior day's data at 1 AM UTC, then you may miss some sessions that haven't closed yet. These sessions are available in the following day's files.

Programmatically Downloading Session Stream Files

As an alternative to using an SFTP client, you can use the sample Python script included in this documentation to download your files.



You must have [Paramiko 2.1.1](#) or later to run this example script.

When run more than once for a particular date, this script only downloads files that you haven't already retrieved. You can input as a parameter the date for which you want to fetch data. The Session Stream files are downloaded to the location you specified with the same date-based directory structure that Monetate uses to organize the files. You can schedule this script to run daily or write a similar application that better meets your needs.

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

from argparse import ArgumentParser
from datetime import datetime, timedelta
import paramiko
import os

PATH_FORMAT = "%Y/%m/%d"

def main():
    yesterday = datetime.now() - timedelta(days=1)
    parser = ArgumentParser(description="Downloads monetate session stream via FTP for the specified date")
    parser.add_argument("--host", default="download.monetate.net", help="The host to connect to")
    parser.add_argument("--port", default=22, type=int, help="The port to connect on (default: %(default)s)")
    parser.add_argument("-u", "--username", required=True, help="FTP user to connect as")
    parser.add_argument("-p", "--password", required=True, help="FTP user password")
    parser.add_argument("-d", "--date", default=yesterday.strftime(PATH_FORMAT),
                        help="The date to sync locally, format YYYY/MM/DD (default: %(default)s)")
    parser.add_argument("--root_output_dir", default=os.path.abspath(os.path.dirname(__file__)),
                        help="The root directory where files will be synced locally (default: %(default)s)")
    args = parser.parse_args()

    base_path = "raw_data/session_stream/{0}".format(args.date)

    # create local paths if they don't exist
    try:
        output_path = os.path.join(args.root_output_dir, base_path)
```

```

    os.makedirs(output_path)
except OSError:
    pass

# list the files in the currently directory
local_files = frozenset(os.listdir(output_path))

# connect via ftp and list the files
transport = paramiko.Transport((args.host, args.port))
transport.connect(username=args.username, password=args.password)
sftp = paramiko.SFTPClient.from_transport(transport)

# list the files we need to download
print "Calculating files to download..."
current_files = frozenset(sftp.listdir(base_path))
files_to_download = current_files - local_files

# download all the files needed
print "Fetching {} files...".format(len(files_to_download))
for filename in files_to_download:
    remote_path = "{}/{}".format(base_path, filename)
    local_path = os.path.join(output_path, filename)
    sftp.get(remote_path, local_path)

if __name__ == '__main__':
    main()

```

Session Data Description

Session Stream contains data for any session that has ended, which occurs after 30 minutes of inactivity and persists for a maximum of 12 hours if a visitor is active at least once every 30 minutes. Each line in a file contains one JSON object that represents one session.

Nested Data

Each session is not a flat structure. In addition to basic information (browser, OS, and session time), the fields in the object and array types contain more detailed data and may each have no entries, one entry, or many entries per session. Variable-size fields include the following:

- custom_targets
- offers
- page_event_ids
- purchases
- cart_lines
- view_lines
- purchase_lines

The `purchase_lines` field is doubly nested, and each value has the `items` field, which is variable-size.

Field Size Limits

The object and array fields described are limited to 1,000 items each to protect against tasks performed at a higher rate. A higher rate indicates the likelihood of a bot and nonhuman interaction.

This upper limit is not hard. You may see sessions with a few extra items. Fields with fewer than 1,000 items are always complete, but you should assume fields at or exceeding the size limit to be incomplete.

This table contains a more detailed description of the fields. Refer to the spec-compliant JSON schema for the precise format of the structure.

Field	Type	Description
<code>account_id</code>	Number	The unique numeric ID associated with your Monetate account.
<code>browser</code>	String	The browser used within the session.
<code>browser_version</code>	String	The specific version of the browser used in the session.
<code>cart_lines</code>	Array	A list of items carted, including product ID, SKU, and time for each item.
<code>city</code>	String	The visitor's city based on IP address.
<code>country_code</code>	String	ISO 3166-1 two-letter country code.
<code>custom_targets</code>	Object	Custom target IDs and values matched to a visitor in the session based on targets created in Target Builder. If you've configured an ID collector to collect names, e-mail address, and other data, then your custom target values may contain personal identifiable information (PII). PII can be sensitive or non-sensitive; be mindful if you're loading sensitive information into a data warehouse or BI platform.
<code>customer_id</code>	String	The unique customer ID value collected on the page during the session.
<code>customer_link</code>	String	A customer ID that was previously associated with the user in a prior session. This can differ from the <code>customer_id</code> value if the <code>customer_id</code> collected in the current session differed from the <code>customer_id</code> collected in a previous session.
<code>device_type</code>	String	The type of device used during the session deciphered through user agent string.
<code>end_time</code>	Number	Session end time in Unix milliseconds.

Field	Type	Description
<code>guid</code>	String	String that uniquely identifies the Monetate session: <code>account id: Monetate ID</code> .
<code>has_cart</code>	String	String (<code>t</code> or <code>f</code>) to indicate whether the session viewed the cart page with an item in the cart.
<code>has_new_customer</code>	String	String (<code>t</code> or <code>f</code>) to indicate whether the visitor for this session represents a new customer. <code>f</code> indicates a returning customer.
<code>has_product_view</code>	String	String (<code>t</code> or <code>f</code>) to indicate whether a product page was viewed during the session.
<code>has_purchase</code>	String	String (<code>t</code> or <code>f</code>) to indicate whether a product was purchased during this session.
<code>has_recommendation_cart</code>	String	String (<code>t</code> or <code>f</code>) to indicate whether a recommended product was added to cart.
<code>has_recommendation_click</code>	String	String (<code>t</code> or <code>f</code>) to indicate whether a recommendation was clicked.
<code>has_recommendation_impression</code>	String	String (<code>t</code> or <code>f</code>) to indicate whether a recommendation was seen.
<code>has_recommendation_purchase</code>	String	String (<code>t</code> or <code>f</code>) to indicate whether a recommended product was purchased.
<code>has_stealth</code>	String	String (<code>t</code> or <code>f</code>) to indicate whether visitor is in Stealth Mode. This is configured within your account settings.
<code>is_bounce</code>	String	String (<code>t</code> or <code>f</code>) to indicate whether a visitor bounced during this session.
<code>is_closed</code>	String	String (<code>t</code> or <code>f</code>) to indicate whether the session is closed.
<code>offers</code>	Object	Experience split IDs and time when the split was first seen in Unix milliseconds.
<code>os</code>	String	Operating system.
<code>os_version</code>	String	Operating system version.
<code>page_event_ids</code>	Object	Page event IDs and time when that event was seen for the first time in Unix milliseconds.
<code>page_views</code>	Number	Number of page views that occurred within the session.
<code>page_views_ss</code>	Number	Number of page views squared for statistical analysis.
<code>product_view_count</code>	Number	Number of product views.
<code>purchase_count</code>	Number	Number of purchases within the session.

Field	Type	Description
<code>purchase_lines</code>	Object	A map of purchase IDs to maps containing detailed information about each purchase, including total value and time for each purchase, and product ID, SKU, unit price, and quantity for each product in a purchase.
<code>purchase_value_ss</code>	Number	Purchase values squared and summed for statistical analysis.
<code>purchases</code>	Object	Purchase IDs and their monetary values with each key-value pair representing one checkout. The purchase ID is provided by your integration and is not defined by Monetate.
<code>region</code>	String	Region code.
<code>screen_height</code>	Number	Visitor screen height.
<code>screen_width</code>	Number	Visitor screen width.
<code>session_count</code>	Number	Session count (always 1).
<code>session_id</code>	String	The ID that uniquely identifies this session and composed of the GUID, purchase count, start time, and end time.
<code>session_value</code>	Number	Total value of purchases within the session.
<code>session_value_ss</code>	Number	Total value of session purchases squared for statistical analysis.
<code>start_time</code>	Number	Session start time in Unix milliseconds.
<code>time_on_site</code>	Number	Time on site in seconds during the session.
<code>time_on_site_ss</code>	Number	Time on site in seconds squared for statistical analysis.
<code>view_lines</code>	Array	A list of products viewed, including product ID and time for each item.

Retrieving Interpretable Values

Session Stream uses unique IDs rather than interpretable names for values such as offer and page event IDs. Monetate provides unique IDs because these IDs never change, whereas you can change the display name of variants and page events within the platform. This means that data received through Session Stream always refers to the same data objects within Monetate and results in zero duplicate data entries on your end.

Titles can be valuable, so Monetate provides you with access to the Monetate Metadata API. This API allows you to request interpretable values for offers, page events, and target IDs for display in your data warehouse or BI tool.

For more information about the Monetate Metadata API, see [Get Interpretable Values with the Metadata API](#).