

# Update Customer Data via the Data API



This content is only for clients with **Product Data** and **Customer Data** options in the **DATASETS** menu in the top navigation bar. If no menu options appear when you click **DATASETS**, then this content doesn't apply to your account. Instead, see the [alternate version of this documentation](#).

Monetate's APIs empower you to send it data in real time. A RESTful API programmatically transfers information over the Web through a predefined schema. There are many different endpoints with specific requirements that perform various actions or return various data.

The Data API allows you to create and send data to the platform that it can leverage to deliver individual decisions and recommendations across channels. As APIs become the driving force behind continuous and automated data exchange, you can securely connect your applications to the platform.

Monetate currently accepts data that contains a timestamp for the purposes of understanding when specific events or interactions occurred in a time series and non-time series data. These are commonly referred to as entities or attributes. Before sending data to the platform, you must fulfill a few prerequisites.

## Creating API Users and Public Keys

API keys allow you to generate authentication credentials that are separate from your username and password for Monetate. To successfully set up an integration between your Monetate account and your applications, you must generate unique API users and create public keys for those users that you can revoke at any time.

To ensure secure communication, the platform uses public-private key pairs. Public keys are made available to everyone, including Monetate, but private keys remain confidential to their respective owners (just you). Whatever is encrypted with a public key may only be decrypted via the corresponding private key and vice versa.

For more information about configuring API users and public keys, refer to [Manage API Keys](#).

## Establishing Authentication

Logging in with a username and password is one example of authentication. When you authenticate with a server, you prove your identity to the server by telling it information that only you know. Once the server knows who you are, it can trust you and disclose the private data in your account.

Monetate's authentication model allows your applications to authenticate directly to it using a JSON Web token (JWT) signed with your private key. The Auth API allows your application to request these access tokens.

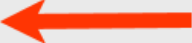
After you create your public-private key pair and submit your public key, use your private key to sign your API request, and then your public key is used to verify it to receive your token.

For help constructing your JWT assertion, refer to the API documentation that is specific to your account. Find it

by first selecting **API Documentation** from the settings cog within the platform, and then clicking the **API Information** option under the Monetate Auth API heading in the left-hand navigation.

## API Documentation

The Monetate Data API allows you to create and send data across any channel in real-time. This data will be leveraged by Monetate's real-time decisioning engine to deliver 1:1 experiences across any channel.

APIs Reference	
Monetate Auth API	
API Information	
Token	>
Monetate Data API	
API Information	
Schema	>
Upload	>
Data	>
Product Catalog	>
Customer Data Privacy	>
Monetate Metadata API	
API Information	
Metadata	>



Multiple access tokens can be live at any given time and the expiration for each token is independent of the expiration of any other access tokens.

## Configuring the Schema, Identifiers, and Collectors

A schema is the organization or structure of the data in a dataset. You must establish the schema for a dataset before sending data to the platform. Follow the steps in [Upload a Customer Dataset](#) to create a schema using the wizard in the platform.



Creating a customer dataset schema via the API is not recommended at this time. You can,

however, add attributes to an existing dataset schema through the Data API.

If you've configured a Customer View in the platform and selected an ID Collector associated with the Customer View when you created the customer dataset, then you can leverage cross-device identification of your site visitors. See [Customer Views](#) for more information.

## Posting Data

Once you create a schema, you can update or create a new data record within that schema with the Data API to immediately react to customer behavior for optimization.

Detailed information about endpoints and request syntax is accessible in the API documentation that is specific to your account. To find it, first select **API Documentation** from the settings cog within the platform, and then click **API Information** under the Monetate Data API heading in the left-hand navigation.

## API Documentation

The Monetate Data API allows you to create and send data across any channel in real-time. This data will be leveraged by Monetate's real-time decisioning engine to deliver 1:1 experiences across any channel.

### APIs Reference

Monetate Auth API

API Information

Token



Monetate Data API

API Information



Schema



Upload



Data



Product Catalog



Customer Data Privacy



Monetate Metadata API

API Information

Metadata





The Data API section is dynamically generated based on your schema. Therefore, you must first create a schema for the Monetate Data API section to appear in the API documentation accessible via the settings cog.

## Validating Data API Records

To verify that dataset records have been updated successfully, you can use a Data API endpoint that supports requests for single records. You cannot use this endpoint to download bulk records or for automated testing, but you can use it to manually spot-check individual records within a dataset.

The endpoint format is as follows:

```
GET https://api.monetate.net/api/data/v1/{schema-name}/production/data/?id={record_id}
```



This GET endpoint is intended for you to manually validate individual records and is strictly rate-limited. A maximum of one request per second is supported.

Ensure you replace `{schema-name}` and `{record_id}` with the specific schema name and record ID that you want to check.