

Monetization API

Current Version: 0.2.2

Used by: [Authenticated Monetization API](#)

Summary

An API format intended to be "sub-classed" by other APIs.

- [Blue items](#) are standard parameters,
- [brown items](#) are constants, and
- [bright-red items](#) are items that are defined by an implementing API.

Any field not marked (Optional) is required.

URL Convention

[<protocol>](#)://[<host>](#)/[<optional-path>](#)/[<API-Name>](#)/[<version>](#)/[<optional-parameters>](#)

- [<optional-path>](#) - any path off the host can be used as long as the path ends with the correct conventions described above
- [<version>](#) - Value that only contains numbers and periods, corresponding to the version of the implementing spec it is built to (**not** the version of this API Interface).
- [<optional-parameters>](#) - API-specific parameters (can be multiple [segments](#)).

Examples:

- <http://mog-fb-qa.playdom.com/itemTransaction/1.0>
- <http://snf-qa01-external.vm.dfw.playdom.com/snc/public/labsgamethree/playdom/itemTransaction/2.0>
- <http://payments.playdom.com/api/imaginaryApi/1.0.23/get/data/foo>

Request

The HTTP method is POST and the request is UTF-8 encoded. All HTTP headers set by the request should be ignored.

Request Body

The request body is a stringified JSON object containing the items to transact and any other relevant data, as follows:

key	value
"system" :	The name of the system, service, or program that is making the request. This can be created by whoever is writing the requester end. (When the auth service becomes available, this should probably be the same as the identifier used by the auth service, if the auth service uses something like that.)
"requester" :	Who is making the request. <ul style="list-style-type: none">• If it is the service itself making the request, please use the same value as the is put for the "system" key.• If it is an individual, use a username of some kind - probably an ldap username.
"comment" :	(Optional) Some description as to why the request is being made.
"info" :	(Optional) JSON object that contains extra information about the request. This is the only place where custom parameters are added in the top-level request object; all undefined keys are reserved for future use.

"t":	Number of seconds from the epoch (unix timestamp in seconds) at the time this message was constructed.
.	
.	More members of <jsonRequest> defined by sub-apis
.	

Example Request Body

```
{
  "system": "virals",
  "requester": "btetrud",
  "t": 1344385436,
  "user": "c28k3fjj9",
  "items": {
    "category": "item",
    "id": "12",
    "amount": 1
  }
}
```

Response

The response body is UTF-8 encoded.

The response body, [<jsonResponse>](#), is a JSON object with the following members when:

- The request was successfully completed
 - "result": "success"
- The request was not completed
 - "result": [<failure-type>](#) - is one of the following:
 - "temporaryFailure" indicates the transaction should be retried
 - "permenantFailure" indicates the transaction should not be retried
 - "message": [<error-message>](#) - a safe but informative error message indicating the problem(s) in the request. This is optional if a "type" is given.
 - "type": [<error-type>](#) - The name of the error. This is intended to be a short name, in camel-case, of an error that might need special handling. This is optional if a "message" is given.
 - The following cases have standard types:
 - "missingParameter" - the request is missing a required parameter. Justification: letting implementations send invalid requests can cause problems down the line.
 - "badRequest" - the request has all the required parameters, but is invalid. Examples of this include invalid argument types, or unexpected values.
 - Sub-APIs can define other standard error-types for certain cases.
 - "info": (Optional) JSON object that contains extra information about the error.
 - Sub-APIs can add additional members to [<jsonResponse>](#)

Appropriate HTTP response codes can be returned in addition to (but not instead of) the request body, but the requester should determine success or failure from the "result" field. If the result is malformed, the result is treated like a "temporaryFailure" response.

Change Log

Note: all items in the change log are reflected in the canonical documentation above.

v0.2.2

- Clarified the use of HTTP response codes as being optional, but ignored by the requester.

v0.2.1

- Updated URL Convention to include optional-parameters

v0.2

- Made either message or type required (must have 1).
- Require that HTTP headers be ignored.

- Added a "missingParameter" error type

v0.1

- Created from the [Item Transaction API v1.0](#)