



THE FINE WINE MARKET

LWIN PUSH services

Document revision 1.2
Date of Issue: 09 March 2020
Date of revision: 06 June 2023

Fred Haselton

Business Analyst

Table of Contents

1. Purpose	3
2. Glossary of Terms	3
3. Technical Standards	3
4. API Listing	4
4.1 LWIN PUSH notifications.....	4
4.1.1 LWIN7 Creation	7
4.1.2 LWIN11 creation	8
4.1.3 LWIN7 update.....	10
4.1.4 LWIN11 update	11
4.1.5 LWIN7 deletion	13
4.1.6 LWIN11 deletion	13
4.1.7 LWIN7 combine	14
5. Response Codes	15
5.1 HTTP Status codes.....	15

1. Purpose

To provide the API end point information and examples of the web services available for LWIN PUSH notifications

2. Glossary of Terms

Term	Meaning
LWIN	LWIN - the Liv-ex Wine Identification Number – serves as a universal wine identifier for the wine trade. LWIN is a unique seven to eighteen-digit numerical code that can be used to quickly and accurately identify a product. LWIN allows wine companies to keep their preferred naming system, while introducing a new universal code.
LWIN7	A 7-digit integer code that describes an individual product.
LWIN11	An 11-digit integer code that describes a specific vintage of an individual product.
Combine	The process by which two LWIN codes (always LWIN7) are merged together. The “leader” LWIN is retained and becomes dominant. The “follower” LWIN is has its status changed, is taken out of the live dataset and is given a pointer to its “leader”.
Delete	An LWIN7 or LWIN11 record has its status changed to deleted and is taken out of the live dataset. Note that LWIN carries out non-destructive deletes.

3. Technical Standards

- Permitted users will be issued with a unique token (CLIENT_KEY) and password (CLIENT_SECRET) combination to control the access for all the web services covered under Exchange Integration.
- The web services will consume and produce both XML and JSON. The user can provide the content type in the request header. If the user does not provide any information, then the default content type will be JSON.
- The project will support ISO 8601.
- The project will only support HTTPS protocol for client and server communications.
- The API will support the following methods:
 1. HEAD to ping consuming systems to check if it alive
 2. POST for update operations
- Pretty printing for output readability only is supported if required
- Compression for bandwidth savings are used
- Authentication mechanism will be custom based on CLIENT_KEY and CLIENT_SECRET
- For PUSH services we require a direct POST URL which should be backed by a service capable of accepting HEAD and POST methods and processes JSON or XML payloads

4. API Listing

4.1 LWIN PUSH notifications

Description

The LWIN application will PUSH messages when events take place on the LWIN dataset. Consumers can specify to Liv-ex one (or more) endpoints where they would like the messages received. Failed messages will be retried up to 3 times.

There are 7 possible event types:

1. LWIN7 creation
2. LWIN11 creation
3. LWIN7 update
4. LWIN11 update
5. LWIN7 deletion
6. LWIN11 deletion
7. LWIN7 combine

We recommend all LWIN7-level events are ingested. Ingesting LWIN11-level event will depend on the needs of specific implementation and whether vintage-accurate naming conventions are required.

Endpoint URL

Consuming systems should provide one or more endpoints that updates should be sent to. The URL should be backed by an HTTP-based service capable of receiving HEAD and POST messages and interpreting incoming JSON or XML payloads.

Both HTTP and HTTPS URLs are supported.

Push service requirements – receiving side

An LWIN PUSH message is comprised of 2 parts:

1. A HEAD request (to check the endpoint is alive).
2. A POST payload

The service will first send a HEAD request to ping the Merchant URL before sending the PUSH payload. If the Merchant URL fails to respond with HTTP Code 200 OK, the PUSH notification will not be sent to it.

The LWIN application will attempt to retry sending each event message for 5 times.

HEAD

Payload headers of note:

Key	Value
User-Agent	Mozilla/5.0 (Macintosh; Intel Mac OS X x.y; rv:42.0) Gecko/20100101 Firefox/42.0

POST

Payload headers of note. In addition to the headers below, it is possible to add optional Key and Value pairs to facilitate authentication of source.

Key	Value
Charset	utf-8
Content-type	application/json <u>or</u> application/xml Note: If no value is provided the default content type will be JSON
User-Agent	Mozilla/5.0 (Macintosh; Intel Mac OS X x.y; rv:42.0) Gecko/20100101 Firefox/42.0
[Optional-Key]	[Optional-Value]

Response parameters

All LWIN notifications share the same set of parameters. Depending on the type of event being communicated, either only some of all of the parameters may be populated.

<i>LWIN event</i>	<i>Parameters populated</i>
LWIN7 creation	New LWIN7 code, all LWIN7 metadata
LWIN7 update	Specific LWIN7 code, all LWIN7 metadata
LWIN7 deletion	Specific LWIN7 code only
LWIN11 creation	New LWIN11 code, all LWIN11 metadata
LWIN11 update	Specific LWIN11 code, all LWIN11 metadata
LWIN11 deletion	Specific LWIN11 code only
LWIN7 combine	"Leader" LWIN7 and "follower" LWIN7

Name	Description
lwin	LWIN7 or LWIN11 code depending on type of event Type: integer(7 11)
changeType	The category of change: "lwin7Creation", "lwin11Creation", "lwin7Update", "lwin11Update", "lwin7Deletion", "lwin11Deletion", "lwin7Combine" Type: alphanumeric
changeDate	The timestamp of when the event took place Type: datetime / epoch
combineReference	The new leader LWIN for the LWIN that has been combined

	Type: integer(7)
producerTitle	Title of producer or owner of wine Type: alphanumeric
producerName	Producer or owner of wine Type: alphanumeric
wine	Name of wine (brand and/or grape and/or technical term) Type: alphanumeric
country	Country of origin Type: alphanumeric
region	Region of origin (where applicable, specific to local laws) Type: alphanumeric
subRegion	Sub-region of origin (where applicable, specific to local laws) Type: alphanumeric
site	Site within sub-region (where applicable, specific to local laws) Type: alphanumeric
parcel	Parcel within site (where applicable, specific to local laws) Type: alphanumeric
colour	Colour of LWIN product Type: alphanumeric Values: "white", "red", "rose", null
type	LWIN beverage type Type: alphanumeric
subType	Subcategory of LWIN type Type: alphanumeric
designation	Officially assigned status (specific to local laws) Type: alphanumeric
classification	Officially declared quality level (where applicable, specific to local laws) Type: alphanumeric
vintageConfiguration	Vintage pattern of the wine (e.g. single vintage only, production ended) Should be null for LWIN11 Type: alphanumeric Values: "sequential", "nonSequential", "singleVintageOnly"
vintageValues	Type: array, integer(4)
firstVintage	The first vintage of the LWIN7 Should be null for LWIN11 Type: integer(4)
finalVintage	The final vintage of the LWIN7 Should be null for LWIN11 Type: integer(4)
childOf	Parent wine LWIN7. This should be used to map associated wines to one another. This is a separate relationship to LWIN merge references Should be null for LWIN11 Type: integer (7)

displayNameType	One out of 8 display name concatenation formulas. Assigned by LWIN administrator Type: alphanumeric
displayName	Type: alphanumeric
status	Type: alphanumeric Values: "live", "deleted", "combined"
requestReference	Reference can be provided with "lwin7Creation", "lwin11Creation" metadata if LWIN 7/11 was created for a specific request. Type: integer (11)
dateCreated	Type: datetime / epoch
lastUpdateDate	Type: datetime / epoch

4.1.1 LWIN7 Creation

A new LWIN7 product has been created. The payload features the new LWIN7 code plus metadata for the most recent vintage of the product

LWIN7 creation event example

```

JSON
{
  "lwinwebhook": {
    "lwin": "2027229",
    "eventType": "lwin7Creation",
    "eventDate": 1583762020000,
    "combineReference": null,
    "metaData": {
      "producerTitle": "Bodegas",
      "producerName": "Test",
      "wine": "AutoDummyWine863936",
      "country": "France",
      "region": "Burgundy",
      "subRegion": "Meursault",
      "site": "Porusot",
      "parcel": "Dessous",
      "colour": "Red",
      "type": "Beer",
      "subType": "Sake",
      "designation": "AOP",
      "classification": "classification",
      "vintageConfiguration": "sequential",
      "vintageValues": [
        "2017"
      ],
      "firstVintage": "2016",
      "finalVintage": "2017",
      "childOf": null,
      "displayNameType": "Type 1",
      "displayName": "AutoDummyWine863936, Bodegas, Test, classification,
Meursault",
      "status": "live",
      "requestReference": null,
      "dateCreated": 1583762020000,
      "lastUpdateDate": 1583762020000
    }
  }
}

```

```

}

XML
<lwinWebhookRequest>
  <lwinWebhook>
    <lwin>2027506</lwin>
    <eventType>lwin7Creation</eventType>
    <eventDate>2020-03-09T14:18:01Z</eventDate>
    <combineReference xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>
    <metaData>
      <producerTitle>Bodegas</producerTitle>
      <producerName>Test</producerName>
      <wine>AutoDummyWine588768</wine>
      <country>France</country>
      <region>Burgundy</region>
      <subRegion>Meursault</subRegion>
      <site>Porusot</site>
      <parcel>Dessous</parcel>
      <colour>Red</colour>
      <type>Beer</type>
      <subType>Sake</subType>
      <designation>AOP</designation>
      <classification>classification</classification>
      <vintageConfiguration>sequential</vintageConfiguration>
      <vintageValues>
        <vintage>2017</vintage>
      </vintageValues>
      <firstVintage xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>
      <finalVintage xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>
      <childOf xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>
      <displayNameType>Type 1</displayNameType>
      <displayName>AutoDummyWine588768, Bodegas, Test, classification,
Meursault</displayName>
      <status>live</status>
      <requestReference>3781</requestReference>
      <dateCreated>2020-03-09T14:18:01Z</dateCreated>
      <lastUpdateDate>2020-03-09T14:18:01Z</lastUpdateDate>
    </metaData>
  </lwinWebhook>
</lwinWebhookRequest>

```

4.1.2 LWIN11 creation

A new LWIN11 (i.e. a new vintage) for an existing LWIN has been created. The payload features the new LWIN11 code and the specific metadata associated with it.

LWIN11 creation event example

```

JSON
{
  "lwinWebhook": {
    "lwin": "20271441988",
    "eventType": "lwin11Creation",
    "eventDate": "1583761710000",
    "combineReference": null,
    "metaData": {
      "producerTitle": "Chateau",
      "producerName": "PUSHtest",
      "wine": "Test A1",

```



```

    "country": "France",
    "region": "Alsace",
    "subRegion": "Mambourg",
    "site": null,
    "parcel": null,
    "colour": "Red",
    "type": "Wine",
    "subType": "Still",
    "designation": "VT",
    "classification": "1er Cru",
    "vintageConfiguration": null,
    "vintageValues": [
      "1988"
    ],
    "firstVintage": null,
    "finalVintage": null,
    "childOf": null,
    "displayNameType": "Type 1",
    "displayName": "Test A1, Chateau, PUSHtest, 1er Cru, Mambourg",
    "status": "live",
    "requestReference": null,
    "dateCreated": 1583761710000,
    "lastUpdateDate": 1583761710000
  }
}
}

XML
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<lwinWebhookRequest>
  <lwinWebhook>
    <lwin>20274342017</lwin>
    <eventType>lwin11Creation</eventType>
    <eventDate>2020-03-09T14:13:58Z</eventDate>
    <combineReference xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>
    <metaData>
      <producerTitle>Bodegas</producerTitle>
      <producerName>Test</producerName>
      <wine>AutoDummyWine113102</wine>
      <country>France</country>
      <region>Burgundy</region>

      <subRegion>Meursault</subRegion>
      <site>Porusot</site>
      <parcel>Dessous</parcel>
      <colour>Red</colour>
      <type>Beer</type>
      <subType>Sake</subType>
      <designation>AOP</designation>
      <classification>classification</classification>
      <vintageConfiguration xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:nil="true"/>
      <vintageValues>
        <vintage>2017</vintage>
      </vintageValues>
      <firstVintage xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>
      <finalVintage xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>
      <childOf xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>
      <displayNameType>Type 1</displayNameType>
      <displayName>AutoDummyWine113102, Bodegas, Test, classification,
Meursault</displayName>

```

```

        <status>live</status>
        <requestReference xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>
        <dateCreated>2020-03-09T14:13:58Z</dateCreated>
        <lastUpdateDate>2020-03-09T14:13:58Z</lastUpdateDate>
        </metaData>
    </lwinWebhook>
</lwinWebhookRequest>

```

4.1.3 LWIN7 update

An edit has taken place to an existing LWIN7. This could be a change to the product's type, colour information, or the addition of a vintageValue. The payload features the latest view of the LWIN7 metadata [in full](#).

LWIN11 update event example

```

JSON
{
  "lwinWebhook": {
    "lwin": "2027229",
    "eventType": "lwin7Update",
    "eventDate": 1583762273000,
    "combineReference": null,
    "metaData": {
      "producerTitle": "Bodegas",
      "producerName": "Test",
      "wine": "AutoDummyWine863936",
      "country": "France",
      "region": "Burgundy",
      "subRegion": "Meursault",
      "site": "Porusot",
      "parcel": "Dessous",
      "colour": "Red",
      "type": "Wine",
      "subType": "Still",
      "designation": "AOP",
      "classification": "1er Cru",
      "vintageConfiguration": null,
      "vintageValues": [
        "2017"
      ],
      "firstVintage": null,
      "finalVintage": null,
      "childOf": null,
      "displayNameType": "Type 1",
      "displayName": "AutoDummyWine863936, Bodegas, Test, 1er Cru, Meursault",
      "status": "live",
      "requestReference": null,
      "dateCreated": 1583762020000,
      "lastUpdateDate": 1583762273000
    }
  }
}

XML
<lwinWebhookRequest>
  <lwinWebhook>
    <lwin>2027144</lwin>
    <eventType>lwin7Update</eventType>
    <eventDate>2020-03-09T14:18:19Z</eventDate>

```

```

        <combineReference xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>
        <metaData>
            <producerTitle>Chateau</producerTitle>
            <producerName>PUSHtest</producerName>
            <wine>Test A1</wine>
            <country>France</country>
            <region>Alsace</region>
            <subRegion>Mambourg</subRegion>
            <site xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>
            <parcel xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>
                <colour>Rose</colour>
                <type>Wine</type>
                <subType>Still</subType>
                <designation>VT</designation>
                <classification>1er Cru</classification>
                <vintageConfiguration>sequential</vintageConfiguration>
                <vintageValues>
                    <vintage>2012</vintage>
                    <vintage>2011</vintage>
                    <vintage>2010</vintage>
                    <vintage>2009</vintage>
                    <vintage>2008</vintage>
                </vintageValues>
                <firstVintage xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>
                <finalVintage xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>
                <childOf xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>
                    <displayNameType>Type 5</displayNameType>
                    <displayName>Chateau, PUSHtest, Test A1, 1er Cru, Mambourg,
Alsace</displayName>
                    <status>live</status>
                    <requestReference xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>
                    <dateCreated>2020-03-09T09:52:56Z</dateCreated>
                    <lastUpdateDate>2020-03-09T14:18:19Z</lastUpdateDate>
                </metaData>
            </lwinWebhook>
        </lwinWebhookRequest>
    
```

4.1.4 LWIN11 update

An edit has been made at a vintage-specific level. This could be an update to the producer name (e.g a Burgundy estate was inherited) or to the designation of the wine (local laws changed). The payload features the latest view of the LWIN11 metadata in full.

If the LWIN11 is the most recent vintage of the product, an update will also take place to the associated LWIN7 record to reflect these LWIN11 changes.

LWIN11 update event example

```

JSON
{
  "lwinWebhook": {
    "lwin": "20272292017",
    "eventType": "lwin11Update",
    "eventDate": 1583762216000,
    "combineReference": null,
    "metaData": {
      "producerTitle": "Bodegas",
      "producerName": "Test",
      "wine": "AutoDummyWine863936",
      "country": "France",
      "region": "Burgundy",
      "subRegion": "Meursault",
      "site": "Porusot",
      "parcel": "Dessous",
      "colour": "Red",
      "type": "Beer",
      "subType": "Sake",
      "designation": "AOP",
      "classification": "1er Cru",
      "vintageConfiguration": null,
      "vintageValues": [
        "2017"
      ],
      "firstVintage": null,
      "finalVintage": null,
      "childOf": null,
      "displayNameType": "Type 1",
      "displayName": "AutoDummyWine863936, Bodegas, Test, 1er Cru, Meursault",
      "status": "live",
      "requestReference": null,
      "dateCreated": 1583762020000,
      "lastUpdateDate": 1583762216000
    }
  }
}

XML
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<lwinWebhookRequest>
  <lwinWebhook>
    <lwin>20271442015</lwin>
    <eventType>lwin11Update</eventType>
    <eventDate>2020-03-09T14:16:27Z</eventDate>
    <combineReference xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>
    <metaData>
      <producerTitle>Chateau</producerTitle>
      <producerName>PUSHtest</producerName>
      <wine>Test B2</wine>
      <country>France</country>
      <region>Alsace</region>
      <subRegion>Mambourg</subRegion>
      <site xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>
      <parcel xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>
      <colour>White</colour>
      <type>Wine</type>
      <subType>Still</subType>
      <designation>VT</designation>
      <classification>1er Cru</classification>
      <vintageConfiguration xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:nil="true"/>

```

```

        <vintageValues>
          <vintage>2015</vintage>
        </vintageValues>
        <firstVintage xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>
        <finalVintage xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>
        <childOf xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>
        <displayNameType>Type 1</displayNameType>
        <displayName>Test B2, Chateau, PUSHtest, 1er Cru, Mambourg</displayName>
        <status>live</status>
        <requestReference xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>
        <dateCreated>2020-03-09T09:52:57Z</dateCreated>
        <lastUpdateDate>2020-03-09T14:16:27Z</lastUpdateDate>
      </metaData>
    </lwinWebhook>
  </lwinWebhookRequest>

```

4.1.5 LWIN7 deletion

An LWIN7 product has been marked as deleted in the dataset. Note: LWIN7-level deletion events are very rare. In most scenarios LWINs are combined rather than deleted.

LWIN7 delete event example

```

JSON
{
  "lwinWebhook": {
    "lwin": "2548074",
    "changeType": "lwin7Deletion",
    "changeDate": 1579530323000,
    "combineReference": null,
    "metaData": null
  }
}

XML
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<lwinWebhookRequest>
  <lwinWebhook>
    <lwin>2027144</lwin>
    <eventType>lwin7Deletion</eventType>
    <eventDate>2020-03-09T14:22:43Z</eventDate>
    <combineReference xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>
    <metaData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>
  </lwinWebhook>
</lwinWebhookRequest>

```

4.1.6 LWIN11 deletion

A vintage (LWIN11) has been deleted from an existing LWIN7.

LWIN11 delete event example

```

JSON
{
  "lwinWebhook": {
    "lwin": "20271442014",
    "eventType": "lwin11Deletion",
    "eventDate": 1583761602000,
    "combineReference": null,
    "metaData": null
  }
}

XML
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<lwinWebhookRequest>
  <lwinWebhook>
    <lwin>20271442015</lwin>
    <eventType>lwin11Deletion</eventType>
    <eventDate>2020-03-09T14:21:56Z</eventDate>
    <combineReference xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>
    <metaData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>
  </lwinWebhook>
</lwinWebhookRequest>

```

4.1.7 LWIN7 combine

An LWIN7 has been combined into another LWIN7. This usually takes place when a duplicate LWIN record is discovered. The event shows the LWIN7 record that has been combined (the “follower” LWIN) and in **combinedReference** the “leader” LWIN7 that now becomes the primary reference for the product.

LWIN7 combine event example

```

JSON
{
  "lwinWebhook": {
    "lwin": "2027173",
    "changeType": "lwin7Combine",
    "changeDate": 1583755150000,
    "combineReference": "2027160",
    "metaData": null
  },
},

XML
<lwinWebhookRequest>
  <lwinWebhook>
    <lwin>2027535</lwin>
    <eventType>lwin7Combine</eventType>
    <eventDate>2020-03-09T14:26:48Z</eventDate>
    <combineReference>2027522</combineReference>
    <metaData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>
  </lwinWebhook>
</lwinWebhookRequest>","
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<lwinWebhookRequest>
  <lwinWebhook>
    <lwin>20275352015</lwin>
    <eventType>lwin11Update</eventType>
    <eventDate>2020-03-09T14:26:48Z</eventDate>
    <combineReference xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>

```

```

        <metaData>
          <producerTitle>Bodega</producerTitle>
          <producerName>NP</producerName>
          <wine>PUSH_02</wine>
          <country>Italy</country>
          <region>Tuscany</region>
          <subRegion xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>
          <site xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>
          <parcel xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>
          <colour>White</colour>
          <type>Wine</type>
          <subType>Still</subType>
          <designation xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>
          <classification>IGT</classification>
          <vintageConfiguration xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:nil="true"/>
          <vintageValues>
            <vintage>2015</vintage>
          </vintageValues>
          <firstVintage xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>
          <finalVintage xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>
          <childOf xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>
          <displayNameType>Type 1</displayNameType>
          <displayName>PUSH_02, Bodega, NP, IGT</displayName>
          <status>combined</status>
          <requestReference xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>
          <dateCreated>2020-03-09T14:26:11Z</dateCreated>
          <lastUpdateDate>2020-03-09T14:26:48Z</lastUpdateDate>
        </metaData>
      </lwinWebhook>
    </lwinWebhookRequest>

```

5. Response Codes

This section describes the response codes that will be returned by the service

5.1 HTTP Status codes

HTTP defines a bunch of meaningful status codes that can be returned from our API. These can be leveraged to help our API Merchants/consumers route their responses accordingly:

Code	Message
200 OK	Response to a successful GET, POST, PUT, DELETE. Can also be used for a POST that doesn't result in a creation.
201 Created	Response to a POST that results in a creation.

202 Accepted	The request has been accepted and will be processed later. It is a classic answer to asynchronous calls (for better UX or performances).
204 No Content	Response to a successful request that won't be returning a body (like a DELETE request)
400 Bad Request	The request is malformed, such as if the body does not parse
401 Unauthorized	When no and/or invalid authentication details are provided. Can also be used to trigger an auth popup if API is used from a browser
403 Forbidden	When authentication succeeded but authenticated user doesn't have access to the resource
404 Not Found	When a non-existent resource is requested
405 Method Not Allowed	When an HTTP method is being requested that isn't allowed for the authenticated user
406 Not Acceptable	Nothing matches the Accept-* Header of the request. As an example, you ask for an XML formatted resource but it is only available as JSON.
409 Conflict	Indicates one or more supplied parameters are triggering a validation error. A relevant TR code should be returned in the response.
410 Gone	Indicates that the resource at this end point is no longer available. Useful as a blanket response for old API versions
415 Unsupported Media Type	If incorrect content type was provided as part of the request
422 Unprocessable Entity	Used for validation errors. Should be used if the server cannot process the entity, e.g. if an image cannot be formatted or mandatory fields are missing in the payload.
429 Too Many Requests	When a request is rejected due to rate limiting
500 Internal Server Error	The general catch-all error when the server-side throws an exception. The request may be correct, but an execution problem has been encountered at our end.