Synkka Data Pool Operations Manual

Document Summary

|  |  |
| --- | --- |
| Document Item | Current Value |
| Document Name | Synkka Data Pool Operations Manual |
| Document Date |  |
| Document Version | 2 |
| Document Status |  |
| Document Description | This document specifies and describes the basic functionalities of Synkka data pool from the integrating partner’s point of view. |

Log of Changes

|  |  |  |  |
| --- | --- | --- | --- |
| Release | Date of Change | Changed By | Summary of Change |
| 0.1 | 20/06/2017 | Hannu Lehtonen | First version of the document. |
| 0.2 | 24/10/2017 | Ari Vuorio | Media publishing and subscriptions added to text |
| 0.3 | 21/05/2019 | Raija Perälä | Updated links and API description |
| 0.4 | 04/12/2019 | Hannu Lehtonen | Updated chapter 6.1.6 Withdraw Item Hierarchy |

Table of Contents

[Synkka Data Pool Operations Manual 1](#_Toc9418528)

[1. Introduction 5](#_Toc9418529)

[1.1. Purpose of this Document 5](#_Toc9418530)

[1.2. Audience 5](#_Toc9418531)

[1.3. What is the GDSN network? 5](#_Toc9418532)

[1.4. GDSN Standards and documentation 6](#_Toc9418533)

[2. Message to Message Communication Interfaces 6](#_Toc9418534)

[3. Message validation 8](#_Toc9418535)

[4. Messages 9](#_Toc9418536)

[4.1. Catalogue Item Notification (CIN) 9](#_Toc9418537)

[4.2. GS1 Response EXCEPTION 13](#_Toc9418538)

[4.3. GS1 Response ACCEPTED 15](#_Toc9418539)

[4.4. CIC Confirmations 16](#_Toc9418540)

[4.4.1. When to expect a CIC message 16](#_Toc9418541)

[4.4.2. Synkka data pool CIC RECEIVED 18](#_Toc9418542)

[4.4.3. Synkka data pool CIC REVIEW 20](#_Toc9418543)

[4.5. Catalogue Item Subscription (CIS) 21](#_Toc9418544)

[4.5.1. Catalogue Item Subscription DELETE 22](#_Toc9418545)

[4.6. Request for Catalogue Item Notification (RFCIN) 23](#_Toc9418546)

[4.7. Catalogue Item Hierarchical Withdrawal (CIHW) 24](#_Toc9418547)

[5. Use cases 25](#_Toc9418548)

[5.1. Data Source use cases 25](#_Toc9418549)

[5.1.1. Publish new item hierarchy 25](#_Toc9418550)

[5.1.2. Change Item Hierarchy 25](#_Toc9418551)

[5.1.3. Correct Item Hierarchy 25](#_Toc9418552)

[5.1.4. Cancel Item Hierarchy 26](#_Toc9418553)

[5.1.5. Discontinue Item Hierarchy 26](#_Toc9418554)

[5.1.6. Withdraw Item Hierarchy 27](#_Toc9418555)

[5.2. Data recipient use cases 27](#_Toc9418556)

[5.2.1. Create a subscription 27](#_Toc9418557)

[5.2.2. Remove subscription 27](#_Toc9418558)

[5.2.3. Create a RFCIN 28](#_Toc9418559)

[6. JSON API 28](#_Toc9418560)

[6.1. PUBLIC 28](#_Toc9418561)

[6.2. GDSN 28](#_Toc9418562)

[7. Media publishing and subscriptions 28](#_Toc9418563)

[7.1. Overview 28](#_Toc9418564)

[7.2. Main entities 29](#_Toc9418565)

[7.2.1. MediaItem 29](#_Toc9418566)

[7.2.2. MediaItemWithExtendedInfo 29](#_Toc9418567)

[7.2.3. MediaSubscription 29](#_Toc9418568)

[7.2.4. MediaCollection 30](#_Toc9418569)

[7.2.5. MediaShare 30](#_Toc9418570)

[7.3. Entity versioning 30](#_Toc9418571)

[7.4. Authentication 30](#_Toc9418572)

[7.5. Errors 30](#_Toc9418573)

[8. Media API 30](#_Toc9418574)

[8.1. Account 31](#_Toc9418575)

[8.1.1. Log in into the api 31](#_Toc9418576)

[8.1.2. Log Out from the api 31](#_Toc9418577)

[8.1.3. Switch between companies if the user has more than one 32](#_Toc9418578)

[8.2. MediaItem 32](#_Toc9418579)

[8.2.1. List MediaItems 32](#_Toc9418580)

[8.2.2. Create or Update a MediaItem 33](#_Toc9418581)

[8.2.3. Get a MediaItem by id 35](#_Toc9418582)

[8.2.4. Delete a MediaItem by Id 36](#_Toc9418583)

[8.2.5. Download the binary data of a MediaItem 36](#_Toc9418584)

[8.2.6. Add/Remove a MediaItem to/from products 38](#_Toc9418585)

[8.2.7. Add/Remove several MediaItems to/from products 39](#_Toc9418586)

[8.2.8. Create MediaItems with posting multipart/form-data 40](#_Toc9418587)

[8.3. MediaSubscription 40](#_Toc9418588)

[8.3.1. List MediaSubscriptions 40](#_Toc9418589)

[8.3.2. Create or Update a MediaSubscription 41](#_Toc9418590)

[8.3.3. Get a MediaSubscription by id. 43](#_Toc9418591)

[8.3.4. Delete a MediaSubscription by Id 44](#_Toc9418592)

[8.3.5. List MediaSubscriptionMatches for a MediaSubscription 44](#_Toc9418593)

[8.3.6. Confirm a single subscription match 45](#_Toc9418594)

[8.3.7. Confirm all matches for a given subscription until (and including) a matchId 46](#_Toc9418595)

[8.3.8. Clear all matches and item state associated to a MediaSubscription 46](#_Toc9418596)

[9. Appendix 48](#_Toc9418597)

[9.1. Abbreviations 48](#_Toc9418598)

[9.2. Glossary of terms 48](#_Toc9418599)

# Introduction

## Purpose of this Document

This document specifies and describes the basic functionalities of Synkka data pool.

Synkka data pool is based on the Global Data Synchronisation Network (GDSN) standard which is maintained by GS1 globally. The platfrom GS1Trade Sync is maintained by Trade Connectors.

All the screenshots in this document are only to give the reader a generic picture of the XML messages. For more comprehensive example messages please visit GS1 Finland’s web page.

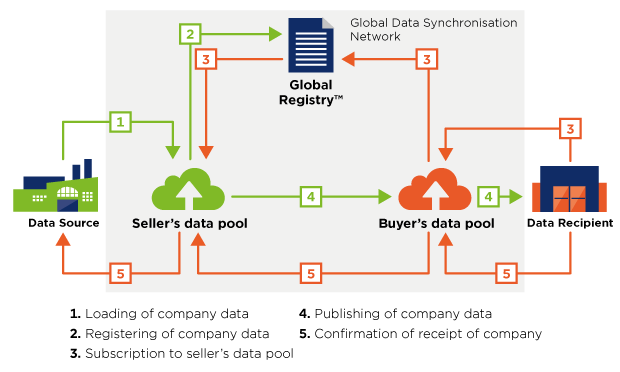
## Audience

This document is intended for business persons and developers who will implement GDSN in their own systems and already have a general knowledge of the GDSN and GS1 standards.

## What is the GDSN network?

The GS1 Global Data Synchronisation Network® (GDSN®) is a network of interoperable data pools enabling collaborating users to securely synchronise master data based on GS1 standards. GDSN supports accurate, real-time data sharing and trade item updates among subscribed trading partners.

Synkka data pool is a certified GDSN data pool which is connected to Global Registry, a central directory which keeps track of connections, guarantees the uniqueness of data and ensures compliance with shared GS1 standards. There are many GDSN data pools spread across the world. All items will be registered to the Global Registry.



*(*[*https://www.gs1.org/how-gdsn-works*](https://www.gs1.org/how-gdsn-works)*)*

More information including case studies and detailed documentation can be found from GS1 global web pages <https://www.gs1.org/gdsn>

## GDSN Standards and documentation

We recommend that all developers read at least the GDSN Operations Manual Release 3.1 and GDSN 3.1 Trade Item Implementation Guide documents which can be found from GS1 Global’s web pages too. All other relevant and useful documentation can be found from the following link: <https://www.gs1.org/standards/gdsn>.

The M2M partners should also be aware that the GDSN standard is a developing standard and is currently being updated twice a year. These updates affect the code lists, Global Product Classification, XML schemas, attributes and validations.

# Message to Message Communication Interfaces

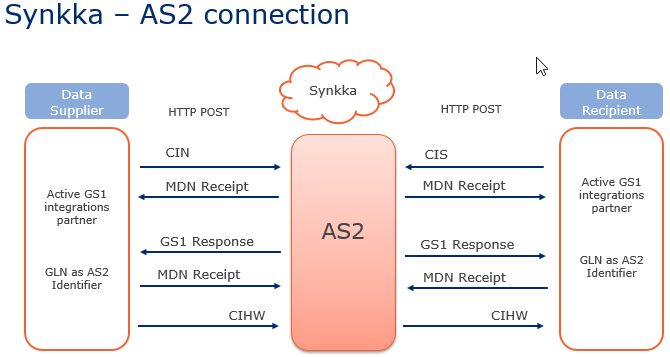
Synkka data pool supports AS2 and FTPS communication standards. The message format used in M2M communication is XML which is based on the GDSN XML schema. Further information about these standards can be found from the AS2 and FTPS connectivity guides which can be downloaded from GS1 Finland’s web pages.

GS1 Finland Synkka Data Pool AS2 Connectivity Guide download link:

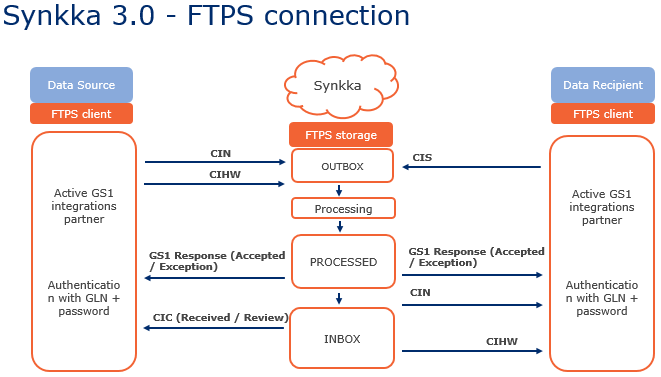
<https://asiakas.gs1.fi/content/download/27742/311601/file/GS1+Finland_Synkka_Data_Pool_AS2_Connectivity_Guide.pdf>

GS1 Finland Synkka  Data Pool FTPS Connectivity Guide download link:

<https://asiakas.gs1.fi/content/download/27743/311606/file/GS1_Finland_Synkka_Data_Pool_FTPS_Connectivity_Guide.pdf>



*Summary of the AS2 communication between the parties in the local network*



*Summary of the FTPS communication between the parties in the local network*

The purpose of the different business messages (CIN, CIS, CIHW etc.) are described later in this document.

# Message validation

Every message the data pool receives is being validated. Depending of the message in question the validations will include schema validation, GDSN validation rule and/or local validation rule validations.

The schema validation basically means that the structure of the message is checked in to order to ensure it is according to the schema that is specified in the GDSN standard. The schemas are available on the GS1 Global’s home page.

GDSN validation rules are common for all the data pools within the GDSN network. This means for example in case the data source would like to publish an item data for a particular data recipient operating on the Finnish target market, the message must pass these validations in order to be published in Synkka data pool.

Local validation rules are rules which have been specified due to national legislations, agreements and to enhance consistency of the data within the local network. These rules also give guidance for the data sources so that they could provide all the relevant data regarding each item.

Here is a list of the most common GDSN messages in Synkka data pool. These messages will be covered in detail later in this document:

* Catalogue Item Notification (CIN)
* GS1 Response ACCEPTED and GS1 Response EXCEPTION
* Catalogue Item Confirmation (CIC)
* Catalogue Item Subscription (CIS)
* Catalogue Item Subscription (CIS) DELETE
* Request for Catalogue Item Notification (RFCIN)
* Catalogue Item Hierarchical Withdrawal (CIHW)

# Messages

This chapter describes briefly the messages the data source or the data recipient should be prepared to send and receive. All the GDSN messages are based on the schema.

## Catalogue Item Notification (CIN)

A CIN is a carrier message which carries the information of the trade item hierarchy. All the actual item data is included in this message. Each message can include several CINs with the following restrictions:

1. There is a limit of 1 Document type within 1 Message

2. There is a limit of 1000 Transactions within 1 Message

3. There is a limit of 1 Command type within 1 Transaction

4. There is a limit of 100 Documents within 1 Transaction

For more information about the principals of the CIN message please see the GDSN Operations Manual Release 3.1 chapter 6. Example messages can be downloaded from GS1 Finland’s web page or by requesting them from GS1 Finland’s customer support.

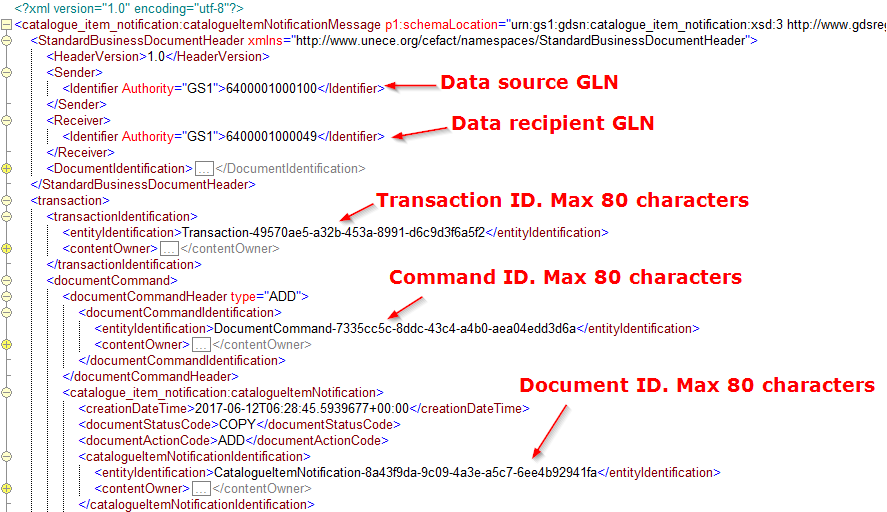
Synkka data pool validates the message and returns the data source a GS1 Response with the status EXCEPTION or ACCEPTED. FTPS users will receive a copy of the sent message and the GS1 Response message in the Processed-folder on the FTPS server once the file has been processed.

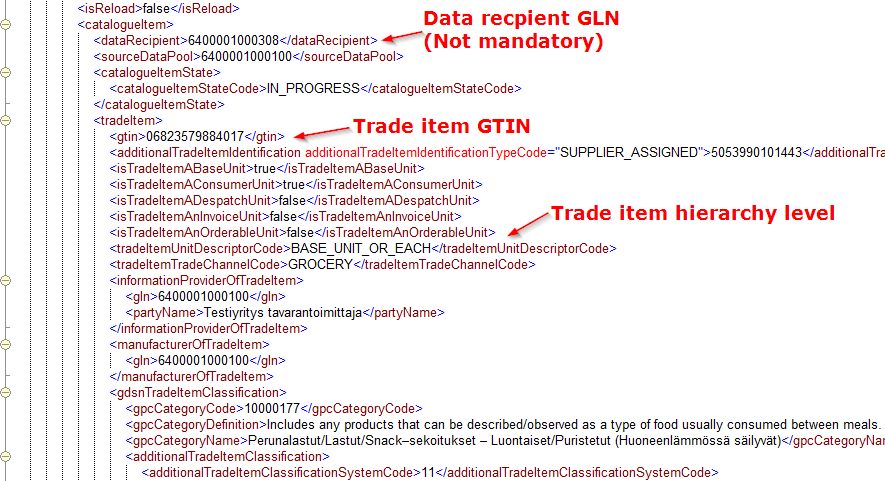
Data recipients using FTPS will receive these messages to their Inbox-folder on the FTPS server.

When sending data to an external data pool the validation errors are often included on a CIC REVIEW message. So, if the data source is sending data to an external data pool they should implement the CIC messages in their own system as well. FTPS users will receive these message to their Inbox-folder.

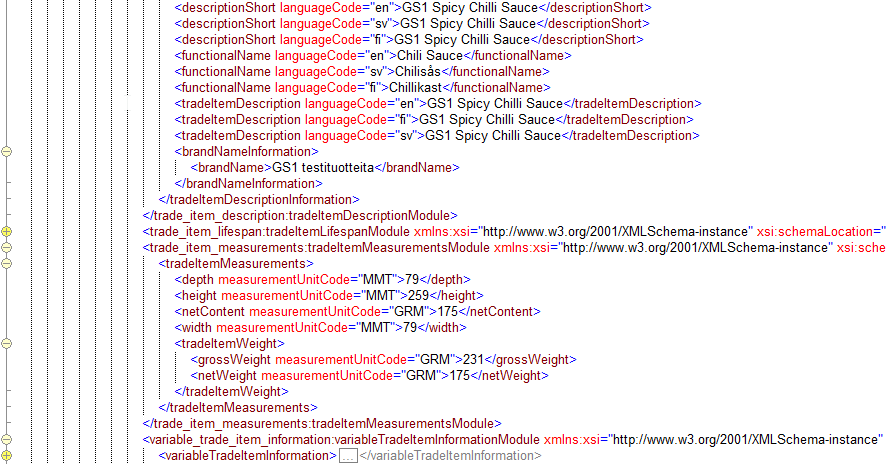
**IMPORTANT NOTICE FOR DATA RECIPIENTS:** Because the GDSN standard makes it possible to send product information from one data pool to another, it should be noticed that the CIN messages may include some attributes that are not specified in Finnish item profile. This is a normal phenomenon in the GDSN network and can be handled by reading in only the specified XML modules and attributes. This way the additional attributes that may come within a CIN message can be ignored and should not cause an error in the receiving systems.

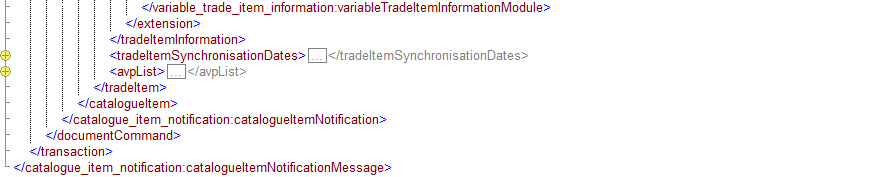
*Example of a CIN message. Please note that some elements are folded.*











## GS1 Response EXCEPTION

If the message fails the validations (either GDSN or local validations), Synkka data pool will return a GS1 Response message with the status “EXCEPTION”. The message includes a description of the error and indicates the GTIN of the item where the error occurred.

FTPS users will receive this message in the Processed-folder of the FTPS server once the sent file has been processed.

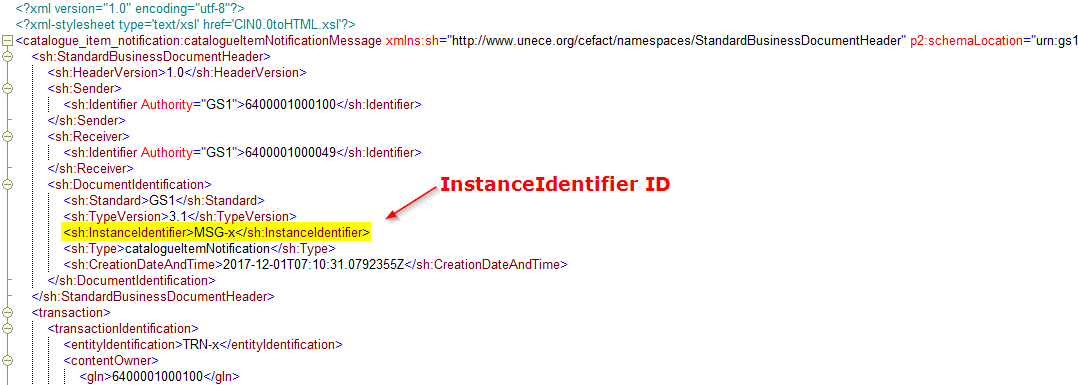
*Example of a GS1 Response message with the status EXCEPTION.*



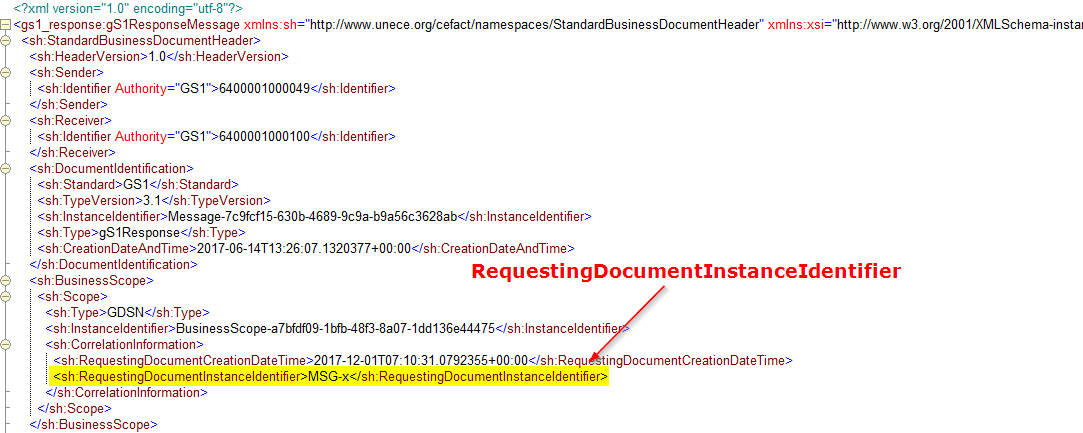


The GS1 Response EXCEPTION message includes the GTIN where the error occurred. In addition the sent message (CIN) and the GS1 Response message can be matched between each other by the InstanceIdentfier ID. The CIN message’s InstanceIdentifier ID is placed in the RequestingDocumentInstanceIdentifier in the GS1 Response message.

CIN:

**

GS1 Response:

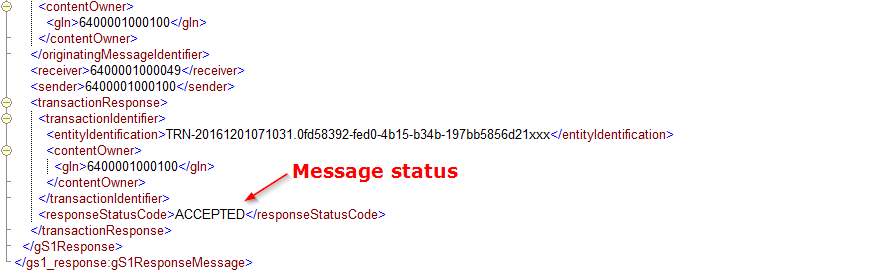
**

## GS1 Response ACCEPTED

If the message passes all validations, Synkka data pool will return a GS1 Response message with the status “ACCEPTED”.

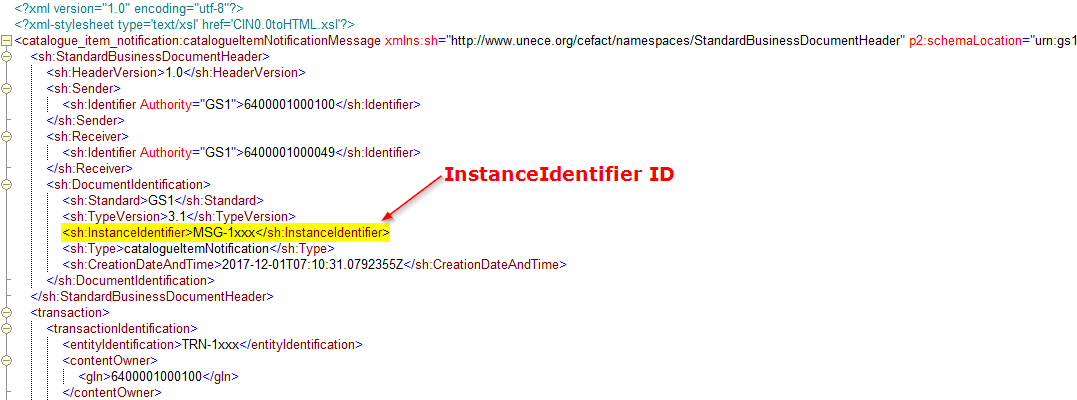
FTPS users will receive this message in the Processed-folder of the FTPS server once the sent file has been processed.

*Example of a GS1 Response message with the status ACCEPTED.*

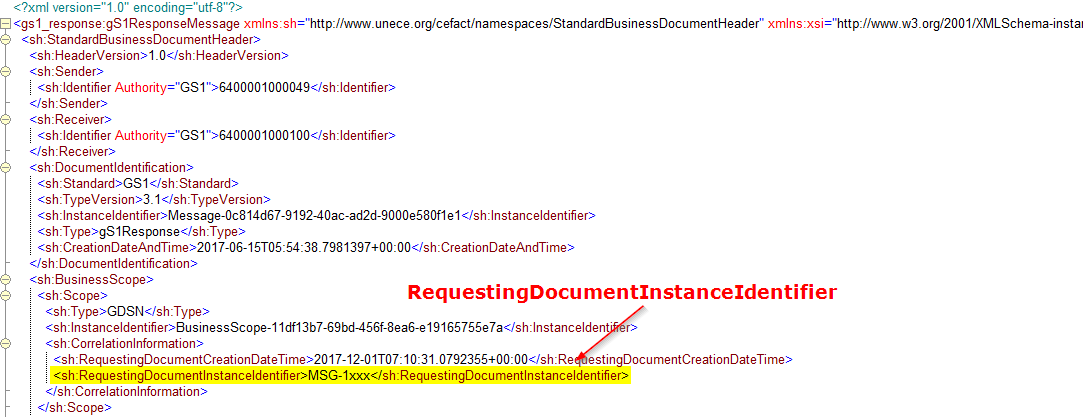


The GS1 Response ACCEPTED message does not include the GTIN of the accepted item. The sent message (CIN) and the GS1 Response message can be matched between each other by the InstanceIdentfier ID. The CIN messages InstanceIdentifier ID is placed in the RequestingDocumentInstanceIdentifier in the GS1 Response message. See next page.

CIN:

**

GS1 Response:

**

## CIC Confirmations

The data recipient can send a CIC message to the data source as a verification that the data recipient received the CIN message, and as a feedback on the CIN message. This CIC message is not mandatory. The feedback can be positive (status “RECEIVED” or “SYNCHRONIZED”) or negative (status “REVIEW” or “REJECTED”):

* **RECEIVED** – The received CIN message follows the data recipient’s specific rules, but no business decision has been made on the data. NOTE: There are no recipient specific rules in the Finnish target market at the moment.
* **REVIEW** - The data recipient can comment the trade item data and request that the data source changes or corrects the trade item characteristics.
* **REJECTED** - The data recipient is not interested in the trade item information and does not want to receive the CIN of the item anymore.
* **SYNCHRONIZED** – item information is approved by the data recipient and is integrated into the data recipient’s internal systems.

NOTE: GDSN standard includes a possibility to have trading partner dependent validation rules. These validation errors would be communicated with CIC messages.

### When to expect a CIC message

* For every sent CIN, the data recipient can send several CIC messages.
* When an item is sent to a recipient, Synkka will always send a CIC Received message to the data source.
* The CIC can be positive from one data recipient but negative from another.
* The data recipient can decide to send only positive or only negative CIC messages.
* If no specific data recipient is specified in the CIN, a CIC can be sent from any data recipient that subscribes on the trade item data.
* The data recipient can decide not to send CIC messages, because it is not mandatory.

### Synkka data pool CIC RECEIVED

Synkka data pool sends the data source a CIC message with the confirmation state RECEIVED when the CIN is forwarded to a data recipient. If the item has been subscribed by e.g. 20 data recipients, the system will send the CIC message to these 20 data sources. The purpose of this message is to indicate the data source that message has been send forward to the trading partner.

FTPS users will receive this message in the Inbox-folder of the FTPS server.

*Example of a CIC message with the confirmation state RECEIVED.*





### Synkka data pool CIC REVIEW

The data recipient (from the local or external data pool) can send CIC messages to the data sources e.g. if they want to confirm that the item data has been received or if they want that the data source corrects some information on the item data. The Finnish data recipients are not using this functionality at the moment.

FTPS users will receive this message in the Inbox-folder of the FTPS server.

*Example of a CIC message with the confirmation state REVIEW.*



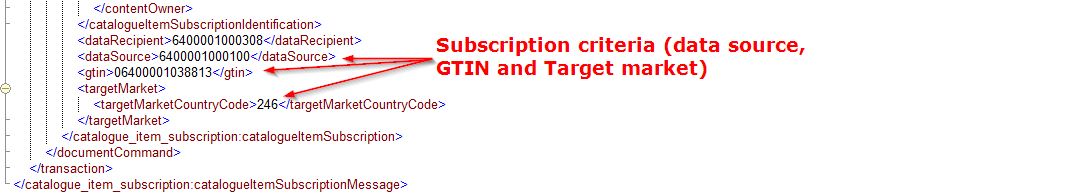
## Catalogue Item Subscription (CIS)

The data recipient can create a subscription either via the Synkka data pool’s user interface or by sending Catalogue Item Subscription (CIS) message to the data pool. The subscription criterions available are the same on the user interface and the CIS message: Data Source, GPC (Global Product Classification), GTIN and Target Market. Each message can have more than one subscription (CIS) included as long as the message follows the general rules of the GDSN business messages described in the GDSN Operations Manual Release 3.1 chapter 6. Note that the Target Market cannot be used alone as a subscription criteria.

Synkka data pool validates the CIS message and returns the data recipient a GS1 Response message with the status EXCEPTION or ACCEPTED. FTPS users will receive a copy the sent message and the GS1 Response message in the Processed-folder of the FTPS server once the file has been processed. The system will add the subscription match (CIN) to the Inbox-folder of the FTPS server.

*Example of a message with two CIS messages included*

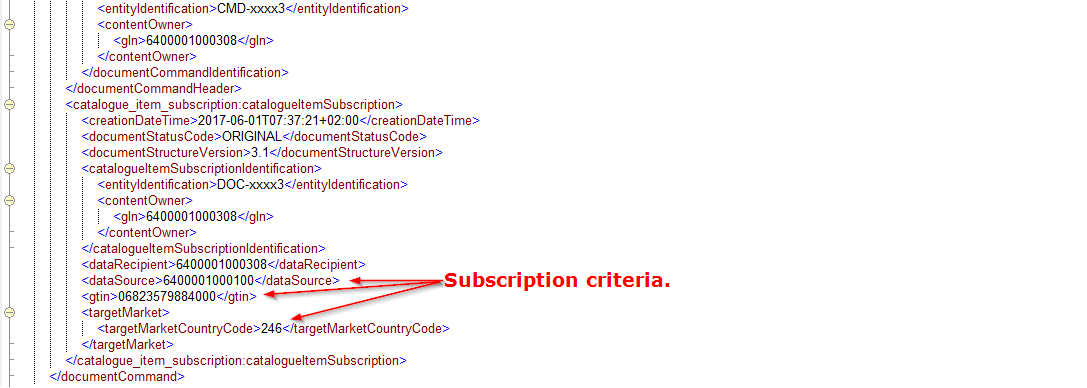




### Catalogue Item Subscription DELETE

The message for removing a subscription is like the subscription message. The criteria should match with the existing subscription. In this case the command should be DELETE.

*Example of a CIS DELETE message*

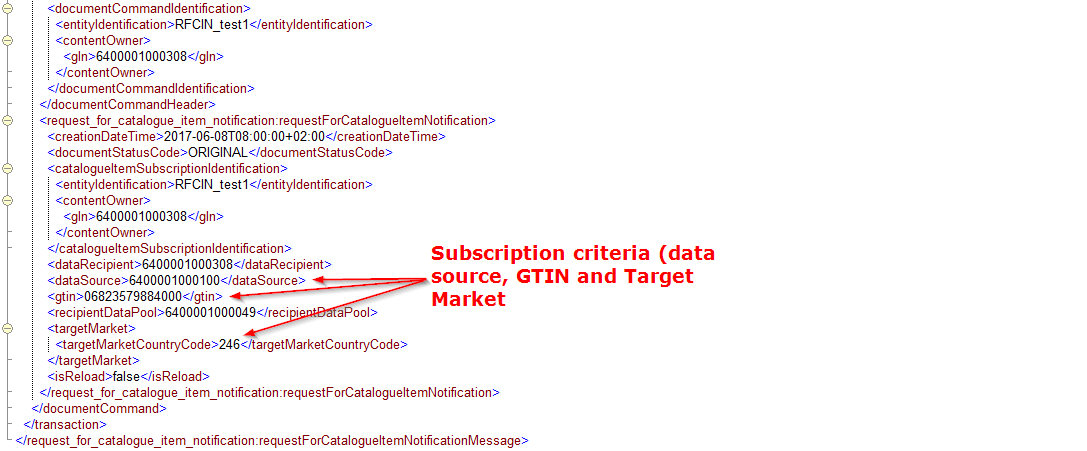


## Request for Catalogue Item Notification (RFCIN)

Sometimes the data recipient needs to get some trade item’s information again although the item GTIN or source GLN may already exist in their synchronisation list (Synchronisation list is created based on the subscription matches between the published item data and data recipient’s subscriptions). If an item already exists in the synchronisation list it cannot be subscribed again by creating a new subscription. Instead the data recipient must create a Request for Catalogue Item Notification message and send it to the data pool. This message will trigger a one-time delivery of the requested item data. Items requested with RFCIN are not added to the synchronisation list. Target Market cannot be used alone as a subscription criteria.

Synkka data pool validates the RFCIN message and returns the data recipient a GS1 Response message with the status EXCEPTION or ACCEPTED. FTPS users will receive a copy of the sent message and the GS1 Response message in the Processed-folder of the FTPS server once the file has been processed. The system will add the subscription match (CIN) to the Inbox-folder of the FTPS server.

*Example of a RFCIN message*



## Catalogue Item Hierarchical Withdrawal (CIHW)

When a data source wants to stop the synchronization of an item hierarchy to some data recipient, it can be done by sending a Catalogue Item Hierarchical Withdrawal message to Synkka data pool. In release 2.8 this was achieved by sending a CIN with the document command header DELETE to Synkka. However, this method should not be used in version 3.1. The GTIN to be used in the message is the top level GTIN of the hierarchy.

Synkka data pool validates the message and returns the data source a GS1 Response with the status EXCEPTION or ACCEPTED. FTPS users will receive a copy of the sent message and the GS1 response message in the Processed-folder on the FTPS server once the file has been processed. The status of the item will change to DRAFT after it has been withdrawn. All items in DRAFT-state can be removed from the data pool manually by the user via the user interface.

Data recipients using FTPS will receive the CIHW messages in to their Inbox-folder on the FTPS server.

*Example of a CIHW message*



# Use cases

This chapter includes examples of the common use cases and explains the basic steps that the data source or data recipient needs to take to use the main functionalities of the system. For more detailed descriptions please see the Trade Connectors – Functional Specification documents.

## Data Source use cases

### Publish new item hierarchy

The process starts when the data source has a new item hierarchy they want to add and publish. The actors in this scenario are the data source, Synkka data pool, Global Registry and the data recipient.

1. The data source creates the message including the item data (CIN) and sends it to Synkka data pool. The command to be used is ADD.
2. Synkka data pool receives the message and validates it.
3. A. If the validation was successful Synkka data pool sends a GS1 Response message with the status ACCEPTED to the data source. Now the data source will now that the item passed the validations and was stored to the data pool. After this the item data is send for validation and storing to Global Registry.

B. If the validation was not successful, Synkka data pool will send a GS1 Response message with the status EXCEPTION to the data source. The data source will need to fix the errors and send the item data again.

After a successful validation, the item will be automatically published in the data pool and all data recipients can see and subscribe the item data (if the item is not set as private). If a data recipient already had a subscription matching the item data prior its publication, the item data will be send to the recipient without any further actions needed.

Synkka data pool will send a CIC message with the confirmation state RECEIVED to the data source when the item data is forwarded to a recipient. Each delivery (data recipient) will trigger a unique CIC RECEIVED message.

### Change Item Hierarchy

The process starts when the data source has a need to update the data of an existing item which is already stored in Synkka data pool. The actors in this scenario are the data source, Synkka data pool, Global Registry and the data recipient. The changes must comply with the validation rules.

1. The data source creates the message including the updated item data (CIN) and sends it to Synkka data pool. The command to be used is CHANGE\_BY\_REFRESH.
2. Synkka data pool receives the message and validates it.
3. A. If the validation was successful Synkka data pool sends a GS1 Response message with the status ACCEPTED to the data source. Now the data source will now that the item passed the validations and was stored to the data pool. After this the item data is send for validation and storing to Global Registry.

B. If the validation was not successful, Synkka data pool will send a GS1 Response message with the status EXCEPTION to the data source. The data source will need to fix the errors and send the item data again.

1. After a successful validation, the item data will be automatically published and an updated message is send to the subscribed data recipients. Note that the updated item message will override the existing data entirely.
2. Synkka data pool will send a CIC message with the confirmation state RECEIVED to the data source when the item data is forwarded to a recipient. Each delivery (data recipient) will trigger a unique CIC RECEIVED message.

### Correct Item Hierarchy

The process starts when the data source has a need to correct the data of an existing item which is already stored in Synkka data pool. The actors in this scenario are the data source, Synkka data pool, Global Registry and the data recipient.

Correction is the update of data in ways that would not be allowed by the GTIN Management Standard (i.e. changes that would otherwise require the allocation of a new GTIN). All other validations (i.e. syntax, consistency, legal compliancy) still apply. Correction will trigger a different process at the data recipient’s end.

This process is intended to correct errors, not to circumvent the validation process as part of a standard data update.

1. The data source creates the message including the updated item data (CIN) and sends it to Synkka data pool. The command to be used is CORRECT.
2. Synkka data pool receives the message and validates it.
3. A. If the validation was successful Synkka data pool sends a GS1 Response message with the status ACCEPTED to the data source. Now the data source will now that the item passed the validations and was stored to the data pool. After this the item data is send for validation and storing to Global Registry.

B. If the validation was not successful, Synkka data pool will send a GS1 Response message with the status EXCEPTION to the data source. The data source will need to fix the errors and send the item data again.

1. After a successful validation, the item data will be automatically published and an updated message is send to the subscribed data recipients. Note that the new item message will override the existing data entirely.
2. Synkka data pool will send a CIC message with the confirmation state RECEIVED to the data source when the item data is forwarded to a recipient. Each delivery (data recipient) will trigger a unique CIC RECEIVED message.

### Cancel Item Hierarchy

Cancel communicates that the item/hierarchy was never manufactured. This can be achieved by maintaining the attribute Cancelled Date Time. This allows the reuse of the GTIN 12 months after cancellation instead of 48 months.

Note: When an item is cancelled in the GDSN, the waiting period for an item may have to be aligned with the specific industry requirement.

1. The data source creates the message including the Cancelled Date Time- information (CIN) and sends it to Synkka data pool. The command to be used is CORRECT.
2. Synkka data pool receives the message and validates it.
3. A. If the validation was successful, Synkka data pool sends a GS1 Response message with the status ACCEPTED to the data source. Now the data source will now that the item passed the validations and was stored to the data pool. After this the item data is send for validation and storing to Global Registry.

B. If the validation was not successful Synkka data pool will send a GS1 Response message with the status EXCEPTION to the data source. The data source will need to fix the errors and send the item data again.

1. After a successful validation, the new item data will be automatically published and the updated item data is send to the subscribed data recipients. Note that the new item message will override the existing data completely.
2. Synkka data pool will send a CIC message with the confirmation state RECEIVED to the data source when the item data is forwarded to a recipient. Each delivery (data recipient) will trigger a unique CIC RECEIVED message.
3. The item/hierarchy will be cancelled when the set Cancel date time is passed. After this the system will automatically send a CIHW message for the data recipient.

### Discontinue Item Hierarchy

The process starts when the data source wants to discontinue an item hierarchy or a single trade item. This can be achieved by maintaining the attribute Discontinued Date Time. This use case describes the process to flag a Catalogue Item for deletion, authorising the deletion of the Catalogue Item Data. When an item is discontinued in the GDSN, the waiting period for the GTIN before it can be reused for an item must be aligned with the specific industry requirement (as defined by GTIN Management Standard). After the discontinuation period lapses, all parties are free to delete the Item from their databases.

1. The data source creates the message including the Discontinued Date Time -information (CIN) and sends it to Synkka data pool. The command to be used is CORRECT.
2. Synkka data pool receives the message and validates it.
3. A. If the validation was successful Synkka data pool sends a GS1 Response message with the status ACCEPTED to the data source. Now the data source will now that the item passed the validations and was stored to the data pool. After this the item data is send for validation and storing to Global Registry.

B. If the validation was not successful Synkka data pool will send a GS1 Response message with the status EXCEPTION to the data source. The data source will need to fix the errors and send the item data again.

1. After a successful validation, the new item data will be automatically published and the updated item data is send to the subscribed data recipients. Note that the new item message will override the existing data completely.
2. Synkka data pool will send a CIC message with the confirmation state RECEIVED to the data source when the item data is forwarded to a recipient. Each delivery (data recipient) will trigger a unique CIC RECEIVED message.
3. The item/hierarchy will be discontinued when the set Discontinued date time is passed. After this the system will automatically send a CIHW message for the data recipient.

### Withdraw Item Hierarchy

When a supplier wants to stop the synchronization of a private item hierarchy with a data recipient, it can be done by sending a Catalogue Item Hierarchical Withdrawal message to Synkka data pool. The GTIN to be used in the message is the top level GTIN of the hierarchy.

After the withdraw, the data recipient specific item version will be removed and in case the item had a public version as well it will be kept. The integrating data sources can’t have draft versions of items in the system.

## Data recipient use cases

### Create a subscription

To receive trade item data from Synkka data pool the data recipient must have a subscription for the data. The subscription can be created with a message (CIS) or via the user interface of Synkka data pool. The criteria that can be used for the subscriptions are:

* Data source
* GPC (Global Product Classification)
* GTIN
* Target market

The data recipient can use each one of these parameters on their own or combinations of them in the same subscription with the exception that the GTIN and GPC should not be used together as the only parameters in the same subscription and the usage of Target Market alone is not allowed.

The steps are the following:

1. The data recipient creates a CIS Message in their own system and sends it to Synkka data pool
2. Synkka data pool receives the message and validates it.
3. A. Validation not ok – The data receiver receives a GS1 Response message with the status “EXCEPTION”. The message includes details of the validation rules which were triggered in the validation. The data receiver needs to correct the errors and send the CIS message again to the data pool.

B. Validation ok - If no errors were triggered the data receiver will receive a GS1 Response message with the status “ACCEPTED”.

1. Synkka data pool stores the CIS Message. The subscription is also send and stored in Global Registry.
2. Synkka data pool looks up for items matching the subscription and sends them to the data recipient.
3. The data recipient can send a CIC Message to the data source regarding the received item data.

Depending of the current message traffic and Global Registry processes, the CIN should be sent to the data recipient within 30 minutes.

### Remove subscription

The data recipient can remove subscriptions from the data pool. A situation could be e.g. when the data recipient decides that some item will no longer be included in their catalogue. After the removal, the data recipient will no longer receive updates of the item hierarchy from Synkka data pool.

The steps are the following:

1. The data recipient creates a CIS message with matching criteria to the subscription they want to remove. The command to be used is DELETE.
2. Synkka data pool receives the message and validates it.
3. A. Validation not ok – The data receiver receives a GS1 Response message with the status “EXCEPTION”. The message includes details of the validation rules which were triggered in the validation. The data receiver needs to correct the errors and send the CIS again to the data pool.

B. Validation ok - If no errors were triggered the data receiver will receive a GS1 Response message with the status “ACCEPTED”.

1. Synkka data pool removes the subscription from the system.

### Create a RFCIN

When the data recipient needs to get some trade item’s CIN message again and they already have a subscription e.g. for the GTIN or data source’s GLN, they need to send a RFCIN message to Synkka data pool. In other words, if an item is already in the data recipient’s synchronisation list, it cannot be subscribed again with a new CIS message.

The steps are:

1. The data recipient creates a RFCIN in their own system and sends it to Synkka data pool
2. Synkka data pool receives the message and validates it.
3. A. Validation not ok – The data receiver receives a GS1 Response message with the status “EXCEPTION”. The message includes details of the validation rules which were triggered in the validation. The data receiver needs to correct the errors and send the RFCIN again to the data pool.

B. Validation ok - If no errors were triggered the data receiver will receive a GS1 Response message with the status “ACCEPTED”.

1. Synkka data pool looks up for items matching the subscription and trigger a one-time delivery of the requested item data.

# JSON API

## PUBLIC

Public item Api was designed to make the public products / hierarchies available via JSON API. Public products / hierarhies are the ones which were published publicly in the data pool. API usage requires authorization that can be given by the data pool support team.

The documentation can be found <https://pp-synkka.gs1.fi:5443/helpPages/publicitemapi/index>

## GDSN

GDSN Api was designed to support product data retrieval for data source and data recipient companies. The API is an alternate way of product data retrieval from the data pool. For data source companies, only the published products are available for retrieval. Draft,cancelled, discontinued products are not available. For data recipient companies item availability is based on the GDSN publication / subscription process. That means exactly the same item information can be retrived via the API as via an M2M channel / is available on the Web User interface

The documentation can be found <https://pp-synkka.gs1.fi:5443/helpPages/gdsnapi/index>

# Media publishing and subscriptions

In the following chapters are the basic use cases for subscribing and publishing media files to Synkka using API. This document focuses on requests and responses in JSON through few of the most common use cases partners will face with Synkka, including subscribing for an asset and sending assets into the system.

The documentation with further request and response examples can be found <https://pp-synkka.gs1.fi/assets/html/redoc.html>

## Overview

The GDSN Media Solution API is a RESTful service that uses JSON messages. The high level goals for this API are the following:

* Upload/Modify/Get MediaItems
* Add/Remove MediaItems to GDSN products
* Handle subscriptions to MediaItems

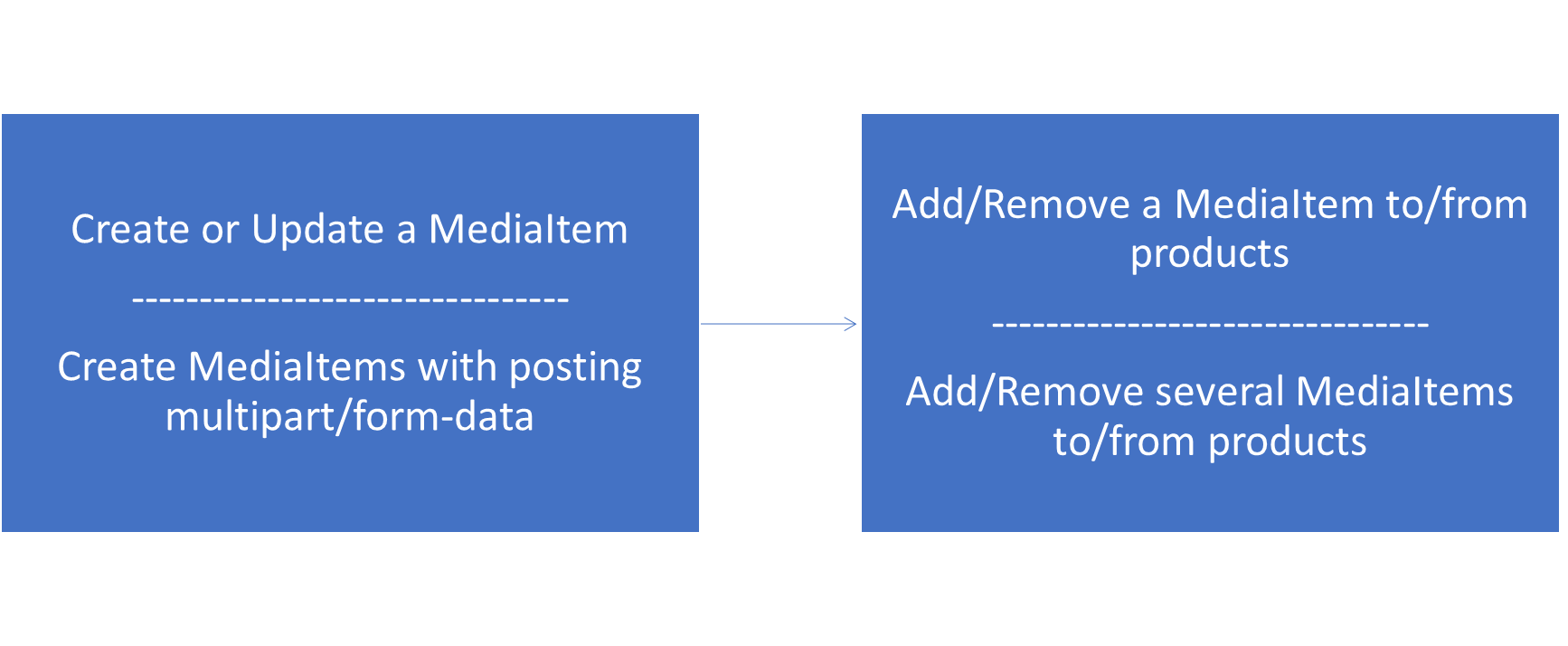
Download OpenAPI specification: <https://pp-synkka.gs1.fi/swagger/ui/index>

## Main entities

### MediaItem

A MediaItem is an entity that contains a file and some associated metadata that can be published into the GDSN network. The file can be an image a document or anything that can be published eventually into the network. Some of the properties of MediaItems are readonly and are not ment to be modified directly via simple updates. For exmaple properties that are calculated from the binary data of an image ( FilePixelWidth , FilePixelHeight , FileAspectRatio , FileResolutionDescription , FileSize etc.) are readonly.

When a MediaItem is first created it won't be added to any products ( IsAddedToProducts = false , IsArchived = false ). When it is added to one or more products it will have IsAddedToProducts = true , IsArchived = false . If it is removed from the last product it will become archived ( IsAddedToProducts = false , IsArchived = true ). If an archived item is added to a product again it will become IsAddedToProducts = true , IsArchived = false . Withdrawing/Cancelling/Discontinuing products that have MediaItems added to them will be automatically removed from the given items. That can also cause MediaItems to be acrhived. Archived items will be deleted after approximately 180 days.

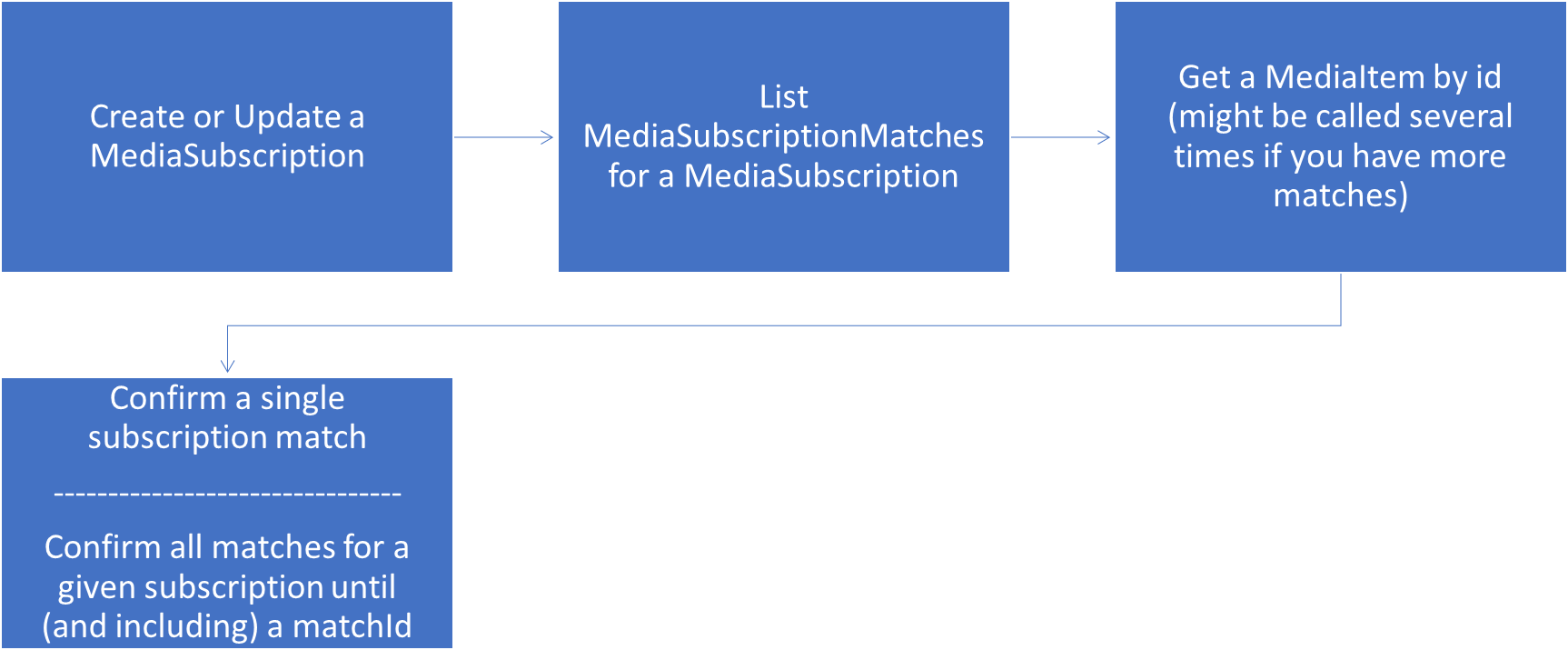


### MediaItemWithExtendedInfo

A MediaItemWithExtendedInfo is a wrapper object to provide additional information for MediaItems such as validity or available GDSN products. The IsValid property will indicate whether the item is valid for adding it to products. There are also partially valid items that can be added to one DataSource and TargetMarket but not to another (this is indicated as a warning).

### MediaSubscription

A MediaSubscription will represent a media item subscription with some subscription criteria. If a MediaSubscription matches (or no longer mathes) a MediaItem then a MediaSubscriptionMatch is created with the appropriate MatchType:   
Created - The media item is matched for the first time for this subscription.   
Updated - The media item is matched again after an item or a subscription change.   
Deleted - The media item is not longer matched for this subscription.



### MediaCollection

A MediaCollection is for grouping MediaItems together so they can be viewed or shared conveniently.

(SEE THE [ONLINE DOCUMENTATION](https://pp-synkka.gs1.fi/assets/html/redoc.html))

### MediaShare

With a MediaShare one can share either a MediaItem or a MediaCollection. The MediaShare can be public or only available to some GLNs. It can also have and effective date timeframe when it can be accessed. For every MediaShare there is a generated unique link, from where the binary content of the shared MediaItems can be downloaded (in a zip file if there are more than one item). A MediaShare can have some default conversions that enables to convert images before downloading them.

(SEE THE [ONLINE DOCUMENTATION](https://pp-synkka.gs1.fi/assets/html/redoc.html))

## Entity versioning

MediaItems and MediaSubscriptions are versioned entities, so all of them have a Version property. With this property one can retrieve historical versions of the entity and this property can be used for optimistic concurrency control. So if you update a versioned entity and the Version property is set the server will make sure that it will only update the entity if the Version still matches the current version on the server. Otherwise the update will fail. If you don't want this check just don't set the Version property (or set it to *null* ).

## Authentication

The API uses simple cookie authentication. A successful Account/LogIn method call will set the .ASPXAUTH cookie which should be sent with all further requests.

## Errors

The API uses standard HTTP status codes to indicate the success or failure of the API call.

# Media API

## Account

### Log in into the api

/api/Account/LogIn

Response will contain the .ASPXAUTH cookie. Which should be sent with all further requests.

Request Body

Login information to use.

|  |  |
| --- | --- |
| UserEmail | string **Required**  Email of the user. |
|  |
| Password | string **Required**  Password of the user. |
|  |
| Gln | string 13 characters  GLN of the selected company if the user have more than one. |
|  |

Responses

200 OK

400 The request is invalid.

401 Login failed.

404 Company was specified but not found.

500 Internal Server Error.

}

### Log Out from the api

/api/Account/LogOut

Log Out from the api, clears the the atuh cookie.

Responses

200 OK

400 The request is invalid.

401 Not logged in.

500 Internal Server Error.

### Switch between companies if the user has more than one

/api/Account/SwitchCompany

Response will contain the new .ASPXAUTH cookie. Which should be sent with all further requests.

Parameters

query Parameters ?

gln

string **Required**

The GLN of the selected company.

Responses

200 OK

400 The request is invalid.

401 Not logged in.

404 Company is not found.

500 Internal Server Error.

}

## MediaItem

### List MediaItems

/api/MediaItem

Get pagable list of MediaItems with extended info.

Parameters

query Parameters ?

pageIndex

integer <int32>

Page number to get.

pageSize

integer <int32>

Size of the page.

uploadSessionId

string <uuid>

Search filter for upload session id.

Responses

200 OK - Returns: List of MediaItems with extended info.

400 The request is invalid.

500 Internal Server Error.

}

### Create or Update a MediaItem

/api/MediaItem

This method is used for both updating and creating MediaItems. If Id is zero then a new item will be created, otherwise an existing item will be updated. With the Version property set you opt-in for optimistic concurrency control, so item will be updated only if there were no changes after the specified version.

The simplest message to create a MediaItem needs only two properties (OriginalFileName, and BinaryData).

=AAAAAAAK5ac%3D" }

Request Body

MediaItem to save.

|  |  |
| --- | --- |
| Id | integer <int32>  The Id of the MediaItem. |
|  |
| Version | string  The Version of the MediaItem. |
|  |
| CreateDate | string <date-time>  Creation date of the MediaItem. |
|  |
| UpdateDate | string <date-time>  Last modification date of the MediaItem. |
|  |
| IsAddedToProducts | boolean  Indicates whether the MediaItem is added to any products or not. There can be more than one Product for a media item. Only Valid items can be added to a Product. Only items that are added to atleast one product can be subscribed to. |
|  |
| IsQualified | boolean  Indicates whether the MediaItem is qualified (uploaded by a qualified photographer) or not. |
|  |
| IsPrimary | boolean  Indicates whether this is a primary file or not. |
|  |
| IsArchived | boolean  Indicates whether the MediaItem is archived or not. (Readonly - automatically set to true when the media item is removed from the last product and to false otherwise) |
|  |
| ArchivalDateTime | string <date-time>  Indicates when the MediaItem was last archived. (Readonly - automatically set to the current time when the media item is removed from the last product and to null otherwise) |
|  |
| Gtin | string 14 characters  The Gtin of the MediaItem. (Readonly if IsAddedToProducts is true) |
|  |
| OriginalFileName | string [ 0 .. 260 ] characters  The original file name. (Readonly after created) |
|  |
| **MarketingImageProperties** | MarketingProductImageProperties  Marketing image properties. |
|  |
| **PlanogramImageProperties** | PlanogramProductImageProperties  Planogram image properties. |
|  |
| **ReferencedFileHeader** | ReferencedFileHeader  Extended file information. |
|  |
| BinaryDataChecksum | string  MD5 checksum of the BinaryData. (Readonly) |
|  |
| BinaryData | string <byte>  File contents for a media item. (It is not always populated depends on the query parameters) |
|  |
| BinaryDataUpdateDate | string <date-time>  The date when the BinaryData was last modified. (Readonly) |
|  |
| UploadSessionId | string <uuid>  The client generated id of the session for the first upload. (Readonly after created) |
|  |
| **AddedToProducts** | ProductInfoForMedia  List of products that contains this MediaItem. (Readonly via simple update) |
|  |

Responses

200 OK - Returns: The saved MediaItem with extended info.

400 The item was not in the correct form.

403 Access denied.

404 Item not found.

409 The item was modified by someone else.

500 Internal Server Error.

}

### Get a MediaItem by id

/api/MediaItem/{id}

Gets a MediaItem by id (and version). The BinaryData property of the MediaItem will only have value if the includeBinary parameter is set to true. If the version parameter is not set then the latest version will be retrieved, otherwise the one with that particular version.

Parameters

path Parameters ?

id

integer <int32> **Required**

The id of the item.

query Parameters ?

version

string

The version of the item.

includeBinary

boolean

Get binary data alongside the metadata.

Responses

200 OK - Returns: The MediaItem with extended info.

400 The request is invalid.

403 Access denied.

404 Item is not found.

500 Internal Server Error.

### Delete a MediaItem by Id

/api/MediaItem/{id}

Deletes a MediaItem by id (and version). If the version parameter is not set then the latest version will be deleted, otherwise the one with that particular version.

Parameters

path Parameters ?

id

integer <int32> **Required**

The id of the item.

query Parameters ?

version

string

The version of the item.

Responses

200 OK

400 The request is invalid.

403 Access denied.

404 Item not found (might already be deleted).

409 The item was modified by someone else.

500 Internal Server Error.

}

### Download the binary data of a MediaItem

/api/MediaItem/{id}/Download

Downloads the binary data of a MediaItem. If no conversion is applied the ContentDisposition header will be set and containing an 'attachment' type with the 'filename' set. If conversion is applied there will be no ContentDisposition header set. If width and height are specified the system will resize the image so that it will fit into that frame while keeping its aspect ratio. If width, height and scale are specified the most restrictive one will prevail.

Example resize for a 200 x 100 image:

| **Width** | **Height** | **Scale** | **New size** | **Comment** |
| --- | --- | --- | --- | --- |
| 100 | 50 |  | 100 x 50 | Shrink to fit exactly |
|  |  | 50 | 100 x 50 | Shrink to half |
| 100 | 100 |  | 100 x 50 | Shrink to fit but keep aspect ratio |
| 400 | 200 |  | 400 x 200 | Enlarge to fit exactly |
|  |  | 200 | 400 x 200 | Enlarge to twice |
| 400 | 400 |  | 400 x 200 | Enlarge to fit but keep aspect ratio |
| 100 | 100 | 100 | 100 x 50 | Shrink only if larger |
| 400 | 400 | 100 | 200 x 100 | Shrink only if larger |

As seen by the last two examples if you only want to shrink images (but don't enlarge them) to fit into a certian frame, then set width and height as desired and Scale to 100.

Parameters

path Parameters ?

id

integer <int32> **Required**

The id of the item.

query Parameters ?

version

string

The version of the item.

width

integer <int32>

Maximum width of the converted image.

height

integer <int32>

Maximum height of the converted image.

scale

number <float>

Image scale in percentages (50 means half the size).

ppi

number <float>

Image resolution to use.

format

string

"Unknown" "Bmp" "Jpeg" "Png" "Gif" "Tiff"

Image format to use.

quality

string

"Low" "Medium" "High"

Quality of the conversion.

compress

boolean

Zip the output file.

Responses

200 OK - Returns: The binary data of a MediaItem.

400 Invalid parameters.

403 Access denied.

404 Item is not found.

500 Internal Server Error.

Collapse all

### Add/Remove a MediaItem to/from products

/api/MediaItem/{id}/UpdateProduct

Add/Remove a MediaItem to/from products. Only valid items can be added to products.

Parameters

path Parameters ?

id

integer <int32> **Required**

The id of the item.

Request Body

Datasource and TargetMarket paris of products to add or remove the specified MediaItem.

|  |  |
| --- | --- |
| **ProductsToAdd** | DataSourceAndTargetMarket  DataSource and TargetMarket pairs that should be added to a MediaItem. |
|  |
| **ProductsToRemove** | DataSourceAndTargetMarket  DataSource and TargetMarket pairs that should be removed from a MediaItem. |
|  |

Responses

200 OK - Returns: The modified MediaItem with extended info.

400 The request is invalid.

403 Access denied.

404 Item not found.

422 The item is invalid and cannot be added to or removed from products.

500 Internal Server Error.

### Add/Remove several MediaItems to/from products

/api/MediaItem/UpdateProductForMediaItems

Add/Remove several MediaItems to/from products with one round-trip. Only valid items can be added to products.

Request Body

The modification parameters.

|  |  |
| --- | --- |
| MediaItemIds | integer <int32>  Ids of the MediaItems that should be added or removed to/from products. |
|  |
| **ProductsToAdd** | DataSourceAndTargetMarket  DataSource and TargetMarket pairs that should be added to all MediaItems. |
|  |
| **ProductsToRemove** | DataSourceAndTargetMarket  DataSource and TargetMarket pairs that should be removed from all MediaItems. |
|  |

Responses

200 OK - Returns: The modified MediaItems with extended info.

400 The request is invalid.

403 Access denied.

422 One or more items are invalid and cannot be added to or removed from products.

500 Internal Server Error.

### Create MediaItems with posting multipart/form-data

/api/MediaItem/Upload/{uploadSessionId}

Creates MediaItems with posting multipart/form-data. Any reasonable number of files could be uploaded. If uploadSessionId is not specified then the server will generate one and use that for all the file in the request. If a zip file is uploaded its content will be extracted (file names starting with dots are ignored).

Parameters

path Parameters ?

uploadSessionId

string <uuid> **Required**

The uploadSessionId of the created item.

formData Parameters ?

file

file **Required**

file2

file

file3

file

Responses

200 OK - Returns: MediaItems created with extended info.

400 The request is invalid.

403 Access denied.

415 Media type is not supported.

1. ternal Server Error.

## MediaSubscription

### List MediaSubscriptions

/api/MediaSubscription

Get pagable list of MediaSubscriptions.

Parameters

query Parameters ?

pageIndex

integer <int32>

Page number to get.

pageSize

integer <int32>

Size of the page.

Responses

200 OK - Returns: List of MediaSubscriptions.

400 The request is invalid.

500 Internal Server Error.

}

### Create or Update a MediaSubscription

/api/MediaSubscription

This method is used for both updating and creating MediaSubscriptions. If Id is zero then a new item will be created, otherwise an existing item will be updated. With the Version property set you opt-in for optimistic concurrency control, so item will be updated only if there were no changes after the specified version.

Request Body

MediaSubscription to save.

|  |  |
| --- | --- |
| Id | integer <int32>  The Id of the MediaSubscription. |
|  |
| Version | string  The Version of the MediaSubscription. |
|  |
| CreateDate | string <date-time>  Creation date of the MediaSubscription. |
|  |
| UpdateDate | string <date-time>  Last modification date of the MediaSubscription. |
|  |
| Name | string [ 1 .. 200 ] characters **Required**  The Name of the MediaSubscription. |
|  |
| LastMatchClearDate | string <date-time>  The last time the matches were cleared for this subscription. |
|  |
| DataSource | string 13 characters  DataSource to match. |
|  |
| Gtin | string 14 characters  Gtin to match. |
|  |
| TargetMarket | string 3 characters  TargetMarket to match. |
|  |
| ReferencedFileTypeCode | string  "PRODUCT\_IMAGE" "PLANOGRAM" "LOGO" "DOCUMENT" "WARRANTY\_INFORMATION" "ASSEMBLY\_INSTRUCTIONS" "SAFETY\_DATA\_SHEET"  ReferencedFileTypeCode to match. |
|  |
| FileFormatName | string [ 0 .. 50 ] characters  FileFormatName to match. |
|  |
| MinFileEffectiveStartDateTime | string <date-time>  Minimum effective stat date to match. |
|  |
| MaxFileEffectiveEndDateTime | string <date-time>  Maximum effective end date to match. |
|  |
| IsPrimary | boolean  IsPrimary to match. |
|  |
| Nature | string  "Undetermined" "StillShotSingleGtin" "StillShotSingleGtinWithSupportingElements" "StillShotSingleGtinHighResolution" "StillShotSingleGtinWithSupportingElementsHighResolution"  Nature to match. |
|  |
| Direction | string  "None" "Front" "Left" "Top" "Back" "Right" "Bottom"  Direction to match. |
|  |
| Angle | string  "Center" "Left" "Right" "NoPlunge"  Angle to match. |
|  |
| Packaging | string  "InPackaging" "OutOfPackaging" "Case" "Innerpack" "RawUncooked" "Prepared" "Plated" "Styled" "Staged" "Held" "Worn" "Used" "Family" "OpenCase"  Packaging to match. |
|  |
| Language | string [ 0 .. 200 ] characters  Language to match. |
|  |
| ValidUntil | string <date-time>  ValidUntil to match. |
|  |
| Sequence | integer <int32>  Sequence to match. |
|  |
| Display | string  "Default" "Alternative" "Tray" "Display"  Display to match. |
|  |
| Gpc | string 8 characters  Gpc to match. |
|  |

Responses

200 OK - Returns: The saved MediaSubscription.

400 The item was not in the correct form.

403 Access denied.

404 Item not found.

409 The item was modified by someone else.

500 Internal Server Error.

}

### Get a MediaSubscription by id.

/api/MediaSubscription/{id}

Gets a MediaSubscription by id (and version). If the version parameter is not set then the latest version will be retrieved, otherwise the one with that particular version.

Parameters

path Parameters ?

id

integer <int32> **Required**

The id of the item.

query Parameters ?

version

string

The version of the item.

Responses

200 OK - Returns: The MediaSubscription.

400 The request is invalid.

403 Access denied.

404 Item is not found.

500 Internal Server Error.

### Delete a MediaSubscription by Id

/api/MediaSubscription/{id}

Deletes a MediaSubscription by id (and version). If the version parameter is not set then the latest version will be deleted, otherwise the one with that particular version.

Parameters

path Parameters ?

id

integer <int32> **Required**

The id of the item.

query Parameters ?

version

string

The version of the item.

Responses

200 OK

400 The request is invalid.

403 Access denied.

404 Item not found (might already be deleted).

409 The item was modified by someone else.

500 Internal Server Error.

### List MediaSubscriptionMatches for a MediaSubscription

/api/MediaSubscription/{id}/Match

Get pagable list of MediaSubscriptionMatches for a MediaSubscription. MediaSubscriptionMatches are created asynchronously triggered by changes to MediaItems and MediaSubscriptions. Not all versions will be visible via a MediaSubscriptionMatch, but eventually all the latest versions will be matched. (i.e. if you update an item 100 times in one second, and all 100 versions are matched by a subscription you might only see the last version of the item as a MediaSubscriptionMatch not 100 of them.)

Parameters

path Parameters ?

id

integer <int32> **Required**

The id the MediaSubscription.

query Parameters ?

pageIndex

integer <int32>

Page number to get.

pageSize

integer <int32>

Size of the page.

Responses

200 OK - Returns: List of MediaSubscriptionMatches.

400 The request is invalid.

404 Item not found.

500 Internal Server Error.

### Confirm a single subscription match

/api/MediaSubscription/{id}/Match/{matchId}/Confirm

Confirms a single match id for the given MediaSubscription Id. A confirmed match will no longer appear in the list of MediaSubscriptionMatches.

Parameters

path Parameters ?

id

integer <int32> **Required**

Id of the MediaSubscription.

matchId

integer <int32> **Required**

Id of the MediaSubscriptionMatch to confirm.

Responses

200 OK - Returns: Number of matches confirmed.

400 The request is invalid.

404 Item not found.

500 Internal Server Error.

### Confirm all matches for a given subscription until (and including) a matchId

/api/MediaSubscription/{id}/Match/{matchId}/ConfirmUntil

Confirms all existing matches that have an Id less than equal to the matchId specified for the given MediaSubscription Id. (This won't affect matches that have an Id less than equal to the matchId specified but with different MediaSubscription Id.) The confirmed matches will no longer appear in the list of MediaSubscriptionMatches.

Parameters

path Parameters ?

id

integer <int32> **Required**

Id of the MediaSubscription.

matchId

integer <int32> **Required**

Largest MediaSubscriptionMatch id to confirm.

Responses

200 OK - Returns: Number of matches confirmed.

400 The request is invalid.

404 Item not found.

500 Internal Server Error.

### Clear all matches and item state associated to a MediaSubscription

/api/MediaSubscription/{id}/Match/Reset

Clears all matches and item state associated to a MediaSubscription. After calling this method, the subscription will be checked against all media items (asynchronously) and all new matches will have match type 'Created' (as if this was a brand new subscription).

Parameters

path Parameters ?

id

integer <int32> **Required**

Id of the MediaSubscription.

Responses

200 OK - Returns: The modified MediaSubscription.

400 The request is invalid.

404 Item not found.

500 Internal Server Error.

# Appendix

## Abbreviations

|  |  |
| --- | --- |
| **Abbreviation** | **Term** |
| AS2 | Applicability Statement 2 |
| CIC | Catalogue Item Confirmation |
| CIHW | Catalogue Item Hierarchical Withdrawal |
| CIN | Catalogue Item Notification |
| CIS | Catalogue Item Subscription |
| FTPS | FTP Secure |
| GDSN | Global Data Synchronisation Network |
| GLN | Global Location Number |
| GPC | Global Product Classification |
| GTIN | Global Trade Item Number |
| MDN | Message Disposition Notification |
| RFCIN | Request for Catalogue Item Notification |
| XML | eXtensible Markup Language |

## Glossary of terms

|  |  |
| --- | --- |
| **Term** | **Definiton** |
| Add catalogue item | The command to create a new catalogue item record. |
| AS2 | A specification to transport data securely and reliably over the internet. Security is achieved by using digital certificates and encryption. The AS2 protocol is based on HTTP and S/MIME. It was GS1 Hungary GS1Trade Sync Data Pool – Functional Specification Version 0.95, Draft/Approved, 2016.06.24. All rights reserved @ GS1 Hungary Page 90 of 94 the second AS protocol developed and uses the same signing, encryption and MDN (as defined by RFC 3798) conventions used in the original AS1 protocol introduced in the late 90s by IETF. Like any other AS file transfer, AS2 file transfers typically require both sides of the exchange to trade SSL certificates and specific "trading partner" names before any transfers can take place. AS2 trading partner names can usually be any valid phrase. |
| Cancel Date | Date assigned by data source and stored in the source data pool reflecting the date the catalogue item was cancelled. This date will also be stored in the Registry. |
| Cancel Item | Global data synchronization term describing a maintenance function used to communicate that a catalogue item was never manufactured. This allows reuse of the GTIN 12 months after cancellation. |
| Catalogue Item | The item as it is stored in a catalogue or data pool. This is uniquely identified by (GTIN + GLN + Target Market). |
| Catalogue Item Confirmation | This refers to electronic communication from the Data Recipient to the Data Source indicating what action has been taken on the item. The confirmation process occurs in the recipient’s data pool. Confirmation is not mandatory. When used, it provides for the following outcomes: Synchronized: data is integrated, in synch and added to the synchronization list. Received: Data has been received by the Recipient, but no business decision has been made on the data. Rejected: recipient requests that no further updates are desired. Data will no longer be synchronized or updates will no longer be provided. Review: a request to the data source to “review” their data because the data recipient has received discrepant data which they cannot synchronize. |
| Catalogue Item Confirmation State | The four states reflected by a Recipient Data Pool are: Received, Rejected, Review and Synchronised |
| Catalogue Item Notification | A business message used to transmit trade item information from a data source or a data pool to a data recipient with the Global Data Synchronisation Network. |
| Catalogue Item Publication | A business message standard used to distribute trade item information within the Global Data Synchronisation Network. |
| Catalogue Item State | The four states are: Registered , Cancelled, In Progress and Discontinued. |
| Catalogue Item Subscription | A business message used to establish a request for the update of trade item information from an end recipient on a continuous basis. |
| Change Catalogue Item | The command to update an existing catalogue item record. |
| Correct Item | Refers to a command that allows incorrect data to be altered in ways that would not normally be allowed by standard GTIN allocation rules. All other validations still apply. This process is intended to correct errors, not to circumvent the validation process. |
| Data Pool | A repository of Data where trading partners can obtain, maintain and exchange information on items and parties in a standard format through electronic means. |
| Data Recipient | Party, which is authorized to view, use, download a set of Master Data provided by a Data Source. |
| Data Recipient Specific Item Version | An item version which is addressed to a specific data recipient (retailer). Each data recipient specific item version is an unique set of item data and is considered as an own item/hierarchy by the system. |
| Data Source | Entity that provides the global data synchronization network with Master Data. The Data Source is officially recognized as the owner of this data. For a given Item or Party, the source of data is responsible for permanent updates of the information under its responsibility. |
| Delete Catalogue Item | The command to flag the existing catalogue item record for deletion (The objective is to enable the eventual removal of the catalogue item record from the data pool.) |
| Global Data Synchronization Network (GDSN) | The Global Data Synchronization Network is a federation of interoperable certified Data Pools and a certified Global Registry that collectively provide for the synchronization of Master Data between trading partners on a global basis. |
| Global Location Number (GLN) | Unique location number mandatory within the Global Data Synchronization process to identify data owners/info providers, etc. such as Distributors, brokers, manufacturers. |
| Global Registry | A directory for the registration of unique catalogue items and parties. It contains a limited data set certified to be GS1 compliant and acts as a pointer to source data pools where master data is housed. |
| Global Trade Item Number GTIN | A particular Global trade item Number, a numerical value used to uniquely identify a trade item. A trade item is any trade item (trade item or service) upon which there is a need to retrieve pre-defined information and that may be planned, priced, ordered, delivered and or invoiced at any point in any supply chain. |
| GTIN Management Standard | The Global Trade Item Number (GTIN) provides a global supply chain solution by identifying any trade item that may be priced, or ordered, or invoiced at any point in the supply chain upon which there is a need to retrieve pre-defined information.  The Global Trade Item Number (GTIN) Management Standardis designed to help industry make consistent decisions about the unique identification of trade items in open supply chains. This standard has been developed in accordance with the GS1 Global Standards Management Process (GSMP) and is considered a part of the GS1 system of standards. Overall, costs are minimised when all partners in the supply chain adhere to the GTIN Management Standard. |
| Matching Process | A critical step within the data synchronization process that is owned and developed by each source data pool in order to trigger data distribution based on publication and subscription data. The matching process can be triggered either by publication, subscription or as a scheduled event. |
| New Item Hierarchy | A new construct of data containing a set of GTINs and Links that make up a unique relationship from the highest level GTIN with no parent to the lowest level with no children. To create a new Item Hierarchy, the Information Provider (or data owner) enters Item and Item Link data into the Source Data Pool. The data pool verifies that the information loaded is “correct” and then sends the relevant Item data to the registry for registration. Item link data is not sent to the Registry as links are not registered. |
| Publication | To prepare and issue data for distribution to one or a group of trading partners. A function within the Data Synchronization process whereby the Data Source grants visibility of item, party and partner profiles including party capabilities data to a given list of parties (identified by their GLNs) or to all parties in a given Market. It also will trigger the matching process that is the precursor to the distribution of data |
| Recipient Data Pool | A data pool that supports the functionality of the Data Recipient (Subscription, Confirmation, Search, Request for Notification, etc.) |
| Registration | Registration is the process, which references all items and parties prior to publication by all GS1 certified data pools and on which there is a need to synchronize information. Registering a Trade Item involves validation by the Registry for product uniqueness. The combination of attributes used to ensure unique records includes GTIN, GLN and Target Market. |
| Request for Catalogue Item Notification | A business message used to establish a subscription to trade item information for a data recipient within the Global Data Synchronisation Network. |
| Source Data Pool | A data pool that supports the functionality required by a Data Source such as Data Loading, Publication, Notification, Registration, etc. |
| Subscribe | A data synchronization function that refers to the creation of a subscription that lists the criteria for receiving publications. |
| Subscription | GTIN, GLN of Information Provider, Target market and Product Classification or any combination of these can maintain subscriptions. When a subscription is established, a Data Recipient sets a profile to receive on-going updates of the matching data. Subscriptions remain valid until they are deleted. Subscriptions are created by data recipients in their home data pool and sent to the registry. The Registry maintains a subscription list that is used to route relevant subscriptions to appropriate Source Data Pools. |
| Synchronisation | The process of continuous harmonization of information between all trading partners within the supply chain through the use of Align Data standards as published by GS1. |
| Synchronisation List | This is a subset of the Notification List maintained by the source data pool to keep track of where data has been notified - independent of the confirmations received. The list includes every Catalogue Item (GTIN+GLN+TM) that is synchronized. |
| Target Market | The Target Market is a geographical region based upon geographical boundaries sanctioned by the United Nations. There is one international system to describe geographical regions, the ISO-3166-code system. |
| Target Market Country Code | The country level or higher geographical definition in which the Information Provider will make the GTIN available to buyers. This does not in any way govern where the buyer may re-sell the GTIN to consumers. This code can be repeated as many times as needed. This code is represented by the 2-character ISO 3166-1 code. It is a mandatory attribute. Additionally, Target Market Subdivision Code indicates country subdivision where the trade item is intended to be sold. This code is represented by the 3-character ISO 3166-2 code. |
| Trading Partners | One or more parties engaged in trade. In the context of GS1 business models any  combination of Buyer, Seller, or Third Party. |
| Validation | The compliance checking of new or changed data against GS1 Global Data Standards, principles, rules and models. |