Trust.med

REST API Documentation

Back To Top

Introduction

The Trust.med API Allows you to gather and edit multiple different facets of data within your Trust.med account.

As a manufacturer, you will be able to gather information about your recalls and the notifications sent for those recalls. You will also be able to add a GTIN/NDC to your account, assign resolvers to those GTIN/NDCs and create/edit Resolver Links tied to those resolvers.

As a Distributor or Dispenser, you will be able to gather the notifications sent to you and acknowledge those recall notifications.

As a Third Party, you will be able to do the above for the appropriate type of account that you are responsible for.

Overview of the API

The API uses token based authentication.

Other forms of authentication, ATP for example, may be added later and this document will be updated accordingly.

This document will show all API calls with examples in both Python and CURL. The Python version is expected to use the latest version of the Python library.

Using The API

Authentication

The Trust.med API uses Token based Authentication. In order to use the API, you will need to be provided with a username and password by your Trust.med Representative.

Sandbox Testing

In order to test the Trust.med API within a Sandbox environment, there are a few extra steps:

1. You will need to have your Trust.med representative provision a *TESTING* API username and password for the system. This is a different set of credentials than those mentioned in the “Before You Begin” section.

2. You will need to provide the IP addresses for all of the test machines to your Trust.med representative.

3. **⚠️ All endpoints in this document will need to have the “https://atp.med” replaced with “https://staging.atp.med”⚠️**

Generating A Token

`POST /token`

The Trust.med API uses the Authorization token sent in the Header of a request to authenticate a user.

Accepted Methods: POST

The header will look like the following:

```
"Authorization: Bearer 123abc...987zyx"
```

In order to generate the access token, follow these steps:

1. Gather your username and password.
2. Send a request to the Access Token Generation endpoint.
3. You will send the username, password, scope, grant_type and client_id as Post Body elements.

Back To Top

Example Body

```
{
  "client_id": "dotmed",
  "grant_type": "password",
  "scope": "openid",
  "username": "your-username",
  "password": "your-password",
}
```

```
import requests

payload = {
    'client_id': 'dotmed',
    'grant_type': 'password',
    'scope': 'openid',
    'username': 'your-username',
    'password': 'your-password',
}

r = requests.post("https://atp.med/api/v1.0/token", data=payload)
```

cURL

```
curl --request POST \
--url 'http://api.med/api/v1.0/token' \
--header 'content-type: application/json' \
--data '{"client_id": "dotmed", "grant_type": "password", "scope": "openid", "username":
"your-username", "password": "your-password"}'
```

Response Data

The response to this endpoint will return with the following schema:

Item	Format	Description
access_token	String	The access token value
expires_in	Integer	How long the token lives, in seconds
token_type	String	Bearer

Back To Top

Recall Endpoints

List Manufacturer Recalls

`GET /recall`

This endpoint provides a list of active recalls.

Manufacturers engaging with this endpoint will receive a list of their active recalls. Dispensers and Wholesalers engaging with this endpoint will receive a list of active recalls initiated by manufacturers.

Example Request

Python

```
import requests

token = getAccessToken() # Defined by GenerateAccessToken endpoint
url = f'https://atp.med/api/v1.0/recall/'
headers = {'Authorization': f'Bearer {token}'}

r = requests.get(url, headers=headers)
```

cURL

```
curl --request GET \
--url 'http://atp.med/api/v1.0/recall/' \
--header 'Authorization: Bearer abc123...x_yz'
```

Response Data

The response to this endpoint will return with the following schema:

Item	Format	Description
id	Integer	The numerical ID for the Notification
recall_id	Integer	The numerical ID for the Recall
acknowledge	Boolean	If the notification has been acknowledged
acknowledge_details	String	The reason given for the acknowledgement

Back To Top

ack_date	DateTime	When the notification was first acknowledged
created_at	DateTime	When the notification was created
last_notified	DateTime	Date Time of last notification
company	Object	The Company info the notification was sent to
location	Object	The Location info the notification was sent to

List Recall Notifications

`GET /recall/{id([0-9]+)}/notification`

This endpoint uses an “id” that corresponds to the identifier for the recall that you wish to gather the notifications for.

This endpoint is used to generate a list of all of the notifications distributed for a particular recall.

When notifications are gathered via this endpoint by a third-party, the notification is auto-acknowledged for the user. The expectation is that the third-party is sending the information to the distributor or dispenser and is taking responsibility for the delivery of the notification downstream.

Example Request

Python

```python
import requests

token = getAccessToken() # Defined by GenerateAccessToken endpoint
recall_id = 123
url = f'https://atp.med/api/v1.0/recall/{recall_id}/notification/'
headers = {'Authorization': f'Bearer {token}'}

r = requests.get(url, headers=headers)
```

cURL

```
curl --request GET \
--url 'http://atp.med/api/v1.0/recall/:recall_id/notification/' \
--header 'Authorization: Bearer abc123...x_yz'
```

Back To Top

Response Data

The response to this endpoint will return with the following schema:

Item	Format	Description
id	Integer	The numerical ID for the Notification
recall_id	Integer	The numerical ID for the Recall
acknowledge	Boolean	If the notification has been acknowledged
acknowledge_details	String	The reason given for the acknowledgement
ack_date	DateTime	When the notification was first acknowledged
created_at	DateTime	When the notification was created
last_notified	DateTime	Date Time of last notification
company	Object	The Company info the notification was sent to
location	Object	The Location info the notification was sent to

List New Recall Notifications

`GET /recall-notifications`

Utilize this endpoint to gather all new recall notifications tied to an account. This can be used to gather the new recall notifications in bulk without the need to gather each recall identifier and hit the recall notification endpoint multiple times.

Once a notification is returned via this endpoint, it will not be part of the data returned if the endpoint is hit a second time since it only returns “fresh” data.

Back To Top

Example Request

Python

```
import requests

token = getAccessToken() # Defined by GenerateAccessToken endpoint
url = f'https://atp.med/api/v1.0/recall-notifications/'
headers = {'Authorization': f'Bearer {token}'}

r = requests.get(url, headers=headers)
```

cURL

```
curl --request GET \
--url 'http://atp.med/api/v1.0/recall-notifications/' \
--header 'Authorization: Bearer abc123...x_yz'
```

Response Data

The response to this endpoint will return with the following schema:

Item	Format	Description
id	Integer	The numerical ID for the Recall
identifier	String	The manufacturer provided name for the Recall
manufacturer	String	The Manufacturer’s name
manufacturer_event_number	String	The Manufacturer’s event number
manufacturer_phone	String	The contact number for the manufacturer
inititation_date	DateTime	The initiation date of the Recall
root_cause	String	URL for Root Cause document
reason	String	The reason for the Recall
recall_class	String	The class of the Recall
recall_level	String	The level of the Recall
stop_immediately	Boolean	Stop Immediately status
letter_wholesale	String	The URL for the Wholesale Letter if available

Back To Top

letter_pharmacy	String	The URL for the Pharmacy Letter if available
letter_patient	String	The URL for the Patient Letter if available
products	Array	A list of products, NDCs, Lots included in the Recall
company	Integer	The Company ID for the Recall
created_by	Integer	The User ID for the created Recall
creator_email	String	The email for the Recall Initiator

Resolver Endpoints

List/Create Resolver Objects

```
GET /company/{company_id}([0-9])/resolver
POST /company/{company_id}([0-9])/resolver
```

This endpoint can accept both a GET or a POST request. The GET request will return all resolvers tied to the specified company identifier. The POST request will create a new Resolver with the provided name for the company tied to the identifier.

Example Request

Python

```
import requests
token = getAccessToken() # Defined by GenerateAccessToken endpoint
url = f'https://atp.med/api/v1.0/company/1/resolver/'
headers = {'Authorization': f'Bearer {token}'}

# Example GET
r = requests.get(url, headers=headers)

# Example POST
payload= { 'name': 'My New Resolver' }
r2 = requests.post(url, headers=headers, data=payload)
```

Back To Top

cURL

```
curl --request GET \
--url 'http://atp.med/api/v1.0/company/1/resolver/' \
--header 'Authorization: Bearer abc123...x_yz'

curl --request POST \
--url 'http://atp.med/api/v1.0/company/1/resolver/' \
--header 'Authorization: Bearer abc123...x_yz' \
--data '{"name": "My New Resolver"}'
```

Response Data

The response to this endpoint will return either a list of resolvers or a single resolver with the following schema:

Item	Format	Description
id	([a-zA-Z0-9-)+	The UUID4 id for the Resolver
name	String	The name of the Resolver
company_id	Integer	The identifier for the Company
created_at	DateTime	When the Resolver was created
resolver_links	List: Resolver Link	List of the Resolver Link objects tied to this Resolver

List/Create Resolver Links

```
GET /company/{company_id}([0-9])/resolver/{resolver_id}([a-zA-z0-9-]+)/links
POST /company/{company_id}([0-9])/resolver/{resolver_id}([a-zA-z0-9-]+)/links
```

This endpoint will allow a company to gather the Resolver Links or create a brand new Resolver Link tied to a resolver. This link is the actual URL that will show within the Linkset for the specific GS1 Linktype (Appendix A).

Back To Top

Example Request

Python

```
import requests
token = getAccessToken() # Defined by GenerateAccessToken endpoint
url =
f'https://atp.med/api/v1.0/company/1/resolver/b0c8cc3f-deeb-4333-8874-2d7480334455/links/'
headers = {'Authorization': f'Bearer {token}'}

# Example GET
r = requests.get(url, headers=headers)

# Example POST
payload = {
  'title': 'Master Data',
  'href': 'https://my.masterdata.endpoint/',
'hreflang': 'en',
'link_type': 'gs1:masterData'
}
r2 = requests.post(url, headers=headers, data=payload)
```

cURL

```
curl --request GET \
--url
'http://atp.med/api/v1.0/company/1/resolver/b0c8cc3f-deeb-4333-8874-2d7480334455/links/' \
--header 'Authorization: Bearer abc123...x_yz'

curl --request POST \
--url
'http://atp.med/api/v1.0/company/1/resolver/b0c8cc3f-deeb-4333-8874-2d7480334455/links/' \
--header 'Authorization: Bearer abc123...x_yz' \
--data '{"title": "Master Data", "href":
"https://my.masterdata.endpoint/","hreflang":"en","link_type":"gs1:masterData" }'
```

Response Data

The following is the response data schema:

Note: The GET request returns a paginated result with a total result count as “count” and the following schema as a list under the “results” item.

Item	Format	Description
id	([a-zA-Z0-9-)+	The UUID4 id for the Resolver Link

Back To Top

title	String	The title of the Resolver Link
href	String	The URL for the data endpoint
hreflang	String	The language the data endpoint is in (default: en)
link_type	String	The GS1 Link Type for this Resolver Link
created_at	DateTime	When the Resolver Link was created

Update Resolver Link Data URL

`PATCH /company/{company_id}([0-9])/resolver/{resolver_id}([a-zA-z0-9-]+)/links/{link_id}([a-zA-z0-9-]+)`

Use this endpoint to update the Data URL for the Resolver Link.

Example Request

Python

```python
import requests
token = getAccessToken() # Defined by GenerateAccessToken endpoint
url =
f'https://atp.med/api/v1.0/company/1/resolver/b0c8cc3f-deeb-4333-8874-2d7480334455/links/b0c
8cc3f-deeb-4333-8874-2d7480445566/'
headers = {'Authorization': f'Bearer {token}'}

# Example PATCH
payload= { 'href': 'https://my.new.url/' }
r2 = requests.patch(url, headers=headers, data=payload)
```

cURL

```
curl --request PATCH \
--url
'http://atp.med/api/v1.0/company/1/resolver/b0c8cc3f-deeb-4333-8874-2d7480334455/links/b0c8c
c3f-deeb-4333-8874-2d7480445566/' \
--header 'Authorization: Bearer abc123...x_yz' \
--data '{"href": "https://my.new.url/" }'
```

Back To Top

Response Data

The following is the response data schema:

Item	Format	Description
id	([a-zA-Z0-9-)+	The UUID4 id for the Resolver Link
title	String	The title of the Resolver Link
href	String	The URL for the data endpoint
hreflang	String	The language the data endpoint is in (default: en)
link_type	String	The GS1 Link Type for this Resolver Link
created_at	DateTime	When the Resolver Link was created

VRS Endpoints

Add Product to VRS by GTIN

```
POST /vrs/company/{company_id}([0-9])/gtin/{gtin}([0-9])
```

Use this endpoint to add a product to the Trust.Med Lookup Directory, which will be synced to all working VRS partners.

⭐ **NOTE:** This endpoint is currently under development. More methods may become available as development continues.

Request Data

The following is the request data schema, expected as a JSON payload passed in the body:

Item	Format	Required	Description
recordOwner	String	Y	The Global Company Prefix (GCP) of the record owner
ci	String	Y	The connectivity information (e.g. URL, IP address) of the responder.
startExpDate	DateTime	Y	Lower bound of expiration date value.
endExpDate	DateTime	N	Upper bound of expiration date value.

Back To Top

status	String	Y	“active”, “inactive” or “deleted”
nextRecordOwner	String	N	The GCP of the next record owner if the product is being transferred. This should match recordOwner if not in transfer.

Example Request

Python

```
import requests
token = getAccessToken() # Defined by GenerateAccessToken endpoint
url = f'https://atp.med/api/v1.0/vrs/company/1/gtin/12345678901234'
headers = {'Authorization': f'Bearer {token}'}

# Example POST
payload= {
  'recordOwner': '123456',
  'ci': 'https://vrs-endpoint.med/vrs',
  'startExpDate': '200101',
  'endExpDate': '251231',
  status: 'active',
  'nextRecordOwner': '123456'
}
r2 = requests.post(url, headers=headers, data=payload)
```

cURL

```
curl --request POST \
--url 'http://atp.med/api/v1.0/vrs/company/1/gtin/12345678901234' \
--header 'Authorization: Bearer abc123...x_yz' \
--data '{"recordOwner": "123456", "ci": "https://vrs-endpoint.med/vrs", "startExpDate":
"200101", "endExpDate": "251231", "status":"active", "nextRecordOwner": "123456" }'
```

Response Data

The following is the response data schema:

Item	Format	Description
id	([a-zA-Z0-9-)+	The UUID4 id for the record
gtin	String	The GTIN value for the product
recordOwner	String	The Global Company Prefix (GCP) of the record owner
ci	String	The connectivity information (e.g. URL, IP address) of the responder

Back To Top

startExpDate	DateTime	The earliest point for product expiration
endExpDate	DateTime	The latest point for product expiration
nextRecordOwner	String	The GCP of the next record owner if the product is being transferred

Appendix

Appendix A: Accepted GS1 Link Types

Name	Code
Allergy Information	gs1:allergenInfo
Brand Homepage Pro	gs1:brandHomepageClinical
Brand Homepage Patient	gs1:brandHomepagePatient
Consumer Handling & Storage	gs1:consumerHandlingStorageInfo
EPCIS	gs1:epcis
Patient Information Leaflet	gs1:epil
Frequently Asked Questions	gs1:faqs
Org Info Page	gs1:homepage
Ingredients Information	gs1:ingredientsInfo
Instructions	gs1:instructions
Location Information	gs1:locationInfo
Master Data	gs1:masterData
Product Info Page	gs1:pip
Recall Status	gs1:recallStatus
Related Video	gs1:relatedVideo
Reviews	gs1:review
Safety Information	gs1:safetyInfo
Smart Label Page	gs1:smartLabel
Summary Product Characteristics	gs1:smpc

Back To Top

Support Page	gs1:support

Contact Details

If you require assistance please contact Trust.med at:

Email: support@trust.med
Phone: 855-630-0633