7 July 2025
EMA/772575/2022
Information Management Division

EU Implementation Guide (Vet EU IG) on veterinary medicines product data in the Union Product Database

Implementation of the requirements of Regulation (EU) 2019/6 for the Union database on veterinary medicinal products in the European Economic Area

Chapter 5: Technical specifications

Version 3

Overview of changes:

The key changes in this version are listed below:

- In section 3. 'API specifications' removed the whole paragraph referring to documentation and subscription-key enabled endpoints for National Competent Authorities as these have been deprecated at the end of June 2025.
- In annex 1.4.2.2 included information about a parameter (petchapter2) to be used in case a pet product is being validated.

Users are advised to read and consult the whole document.

Official address Domenico Scarlattilaan 6 ● 1083 HS Amsterdam ● The Netherlands
Address for visits and deliveries Refer to www.ema.europa.eu/how-to-find-us
Send us a question Go to www.ema.europa.eu/contact **Telephone** +31 (0)88 781 6000 An agency of the European Union

Acronym key and glossary terms

API	Application Programming Interface	**MS**	Member State
CA	Competent Authority	**NAP**	Nationally Authorised Products
CAP	Centrally Authorised Products	**OPAD**	Other Post Authorisation Data
CMDv	Coordination group for mutual recognition and decentralised procedures for veterinary medicinal products	**PET**	Veterinary medicinal products intended for animals which are exclusively kept as pets: aquarium or pond animals, ornamental fish, cage birds, homing pigeons, terrarium animals, small rodents, ferrets and rabbits
CMS	Concerned Member State	**PMS**	Product Management Service
CSV	Comma-separated values	**RMS**	Reference Member State
DCP	Decentralised Procedure	**SMS**	Substances Management Services
EAM	EMA Account Management	**SPOR**	Substances, Products, Organisations and Referentials
EC	European Commission	**SRP**	Subsequent Recognition Procedure
EEA	European Economic Area	**UAT**	User Acceptance Testing
EMA	European Medicines Agency	**UI**	User Interface
EP	End Point	**UPD**	Union Product Database
EU IG	European Union Implementation Guide	**NCA**	National Competent Authority
FHIR	Fast Healthcare Interoperability Resources	**NP**	National Procedure
HL7	Health Level Seven	**OMS**	Organisation Management Service
JSON	JavaScript Object Notation	**UUID**	Universally Unique Identifier
LOC ID	Location identifier	**VNeeS**	Veterinary Non eCTD Electronic Submission
MAH	Marketing Authorisation Holder	**VNRA**	Variations not requiring assessment
MRP	Mutual Recognition Procedure	**VoS**	Volume of Sales
MRPH	MRP products created after SPC harmonisation procedure	**XML**	eXtensible Markup Language

Table of contents

1. Introduction

The technical mechanism for product information management in the Union Product Database (UPD) is a RESTful Application Programming Interface (API), known as the SPOR API. The SPOR API offers services to work with elements in all four domains (Substances, Products, Organisations and Referentials). Support for Products and Substances has been introduced in version 2 of the SPOR API, whereas support for Organisations and Referentials is already a service publicly available since 2017.

Please also refer to section *References to FHIR versions*, in Vet EU IG Chapter 2.

2. Main API concepts

The SPOR API v2 has been built by using HL7 FHIR R5 Preview #2 (v4.4.0) as an overarching standard[1]. At the same time, it tries to follow the same principles as the existing SPOR API v1 and leverages both existing FHIR and SPOR API v1 resources and endpoints, where applicable. SPOR API v1 offers data from the Organisation Management Services (OMS) and the Referentials Management Service (RMS). The current UPD APIs under the v3 umbrella evolve the concepts of the SPOR API v2, modernising the authentication and authorisation mechanisms and adding extra endpoints delivering information from the UPD database that expands the SPOR API v2 datasets.

The SPOR API v2 is formally presented through an API specification, which is meant to enable developers to build an integration with SPOR services within their own local systems and fulfil their own business and regulatory requirements. The specification is the main reference point and in order to understand how to implement client systems, the following key elements should be considered:

- Resources: this refers to the formal representation of a set of data structures that can be exchanged between the client and the SPOR API in any specific communication. The resources are publicly available in the HL7 FHIR server. For simplicity, all relevant resources are also linked from the SPOR API v2 specification document.
- Services: this refers to the formal representation of the actions that can be executed against the existing Resources. The combination of Services and Resources gives developers the capability to understand the features that the SPOR API offers.
- Common mechanisms: since the API is based on FHIR and RESTful principles, the specification provides overall information regarding how these common mechanisms work. This includes, amongst others: usage of HTTP methods and codes, usage of FHIR bundles, FHIR extensions, etc. Importantly, it also describes the versioning mechanisms that guarantee backwards compatibility within a structured, governed process.

3. API specification

The API specification is mainly composed of:

- A portal with API endpoints description based on Open API specification (see below).
- A document that details all the concepts described in section 2 (main API concepts) *(SPOR API v2 specification*) and the list of resources available live in the HL7 FHIR server.
- A file with a list of sample payloads that can be used as examples of the UPD API utilization.

1 See http://hl7.org/fhir/directory.html for a list of all FHIR versions.

The portal with API endpoints description based on Open API framework is the basis of all the APIs available by UPD and is structured as follows:

- UPD National Competent Authorities APIs: This page presents the UPD APIs intended for use by NCA system users. The endpoints described here provide access to UPD Medicinal Products data, UPD Procedure data, UPD Variations Not Requiring Assessment (VNRA) data and UPD Volume of Sales (VoS) data.
- UPD MAH API: This page presents the UPD API intended for use by MAH system users. The endpoints described here provide read access to UPD Medicinal Products data.
- UPD Public API: This page presents the UPD API intended for use by General Public system users. The endpoints described here provide read access to UPD Medicinal Products data[2].

The recommended entry point to the different sources of documentation is the Open API representation of the API endpoints. The information included in Annex I of this document should be considered as well, since many known questions or issues are already described there.

If further details are required, please see the SPOR API v2 specification document. The relevant resources are part of the SPOR API v2 specification document, whilst better understanding is obtained if resources and services are reviewed in conjunction. The SPOR API v2 specification document also contains additional information that is out of scope for the UPD purposes. For this reason, Annex I for NCA APIs is included under this document, describing the main UPD related concepts and including links to the relevant information in the SPOR API v2 specification. This content has been enriched with clarifications and descriptions derived from feedback from NCA API users.

At last, to further assist the efficient use of the UPD APIs, a list of samples for NCAs related to the various API capabilities is available. Those can be found under the following link: API samples

Existing tools will be enriched, and new tools will be provided in future stages and as the regulatory and technical requirements evolve to further assist organisations that utilize the API.

4. Registration and access to UPD API

Information related to the registration process for the UPD API users can be found in the EMA website under the following link: UPD Registration guide for UI and API users

5. FHIR profiles

The relevant data elements and resources within the overall FHIR model will be enforced within each phase of the development of the UPD. The mechanism to narrow the overall IDMP model down and to describe technically all the business rules to be enforced in the FHIR message is referred as to the *FHIR profiles* *http://hl7.org/fhir/2020May/profiling.html*. A FHIR profile allows the authoring and publishing of a customised, more specific resource definition, by specifying a set of constraints and/or extensions on the base resource. Concrete FHIR resources like e.g. a *MedicinalProductDefinition* resource can express their conformance to a specific profile. This allows a FHIR server to programmatically validate a given resource against the associated profile definition.

More information about FHIR profiles is available within the SPOR API v2 specification document.

[2] Please, note that this version of the API will not give you access to the UPD documents (Product Information and Public assessment report). UPD documents are currently publicly available on the UPD Public Portal

In the present release of the Vet EU IG, no FHIR profile is provided. They are being built based on the business rules contained in this guide and they shall be provided at a later stage.

Annex I: Guidance for API users

Please review first the UPD Registration guide for UI and API users to understand the key concepts regarding access enablement, authentication and authorisation.

1. UPD API to Retrieve and Maintain Products and Product Documents (NCA, MAH, General Public)

1.1. Scope of this release for API

- Create DCP based on Chapter 4 Legacy or Chapter 2 rules
- Create MRP based on Chapter 4 Legacy or Chapter 2 rules
- Create SRP based on Chapter 4 Legacy or Chapter 2 rules
- Create MRPH based on Chapter 4 Legacy or Chapter 2 rules
- RMS can update Common Data for products under DCP/MRP/SRP/MRPH (data and documents)
- RMS and CMS can complement DCP/MRP/SRP/MRPH product with national data
- Create NP & Registered Homeopathic based on Chapter 4 Legacy or Chapter 2 rules
- Update NP & Registered Homeopathic product based on Chapter 4 Legacy or Chapter 2 rules
    - Edit existing, add new, or delete an existing non-mandatory attribute
    - Add new resources. For example: add an Ingredient or add another Package
    - Delete an existing non-mandatory resource. For example: remove an Ingredient
- Create & Update Parallel trade based on Chapter 4 Legacy or Chapter 2 rules
- Create & Update Pet products based on Chapter 4 Legacy or Chapter 2 rules
- Search and retrieve products
- Nullify product
- Upload, search, retrieve, and update Documents (for product under any procedure type)

1.2. General listing of UPD API supported Product Service endpoints per consumer type (NCAs, MAHs and General Public)

The current API version is V3 for NCAs and MAHs and v4 for General Public, only the HTTPS protocol is supported, and the authentication mechanism follows the OAuth2 Client Credentials standard.

API Property	Purpose
Base Path UAT	https://api-uat.pms.ema.europa.eu/
Bashe Path PROD	https://spor.azure-api.net/
Protocol	HTTPS

API Property	Purpose
Authentication	Oauth2 Client Credentials flow
FHIR Version	4.4.0 (R5 Preview #2)

The following list enumerates the available endpoints for Product services. EP302 Search Product Part and EP305 Get Product Part endpoints are no longer available.

SPOR API Specification v2	Allowed Consumers	API Endpoint
EP301 Search Product	NCA MAH PUB	GET MedicinalProductDefinition - Search for a MedicinalProductDefinition resource or resources
EP303 Get Product	NCA MAH PUB	GET MedicinalProductDefinition - Get a MedicinalProductDefinition ID
EP304 Get Product Full	NCA MAH PUB	GET Everything Current - Get $everything for a MedicinalProductDefinition ID
EP306 Get Product Version	NCA MAH	GET MedicinalProductDefinition Version - Get version of MedicinalProductDefinition ID
EP306a Get Product Version Full	NCA MAH	GET Everything Versioned - Get $everything for a version of MedicinalProductDefinition ID
EP307 Get Product Versions	NCA MAH	GET MedicinalProductDefinition - Get history of MedicinalProductDefinition ID

SPOR API Specification v2	Allowed Consumers	API Endpoint
EP309 Create Product	**NCA**	NAP: POST Bundle - Create/Update resources in the bundle DCP: POST dcp-bundle - Submit a Create DCP payload MRP: POST mrp_bundle – Submit a Create MRP payload SRP: POST srp_bundle – Submit a Create SRP payload MRPH: POST mrp-harmonisation-bundle – Submit a Create MRP Harmonised payload Registered homeopathic: POST Bundle - Create/Update resources in the bundle Parallel trade: POST ptp-bundle - Create/Update resources in the bundle Pet: POST pet-bundle - Create/Update resources in the bundle Refer to section 1.3 Endpoints for creating or updating Product and Procedure Data for more details.
EP309 Create Product EP311 Update Product for use with any Create or Update	**NCA**	**GET** OperationOutcome - Get a resource by ID Note: use this to query the outcome of Create or Update when response to Post is "202 Accepted"
EP311 Update Product	**NCA**	NAP: **POST** Bundle - Create/Update resources in the bundle Update National Data: **POST** /upd/api/v3/national-data-bundle/ - Submit an Update National Data payload for DCP/MRP/SRP products Update Common Data: **POST** /upd/api/v3/common-data-bundle/ - Submit an Update Common Data payload for DCP/MRP/SRP products
EP318 Validate Product	**NCA**	**POST** Validate Bundle – To validate a bundle and the resources in the bundle Used for all procedure types; for both chapter 2 or legacy validation rules; and for both Create & Update
EP UC19 Nullify Product	**NCA**	**POST** /upd/api/v3/vmp-nullification/

SPOR API Specification v2	Allowed Consumers	API Endpoint
EP401 Search document	NCA MAH	GET DocumentReference - Search for DocumentReference No
EP402 Get/Retrieve document by Id	NCA MAH	GET DocumentReference - Get a DocumentReference by Id Note
EP403 Create document	NCA	POST DocumentReference - Create a DocumentReference
EP404 Update document by Id	NCA	PUT DocumentReference - Update a DocumentReference

1.3. Endpoints for reading Product Data (NCA, MAH, General Public)

Endpoints EP301, EP303, EP304, EP306, EP306a, EP307, are used to search and list Products, and querying specific product details, with or without versions, depending on the permissions of the requestor.

Request Header: Key	Values	Purpose
Accept	application/fhir+xml application/fhir+json	Specifies the format for the response body of the POST if there are any validation or other errors

Payloads fulfil the FHIR specification of the resources in place. In order to become familiar with the responses and explore the content of the internal referenced resources and attributes, we recommend the execution of the following flow, where the retrieved content will be relevant to the products on the scope of the API consumer. The URLs take the UAT base path as example:

1. Execute a general search, with or without filters (EP301):
    a. For NCA, MAH, use:

        https://api-uat.pms.ema.europa.eu/pms/api/v3/MedicinalProductDefinition

    b. For General Public, use:

        https://api-uat.pms.ema.europa.eu/upd/public/v4/MedicinalProductDefinition

    Then, explore the payload and find a valid *MedicinalProductDefinition.* We will be using

2. Get the details of the product (EP303, EP304):
    a. For NCA, MAH, use
        i. *https://api-uat.pms.ema.europa.eu/pms/api/v3/MedicinalProductDefinition/600000093463*
        ii. *https://api-uat.pms.ema.europa.eu/pms/api/v3/MedicinalProductDefinition/600000093463/$everything*
    b. For General Public, use:
        i. *https://api-uat.pms.ema.europa.eu/upd/public/v4/MedicinalProductDefinition/600000093463*
        ii. *https://api-uat.pms.ema.europa.eu/upd/public/v4/MedicinalProductDefinition/600000093463/$everything*
3. Previous versions of the products can be retrieved by NCA, MAH (EP306, EP306a, EP307). The modifier *$everything* is used in the same way to obtain a detailed payload of a specific version
    a. Get the versions of the product:

    https://api-uat.pms.ema.europa.eu/pms/api/v3/MedicinalProductDefinition/600000093463/_history
    b. Select a particular version to explore:

    https://api-uat.pms.ema.europa.eu/pms/api/v3/MedicinalProductDefinition/600000093463/_history/1

    https://api-uat.pms.ema.europa.eu/pms/api/v3/MedicinalProductDefinition/600000093463/_history/1$everyting

Use the obtained payloads to navigate through products and versions. For more specific details, please read the SPOR API v2 specification and the official FHIR documentation applicable to the system version.

1.4. Endpoints for creating or updating Product and Procedure Data (NCA)

The entire section is relevant only for NCA

1.4.1. Apply Chapter 4 Legacy or Chapter 2 Validation rules

When submitting a POST for EP309 Create Product or EP311 Update Product, there is a Request header that is used to specify which validation rules are to be applied.

Please note that each type of update may use a different value for the Key.

Value	Validation rules applied
Request header not included	Vet EUIG Chapter 2
false	Vet EUIG Chapter 2
true	Vet EUIG Chapter 4 Legacy

1.4.2. API EP309 Create, EP311 Update & Nullify product endpoints

1.4.2.1. Request headers applicable for all Create, Update & Nullify POST

When submitting a POST for EP309 Create Product or EP311 Update or Nullify Product, the same Request headers are used for all endpoints that specify the format for the request and response.

Request Header: Key	Values	Purpose
Content-type	application/fhir+xml application/fhir+json	Specifies the format of the request body that is being submitted
Accept	application/fhir+xml application/fhir+json	Specifies the format for the response body of the POST if there are any validation or other errors

1.4.2.2. Create and Update endpoints

- As specified in SPOR API v2 Specification section 6.4.12
- Refer to Open API specification
- The Request body is a Bundle (type=transaction) of MedicinalProductDefinition and other resources
- For all the Update endpoints, the Bundle should be based on all data in the existing product. This includes Update Common Data DCP/MRP/SRP where all existing National data should also be included in the bundle even although it is only Common data that will be updated
- Create MRP is an update to an existing NP product. The Bundle should be based on all national data in that product, with the additional Common data added, and the procedure type updated to MRP
- Create SRP is an update to an existing DCP/MRP/SRP product. The Bundle should be based on all national data in that product, with the additional Common data added
- Please refer to the example bundles and recommended approach sections

Type and Procedure	POST Endpoint	Request header Key for validation rules	Additional Request header
Create NP	/pms/api/v3	chapter4	

Type and Procedure	POST Endpoint	Request header Key for validation rules	Additional Request header
Update NP	/pms/api/v3	chapter4	is update = true
Create DCP	/upd/api/v3/dcp-bundle/	chapter4	
Update Common Data DCP/MRP/SRP/MRPH	/upd/api/v3/common-data-bundle/	chapter4	is update = true
Update National Data DCP/MRP/SRP/MRPH	/upd/api/v3/national-data-bundle/	chapter4	is update = true
Create MRP	/upd/api/v3/mrp-bundle/	chapter4	
Create SRP	/upd/api/v3/srp-bundle/	chapter4	
Create MRPH	/upd/api/v3/mrp-harmonisation-bundle/	chapter4	
Create Registered Homeopathic	/pms/api/v3	homeopathicschapter2 = true OR homeopathicschapter4 = true	
Update Registered Homeopathic	/pms/api/v3	homeopathicschapter2 = true OR homeopathicschapter4 = true	is update = true
Create Parallel Trade	/upd/api/v3/ptp-bundle/	parallelchapter2 = true OR parallelchapter4 - true	
Update Parallel Trade	/upd/api/v3/ptp-bundle/	parallelchapter2 = true OR parallelchapter4 - true	is update = true

Type and Procedure	POST Endpoint	Request header Key for validation rules	Additional Request header
Create Pet	/upd/api/v3/pet-bundle/	chapter4	
Update Pet	/upd/api/v3/pet-bundle/	chapter4	is update = true
To Validate any Create or Update bundle	/pms/api/v3/$Validate	Use appropriate request header to apply validation rules based on the procedure type	Use is update = true when validating the following bundles: • Update NP • Update Registered Homeopathic • Update Parallel Trade • Update Pet • Update Common Data DCP/MRP/SRP • Update National Data DCP/MRP/SRP • Create MRP • Create SRP If validating a pet product, the following parameter is required to be set: • petchapter2 = true

1.4.2.3. Nullify endpoint

Type and Procedure	POST Endpoint	Request header Key for validation rules	Additional Request header
Nullify product	/upd/api/v3/vmp-nullification/	not required	

Content-Type	Request body

JSON	{
	"permanentId": "*Permanent Identifier*"
	}
	
	---- 	
	For example:
	{
	"permanentId": "600011984989"
	}

XML	<root><permanentId> *Permanent Identifier* </permanentId></root> ---- For example: <root><permanentId>600011353107</permanentId></root>

Response to POST:

- Response code 202 Accepted indicates the nullification has been successfully submitted
- Response code 400 Bad request indicates there is a validation error, and the Response body will contain error message. For example:
  "Resource type 'Bundle' with id '600011984989' couldn't be found."

1.4.2.4. Response to POST for Create, Update or Nullify and use of Get OperationOutcome

When POST for Create, Update or Nullify is successful and it cannot be honoured timely it is automatically queued. The Response header **Content-Location** contains an id that can be used to obtain the status of the operation.

Content-Location has two parts: **post-operation**/**operation-outcome-id**

The status of the operation can be consulted, it is one of:

- QUEUED
- IN_PROGRESS
- MSG_CREATED
- ERROR

Upon successful creation, update or nullification of the medicinal product, the operation outcome will show a status of MSG_CREATED along with the unique Permanent identifier(s) of the product(s).

The endpoint GET OperationOutcome/**operation-outcome-id** is used to query the status of the operation and this should be repeated until it is successful with MSG_CREATED or has ERROR.

The format of the Content-Location is showing in the following table, and the response value can be used for Get OperationOutcome.

POST	Content Location example showing format of the operation-outcome-id
Create NP	OperationOutcome/baab996e-8e58-4825-89d1-90a8f30458db
Update NP	OperationOutcome/c2e2275c-141c-4631-a42e-045726d95adb
Create DCP	OperationOutcome/ddb9f96b-10f5-4428-9503-170feb5c58db-DCP
Update Common Data DCP/MRP/SRP	OperationOutcome/f4d76850-358a-48f1-a9bb-3fb4b1615bdb-CD
Update National Data DCP/MRP/SRP	OperationOutcome/b371f2db-dd29-4c60-b6ab-63b0abf95bdb-ND
Create MRP	OperationOutcome/2f89089c-3ad7-4427-9311-7ea491395ddb-MRP
Create SRP	OperationOutcome/cf7af9a9-b34d-4db9-a551-89d40c077306-SRP
Create & Update Registered Homeopathic	OperationOutcome/a588416b-7a0b-40b1-8d03-a88ea4668f8f
Create & Update Parallel Trade	OperationOutcome/04b5bc00-16f4-4ea0-b33e-1a95029d8f8f-PTP
Create & Update Pet	OperationOutcome/2664fdf2-6aef-4540-8254-b6df6451b8af-PET

1.4.2.5. Creating products for DCP or Update Common Data if national data is provided

When the RMS submits a request bundle to create DCP products, they should only provide Common Data. Refer to Annex 1 of Vet EU IG Chapter 2.

If any National data attributes are populated in the create request bundle this does not result in a validation error. The products for the RMS and each CMS will be created, and any national data entered will be silently ignored.

The same applies for Update Common Data. The RMS should populate the complete Update bundle for their RMS product containing all existing Common and National Data. Only Common Data will be updated to the RMS product and the CMS products under the Product identifier.

1.4.3. API EP309 Create product example request bundles

Examples for EP309 Create Product for NP and DCP. Please note that the purpose of these examples is as illustration of the FHIR attributes to be populated.

The value for MedicinalProductDefinition as a cross-referenced product is a valid permanent identifier from UAT.

Procedure type	Validation rules	Example file
DCP	Chapter 2	UPD_1.6.5-6_DCP_Chpt2_C2_Mandatory_VetIG.JSON

Procedure type	Validation rules	Example file
		UPD_1.6.5-6_DCP_Chpt2_C2_Mandatory_VetIG.XML UPD_1.6.5-6_DCP_Chpt2_C110_VetEUIG_AllData.JSON UPD_1.6.5-6_DCP_Chpt2_C110_VetEUIG_AllData.XML
DCP	Chapter 4 Legacy	UPD_1.6.5-6_DCP_Legacy_C2_Mandatory_VetIG.JSON UPD_1.6.5-6_DCP_Legacy_C2_Mandatory_VetIG.XML UPD_1.6.5-6_DCP_Legacy_C110_VetEUIG_AllData.JSON UPD_1.6.5-6_DCP_Legacy_C110_VetEUIG_AllData.XML
NAP	Chapter 2	2.2 Authorisation/registration/entitlement number is specified at Product level UPD_1.6.1-4_NAP_Chpt2_C2_Mandatory_VetIG_MANumber_AtMedicinalProductLevel.JSON UPD_1.6.1-4_NAP_Chpt2_C2_Mandatory_VetIG_MANumber_AtMedicinalProductLevel.XML UPD_1.5.1-0_NAP_Chpt2_C110_VetEUIG_AllData_MANumber_AtMedicinalProductLevel.JSON UPD_1.5.1-0_NAP_Chpt2_C110_VetEUIG_AllData_MANumber_AtMedicinalProductLevel.XML --- 5.5 Marketing authorisation (package level) UPD_1.5.1-0_NAP_Chpt2_C111_VetEUIG_AllData_MANumber_AtPackageLevel.JSON This example contains 2 packages. There are 3 RegulatedAuthorization resources: • One with subject reference = MedicinalProductDefinition resource; populated with attributes from Section 2 (Vet EU IG Chapter 2), excluding the marketing authorisation number • One with subject reference = 1st PackagedProductDefinition resource; populated with the Marketing authorisation number for Package 1

Procedure type	Validation rules	Example file
		• One with subject reference = 2nd PackagedProductDefinition resource; populate with the Marketing authorisation number for Package 2
NAP	Chapter 4 Legacy	UPD_1.6.1-4_NAP_Legacy_C2_Mandatory_VetIG_MANumber_AtMedicinalProductLevel.JSON UPD_1.6.1-4_NAP_Legacy_C2_Mandatory_VetIG_MANumber_AtMedicinalProductLevel.XML UPD_1.5.1-0_NAP_Legacy_C110_VetEUIG_AllData_MANumber_AtMedicinalProductLevel.JSON UPD_1.5.1-0_NAP_Legacy_C110_VetEUIG_AllData_MANumber_AtMedicinalProductLevel.XML
NAP	Chapter 4 Legacy	UPD_1.5.1-0_NAP_Legacy_Cx_ManyAttributesAndResources_MANumberAtMedicinalProductLevel.XML This example contains: • 2 or more values for those attributes that are repeatable. For example, Product name, ATC Vet Code, Manufacturing Business Operation • 2 Packages (PackagedProductDefinition) • 2 Manufactured Items (ManufacturedItemDefinition) • 3 Ingredients (Ingredient)
NAP	Chapter 2	UPD_1.5.1-0_NAP_Chpt2_ExampleForStrengthAsPresentationOrConcentration.XML This example contains Ingredient resources that illustrate how to specify Substance and Reference Strength as either Presentation or Concentration.
NAP	Chapter 2	NAP_Chpt2_Create_BR-178_StrengthFreeTextExample_1.6.22-6.XML F178: This example contains Ingredient resources that illustrate how to specify free-text substance or reference substance strength
Registered Homeopathic	Chapter 2	UPD_1.6.1-4_HOM_Chpt2_C2_Mandatory_VetIG_MANumber_AtMedicinalProductLevel.JSON

Procedure type	Validation rules	Example file
		UPD_1.6.1-4_HOM_Chpt2_C110_VetEUIG_AllData_MANumber_AtMedicinalProductLevel.JSON
Parallel Trade	Chapter 2	UPD_1.6.8-4_PAT_Chpt2_C2_Mandatory_VetIGl.JSON UPD_1.6.8-4_PAT_Chpt2_C110_VetEUIG_AllData.JSON
Pet	Chapter 2	PET_Chpt2_C2_Mandatory_VetIG_MANumber_AtMedicinalProductLevel_1.6.34-5.json PET_Chpt2_C110_AllData_VetIG_MANumber_AtMedicinalProductLevel_1.6.34-5.json

1.4.3.1. Recommended approach to prepare update request bundle

The recommended approach for preparing a request bundle to update a product (any procedure type) is:

- Use the response from EP304 GET MedicinalProductDefinition/{permanent identifier}/$everything as a starting point
- Add Bundle.entry.request for each resource and update Bundle.type

Attribute	Change
Bundle.type	Must be "transaction"
For every Bundle.entry	Bundle.entry.request must also be populated. Bundle.entry.request.method should be: • PUT to update an existing resource • POST to add a new resource Bundle.entry.request.url should be: • Same value as Bundle.entry.fullUrl

For example:

```
<?xml version="1.0" encoding="utf-8"?>
<Bundle xmlns="http://hl7.org/fhir">
    <id value="600000022531" />
    <meta>
        <versionId value="1" />
        <lastUpdated value="2021-07-07T08:52:51.607+00:00" />
    </meta>
    <type value="transaction" />
    <entry>
        <fullUrl value="MedicinalProductDefinition/600000022531" />
        <resource>
            <MedicinalProductDefinition>
        </resource>
        <request>
            <method value="PUT" />
            <url value="MedicinalProductDefinition/600000022531" />
        </request>
    </entry>
    <entry>
        <fullUrl value="PackagedProductDefinition/170427" />
        <resource>
            <PackagedProductDefinition>
        </resource>
        <request>
            <method value="PUT" />
            <url value="PackagedProductDefinition/170427" />
        </request>
    </entry>
```

- DO NOT edit or remove the IDs for each resource and in-line within each resource in the EP304 Get $everything response

1.4.3.2. How to use Update NP product endpoint and example bundle

Create product via API	POST Bundle	Sample XML bundle used: UPD_1.5.1-0_NAP_Legacy_C110_VetEUIG_AllData_MANumber_AtMedicinalProductLevel.XML
Check operation outcome	MSG_CREATED message expected containing Permanent identifier	
EP304 Get Product Full	Prepare update bundle based on the response by updating Bundle.type to transaction and adding Bundle.entry.request.method for each resource. Edit the payload e.g. - modify product name - add another ATC Vet code - add another ManufacturedIte	Sample XML of Get Everything response used as a starting point: UPD_1.5.1-0_EP311_UpdateProduct_GetEverything_version1.XML Update bundle prepared: UPD_1.5.1-0_EP311_UpdateProduct_RequestBundle.XML

	mDefinition including this into the existing PackagedProduct Definition	
Update product via API	POST Bundle with request headers to /pms/api/v3 • "is_update=true" • "chapter4" = true or false for the validation rules to apply	
Check operation outcome	MSG_CREATED message expected containing Permanent identifier	
EP304 Get Product Full	Check the response for modifications	Sample XML of GET everything after update: UPD_1.5.1-0_EP311_UpdateProduct_GetEverything_version2.XML

1.4.3.3. How to use Update National Data DCP/MRP/SRP product endpoint and example bundle

EP304 Get Product Full	Prepare update bundle based on the response by updating Bundle.type to transaction and adding Bundle.entry.request.method for each resource. Edit the payload and add national data e.g. - Product name - Legal status of supply (product level) - Package description - Marketing authorisation number (product level) - Marketing authorisation status & dates - Responsible authority	Create DCP using this example file: UPD_1.6.16-5_CreateDCPForUpdateNationalData.XML Product Identifier: d0f4414c-cd65-478b-921e-f107c66f7a85 CMS for Italy **Permanent identifier:** 600000251886 Sample XML of Get Everything response used as a starting point: UPD_1.6.16_DCP_UpdateNationalData_600000251886_GetEverything_v1.XML Update bundle prepared: UPD_1.6.16_DCP_UpdateNationalData_600000251886_BasedOn_v1.XML
Update produc	POST Bundle with request headers to	

t via API	/upd/api/v3/national-data-bundle/ • "is_update=true" • "chapter4" = true or false for the validation rules to apply	
Check operation outcome	MSG_CREATED message expected containing Permanent identifier	
EP304 Get Product Full	Check the response for modifications	Sample XML of GET everything after update: UPD_1.6.16_DCP_UpdateNationalData_600000251886_GetEverything_v2.XML

1.4.3.4. How to use Update Common Data DCP/MRP/SRP product endpoint and example bundle

EP304 Get Product Full	Prepare update bundle based on the response by updating Bundle.type to transaction and adding Bundle.entry.request.method for each resource. Edit the payload e.g. - modify common product name - add another ATC Vet code Important: any national data that has been populated should be also included in the update bundle.	Sample XML of Get Everything response used as a starting point: UPD_1.5.3-4_DCP_UpdateCommonData_Product_600000149642_GetEverything_Version1.XML Update bundle prepared: UPD_1.5.3-4_DCP_UpdateCommonData_Product_600000149642_UpdateBundleBasedOnVersion1.XML
Update product via API	POST Bundle with request headers to /upd/api/v3/common-data-bundle/	

	• "is_update=true" • "chapter4" = true or false for the validation rules to apply	
Check operation outcome	MSG_CREATED message expected containing Permanent identifiers	
EP304 Get Product Full	Only the Common data in the RMS and CMS products under that Product Identifier will be updated	Please refer to Known issues section for any outstanding issues where national data submitted when updating common data is not being ignored.

1.4.3.5. How to use Create MRP product endpoint and example bundle

EP304 Get Product Full	Prepare update bundle based on the response by updating Bundle.type to transaction and adding Bundle.entry.request.method for each resource.	Sample XML of Get Everything response used as a starting point: UPD_1.5.3-4_CreateMRP_NP_600000184179_GetEverything_version1.XML
Prepare Create MRP Bundle	• Change procedure type from NP to MRP • Add Common Name with Country = EU and Language = English • Add Reference member state and Concerned member state • Add Common package description in English (if doesn't exist)	Create MRP bundle prepared: UPD_1.5.3-4_CreateMRP_BasedOn_NP_version1.XML
Create MRP via API	POST Bundle with request headers to /upd/api/v3/mrp-bundle/ • "chapter4" = true or false for the	

	validation rules to apply	
Check operation outcome	MSG_CREATED message expected containing Permanent identifiers for RMS NP product and products created for each CMS	
EP304 Get Product Full	RMS: • Contains the Common data that was added CMS: • Each new product is only populated with Common data, with status of Provisional	

1.4.3.6. How to use Create SRP product endpoint and example bundle

EP304 Get Product Full	Prepare update bundle based on the response by updating Bundle.type to transaction and adding Bundle.entry.request.method for each resource.	Sample XML of Get Everything response used as a starting point: UPD_1.6.1-4_CreateSRP_RMSProduct_GetEverything_version1.XML
Prepare Create SRP Bundle	• Add new Concerned member state(s) • Update common data as required	Create SRP bundle prepared: UPD_1.6.1-4_CreateSRP_BasedOnRMSProduct_version1.XML
Create SRP via API	POST Bundle with request headers to /upd/api/v3/srp-bundle/ • "chapter4" = true or false for the validation rules to apply	
Check operation outcome	MSG_CREATED message expected containing Permanent identifiers for existing RMS & CMS products and products created for each new CMS	
EP304 Get Product Full	RMS & existing CMS: • Contains the new CMS • Procedure type remains unchanged	

	• Contains the Common data that was updated New CMS: • Each new product is only populated with Common data, with status of Provisional, and procedure type of SRP	

1.5. API Manage document (NCA, MAH)

1.5.1. EP403 Create document

Available for NCA

Resource Information

Endpoint	POST /pms/api/v3/DocumentReference
Request	
Accept	application/fhir+xml application/fhir+json
Body	<DocumentReference .. </DocumentReference>
Content-type	application/fhir+xml application/fhir+json

Response	
Body	Document with version 1 and document ID returned Note: ID expected format example: 3c46270e-3c3d-4869-a73c-ad4d7c3f2893

Query Parameters

None

Example Request

For UAT environment:

POST https://api-uat.pms.ema.europa.eu/pms/api/v3/DocumentReference

Example file for request body: UPD_1.6.1-4_Doc_EP403_CreateDocument.XML

PDF document that was converted to base64: EP403_UploadDocument.PDF

- Document status value is case-sensitive (e.g.: current will work; CURRENT will fail)
- Document language value is case-sensitive (e.g.: en will work; EN will fail)

1.5.2. EP401 Search document

Available for NCA MAH

Resource Information

Endpoint	GET /pms/api/v3/DocumentReference?{ param}={value}[&{param}={value}]
Request	
Accept	application/fhir+xml application/fhir+json
Body	n/a
Content-Type	n/a
Response	
Body	Bundle of <DocumentReference>(s) e.g. Bundle Total value=N [entry {DocumentReference Resource Type}] *

Path Parameters

Name	Description
Version	Service version number **Example value:** 2

Query Parameters

Name	Description
related	Permanent identifier of the product the document is related to
type	Type of document
_summary	Boolean set to true or false. If set to true, the contents of the document is not populated in the response in DocumentReference.content.atttachement,data. There is a url provided but it is not intended that you can use this to retrieve the document.

Example request

GET /pms/api/v3/DocumentReference?related=MedicinalProductDefinition/600000216133
GET /pms/api/v3/DocumentReference?type=100000155538

GET
/pms/api/v3/DocumentReference?related=MedicinalProductDefinition/600000216133&_summary=true

1.5.3. EP402 Get/retrieve document

Available for NCA, MAH

Resource Information

Endpoint	GET /pms/api/v3/DocumentReference/{document-id}
Request	
Accept	application/fhir+xml application/fhir+json
Body	n/a
Content-Type	n/a
Response	
Body	Resource of type MedicinalProductDefinition

Path Parameters

Name	Description
Document id	A unique document identifier UUID **Example value:** 7a88176d-10f9-4db3-8fa0-4e4ae4594df7
version	Service version number **Example value:** 2

Query Parameters

None

Example Request

GET /v3/DocumentReference/3c46270e-3c3d-4869-a73c-ad4d7c3f2893

1.5.4. EP404 Update document

Available for NCA

Resource Information

Endpoint	POST /pms/api/v3/DocumentReference
Request	
Accept	application/fhir+xml application/fhir+json
Body	<DocumentReference> <id value="fcd2c31c-0ef9-455c-99a0-75149b888a27"/> .. </DocumentReference>
Content-type	application/fhir+xml application/fhir+json
is_update	true

Response	
Body	Document with version number incremented by 1

Query Parameters

None

Example Request

For UAT environment: POST https://api-uat.pms.ema.europa.eu/pms/api/v3/DocumentReference

Example file for request body:

- GET of document before update: UPD_1.6.1-4_Doc_EP402_GetDocument_version1.XML
- Update posted: UPD_1.6.1-4_Doc_EP404_UpdateDocument_BasedOnVersion1.XML
- Response to POST: UPD_1.6.1-4_Doc_EP404_ResponseAfterUpdate.XML
- GET of document after update: UPD_1.6.1-4_Doc_EP402_GetDocument_AfterEP404Update_version2.XML

1.5.5. Changes for Create and Update document payload

- There are no changes to payload

2. UPD API for VNRA (NCA)

API for VNRA is only available for NCA

2.1. Scope of this release for VNRA API

UPD-UC31 Manage VNRA Submissions via API

- Search and Retrieve VNRA
- Approve/Reject VNRA

2.2. UPD API supported VNRA endpoints

2.2.1. Query / Retrieve VNRA Submission

Query / Retrieve VNRA Submission	GET	***Returns the complete collection of submissions which the caller is entitled to view*** ***/vnra/v3?permanentIdentifier={permanentId}*** ***upd/api/vnra/v3?permanentIdentifier=600013438271***
API entry point	UAT	***https://api-uat.pms.ema.europa.eu/upd/api/vnra/v3?permanentIdentifier=600013438271***

API entry point	PROD	***https://spor.azure-api.net/upd/api/vnra/v3?permanentIdentifier=600013438271***
Query Parameters		**Query Parameters (All Are Optional)** ***Note: Calls to base url , (without parameters) /vnra/v3 will return the complete collection of submissions which the caller is entitled to view*** 1. **productName :** Product name – free text field and case insensitive 2. **productIdentifier :** Product identifier – free text field 3. **permanentIdentifier :** Permanent identifier – free text field 4. **mah :** OMS LOC_ID of Product owner – LOC-100005358 5. **responsibleAuthority :** OMS LOC_ID of Responsible authority (organisation) – LOC-100001603 6. **maNumber :** Authorisation/registration/entitlement number – free text field 7. **procedureType :** Procedure type – RMS Code 8. **procedureNumber :** Procedure number – free text field with "Starts with" and "Contains" and case insensitive 9. **submissionIdentifier :** Submission identifier – free text field 10. **submissionStatus :** Submission status – PENDING | APPROVED | PARTIALLY_APPROVED | REJECTED 11. **dateFrom :** Date From-To – calendar field to add interval "from" 12. **dateTo :** Date From-To – calendar field to add interval "to" 13. **vnraStatus :** VNRA Status – single selection field with list of VNRA status -PENDING | APPROVED | REJECTED 14. **vnraClassificationIdentifier :** vnraClassificationIdentifierClassification – field with list of VNRA classifications - RMS Code
Headers		**Security Headers (Mandatory)** v3 of the API require a mandatory Bearer Token which is passed via the Authorization header **Oauth Bearer Token** curl -X GET \ -H "Authorization: Bearer $(oauth-access-token)" \ https://spor.azure-api.net/upd/api/vnra/v3

Pagination		**Pagination**
		Pagination is implemented using Spring Boot Pagination which returns the following standard **Pagination Payload.**
		submission data are returned with in "content": [...],
		
		PageSize is set using the _size parameter.
		Iterating through the pages is managed via _page=x totalPages: y evaluation,
		If totalPages=y and the consumer searches for the last page, then _number should be set to y-1.
		https://spor.azure-api.net/*upd/api/vnra/v3?_size=5*
		https://spor.azure-api.net/*upd/api/vnra/v3?_size=5&_page=2*
		
		Pagination Payload
		{
		"content": [...],
		"pageable": {
		"sort": {
		"empty": **false**,
		"sorted": **true**,
		"unsorted": **false**
		},

```
    "offset": 0,
    "pageNumber": 0,
    "pageSize": 1,
    "paged": true,
    "unpaged": false
  },
  "totalPages": 485,
  "totalElements": 485,
  "last": false,
  "sort": {
    "empty": false,
    "sorted": true,
    "unsorted": false
  },
  "size": 1,
  "number": 0,
  "first": true,
  "numberOfElements": 1,
  "empty": false
}
```


Sample Payload


```
{
  "content": [
    {
      "submissionId": 12192,
      "submissionDate": 1731690475360,
      "submissionComment": "VNRA Submission Example",
      "submissionStatus": "PENDING",
      "potentialDecisionMaker": false,
      "decisionMakers": [
        "ORG-100003918"
      ],
      "vnessFileName": "File02.zip",
      "decisionMaker": false
    }
  ],
  "pageable": {
    "sort": {
      "empty": false,
      "sorted": true,
      "unsorted": false
    },
    "offset": 0,
    "pageNumber": 0,
    "pageSize": 1,
    "paged": true,
    "unpaged": false
  },
  "totalPages": 11719,
  "last": false,
  "totalElements": 11719,
  "sort": {
    "empty": false,
```

```
      "sorted": true,
      "unsorted": false
    },
    "size": 1,
    "number": 0,
    "first": true,
    "numberOfElements": 1,
    "empty": false
}
```


2.2.2. Retrieve a VNRA Submission

Retrieve a VNRA Submission	GET	Retrieve a specific VNRA submission identified by its submissionId ***/<submissionId>?summary={true|false}*** ***upd/api/vnra/v3/456?summary=true***
API entry point	UAT	***https://api-uat.pms.ema.europa.eu/upd/api/vnra/v3/456?summary=false***
API entry point	PROD	***https://spor.azure-api.net/upd/api/vnra/v3/456?summary=false***
Path Parameter		***/<submissionId>*** ***<SubmissionId> is the ID of the submission to retrieve***
Query Parameters		**Query Parameter (All Are Optional)** summary (Optional) : _(true|false) Returns a summary view of the submission else a full view
Headers		**Security Headers (Mandatory)** v3 of the API require a mandatory Bearer Token which is passed via the Authorization header **Oauth Bearer Token** curl -X GET \ -H "Authorization: Bearer $(oauth-access-token)" \ https://spor.azure-api.net/upd/api/vnra/v3/456?summary=false
Sample Payload **Summary=false**		{ "submissionId": 12193, "submissionDate": 1731690732660, "submissionComment": "Example VNRA Submission", "submissionStatus": "PENDING", "variations": [{ "variationId": 39103, "vnraGroup": "45c0d03d-a091-43bb-b6cc-fffb7ac4e5b8", "vnraCode": "200000018623", "implementationDate": 1731625200000, "decisionDate": null, "decisionAuthor": null, "decisionMaker": "ORG-100003918", "decisionComment": null, "fieldChanges": [{ "variationFieldChangeId": 17006,

```
                "key": "A1a-MarketingAuthorisationHolder-LocId",
                "currentValue": "600011430510",
                "currentValueAdditional": "\"600011430510\"",
                "proposedValue": "LOC-100002852",
                "proposedValueAdditional": "{\"locationId\":\"LOC-
100002852\",\"name\":\"Pfizer Manufacturing Deutschland GmbH - Heinrich-
Mack-Strasse 35, Illertissen, 89257, Germany\",\"targetField\":\"A1a-
MarketingAuthorisationHolder-LocId\",\"location\":{\"locationId\":\"LOC-
100002852\",\"organisationId\":\"ORG-
100004089\",\"organisationName\":\"Pfizer Manufacturing Deutschland
GmbH\",\"shortName\":\"Pfizer Manufacturing Deutschland
GmbH\",\"country\":\"DE\",\"locationIdsArray\":[\"LOC-
100002852\"],\"city\":\"Illertissen\",\"address\":\"Heinrich-Mack-Strasse
35\",\"postCode\":\"89257\",\"status\":\"ACTIVE\"}}",
                "changeProposedDate": 1731690742783,
                "changeImplementedDate": null
            }
        ],
        "productName": "Example Product Name",
        "productIdentifier": "aef948ee-a9a0-4a28-8b14-2f7896c8c614",
        "permanentIdentifier": "600011430510",
        "procedureNumber": "1",
        "status": "PENDING",
        "productStatus": "200000005004",
        "marketingAuthorisationHolder": "LOC-100002852",
        "responsibleAuthority": "LOC-100002852",
        "authorisationCountry": "100000000377",
        "marketingAuthorisationNumber": "1",
        "rmsCode": "",
        "cmsCodes": [],
        "procedureType": "100000155062",
        "availabilityStatus": "100000072075",
        "pendingVNRAsForSameProduct": [
            5675,
            12191
        ]
    }
  ], ,
  "foreseenDecisionMaker": "ORG-100003918",
  "vneesFileName": "File01.zip"
}
```


Sample Payload

Summary=true


```
{
  "submissionId": 12193,
  "submissionDate": 1731690732660,
  "submissionComment": "Example VNRA Submission",
  "submissionStatus": "PENDING",
  "variations": [],
  "vneesFileName": "File01.zip"
}
```

2.2.3. Download a VNeeS

Download a VNeeS	GET	Download a VNeeS linked to a VNRA Submission ***/<submissionId>/vnees*** ***upd/api/vnra/v3/456/vnees***
API entry point	UAT	***https://api-uat.pms.ema.europa.eu/upd/api/vnra/v3/456/vnees***
API entry point	PROD	***https://spor.azure-api.net/upd/api/vnra/v3/456/vnees***
Path Parameter		***/<submissionId>*** ***<SubmissionId> is the ID of the submission to retrieve***
Query Parameters		None
Headers		**Security Headers (Mandatory)** v3 of the API require a mandatory Bearer Token which is passed via the Authorization header **Oauth Bearer Token** curl -X GET \ -H "Authorization: Bearer $(oauth-access-token)" \ https://spor.azure-api.net/upd/api/vnra/v3/456/vnees

2.2.3.1. Submit a decision for the VNRA

Submit a decision for the VNRA	PUT	VNRA submit decision - Approve/Reject VNRA ***/<submissionId>/decision*** ***upd/api/vnra/v3/456/decision***
API entry point	UAT	***https://api-uat.pms.ema.europa.eu/upd/api/vnra/v3/456/decision***
API entry point	PROD	***https://spor.azure-api.net/upd/api/vnra/v3/456/decision***
Path Parameter		***/<submissionId>*** ***<SubmissionId> is the ID of the submission containing the variation to approve***
Values		**Decision** APPROVED | REJECTED **Decision Date** Date format should be provided as a string following SO 8601 Format. Examples: Date and time in UTC: *2024-11-21T11:34:35Z* Date and time with the offset: *2024-11-20T23:34:35+02:00* Date: *2024-11-21*
Query Parameters		**None**
Headers		**Security Headers (Mandatory)** v3 of the API require a mandatory Bearer Token which is passed via the Authorization header **Oauth Bearer Token** curl -X GET \ -H "Authorization: Bearer $(oauth-access-token)" \ https://spor.azure-api.net/upd/api/vnra/v3/456/decision
Sample Payload		{ "vnraDecisionItems": [{ "variationId": 40378,

```
            "vnraDecision": "APPROVED",
            "decisionComment": "This change is being approved",
            "decisionAuthor": "John",
            "decisionDate": "2024-11-21T00:00:00.000Z",
            "decisionMaker": "ORG-100003944"
        }
    ]
}
```


Sample Response


```
{
    "submissionId": 12630,
    "submissionDate": 1732116222663,
    "submissionComment": "Example VNRA Submission",
    "submissionStatus": "PENDING",
    "variations": [
        {
            "variationId": 40378,
            "vnraGroup": "5468f0a2-c218-470b-b523-45d4ecef2ff9",
            "productName": "Example NAP Product",
            "productIdentifier": "c0d2abcb-d0a0-4f0f-b32e-045849846a55",
            "permanentIdentifier": "700000047035",
            "procedureNumber": "EMEA/V/C/777777",
            "responsibleAuthority": "LOC-100000064",
            "authorisationCountry": "100000000535",
            "marketingAuthorisationNumber": "EMEA/V/C/777777",
            "vnraCode": "200000018632",
            "implementationDate": 1651446000000,
            "decisionDate": 1732186836,
            "decisionAuthor": "John",
            "decisionMaker": "ORG-100003944",
            "decisionComment": "This change is being approved",
            "status": "APPROVED",
            "marketingAuthorisationHolder": "LOC-100002852",
            "fieldChanges": []
        },
        {…}
    ],
    "potentialDecisionMaker": false,
    "decisionMakers": [
        "ORG-100003944"
    ],
    "decisionMaker": true
}
```

3. UPD API for Volume of Sales Data (NCA)

Available for NCA

3.1. Scope of this release for Volume of Sales API

- Retrieve Volume of Sales Data

3.2. Retrieve Volume of Sales Data Endpoint Description

Verb and Path	**GET**	***/upd/api/vos/v3/vos-sales-json***
Full Path	UAT	***https://api-uat.pms.ema.europa.eu/upd/api/vos/v3/vos-sales-json***
	PROD	***https://spor.azure-api.net/upd/api/vos/v3/vos-sales-json***
Headers	**Security Header** (mandatory)	This endpoint requires a mandatory OAuth Bearer Token which is passed via the Authorization header
Query Params	*(none)*	Calls to the base URL without any parameters will return the complete collection of sales data for all products.
	permanentId (optional)	Permanent identifier of Medicinal Product. Will return sales for the provided Permanent identifier e.g. permanentId=600000225806
	yearFrom (optional)	yearFrom={year-month} Start date for range of sales data to be returned e.g. yearFrom=2020-01&yearTo=2021-07
	yearTo (optional)	yearTo={year-month} End date for range of sales data to be returned e.g. yearFrom=2020-01&yearTo=2021-07
	modifiedDate (optional)	Modified Date of Sales data of Medicinal Product. Will return sales modified since a date. The following prefixes apply to date comparisons against a stored (modified date) value. If no prefixes are specified, the default is eq. • eq: equals, the exact stored value is inside the range defined by the precision of the parameter value • gt: the exact stored value is greater than the exact parameter value e.g. modifiedDate=2023-03-01 or with prefix

		e.g. modifiedDate=gt2023-03-01
	Full structure	/upd/api/vos/v3/vos-sales-json?permanentId={permanentID}&yearFrom={yearFrom}&yearTo={yearTo}&modifiedDate={modifiedDate}
	Example	GET *https://api-uat.pms.ema.europa.eu/upd/api/vos/v3/vos-sales-json?permanentId=600000225806&yearFrom=2020-01&yearTo=2021-07&modifiedDate=gt2023-01-01* GET *https://api-uat.pms.ema.europa.eu/upd/api/vos/v3/vos-sales-json?yearFrom=2020-01&yearTo=2021-07* GET *https://api-uat.pms.ema.europa.eu/upd/api/vos/v3/vos-sales-json?permanentId=600000225806*
Pagination		The pagination payload contains the following fields: • sales data is returned within "content": [...], • *pageSize* is set using the _size parameter • iterating through the pages is managed using the *_page* parameter • *totalPages* represents the total number of pages comprising the response. The pages are numbered from 0 to *totalPages - 1*. **Examples:** GET *https://api-uat.pms.ema.europa.eu/upd/api/vos/v3/vos-sales-json?_size=5* GET https://api-uat.pms.ema.europa.eu/upd/api/vos/v3/vos-sales-json?_size=5&_page=2

	Pagination Payload Example	"content": […],
		"pageable": {
		"sort": {
		"empty": **true**,
		"sorted": **false**,
		"unsorted": **true**
		},
		"offset": 0,
		"pageNumber": 0,
		"pageSize": 100,
		"paged": **true**,
		"unpaged": **false**
		},
		"totalElements": 3823,
		"totalPages": 39,
		"last": **false**,
		"sort": {
		"empty": **true**,
		"sorted": **false**,
		"unsorted": **true**
		},
		"size": 100,
		"number": 0,
		"first": **true**,
		"numberOfElements": 100,

```
    "empty": false
}
```


Response

Full

Sample Response Payload

Sample Response Payload


```
{
    "content": [
        {
            "productIdentifier": "aa4f3491-ea17-4b38-af07-41eaf4f1b55d",
            "productName": "Example NAP Product",
            "permanentIdentifier": "600010556679",
            "authorisationProcedureNumber": "EMEA/V/C/777777",
            "packageIdentifier": "68a997f8-2296-4210-9e78-8606b7e8dead",
            "packageDescription": "Example Package",
            "packSizeNumericValue": "600",
            "packSizeUnitOfPresentation": "Actuation",
            "packSizeUnitOfPresentationIdentifier": "200000002163",
            "country": "Sweden",
            "countryIdentifier": "100000000535",
            "marketingAuthorisationNumber": "EMEA/V/C/777777",
            "creationDateOfProduct": "2024-01-13",
            "creationDateOfPackage": "2024-01-13",
            "deletionDateOfPackage": null,
            "yearMonth": "2024-09",
            "volumeOfSales": "170",
            "speciesIdentifier": "200000000205",
            "speciesPercent": "100.00",
            "doseFactor": "1.00",
            "comment": "",
            "modifiedDate": "2024-10-11 08:36:30.583"
        },
        (…)
    ],
    "pageable": {
        "sort": {
            "empty": true,
            "unsorted": true,
            "sorted": false
        },
        "offset": 0,
        "pageNumber": 0,
        "pageSize": 100,
        "paged": true,
        "unpaged": false
    },
    "totalElements": 2,
    "totalPages": 1,
    "last": true,
    "sort": {
        "empty": true,
```

```
            "unsorted": true,
            "sorted": false
        },
        "size": 100,
        "number": 0,
        "first": true,
        "numberOfElements": 2,
        "empty": false
    }
```