EpiPulse Cases

Machine to Machine Communication
API Specification

Document Version

Version	Date	Comments
1.0	2024-07-06	Technical Documentation for Machine to Machine Interconnection

1. Introduction

1.1. Purpose

The Technical Design (TD) document provides an overview of the interfaces that are available for external systems to interact with EpiPulse Cases.

1.2. Definitions

This table describes terms and abbreviations used in this document. Commonly used IT terms and abbreviations can be found in ECDC's IT Glossary **Error! Bookmark not defined.**.

Term/Acronym	Definition
JWT	JSON Web Token **Error! Reference source not found.**
OIDC	Open ID Connect
FME	Feature Manipulation Engine
API	Application Programming Interface
DUAPI	Data Upload API
TV	Technical Validation
EV	Epidemiological Validation
WS	Web Service
M2M	Machine-to-Machine
WGS	Whole Genome Sequencing
AWS	Amazon Web Services
FMV	File Mapping Validation

1.3. Provide API for machine-to-machine interfacing.

The DUAPI is the externally exposed endpoint of the EpiPulse Cases architecture. It's utilized by the DUUI to allow users to access the system and it can also be used by external systems to interface as M2M. The interfaces provided can be split in 3 major groups and are described in detail in **Error! Reference source not found.**, section **Error! Reference source not found.**.

- Uploads, which provides functionality related to uploaded files (submitting, searching, retrieving)

- Technical Validation, which provides functionality related to the flow of technical validation (starting, retrieving status and output, etc)

- Epidemiological Validation, which provides similar functionality to TV for EV

Appendix A – APIs

This section describes the various interfaces of the system grouped per component and per area of functionality provided. Payload or response attributes in grey are optional and may be omitted.

A.1 Authentication

This is the first call we have to do in order to gain access to the EpiPulse Cases. This call ADFS to get a JWT token.

URL:	`https://zfs.ecdc.europa.eu/adfs/oauth2/token`
Authentication:	
Authorization:	
HTTP Method:	POST

Payload
as **urlencoded**:
	```{```
	```grant_type:password```
	```client_id:f142116f-a5d0-4089-bfaf-46bb0e5dd340```
	```client_secret:a0P5pZqkIJKORIqoLzwoYr_PJDZdpv86nA7k_48u```
	```username:ecdcdmz\EPC_NL_GENERAL-U```
	```password:P@ssw0rd```
	```resource: https://epipulse.ecdc.europa.eu/api/epipulsecases/dataupload```
	```scope:openid```
	```}```

Response:	```{```
	``` "access_token": "TOKEN",```
	``` "token_type": "bearer",```
	``` "expires_in": 3600,```
	``` "resource":```
	```"https://epipulse.ecdc.europa.eu/api/epipulsecases/dataupload",```
	``` "refresh_token": "",```
	``` "refresh_token_expires_in": 28800,```
	``` "scope": "user_impersonation openid",```

	``
	    "id_token": ""
	}
	``

HTTP codes:	• 200 Success. Response returned as documented above • 400 Bad Request. If the request failed to process (but not due to an exception) • 401 Unauthorized. If authorization checks fail

A.2 Data Upload

URL:	`https://epipulse.ecdc.europa.eu/api/v1/DataUploadAPI/Uploads`
Authentication:	JWT Barer Token
Authorization:	Environment Access + Upload
HTTP Method:	POST

Payload:	```
	{
	    "stream": "BASE64",
	    "fileName": "STRING",
	    "start": "DATETIME",
	    "end": "DATETIME",
	    "ignoreDateStartEnd": "BOOLEAN"
	    "uploadType": "STRING",
	    "chunkIdentifier": "GUID"
	}
	``

Response:	```
	{
	    "fileName": "STRING",
	    "uploadGuid": "GUID",
	    "children": [{
	        "fileName": "STRING",
	        "uploadGuid": "GUID"
	      },...]
	}
	``

HTTP codes:	• 200 Success. Response returned as documented above • 400 Bad Request. If the request failed to process (but not due to an exception) • 401 Unauthorized. If authorization checks fail • 409 Conflict. If there is an ongoing metadata synchronization • 500 Internal Server Error. If an exception is thrown while servicing the request
Caller	• DUUI
Comments:	Uploads a file and returns one or more (in case of a zip file containing multiple files) GUID(s). If the file is a zip, then the first entry returned will correspond to the zip file itself and the remaining to each of the children
URL:	https://epipulse.ecdc.europa.eu /api/v1/DataUploadAPI/Uploads/UploadSaveStatus
Authentication	JWT Barer Token

ECDC NORMAL

Authorization:	Environment Access + Upload
HTTP Method:	POST

Payload:	{
	  "uploadGuids": [
	    "GUID", …],
	}

Response:	{
	  "uploadGuids": [
	    "GUID", …],
	}

HTTP codes:	• 200 Success. Response returned as documented above • 400 Bad Request. If the content is 0 length or if the request failed to process (but not due to an exception) • 401 Unauthorized. If authorization checks fail • 500 Internal Server Error. If an exception is thrown while servicing the request
Caller	
Comments:	Checks if the save was completed in the database for the given Guids and returns the list of Guids to start the technical validation if the file(s) was successfully saved.

URL:	https://epipulse.ecdc.europa.eu/api/v1/DataUploadAPI/Uploads/{uploadGuid}
Authentication	JWT Barer Token
Authorization:	Environment Access + UploadApprove
HTTP Method:	PUT

Payload:	{
	    "reportingPeriodStart": "DATETIME",
	    "reportingPeriodEnd": "DATETIME"
	}

Response:	{
	    "isSuccessful": BOOLEAN,
	    "taskCorrelationGuid": "GUID"
	}

HTTP codes:	• 200 Success. Response returned as documented above • 400 Bad Request. If the uploadGuid is not provided • 401 Unauthorized. If authorization checks fail • 404 Not Found. If the provided uploadGuid is not found in the Upload database • 500 Internal Server Error. If an exception is thrown while servicing the request

ECDC NORMAL

Caller	
Comments:	Allows the UI to update the reporting period start and end date.

URL:	`https://epipulse.ecdc.europa.eu/api/v1/DataUploadAPI/Uploads/Search`
Authentication	JWT Barer Token
Authorization:	Environment Access
HTTP Method:	GET
| Payload: | ```json
{
    "uploadDate": "STRING",
    "subjectCode": "STRING",
    "diseaseCode": "STRING",
    "healthTopicCode": "STRING",
    "fileName": "STRING",
    "userName": "STRING",
    "hasErrors": "BOOLEAN",
    "hasWarnings": "BOOLEAN",
    "uploadGuids": [
        "GUID", …],
    "onlySubmittedByMe", "BOOLEAN"
    "orderBy": "STRING",
    "pageNumber": INTEGER,
    "pageSize": INTEGER
    "allowedStatuses": ["STRING", …],
    "notAllowedStatuses": ["STRING", …]
}
``` |
| Response: | ```json
{
    "uploads": [ {
        "uploadGuid": "GUID",
        "technicalValidationJobGroupGuid": "GUID",
        "epidemiologicalValidationGuid": "GUID",
        "dateUploaded": "DATETIME",
        "dateEpiStarted": "DATETIME",
        "dateTechnicalValidationStarted": "DATETIME",
        "subjectCode": "STRING",
        "fileName": "STRING",
        "userName": "STRING",
        "uploadState": "STRING",
        "recordCount": INTEGER,
        "deletedRowCount": INTEGER,
        "numberOfErrors": INTEGER,
        "numberOfWarnings": INTEGER,
        "technicalValidationProgressPercentage": INTEGER,
``` |

```
            "epidemiologicalValidationProgressPercentage": INTEGER,

            "parentUploadGuid": "GUID",
            "parentUploadFileName": "STRING",
            "hasComplesSubjectCode": "BOOLEAN",      "reportingPeriodStartDate":
"DATETIME",
            "reportingPeriodEndDate": "DATETIME",
            "minDateUsedForStatistics": "DATETIME",
            "maxDateUsedForStatistics": "DATETIME",
            "uploadType": "STRING",
            "uploadTypeName": "STRING",
            "hasOngoingEpiValidation": "BOOLEAN",
            "diseaceCode": "STRING",
            "fileSize": LONG,
            "fileTypeID": SHORT,
            "reportingCountry": "STRING",
            "ignoreDateStartEnd: "BOOLEAN",
            "parentComplexUploadGuid": "GUID",
            "uiSubjectCodeColumn": "STRING",
            "healthTopicCode": "STRING"
        }, … ],
      "totalUploadsCount": INTEGER
    }
```

HTTP codes:	• 200 Success. Response returned as documented above • 401 Unauthorized. If authorization checks fail • 500 Internal Server Error. If an exception is thrown while servicing the request
Caller	
Comments:	Performs a query on the uploaded files based on provided filters providing ordering and paging functionality

A.3 Technical validation

URL:	https://epipulse.ecdc.europa.eu/api/v1/DataUploadAPI/TechnicalValidations
Authentication	JWT Barer Token
Authorization:	Environment Access + UploadApprove
HTTP Method:	POST

Payload:	```
	{
	   "uploadGuids": [
	     "GUID", …],
	     "autoStartEpidemiologicalValidation": BOOLEAN
	}
	``

Response:	```
	{
	   "technicalValidationJobGroupCount": INTEGER
	}
	``

ECDC NORMAL

HTTP codes:	• 200 Success. Response returned as documented above • 401 Unauthorized. If authorization checks fail • 500 Internal Server Error. If an exception is thrown while servicing the request
Caller	
Comments:	Starts the Technical Validation for an upload.

URL:	https://epipulse.ecdc.europa.eu/api/v1/DataUploadAPI/TechnicalValidations/ {technicalValidationJobGroupGuid}
Authentication	JWT Barer Token
Authorization:	Environment Access
HTTP Method:	GET

Response:	{
	    "startDateTime": "DATETIME",
	    "currentStep": "STRING",
	    "progressPercentage": INTEGER
	}

HTTP codes:	• 200 Success. Response returned as documented above • 400 Bad Request. If the request failed to process (but not due to an exception) • 401 Unauthorized. If authorization checks fail • 404 Not Found. If technicalValidationJobGroupGuid is not found in the database • 500 Internal Server Error. If an exception is thrown while servicing the request
Caller	
Comments:	Queries the status/progress of a technical validation job

URL:	https://epipulse.ecdc.europa.eu/api/v1/DataUploadAPI/TechnicalValidations/ {technicalValidationJobGroupGuid}/Results
Authentication	JWT Barer Token
Authorization:	Environment Access
HTTP Method:	GET

Response:	{
	    "messages": [{
	        "location": {
	            "recordIdentifier": "STRING",
	            "variable": "STRING",
	            "rowNumber": "STRING",

<table>
<tr><td rowspan="2"></td><td>

```
        "columnIndex": "STRING"
      },
      "severity": "STRING",
      "subjectCode": "STRING",
      "validationRuleCode": "STRING",
      "validationMessage": "STRING",
      "solutionMessage": "STRING"
    }, … ],
    "identityFieldName": "STRING"
}
```

</td></tr>
</table>

HTTP codes:	200 Success. Response returned as documented above 400 Bad Request. If the request failed to process (but not due to an exception) 401 Unauthorized. If authorization checks fail 404 Not Found. If technicalValidationJobGroupGuid is not found in the database 500 Internal Server Error. If an exception is thrown while servicing the request
Caller	
Comments:	Retrieves the messages generated as a result of the technical validation process directly from the EpiPulseCasesTechnicalValidation database

URL:	https://epipulse.ecdc.europa.eu/api/v1/DataUploadAPI/TechnicalValidations/ {uploadGuid}
Authentication	JWT Barer Token
Authorization:	Environment Access + UploadApprove
HTTP Method:	DELETE
Response:	```{ "isSuccessful": BOOLEAN, "taskCorrelationGuid": "GUID" }```
HTTP codes:	200 Success. Response returned as documented above 400 Bad Request. If the provided uploadGuid is empty 401 Unauthorized. If authorization checks fail 404 Not Found. If uploadGuid is not found in the database 500 Internal Server Error. If an exception is thrown while servicing the request
Caller	
Comments:	Cancel a running technical validation

ECDC NORMAL

URL:	https://epipulse.ecdc.europa.eu/api/v1/DataUploadAPI/TechnicalValidations/ {technicalValidationJobGroupGuid}?Action=Reject
Authentication	JWT Barer Token
Authorization:	Environment Access + UploadApprove
HTTP Method:	PUT

Response:	{
	    "isSuccessful": BOOLEAN,
	    "taskCorrelationGuid": "GUID"
	 }

HTTP codes:	• 200 Success. Response returned as documented above • 400 Bad Request. If the provided technicalValidationJobGroupGuid is empty or the action is not "Reject" • 401 Unauthorized. If authorization checks fail • 404 Not Found. If technicalValidationJobGroupGuid is not found in the database • 500 Internal Server Error. If an exception is thrown while servicing the request
Caller	
Comments:	Reject a completed technical validation

URL:	https://epipulse.ecdc.europa.eu/api/v1/DataUploadAPI/TechnicalValidations/ {technicalValidationJobGroupGuid}/Results/csv
Authentication	JWT Barer Token
Authorization:	Environment Access
HTTP Method:	GET
Response:	CSV file containing the technical validation messages for a given technicalValidationJobGroupGuid. The MIME types used is "Text/csv" and the file name "TechnicalValidationMessages – {technicalValidationJobGroupGuid}.csv"
HTTP codes:	• 200 Success. Response returned as documented above • 400 Bad Request. If the request failed to process (but not due to an exception) • 401 Unauthorized. If authorization checks fail • 404 Not Found. If technicalValidationJobGroupGuid is not found in the database • 500 Internal Server Error. If an exception is thrown while servicing the request
Caller	• DUUI
Comments:	Retrieves the messages generated as a result of the technical validation process as a CSV file directly from the EpiPulseCasesTechnicalValidation database

A.4 Data (Epidemiological) Validation

URL:	`https://epipulse.ecdc.europa.eu/api/v1/DataUploadAPI/EpidemiologicalValidations`
Authentication	JWT Barer Token
Authorization:	Environment Access + Upload
HTTP Method:	POST

Payload:	`{`
	`  "uploadGuids": [`
	`    "GUID"`
	`    ]`
	`}`

Response:	`{`
	
	`    "isSuccessful": BOOLEAN,`
	
	`    "taskCorrelationGuid": "GUID"`
	
	`}`

HTTP codes:	• 200 Success. Response returned as documented above • 400 Bad Request. If the request failed to process (but not due to an exception) • 401 Unauthorized. If authorization checks fail • 409 Conflict. If there is another running epidemiological validation for the same country and disease • 500 Internal Server Error. If an exception is thrown while servicing the request
Caller	• DUUI
Comments:	Starts the Epidemiological Validation for an upload.

URL:	`https://epipulse.ecdc.europa.eu/api/v1/DataUploadAPI/EpidemiologicalValidations/` `OngoingStatus/{subjectGuid}`
Authentication	JWT Barer Token
Authorization:	Environment Access
HTTP Method:	GET
Response:	`BOOLEAN`
HTTP codes:	• 200 Success. Response returned as documented above • 400 Bad Request. If the request failed to process (but not due to an exception) or the provide subjectGuid was not a valid GUID • 401 Unauthorized. If authorization checks fail • 404 Not Found. If subjectGuid is not found in the database • 500 Internal Server Error. If an exception is thrown while servicing the request

ECDC NORMAL

Caller	
Comments:	Returns if a particular subject has an ongoing epidemiological validation (for the country of the caller)