D7.5 e-VITA API and technical guidelines for third party access

Deliverable D7.5		
Authors and institution	ENG: Martino Maggio, Francesca D'Agresti.	
Date	M24	
Dissemination level		
PU	Public, fully open, e.g. web	PU
CO	Confidential, restricted under conditions set out in Model Grant Agreement	
CI	Classified, information as referred to in Commission Decision 2001/844/EC	

The project has received funding from the European Union H2020 Programme under grant agreement n° 101016453. The Japanese consortium received funding by the Japanese Ministry of Internal Affairs and Communication (MIC).

Document change history			
Date	Version	Authors	Description
15/12/2023	v0.1	Martino Maggio (ENG)	First Version of Contents and Structure
11/01/2023	v0.2	Martino Maggio (ENG), Francesca D'Agresti (ENG)	Update of Contents in section 1, 2, 3
16/01/2023	v0.3	Martino Maggio (ENG), Francesca D'Agresti (ENG)	Update of Contents in section 1, 2, 3, 4, 5, 6
26/01/2023	v0.4	Martino Maggio (ENG), Francesca D'Agresti (ENG), Jerome Boudy (IMT), Rainer Wieching (USI)	Last draft for revision
31/01/2023	v1.0	Martino Maggio (ENG), Francesca D'Agresti (ENG), Jerome Boudy (IMT), Rainer Wieching (USI)	Final version

Ministry of Internal Affairs and Communications, JAPAN

The project has received funding from the European Union H2020 Programme under grant agreement n° 101016453. The Japanese consortium received funding by the Japanese Ministry of Internal Affairs and Communication (MIC).

Table of Contents

Ministry of Internal Affairs and Communications, JAPAN

The project has received funding from the European Union H2020 Programme under grant agreement n° 101016453. The Japanese consortium received funding by the Japanese Ministry of Internal Affairs and Communication (MIC).

List of Tables

The project has received funding from the European Union H2020 Programme under grant agreement n° 101016453. The Japanese consortium received funding by the Japanese Ministry of Internal Affairs and Communication (MIC).

Acronyms and Abbreviations

API	Application Programming Interface
CSV	Comma-Separated Values
HTTP	Hypertext Transfer Protocol
ID	Identifier
JSON	JavaScript Object Notation
JWT	JSON Web Token
REST	Representational State Transfer

The project has received funding from the European Union H2020 Programme under grant agreement n° 101016453. The Japanese consortium received funding by the Japanese Ministry of Internal Affairs and Communication (MIC).

Executive Summary

This document describes the e-VITA APIs the technical interfaces that allow to access to the main functionalities of the e-VITA platform. The document includes a detailed structure of the API and the way to access them. The deliverable covers five categories of API: User API that provides access to user information, Device API that allows the visualization and management of devices, Service API that provides access and management of the cloud services, Researchers API allows to get historical data and Clients API to allows access to third-party applications.

Ministry of Internal Affairs and Communications, JAPAN

The project has received funding from the European Union H2020 Programme under grant agreement n° 101016453. The Japanese consortium received funding by the Japanese Ministry of Internal Affairs and Communication (MIC).

Disclaimer

This document contains material, which is the copyright of certain e-VITA consortium parties and may not be reproduced or copied without permission.

The information contained in this document is the proprietary CO information of the e-VITA consortium and may not be disclosed except in accordance with the consortium agreement.

The commercial use of any information contained in this document may require a license from the proprietor of that information.

Neither the e-VITA consortium as a whole, nor a certain party of the e-VITA consortium warrant that the information contained in this document is capable of use, or that use of the information is free from risk and accept no liability for loss or damage suffered by any person using this information.

E-VITA – **European-Japanese Virtual Coach for Smart Ageing**

E-VITA (EU PROJECT NUMBER 101016453)

[Number and title of work-package] **WP7 – Integration, Tools & Services**

Document title: **D7.5 e-VITA API and technical guidelines for third party access**

Editors: ENG: Martino Maggio, Francesca D'Agresti, Jerome Boudy (IMT), Rainer Wieching (USI)

Work-package leader: ENG, AIST

Copyright notice

Ministry of Internal Affairs and Communications, JAPAN

The project has received funding from the European Union H2020 Programme under grant agreement n° 101016453. The Japanese consortium received funding by the Japanese Ministry of Internal Affairs and Communication (MIC).

1. Introduction

e-VITA Manager APIs are the interfaces of the *e-VITA Manager*.

The *e-VITA Manager* can be considered the central part and the key component of the e-VITA platform, developed within the project and orchestrating the integration between devices and e-VITA specific modules. Through a set of appropriately implemented connectors, it allows the correct integration of the IoT gateways, wearables, devices and robots sensor data.

The *e-VITA Manager* is the central component of e-VITA platform managing the main communication among the different other element of the architecture and storing the information. In particular, it represents one the components completely developed within e-VITA project allowing to reuse some of the capabilities of the Digital Enabler platform, for the e-VITA project objectives. The e-VITA Manager provides different capabilities and interfaces, via REST APIs, to access them. The component exposes a set of REST APIs that can be consumed by any client. It works as a middleware component to interact with the components of the Digital Enabler (Engineering Ingegneria Informatica Spa, 2022).

From the user point of view, the main user interface to access e-VITA platform functionalities is *e-VITA Dashboard*. It allows to access and use all the functionalities of the e-VITA Manager: to manage user profile information, to register and configure devices, to consult all the information of the registered devices, to monitor user data coming from devices. All functionalities exposed as API by the e-VITA Manager are described in detail in this document, in the following paragraphs.

All the architectural details regarding the e-VITA platform, a detailed description of the e-VITA Manager, the e-VITA Dashboard and all the components they communicate with, in *D7.4 – e-VITA Platform Architecture – Final Version* (e-VITA, 2022).

There are five sets of APIs exposed by e-VITA Manager:

- User API provides access to user information and allows modification or creation of a new user. The API are based on OAuth 2.0 authorization protocol (*Password Grant Type* (OAuth, 2022)).
- Device API allows the visualization and management of devices belonging to a user. functionalities to allow devices to send their measurements or files to the platform and provides access to these measurements, stored as historical data within it. The API are based on OAuth 2.0 authorization protocol (*Password Grant Type* (OAuth, 2022)).
- Service API provides access and management of the cloud services in which a specific category of devices stores its measurements. The API are based on OAuth 2.0 authorization protocol (*Password Grant Type* (OAuth, 2022)).
- Researchers API allows a certain category of users, those who have the role of *researcher*, to obtain the historical data of the devices they need. The API are based on OAuth 2.0 authorization protocol (*Password Grant Type* (OAuth, 2022)).
- Clients API allows access to the e-VITA services to third-party applications, which are authorized to access them. The API are based on OAuth 2.0 authorization protocol (*Client Credentials Grant Type* (OAuth, 2022)).

The online version of this documentation is also available (Apiary, 2022), in which it is possible to directly test the APIs.

Ministry of Internal Affairs and Communications, JAPAN

The project has received funding from the European Union H2020 Programme under grant agreement n° 101016453. The Japanese consortium received funding by the Japanese Ministry of Internal Affairs and Communication (MIC).

1.1 Authentication

e-VITA Manager API uses Header Authentication.

In order to use the API and perform requests, a valid JWT token must be inserted in the request header, as follows:

```
Authorization: Bearer JWT
```

It is needed to replace a valid access token in place of the string `JWT`.

As already mentioned earlier in the Introduction, API controllers must receive a JWT access token within the header of the request. It comes from two different paths, depending on the type of controller, as described below:

- OAuth2.0 Password Grant Type: it is a legacy way to exchange a user's credentials to obtain an access token.
- OAuth2.0 Client Credentials Grant Type: it is used by external client applications to obtain an access token outside of the context of a user.

All the details on how an access token is obtained by the e-VITA platform in *D7.7 - Data Privacy and Security Modules – First release* (e-VITA, 2022).

1.2 Error Responses

The possible HTTP error response codes the requests can return are described in Table.

HTTP Code	Type	Description
400	Bad Request	The content of your request is not correct
401	Unauthorized	You are not logged in or the Authorization token you are providing is not valid
403	Forbidden	You have no rights to perform the request
404	Not Found	The resource you are looking for does not exists
500	Internal server error	There was an internal error in the system so your request cannot be completed

Ministry of Internal Affairs and Communications, JAPAN

The project has received funding from the European Union H2020 Programme under grant agreement n° 101016453. The Japanese consortium received funding by the Japanese Ministry of Internal Affairs and Communication (MIC).

2 e-VITA Manager API – User controller

User APIs represents the way to obtain and manage the information of the user connected to the platform (the user associated with the JWT token inserted in the request header).

Retrieve User		
Request Type	GET	
Request Description	This request does not have any parameters. It allows to obtain all the information of the user corresponding to the JWT authorization token passed within the request header. The information obtained as a response is in JSON format. Response code: - Successful operation uses 200 OK: indicates that the user's information was successfully obtained. - Errors use a non-2xx and (optionally) an error payload. See subsection on "Error Responses" for more details.	
Request Path	`/user`	
Request Headers	`Authorization: Bearer JWT`	
Request Parameters	---	
Request Body	---	
Response Codes	200	400 401 403 404 500

Response Body	{
	"id": "string",
	"country": "string",
	"city": "string",
	"language": "string",
	"gender": "string",
	"username": "string",
	"hasNetatmoToken": true,
	"hasStrapiToken": true,
	"hasOuraringToken": true,
	"modifiedBy": "string",
	"fiwareService": "string",
	"telegramUsername": "string",
	"createdAt": "2023-01-20T14:16:11.216Z",
	"modifiedAt": "2023-01-20T14:16:11.216Z"
	}
		{
		"timestamp": "2023-01-20T17:33:39.710Z",
		"message": "string",
		"status": 0,
		"error": "string"
		}

Table 1: Retrieve User API description

Create a New User		
Request Type	POST	
Request Description	You may create a new user within the platform using this action. It takes a JSON object containing all the information that will be associated with the new user. The information obtained as a response is in JSON format. Response code: • Successful operation uses 200 OK: indicates that the user's information was successfully obtained. • Errors use a non-2xx and (optionally) an error payload. See subsection on "Error Responses" for more details.	
Request Path	`/user`	
Request Headers	`Authorization: Bearer JWT`	
Request Parameters	`---`	

Request Body	`{`
	`  "id": "string",`
	`   "country": "string",`
	`   "city": "string",`
	`   "language": "string",`
	`   "gender": "string",`
	`   "username": "string",`
	`   "hasNetatmoToken": true,`
	`   "hasStrapiToken": true,`
	`   "hasOuraringToken": true,`
	`   "modifiedBy": "string",`
	`   "fiwareService": "string",`
	`   "telegramUsername": "string",`
	`   "createdAt": "2023-01-23T09:35:13.160Z",`
	`   "modifiedAt": "2023-01-23T09:35:13.160Z"`
	`}`	

Response Codes	200	400 401 403 404 500

Response Body	`{`
	`"id": "string",`
	`"country": "string",`
	`"city": "string",`
	`"language": "string",`
	`"gender": "string",`
	`"username": "string",`
	`"hasNetatmoToken": true,`
	`"hasStrapiToken": true,`
	`"hasOuraringToken": true,`
	`"modifiedBy": "string",`
	`"fiwareService": "string",`
	`"telegramUsername": "string",`
	`"createdAt": "2023-01-`
	`20T14:16:11.216Z",`
		`{`
		`  "timestamp": "2023-01-`
		`20T17:33:39.710Z",`
		`    "message": "string",`
		`    "status": 0,`
		`    "error": "string"`
		`  }`

	"modifiedAt": "2023-01-20T14:16:11.216Z" }	

Table 2: Create a New User API description

Update User		
Request Type	PUT	
Request Description	This action gives the possibility to modify/update the information of a user registered on the platform. It takes a JSON object containing the user information to update. The information obtained as a response is in JSON format. Response code: • Successful operation uses 200 OK indicates that the user's information was successfully updated. • Errors use a non-2xx and (optionally) an error payload. See subsection on "Error Responses" for more details.	
Request Path	/user	
Request Headers	Authorization: Bearer JWT	
Request Body	---	
Request Parameters	---	
Response Codes	200	400 401 403 404 500

Response Body	{
	"id": "string",
	"country": "string",
	"city": "string",
	"language": "string",
	"gender": "string",
	"username": "string",
	"hasNetatmoToken": true,
	"hasStrapiToken": true,
	"hasOuraringToken": true,
	"modifiedBy": "string",
	"fiwareService": "string",
	"telegramUsername": "string",
	"createdAt": "2023-01-20T14:16:11.216Z",
	"modifiedAt": "2023-01-20T14:16:11.216Z"
	}
		{
		"timestamp": "2023-01-20T17:33:39.710Z",
		"message": "string",
		"status": 0,
		"error": "string"
		}

Table 3: Update User API description

The project has received funding from the European Union H2020 Programme under grant agreement n° 101016453. The Japanese consortium received funding by the Japanese Ministry of Internal Affairs and Communication (MIC).

Delete User		
Request Type	DELETE	
Request Description	This request does not have any parameters. It allows to remove all the information of the user corresponding to the JWT authorization token passed within the request header. The information obtained as a response is in JSON format. Response code: - Successful operation uses 200 OK: indicates that the user was successfully deleted. - Errors use a non-2xx and (optionally) an error payload. See subsection on "Error Responses" for more details.	
Request Path	`/user`	
Request Headers	`Authorization: Bearer JWT`	
Request Parameters	---	
Request Body	---	
Response Codes	200	400 401 403 404 500

Response Body	`{}`	`{`
		`"timestamp": "2023-01-20T17:33:39.710Z",`
		`"message": "string",`
		`"status": 0,`
		`"error": "string"`
		`}`

Table 4: Delete User API description

Ministry of Internal Affairs and Communications, JAPAN

The project has received funding from the European Union H2020 Programme under grant agreement n° 101016453. The Japanese consortium received funding by the Japanese Ministry of Internal Affairs and Communication (MIC).

Retrieve User Emotions		
Request Type	PUT	
Request Description	This action returns the emotions detected by the external component Emotion Detection, explained in *D7.4 e-VITA Platform Architecture – Final Version* (e-VITA, 2022), which communicates with the e-VITA platform in order to associate specific emotions to audio files previously sent from the user's devices. It returns a JSON array containing an object for each audio file associated with the user. Each JSON object contains: the reference to the device that previously sent the audio file, the language detected in the file which is expressed with the reference code of the country (de, it, en, etc.), the file name and finally the *detectedEmotion* field, which contains the detected emotion associated with the audio file. Examples of emotions detected are *happiness*, *sadness*, etc. Response code: • Successful operation uses 200 OK indicates that the user's emotions was successfully updated. • Errors use a non-2xx and (optionally) an error payload. See subsection on "Error Responses" for more details.	
Request Path	`/user/emotions`	
Request Headers	`Authorization: Bearer JWT`	
Request Parameters	---	
Request Body	---	
Response Codes	200	400 401 403 404 500

Response Body	[
	{
	"id": "string",
	"userId": "string",
	"deviceId": "string",
	"language": "string",
	"file": "string",
	"detectedEmotion": "string",
	"createdAt": "2023-01-23T09:53:11.134Z"
	}
	]
		{
		"timestamp": "2023-01-20T17:33:39.710Z",
		"message": "string",
		"status": 0,
		"error": "string"
		}

Table 5: Retrieve User Emotions API description

3 e-VITA Manager API – Device controller

Device APIs represents the way to obtain and manage all the information relating to the devices belonging to the user, who is identified by the JWT authorization token passed within the request header. Some APIs of the group do not require authorization via access JWT token, as they are used directly by the devices to send their measurements or files to the e-VITA platform.

Retrieve Devices		
Request Type	GET	
Request Description	This request does not have any parameters. It allows to obtain all registered devices belonging to the user specified by the JWT token. The information obtained as a response is in JSON format and it is a JSON Array. Each element of the array identifies a specific device. Response code: • Successful operation uses 200 OK indicates that the user's devices was successfully obtained. • Errors use a non-2xx and (optionally) an error payload. See subsection on "Error Responses" for more details.	
Request Path	`/devices`	
Request Headers	`Authorization: Bearer JWT`	
Request Parameters	---	
Request Body	---	
Response Codes	200	400 401 403 404 500

Response Body	[
	  {
	    "evitaDeviceId": "string",
	    "name": "string",
	    "type": "NETATMO_AIR_QUALITY",
	    "sourceDeviceId": "string",
	    "attributes": [
	      {
	"attributeType": "string",
	"attributeName": "string"
	      }
	    ],
	    "staticAttributes": [
	      {
	"metadataName": "string",
	"metadataType": "string",
		{
		  "timestamp": "2023-01-20T17:33:39.710Z",
		    "message": "string",
		    "status": 0,
		    "error": "string"
		  }

	"metadataValue": "string"
	      }
	    ],
	    "deviceToken": "string",
	    "lastMeasureSent": "2023-01-23T10:35:51.726Z",
	    "evitaUserId": "string",
	    "modifiedBy": "string",
	    "createdAt": "2023-01-23T10:35:51.726Z",
	    "modifiedAt": "2023-01-23T10:35:51.726Z",
	    "enabled": true,
	    "frequency": 0,
	    "measureTime": "string",
	    "trigger": true,
	    "registrationEmail": "string",
	    "registrationCheck": true
	  }
	]	

Table 6: Retrieve Devices API description

Retrieve Device by ID		
Request Type	GET	
Request Description	This request allows to obtain the specific device whose id is passed to the request as a parameter, belonging to the user specified by the JWT token. The information obtained as a response is in JSON format and it is a JSON Object which identifies the device. It takes in input a single parameter: • id: the string with which the device of interest is identified within the e-VITA platform. Response code: • Successful operation uses 200 OK indicates that the user's device was successfully obtained. • Errors use a non-2xx and (optionally) an error payload. See subsection on "Error Responses" for more details.	
Request Path	/devices/{id}	
Request Headers	Authorization: Bearer JWT	
Request Parameters	id (required, string) - Device ID	
Request Body	---	
Response Codes	200	400 401 403 404 500

The project has received funding from the European Union H2020 Programme under grant agreement n° 101016453. The Japanese consortium received funding by the Japanese Ministry of Internal Affairs and Communication (MIC).

Response Body	[
	{
	"evitaDeviceId": "string",
	"name": "string",
	"type": "NETATMO_AIR_QUALITY",
	"sourceDeviceId": "string",
	"attributes": [
	{
	"attributeType": "string",
	"attributeName": "string"
	}
	],
	"staticAttributes": [
	{
	"metadataName": "string",
	"metadataType": "string",
	"metadataValue": "string"
	}
	],
	"deviceToken": "string",
	"lastMeasureSent": "2023-01-23T10:35:51.726Z",
	"evitaUserId": "string",
	"modifiedBy": "string",
	"createdAt": "2023-01-23T10:35:51.726Z",
	"modifiedAt": "2023-01-23T10:35:51.726Z",
	"enabled": true,
	"frequency": 0,
	"measureTime": "string",
	"trigger": true,
	"registrationEmail": "string",
	"registrationCheck": true
	}
	]
		{
		"timestamp": "2023-01-20T17:33:39.710Z",
		"message": "string",
		"status": 0,
		"error": "string"
		}

Table 7: Retrieve Device by ID API description

The project has received funding from the European Union H2020 Programme under grant agreement n° 101016453. The Japanese consortium received funding by the Japanese Ministry of Internal Affairs and Communication (MIC).

Retrieve Device Historical Data by ID		
Request Type	GET	
Request Description	This request allows to obtain the measurements collected by the specific device whose id is passed to the request as a parameter, belonging to the user specified by the JWT token. The information obtained as a response is a *string* containing the past measurements detected by the device and stored within the platform. The format of the string depends on the attributes that the specific device has, but in general it presents the list of device's attributes and the value they have assumed, in relation to the date and time on which the measurement was taken. It takes in input three parameters: • id: the string with which the device of interest is identified within the e-VITA platform. • startDate: it is a string in *DateTime* format which represents the starting date from which to show the measurements detected by the device. • endDate: it is a string in *DateTime* format which represents the end date until which to view the measurements taken by the device. Response code: • Successful operation uses 200 OK indicates that the device's historical data were successfully obtained. • Errors use a non-2xx and (optionally) an error payload. See subsection on "Error Responses" for more details.	
Request Path	/devices/{id}/historical{?startDate,endDate}	
Request Headers	Authorization: Bearer JWT	
Request Parameters	id (required, string) - Device Id. startDate (required, dateTime) - Starting date from which to show the measurements detected by the device. endDate (optional, dateTime) - End date until which to view the measurements taken by the device.	
Request Body	---	
Response Codes	200	400 401 403 404 500

Response Body	{}	{
		  "timestamp": "2023-01-20T17:33:39.710Z",
		    "message": "string",
		    "status": 0,
		    "error": "string"
		  }

Table 8: Retrieve Device Historical Data by ID API description

The project has received funding from the European Union H2020 Programme under grant agreement n° 101016453. The Japanese consortium received funding by the Japanese Ministry of Internal Affairs and Communication (MIC).

Retrieve Device Last Measurement by ID		
Request Type	GET	
Request Description	This request allows to obtain the last measurement detected by the specific device whose id is passed to the request as a parameter, belonging to the user specified by the JWT token. The last measurement of the device is taken by the FIWARE Context Broker and it represents, based on the typology of device, either the last measurement sent directly by the device to the e-VITA platform or the last measurement taken from the e-VITA platform from the cloud service associated with the device. The information obtained as a response is in JSON format and it is a JSON Object containing the measurement. It takes in input a single parameter: • id: the string with which the device of interest is identified within the e-VITA platform. Response code: • Successful operation uses 200 OK: indicates that the devices's last measurement was successfully obtained. • Errors use a non-2xx and (optionally) an error payload. See subsection on "Error Responses" for more details.	
Request Path	/devices/{id}/get_last_measurement	
Request Headers	Authorization: Bearer JWT	
Request Parameters	id (required, string) - Device Id.	
Request Body	---	
Response Codes	200	400 401 403 404 500

Response Body	{}	{
		"timestamp": "2023-01-20T17:33:39.710Z",
		"message": "string",
		"status": 0,
		"error": "string"
		}

Table 9: Retrieve Device Last Measurement by ID API description

The project has received funding from the European Union H2020 Programme under grant agreement n° 101016453. The Japanese consortium received funding by the Japanese Ministry of Internal Affairs and Communication (MIC).

Retrieve Device by ID		
Request Type	GET	
Request Description	This action allows to obtain the information of the user who owns the device uniquely identified by the pair *deviceId* and *deviceToken*, passed as parameters to the request. The information obtained as a response is in JSON format and it is a JSON Object containing the user information. It takes in input two parameters: • deviceId: the string with which the device of interest is identified within the e-VITA platform. • deviceToken: the string that together with the *deviceId*, identifies the device of interest within the e-VITA platform. Response code: • Successful operation uses 200 OK: indicates that the user's information was successfully obtained. • Errors use a non-2xx and (optionally) an error payload. See subsection on "Error Responses" for more details.	
Request Path	/devices/user{?deviceId,deviceToken}	
Request Headers	Authorization: Bearer JWT	
Request Parameters	id (required, string) - Device Id. deviceToken (required, string) - Device Token	
Request Body	---	
Response Codes	200	400 401 403 404 500

Response Body	{
	"id": "string",
	"country": "string",
	"city": "string",
	"language": "string",
	"username": "string",
	"gender": "string",
	"telegramUsername": "string"
	}
		{
		"timestamp": "2023-01-20T17:33:39.710Z",
		"message": "string",
		"status": 0,
		"error": "string"
		}

Table 10: Retrieve Device by ID API description

Retrieve Dashboard Device Type		
Request Type	GET	
Request Description	This request does not receive any parameters and allows to obtain the device which has the specific type *DASHBOARD*, belonging to the user specified by the JWT token. For each user there is a single device of this type; it allows the test communication with the Rasa Dialogue Manager. The information obtained as a response is in JSON format and it is a JSON Object which identifies the device. Response code: • Successful operation uses 200 OK: indicates that the user's Dashboard device was successfully obtained. • Errors use a non-2xx and (optionally) an error payload. See subsection on "Error Responses" for more details.	
Request Path	/devices/dashboard	
Request Headers	Authorization: Bearer JWT	
Request Parameters	---	
Request Body	---	
Response Codes	200	400 401 403 404 500

Response Body	[
	{
	"evitaDeviceId":
	"string",
	"name": "string",
	"type": "DASHBOARD",
	"sourceDeviceId":
	"string",
	"attributes": [
	{
	"attributeType":
	"string",
	"attributeName":
	"string"
	}
	],
	"staticAttributes": [
	{
	"metadataName":
	"string",
	"metadataType":
	"string",
	"metadataValue":
	"string"
	}
		{
		"timestamp": "2023-01-
		20T17:33:39.710Z",
		"message": "string",
		"status": 0,
		"error": "string"
		}

The project has received funding from the European Union H2020 Programme under grant agreement n° 101016453. The Japanese consortium received funding by the Japanese Ministry of Internal Affairs and Communication (MIC).

```
    ],
    "deviceToken": "string",
    "lastMeasureSent": "2023-
01-23T10:35:51.726Z",
    "evitaUserId": "string",
    "modifiedBy": "string",
    "createdAt": "2023-01-
23T10:35:51.726Z",
    "modifiedAt": "2023-01-
23T10:35:51.726Z",
    "enabled": true,
    "frequency": 0,
    "measureTime": "string",
    "trigger": true,
    "registrationEmail":
"string",
    "registrationCheck": true
  }
]
```

Table 11: Retrieve Dashboard Device Type API description

Update Device Name		
Request Type	PUT	
Request Description	This action gives the possibility to modify/update the information of a user's device registered on the platform. In this case, the action allows to update the device name. It takes in input two parameters: • id: the string with which the device of interest is identified within the e-VITA platform. • name: the new name you want to associate with the device. Response code: • Successful operation uses 200 OK: indicates that the device was successfully updated. • Errors use a non-2xx and (optionally) an error payload. See subsection on "Error Responses" for more details.	
Request Path	/devices/{id}/update_name{?name}	
Request Headers	Authorization: Bearer JWT	
Request Parameters	id (required, string) - Device Id. name(required, string) - Device Name to update	
Request Body	- - -	
Response Codes	200	400 401 403 404

The project has received funding from the European Union H2020 Programme under grant agreement n° 101016453. The Japanese consortium received funding by the Japanese Ministry of Internal Affairs and Communication (MIC).

		500

Response Body	{
	"evitaDeviceId": "string",
	"name": "string",
	"type": "NETATMO_AIR_QUALITY",
	"sourceDeviceId": "string",
	"attributes": [
	{
	"attributeType": "string",
	"attributeName": "string"
	}
	],
	"staticAttributes": [
	{
	"metadataName": "string",
	"metadataType": "string",
	"metadataValue": "string"
	}
	],
	"deviceToken": "string",
	"lastMeasureSent": "2023-01-23T11:13:49.586Z",
	"evitaUserId": "string",
	"modifiedBy": "string",
	"createdAt": "2023-01-23T11:13:49.586Z",
	"modifiedAt": "2023-01-23T11:13:49.586Z",
	"enabled": true,
	"frequency": 0,
	"measureTime": "string",
	"trigger": true,
	"registrationEmail": "string",
	"registrationCheck": true
	}
		{
		"timestamp": "2023-01-20T17:33:39.710Z",
		"message": "string",
		"status": 0,
		"error": "string"
		}

Table 12: Update Device Name API description

The project has received funding from the European Union H2020 Programme under grant agreement n° 101016453. The Japanese consortium received funding by the Japanese Ministry of Internal Affairs and Communication (MIC).

Update Device Frequency		
Request Type	PUT	
Request Description	This action gives the possibility to modify/update the information of a user's device registered on the platform. In this case, the action allows to update the device frequency attribute. This attribute has value for devices that are connected to a cloud service and that are configured to send and store their detected measurements in that external service. The e-VITA platform, in the case of this specific category of devices, obtains the measurements of these devices within the respective cloud services, to save such data within the platform. By default, the platform does so every 10 minutes. This API allows to change the frequency with which the platform obtains measurements, with a different value from the default one. It takes in input two parameters: • id: the string with which the device of interest is identified within the e-VITA platform. • frequency: the new frequency value you want to associate with the device. Response code: • Successful operation uses 200 OK: indicates that the device was successfully updated. • Errors use a non-2xx and (optionally) an error payload. See subsection on "Error Responses" for more details.	
Request Path	/devices/{id}/update_measures_frequency{?frequency}	
Request Headers	Authorization: Bearer JWT	
Request Parameters	id (required, string) - Device Id. frequency (required, string) - Device Frequency to update	
Request Body	---	
Response Codes	200	400 401 403 404 500

Response Body	{
	"evitaDeviceId": "string",
	"name": "string",
	"type":
	"NETATMO_AIR_QUALITY",
	"sourceDeviceId": "string",
	"attributes": [
	{
	"attributeType":
	"string",
		{
		"timestamp": "2023-01-
		20T17:33:39.710Z",
		"message": "string",
		"status": 0,
		"error": "string"
		}

	"attributeName":
	"string"
	}
	],
	"staticAttributes": [
	{
	"metadataName":
	"string",
	"metadataType":
	"string",
	"metadataValue":
	"string"
	}
	],
	"deviceToken": "string",
	"lastMeasureSent": "2023-01-23T11:19:23.761Z",
	"evitaUserId": "string",
	"modifiedBy": "string",
	"createdAt": "2023-01-23T11:19:23.762Z",
	"modifiedAt": "2023-01-23T11:19:23.762Z",
	"enabled": true,
	"frequency": 0,
	"measureTime": "string",
	"trigger": true,
	"registrationEmail":
	"string",
	"registrationCheck": true
	}	

Table 13: Update Device Frequency API description

Update Device Static Attributes	
Request Type	PUT
Request Description	This action gives the possibility to modify/update the information of a user's device registered on the platform. In this case, the action allows to update the device static attributes. They are additional attributes of the device that can be useful for attributing further information to the device, such as for example to indicate the position where the device is located by creating a static attribute called location and whose value can be, for example, the room where the device has been installed within the user's home. The body is a JSON Array where each object represents a static attribute. It takes in input one parameter: • id: the string with which the device of interest is identified within the e-VITA platform. Response code: • Successful operation uses 200 OK: indicates that the device was successfully updated. • Errors use a non-2xx and (optionally) an error payload. See subsection on "Error Responses" for more details.

The project has received funding from the European Union H2020 Programme under grant agreement n° 101016453. The Japanese consortium received funding by the Japanese Ministry of Internal Affairs and Communication (MIC).

Request Path	/devices/{id}/staticAttributes	
Request Headers	Authorization: Bearer JWT	
Request Parameters	id (required, string) - Device Id.	

Request Body	[
	{
	"metadataName": "string",
	"metadataType": "string",
	"metadataValue": "string"
	}
	]	

Response Codes	200	400 401 403 404 500

Response Body	{
	"evitaDeviceId": "string",
	"name": "string",
	"type": "NETATMO_AIR_QUALITY",
	"sourceDeviceId": "string",
	"attributes": [
	{
	"attributeType": "string",
	"attributeName": "string"
	}
	],
	"staticAttributes": [
	{
	"metadataName": "string",
	"metadataType": "string",
	"metadataValue": "string"
	}
	],
	"deviceToken": "string",
	"lastMeasureSent": "2023-01-23T11:19:23.761Z",
	"evitaUserId": "string",
	"modifiedBy": "string",
	"createdAt": "2023-01-23T11:19:23.762Z",
	"modifiedAt": "2023-01-23T11:19:23.762Z",
	"enabled": true,
	"frequency": 0,
	"measureTime": "string",
	"trigger": true,
	"registrationEmail": "string",
	"registrationCheck": true
	}
		{
		"timestamp": "2023-01-20T17:33:39.710Z",
		"message": "string",
		"status": 0,
		"error": "string"
		}

Table 14: Update Device Static Attributes API description

The project has received funding from the European Union H2020 Programme under grant agreement n° 101016453. The Japanese consortium received funding by the Japanese Ministry of Internal Affairs and Communication (MIC).

Update Device Token		
Request Type	PUT	
Request Description	This action gives the possibility to modify/update the information of a user's device registered on the platform. In this case, the action allows to update the device token attribute. Through this action, the *deviceToken* field of the device will be automatically updated, which together with the *evitaDeviceId* attribute, allows it to be identified unambiguously within the e-VITA platform. It takes in input one parameter: • id: the string with which the device of interest is identified within the e-VITA platform. Response code: • Successful operation uses 200 OK: indicates that the device was successfully updated. • Errors use a non-2xx and (optionally) an error payload. See subsection on "Error Responses" for more details.	
Request Path	/devices/{id}/refresh_token	
Request Headers	Authorization: Bearer JWT	
Request Parameters	id (required, string) - Device Id.	
Request Body	---	
Response Codes	200	400 401 403 404 500

Response Body	{
	"evitaDeviceId": "string",
	"name": "string",
	"type":
	"NETATMO_AIR_QUALITY",
	"sourceDeviceId": "string",
	"attributes": [
	{
	"attributeType":
	"string",
	"attributeName":
	"string"
	}
	],
	"staticAttributes": [
	{
		{
		"timestamp": "2023-01-
		20T17:33:39.710Z",
		"message": "string",
		"status": 0,
		"error": "string"
		}

	`"metadataName":`
	`"string",`
	`"metadataType":`
	`"string",`
	`"metadataValue":`
	`"string"`
	`}`
	`],`
	`"deviceToken": "string",`
	`"lastMeasureSent": "2023-`
	`01-23T11:19:23.761Z",`
	`"evitaUserId": "string",`
	`"modifiedBy": "string",`
	`"createdAt": "2023-01-`
	`23T11:19:23.762Z",`
	`"modifiedAt": "2023-01-`
	`23T11:19:23.762Z",`
	`"enabled": true,`
	`"frequency": 0,`
	`"measureTime": "string",`
	`"trigger": true,`
	`"registrationEmail":`
	`"string",`
	`"registrationCheck": true`
	`}`	

Table 15: Update Device Token API description

Update Device Enable value	
Request Type	PUT
Request Description	This action gives the possibility to modify/update the information of a user's device registered on the platform. In this case, the action allows to update the device enable attribute. This attribute has a Boolean value which if set to true allows the device to send its detected measurements to the e-VITA platform; in the case of a device configured to save its measurements into an external cloud service, it allows the e-VITA platform to retrieve its measurements from the cloud service. If this attribute is set to false, it disables the device, thus preventing it to save its measurements within the e-VITA platform. It takes in input two parameters: • id: the string with which the device of interest is identified within the e-VITA platform. • enabled: the Boolean value which if true allows to enable the device, if false disables it. Response code: • Successful operation uses 200 OK: indicates that the device was successfully updated. • Errors use a non-2xx and (optionally) an error payload. See subsection on "Error Responses" for more details.

Request Path	/devices/{id}/enable_device{?enabled}	
Request Headers	Authorization: Bearer JWT	
Request Parameters	**id** (required, string) - Device Id. **enabled** (required, boolean) - Device Enable attribute that identifies whether it is enabled to send its measurements to the platform	
Request Body	---	
Response Codes	200	400 401 403 404 500

Response Body	`{`
	`"evitaDeviceId": "string",`
	`"name": "string",`
	`"type": "NETATMO_AIR_QUALITY",`
	`"sourceDeviceId": "string",`
	`"attributes": [`
	`{`
	`"attributeType": "string",`
	`"attributeName": "string"`
	`}`
	`],`
	`"staticAttributes": [`
	`{`
	`"metadataName": "string",`
	`"metadataType": "string",`
	`"metadataValue": "string"`
	`}`
	`],`
	`"deviceToken": "string",`
	`"lastMeasureSent": "2023-01-23T11:19:23.761Z",`
	`"evitaUserId": "string",`
	`"modifiedBy": "string",`
	`"createdAt": "2023-01-23T11:19:23.762Z",`
	`"modifiedAt": "2023-01-23T11:19:23.762Z",`
	`"enabled": true,`
	`"frequency": 0,`
	`"measureTime": "string",`
	`"trigger": true,`
	`"registrationEmail": "string",`
	`"registrationCheck": true`
	`}`
		`{`
		`"timestamp": "2023-01-20T17:33:39.710Z",`
		`"message": "string",`
		`"status": 0,`
		`"error": "string"`
		`}`

Table 16: Update Device Enable value API description

The project has received funding from the European Union H2020 Programme under grant agreement n° 101016453. The Japanese consortium received funding by the Japanese Ministry of Internal Affairs and Communication (MIC).

Delete Device		
Request Type	DELETE	
Request Description	This action removes the device from the platform, whose Id is passed to the request as a parameter. It takes in input one parameter: • id: the string with which the device of interest is identified within the e-VITA platform. Response code: • Successful operation uses 200 OK: indicates that the device was successfully deleted. • Errors use a non-2xx and (optionally) an error payload. See subsection on "Error Responses" for more details.	
Request Path	/devices/{id}	
Request Headers	Authorization: Bearer JWT	
Request Parameters	id (required, string) - Device Id.	
Request Body	---	
Response Codes	200	400 401 403 404 500

Response Body	{}	`{`
		`"timestamp": "2023-01-20T17:33:39.710Z",`
		`"message": "string",`
		`"status": 0,`
		`"error": "string"`
		`}`

Table 17: Delete Device API description

The project has received funding from the European Union H2020 Programme under grant agreement n° 101016453. The Japanese consortium received funding by the Japanese Ministry of Internal Affairs and Communication (MIC).

Create a New Device		
Request Type	POST	
Request Description	You may create a new device that will belong to the user specified by the JWT token. It takes a JSON object containing all the information that contains part of the attributes associated with the new device. If the operation will be successful, the action will return a JSON object also containing other additional attributes associated with the new device, such as: the *evitaDeviceId* and *deviceToken* fields, two strings that uniquely identify the device within the e-VITA platform. Response code: • Successful operation uses 200 OK: indicates that the device was successfully created. • Errors use a non-2xx and (optionally) an error payload. See subsection on "Error Responses" for more details.	
Request Path	/devices	
Request Headers	Authorization: Bearer JWT	
Request Parameters	---	

Request Body	`{`
	`  "type": "NETATMO_AIR_QUALITY",`
	`  "sourceDeviceId": "string",`
	`  "deviceName": "string",`
	`  "frequency": 0,`
	`  "measureTime": "string",`
	`  "registrationEmail": "string",`
	`  "registrationCheck": true,`
	`  "staticAttributes": [`
	`    {`
	`      "metadataName": "string",`
	`      "metadataType": "string",`
	`      "metadataValue": "string"`
	`    }`
	`  ]`
	`}`	

Response Codes	200	400 401 403 404 500

Response Body	`{`
	`  "evitaDeviceId": "string",`
	`  "name": "string",`
	`  "type":`
	`"NETATMO_AIR_QUALITY",`
	`  "sourceDeviceId": "string",`
	`  "attributes": [`
	`    {`
		`{`
		`  "timestamp": "2023-01-`
		`20T17:33:39.710Z",`
		`    "message": "string",`
		`    "status": 0,`
		`    "error": "string"`
		`  }`

	"attributeType": "string",
	"attributeName": "string"
	}
	],
	"staticAttributes": [
	{
	"metadataName": "string",
	"metadataType": "string",
	"metadataValue": "string"
	}
	],
	"deviceToken": "string",
	"lastMeasureSent": "2023-01-23T10:38:00.510Z",
	"evitaUserId": "string",
	"modifiedBy": "string",
	"createdAt": "2023-01-23T10:38:00.510Z",
	"modifiedAt": "2023-01-23T10:38:00.510Z",
	"enabled": true,
	"frequency": 0,
	"measureTime": "string",
	"trigger": true,
	"registrationEmail": "string",
	"registrationCheck": true
	}	

Table 18: Create a New Device API description

The project has received funding from the European Union H2020 Programme under grant agreement n° 101016453. The Japanese consortium received funding by the Japanese Ministry of Internal Affairs and Communication (MIC).

Send Device File		
Request Type	POST	
Request Description	This action allows to send an audio file and associate it with a specific device, uniquely identified by the *deviceId* and *deviceToken* attributes passed as parameters to the request. This action does not require a JWT Authorization token. If the operation will be successful, the file will be sent and stored into the e-VITA platform. Accepted formats for the audio file are audio/wave, audio/wav, audio/x-wav, audio/x-pn-wav. Response code: • Successful operation uses 200 OK: indicates that the device was successfully created. • Errors use a non-2xx and (optionally) an error payload. See subsection on "Error Responses" for more details.	
Request Path	/devices/send_file{?deviceId,deviceToken,file}	
Request Headers	Content-Type: multipart/form-data	
Request Parameters	**deviceId** (required, string) - Device Id **deviceToken** (required, string) - Device Token **file** (required, binary) - File belonging to the Device to send and store into the platform	

Request Body	`-----BOUNDARY`
	`Content-Disposition: form-data; name="audio_file"; filename="audio_file.wav"`
	`Content-Type: audio/wav`
	`Content-Transfer-Encoding: base64`
	`/9j/4AAQSkZJRgABAQEAYABgAAD/2wBDAAgGBgcGBQgHBwcJCQgKDBQNDAsLDBkSEw8UHRofHh0a`
	`HBwgJC4nICIsIxwcKDcpLDAxNDQ0Hyc5PTgyPC4zNDL/2wBDAQkJCQwLDBgNDRgyIRwhMjIyMjIy`
	`MjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjL/wAARCAABAAEDASIA`
	`AhEBAxEB/8QAFQABAQAAAAAAAAAAAAAAAAAAAAAf/xAAUEAEAAAAAAAAAAAAAAAAAAAAAAAAA/8QAFAEB`
	`AAAAAAAAAAAAAAAAAAAAAP/EABQRAQAAAAAAAAAAAAAAAAAAAAAD/2gAMAwEAAhEDEQA/AL+AD//Z`
	`-----BOUNDARY`	

Response Codes	200	400 401 403 404 500

Response Body	`{}`	`{`
		`  "timestamp": "2023-01-20T17:33:39.710Z",`
		`    "message": "string",`
		`    "status": 0,`
		`    "error": "string"`
		`  }`

The project has received funding from the European Union H2020 Programme under grant agreement n° 101016453. The Japanese consortium received funding by the Japanese Ministry of Internal Affairs and Communication (MIC).

Table 19: Send Device File API description

Send Device Data		
Request Type	POST	
Request Description	This action allows to send data to the e-VITA platform from a specific device, uniquely identified by the *deviceId* and *deviceToken* attributes, passed as parameters to the request. This action does not require a JWT Authorization token. Depending on the type of device, the body of the request is a JSON Object containing the specific attributes that compose the measurement sent by the device. The body, therefore, is a JSON object that changes based on the device performing the request. Response code: • Successful operation uses 200 OK: indicates that the data associated with the device was successfully send to the platform. and store into the platform. • Errors use a non-2xx and (optionally) an error payload. See subsection on "Error Responses" for more details.	
Request Path	/devices/send_data{?deviceId,deviceToken}	
Request Headers	Content-Type: application/json	
Request Parameters	deviceId (required, string) - Device Id deviceToken (required, string) - Device Token	

Request Body	`{`
	`     "timestamp": "2023-01-23T18:03:00Z",`
	`     "status": "open"`
	`}`	

Response Codes	200	400 401 403 404 500

Response Body	`{}`	`{`
		`  "timestamp": "2023-01-`
		`20T17:33:39.710Z",`
		`   "message": "string",`
		`   "status": 0,`
		`   "error": "string"`
		`  }`

Table 20: Send Device Data API description

The project has received funding from the European Union H2020 Programme under grant agreement n° 101016453. The Japanese consortium received funding by the Japanese Ministry of Internal Affairs and Communication (MIC).

<table>
<tr><th colspan="2">Send Message From Google Assistant</th></tr>
<tr><td>Request Type</td><td>POST</td></tr>
<tr><td>Request Description</td><td>This action is the webhook API used by the extension implemented in Google Assistant, through the Google Actions Console, to allow the user to make requests through the Google Assistant (integrated within the mobile phone or in Google devices such as the Google Nest) which are forwarded to the Dialogue Manager of the e-VITA platform, which is the brain customized for the platform.

The user can then request information such as the daily news, weather forecasts, but above all information related to the health and well-being.

If the processing is successful, the action returns as output the response to be given to the user coming from the Dialogue Manager of the e-VITA platform; It will be forwarded back and returned to the user via the Google Assistant.

Response code:
• Successful operation uses 200 OK: indicates that the communication between the Google Assistant and the Dialogue Manager was successful.
• Errors use a non-2xx and (optionally) an error payload. See subsection on "Error Responses" for more details.</td></tr>
<tr><td>Request Path</td><td>/devices/google_webhook</td></tr>
<tr><td>Request Headers</td><td>Authorization: Bearer JWT</td></tr>
<tr><td>Request Parameters</td><td>---</td></tr>
<tr><td>Request Body</td><td><pre>{
  "handler": {
    "name": "query"
  },
  "intent": {
    "name": "matchAny",
    "params": {
      "any": {
        "original": "tell me about hypertension",
        "resolved": "tell me about hypertension"
      }
    },
    "query": "tell me about hypertension"
  },
  "scene": {
    "name": "Call_RASA_Scene",
    "slots": {},
    "next": {
      "name": "Call_RASA_Scene"
    },
    "slotFillingStatus": "UNSPECIFIED"
  },</pre></td></tr>
</table>

The project has received funding from the European Union H2020 Programme under grant agreement n° 101016453. The Japanese consortium received funding by the Japanese Ministry of Internal Affairs and Communication (MIC).

	"session": {
	"id": "ABwppHHMmSYVDCDDZDMh4zlafkwSaKgZK1Ywkq9ctA_e5j5e2lawxcntPw_xrC1ky5X09xcF33zGU0s",
	"params": {
	"AccountLinkingSlot": "LINKED"
	},
	"typeOverrides": [],
	"languageCode": ""
	},
	"user": {
	"locale": "it-IT",
	"params": {},
	"accountLinkingStatus": "LINKED",
	"verificationStatus": "VERIFIED",
	"lastSeenTime": "2023-01-22T10:05:19Z",
	"packageEntitlements": [],
	"gaiamint": "",
	"permissions": []
	},
	"device": {
	"capabilities": [
	"SPEECH",
	"RICH_RESPONSE",
	"LONG_FORM_AUDIO"
	],
	"timeZone": {
	"id": "Europe/Rome",
	"version": ""
	}
	},
	"home": {
	"params": {}
	}
	}	

Response Codes	200	400 401 403 404 500

Response Body	{
	"prompt": {
	"override": false,
	"firstSimple": {
	"speech": "Hypertension has the symptoms: Hypertension is called a silent killer. Most people with hypertension are unaware of the problem because it may have no warning signs or symptoms. For this reason, it is essential to measure the blood pressure regularly. However, when symptoms do occur, they can include early morning headaches, Nosebleeds, Irregular heartbeat, Vision problem, Chest pain,
		{
		"timestamp": "2023-01-20T17:33:39.710Z",
		"message": "string",
		"status": 0,
		"error": "string"
		}

The project has received funding from the European Union H2020 Programme under grant agreement n° 101016453. The Japanese consortium received funding by the Japanese Ministry of Internal Affairs and Communication (MIC).

	Breathing difficulty, and buzzing in the ears. Severe hypertension can cause fatigue, nausea, vomiting, confusion, anxiety, chest pain, and muscle tremors.. Did that help? ", "text": "Hypertension has the symptoms: Hypertension is called a silent killer. Most people with hypertension are unaware of the problem because it may have no warning signs or symptoms. For this reason, it is essential to measure the blood pressure regularly. However, when symptoms do occur, they can include early morning headaches, Nosebleeds, Irregular heartbeat, Vision problem, Chest pain, Breathing difficulty, and buzzing in the ears. Severe hypertension can cause fatigue, nausea, vomiting, confusion, anxiety, chest pain, and muscle tremors.. Did that help? " } } }	

Table 21: Send Message From Google Assistant API description

4 e-VITA Manager API – Services controller

Service APIs represents the way to obtain and manage all the information relating to the external cloud services that the user has accessed. The user is identified by the JWT authorization token passed within the request header.

These cloud services are connected to specific types of devices which are configured to store the detected measurements within them.

The services which have so far been configured within the e-VITA platform to allow the user to log in and obtain the measurements of the devices, are as follows:

- *Netatmo Cloud Service*, connected to the Netatmo Air quality sensor.
- *Strapi Cloud Service* connected to the Neu wearable device.
- *OURA Cloud Service* connected to the Oura ring wearable device.

The e-VITA platform, for this particular typology of device, retrieves the measurements from the cloud services with a certain configurable frequency, to store them inside the platform.

Retrieve Services		
Request Type	GET	
Request Description	This request does not have any parameters. It allows to obtain all the cloud services to which the user, corresponding to the JWT authorization token passed within the request header, logged in via the e-VITA platform. The information obtained as a response is in JSON format; it is a JSON Array containing the name of the cloud service and the relative access token to the service, obtained after authentication by the user. Response code: • Successful operation uses 200 OK: indicates that information about cloud services related to the user were successfully obtained. • Errors use a non-2xx and (optionally) an error payload. See subsection on "Error Responses" for more details.	
Request Path	/services	
Request Headers	Authorization: Bearer JWT	
Request Parameters	---	
Request Body	---	
Response Codes	200	400 401 403 404

The project has received funding from the European Union H2020 Programme under grant agreement n° 101016453. The Japanese consortium received funding by the Japanese Ministry of Internal Affairs and Communication (MIC).

		500

Response Body	[
	  {
	    "type": "NETATMO",
	    "token": {}
	  }
	]
		{
		  "timestamp": "2023-01-20T17:33:39.710Z",
		   "message": "string",
		   "status": 0,
		   "error": "string"
		  }

Table 22: Retrieve Services API description

Create a New Service	
Request Type	POST
Request Description	This action creates a new service within the e-VITA platform, which represents the cloud service accessed by the user. After logging in, this action is called; it creates a JSON Object representing the access to the cloud service. The returned object stores the name of the service and the access token that was returned by the service itself after logging in. The map field of the token contains a JSON Object which is the token returned by the external service after user login. It will have a different structure based on the specific service. Response code: • Successful operation uses 200 OK: indicates that information about cloud service related to the user was successfully created. • Errors use a non-2xx and (optionally) an error payload. See subsection on "Error Responses" for more details.
Request Path	/services
Request Headers	Authorization: Bearer JWT
Request Parameters	---

Request Body	{
	  "type": "NETATMO",
	  "token": {
	    "map": {
	      "additionalProp1": {},
	      "additionalProp2": {},
	      "additionalProp3": {}
	    }
	  }
	}

Response Codes	200	400 401 403 404 500
Response Body	`[ { "type": "NETATMO", "token": {} } ]`	`{ "timestamp": "2023-01-20T17:33:39.710Z", "message": "string", "status": 0, "error": "string" }`

Table 23: Create a New Service API description

Retrieve Service By Type		
Request Type	GET	
Request Description	This request allows to obtain all the cloud services to which the user, corresponding to the JWT authorization token passed within the request header, logged in via the e-VITA platform. The information obtained as a response is in JSON format; it is a JSON Array containing the name of the cloud service and the relative access token to the service, obtained after authentication by the user. In the platform it is obviously possible to have only one cloud service for each type. It takes in input one parameter: • type: Typology of the cloud service; it is a string that can only have one of three predefined values. Response code: • Successful operation uses 200 OK: indicates that information about cloud service related to the user was successfully obtained. • Errors use a non-2xx and (optionally) an error payload. See subsection on "Error Responses" for more details.	
Request Path	/services/{type}	
Request Headers	Authorization: Bearer JWT	
Request Parameters	type (required, string) - Typology of the cloud service	
Request Body	- - -	
Response Codes	200	400 401 403 404 500

Ministry of Internal Affairs and Communications, JAPAN

The project has received funding from the European Union H2020 Programme under grant agreement n° 101016453. The Japanese consortium received funding by the Japanese Ministry of Internal Affairs and Communication (MIC).

Response Body	{
	  "type": "NETATMO",
	  "token": {
	    "map": {
	      "additionalProp1": {},
	      "additionalProp2": {},
	      "additionalProp3": {}
	    }
	  }
		{
		  "timestamp": "2023-01-20T17:33:39.710Z",
		   "message": "string",
		   "status": 0,
		   "error": "string"
		  }

Table 24: Retrieve Service By Type API description

Delete Service By Type		
Request Type	DELETE	
Request Description	This action allows to remove the cloud service with the type specified as a parameter, from the e-VITA platform. Since the cloud service is connected to the relative group of devices, using the *keepData* parameter, it is possible to specify whether, in addition to deleting the cloud service, the user also want to delete all the connected devices from the platform. If the user wants to keep the devices related to the cloud service, they will continue to appear as user-registered devices, but them will be disabled. It takes in input two parameters: • type: Typology of the cloud service; it is a string that can only have one of three predefined values. • keepData: Parameter that defines whether, after deleting the service, the devices related to the cloud service to delete must in turn be removed or not. Response code: • Successful operation uses 200 OK: indicates that information about cloud service related to the user (and the related devices, if specified by the user) was successfully deleted. • Errors use a non-2xx and (optionally) an error payload. See subsection on "Error Responses" for more details.	
Request Path	/services/{type}{?keepData}	
Request Headers	Authorization: Bearer JWT	
Request Parameters	type (required, string) - Typology of the cloud service. keepData(required, boolean) - Parameter that defines whether, after deleting the service, the devices related to the cloud service to delete must in turn be removed or not.	
Request Body	---	
Response Codes	200	400 401 403 404 500

The project has received funding from the European Union H2020 Programme under grant agreement n° 101016453. The Japanese consortium received funding by the Japanese Ministry of Internal Affairs and Communication (MIC).

Response Body	{}	{
		  "timestamp": "2023-01-20T17:33:39.710Z",
		   "message": "string",
		   "status": 0,
		   "error": "string"
		  }

Table 25: Delete Service By Type API description

5 e-VITA Manager API – Researchers controller

Researchers APIs represents the way to obtain historical data of all devices belonging to users registered on the platform, by researchers. They are users who are assigned a particular role, thanks to which they can access the historical data of all the devices stored within the e-VITA platform, for analysis and research purposes.

Retrieve Historical Data	
Request Type	GET
Request Description	This request allows to obtain all the information of the user corresponding to the JWT authorization token passed within the request header. The information obtained as a response is a string containing all the required historical data. This output, from the dashboard of the e-VITA platform, can also be downloaded by the user as a CSV file. It takes in input the following parameters: • startDate: it is a string in DateTime format which represents the starting date from which to show the measurements detected by the device. • endDate: it is a string in DateTime format which represents the end date until which to view the measurements taken by the device. • country: the language expressed with the reference code of the country (de, it, en, fr or jp). • group: Device group to consider; it can be a sensor or a coaching device (robot). • type: Typology of the device to consider for obtaining the historical data. it is possible to specify only one type of devices or several. • outputDateFormat: Parameter that specifies the format that the timestamp associated with each measurement must have (24h or 12h format). Default is 24h format. Response code: • Successful operation uses 200 OK: indicates that devices' measurements were successfully obtained. • Errors use a non-2xx and (optionally) an error payload. See subsection on "Error Responses" for more details.
Request Path	/devices/historical{?startDate, endDate, country, group, type, outputDateFormat}
Request Headers	Authorization: Bearer JWT
Request Parameters	startDate (required, datetime) - Starting date from which to show the measurements detected by the device

Ministry of Internal Affairs and Communications, JAPAN

The project has received funding from the European Union H2020 Programme under grant agreement n° 101016453. The Japanese consortium received funding by the Japanese Ministry of Internal Affairs and Communication (MIC).

	endDate (datetime) - End date until which to view the measurements taken by the device. country (string) - Reference country to consider. Possible values: it, en, fr, de, jp. group (boolean) - Typology of the device group to consider. Possible values: SENSOR, COACHING_DEVICE. type (array of string) - Typology of the devices to consider. Possible values: NETATMO_AIR_QUALITY, NEU_TRAINING, NEU_BRAIN, OURARING_SLEEP, OURARING_ACTIVITY, OURARING_READINESS, GATEBOX, NAO_ROBOT, DELTADORE_MOTION_SENSOR, DELTADORE_INTRUSION_SENSOR, ENOCEAN_TEMPERATURE_SENSOR, ENOCEAN_DOOR_SENSOR, ENOCEAN_MOTION_SENSOR, ANDROID_ROBOT, DARUMA, CELESTE, DASHBOARD, GOOGLE_ASSISTANT. outputDateFormat (string) - Format that the timestamp of each measurement must have. Possible values: dd.MM.yyyy; HH:mm:ss, dd.MM.yyyy; hh:mm:ss	
Request Body	---	
Response Codes	200	400 401 403 404 500

Response Body	`string`	`{`
		`  "timestamp": "2023-01-20T17:33:39.710Z",`
		`  "message": "string",`
		`  "status": 0,`
		`  "error": "string"`
		`}`

Table 26: Retrieve Historical Data API description

6 e-VITA Manager API – Clients controller

Clients APIs represents the way to obtain and manage different types of information of the user registered to the e-VITA platform, from an external application (Client) that has the authorization to obtain such information.

Retrieve User by ID		
Request Type	GET	
Request Description	This request allows to obtain all the information of the user corresponding to the identifier passed as a parameter to the request. The request is made by an external application whose identity and authorization are defined by the JWT authorization token passed within the request header. The information obtained as a response is in JSON format; it is a JSON Object which contains user information. It takes in input one parameter: • **userId**: unique identifier of the user within e-VITA platform. Response code: • Successful operation uses 200 OK: indicates that the user's information was successfully obtained. • Errors use a non-2xx and (optionally) an error payload. See subsection on "Error Responses" for more details.	
Request Path	/clients/users{?userId}	
Request Headers	Authorization: Bearer JWT	
Request Parameters	**userId** (required, string) - Unique identifier of the user within e-VITA platform.	
Request Body	---	
Response Codes	200	400 401 403 404 500

Response Body	{
	  "id": "string",
	  "country": "string",
	  "city": "string",
	  "language": "string",
	  "username": "string",
	  "gender": "string",
	  "telegramUsername": "string"
	}
		{
		  "timestamp": "2023-01-20T17:33:39.710Z",
		   "message": "string",
		   "status": 0,
		   "error": "string"
		  }

Table 27: Retrieve User by ID API description

The project has received funding from the European Union H2020 Programme under grant agreement n° 101016453. The Japanese consortium received funding by the Japanese Ministry of Internal Affairs and Communication (MIC).

Retrieve User Emotions by ID		
Request Type	GET	
Request Description	This request allows to obtain all the information of the user corresponding to the identifier passed as a parameter to the request. The user information obtained in this case are the emotions related to audio files previously sent by the user's devices, if any. Emotions are obtained thanks to the interaction of the e-VITA platform with the Emotion Detection external component, which associates a precise emotion (such as happiness, sadness, etc.) to the audio files sent by the devices. The request is made by an external application whose identity and authorization are defined by the JWT authorization token passed within the request header. The information obtained as a response is in JSON format; it is a JSON Array where each element refers to an audio file with the respective emotion detected. It takes in input one parameter: • **userId**: unique identifier of the user within e-VITA platform. Response code: • Successful operation uses 200 OK: indicates that the user's information was successfully obtained. • Errors use a non-2xx and (optionally) an error payload. See subsection on "Error Responses" for more details.	
Request Path	/clients/users/{userId}/emotions	
Request Headers	Authorization: Bearer JWT	
Request Parameters	**userId** (required, string) - Unique identifier of the user within e-VITA platform.	
Request Body	---	
Response Codes	200	400 401 403 404 500

Response Body	`[`
	`  {`
	`    "id": "string",`
	`    "userId": "string",`
	`    "deviceId": "string",`
	`    "language": "string",`
	`    "file": "string",`
	`    "detectedEmotion":`
	`"string",`
	`    "createdAt": "2023-01-`
	`24T10:23:45.854Z"`
		`{`
		`  "timestamp": "2023-01-`
		`20T17:33:39.710Z",`
		`   "message": "string",`
		`   "status": 0,`
		`   "error": "string"`
		`  }`

The project has received funding from the European Union H2020 Programme under grant agreement n° 101016453. The Japanese consortium received funding by the Japanese Ministry of Internal Affairs and Communication (MIC).

	}]	

Table 28: Retrieve User Emotions by ID API description

Retrieve User Devices by ID		
Request Type	GET	
Request Description	This request allows to obtain all the information of the user corresponding to the identifier passed as a parameter to the request. The user information obtained in this case are the devices registered by the specific user. The request is made by an external application whose identity and authorization are defined by the JWT authorization token passed within the request header. The information obtained as a response is in JSON format; it is a JSON Array where each element refers to a single device. It takes in input one parameter: • userId: unique identifier of the user within e-VITA platform. Response code: • Successful operation uses 200 OK: indicates that the user's information was successfully obtained. • Errors use a non-2xx and (optionally) an error payload. See subsection on "Error Responses" for more details.	
Request Path	`/clients/users/{userId}/devices`	
Request Headers	`Authorization: Bearer JWT`	
Request Parameters	**`userId`** `(required, string) - Unique identifier of the user within e-VITA platform.`	
Request Body	---	
Response Codes	200	400 401 403 404 500

Response Body	`[`
	`  {`
	`    "evitaDeviceId":`
	`"string",`
	`    "name": "string",`
	`    "type":`
	`"NETATMO_AIR_QUALITY",`
	`    "attributes": [`
	`      {`
		`{`
		`  "timestamp": "2023-01-`
		`20T17:33:39.710Z",`
		`   "message": "string",`
		`   "status": 0,`
		`   "error": "string"`
		`  }`

The project has received funding from the European Union H2020 Programme under grant agreement n° 101016453. The Japanese consortium received funding by the Japanese Ministry of Internal Affairs and Communication (MIC).

	"attributeType": "string", "attributeName": "string" }], "lastMeasureSent": "2023-01-24T10:28:58.242Z", "evitaUserId": "string", "enabled": true }]	

Table 29: Retrieve User Devices by ID API description

Retrieve User Device by ID		
Request Type	GET	
Request Description	This request allows to obtain all the information of the user corresponding to the identifier passed as a parameter to the request. The user information obtained in this case is the devices registered by the specific user, which has the ID specified as a parameter by the user. The request is made by an external application whose identity and authorization are defined by the JWT authorization token passed within the request header. The information obtained as a response is in JSON format; it is a JSON Object referred to the specific device if present. It takes in input two parameters: • **userId**: unique identifier of the user within e-VITA platform. • **deviceId**: unique identifier of the device within e-VITA platform. Response code: • Successful operation uses 200 OK: indicates that the user's device was successfully obtained. • Errors use a non-2xx and (optionally) an error payload. See subsection on "Error Responses" for more details.	
Request Path	/clients/users/{userId}/devices/{deviceId}	
Request Headers	Authorization: Bearer JWT	
Request Parameters	**userId** (required, string) - Unique identifier of the user within e-VITA platform. **deviceId** (required, string) - Unique identifier of the device within e-VITA platform.	
Request Body	- - -	
Response Codes	200	400 401 403 404

The project has received funding from the European Union H2020 Programme under grant agreement n° 101016453. The Japanese consortium received funding by the Japanese Ministry of Internal Affairs and Communication (MIC).

		500

Response Body	{
	"evitaDeviceId": "string",
	"name": "string",
	"type": "NETATMO_AIR_QUALITY",
	"attributes": [
	{
	"attributeType": "string",
	"attributeName": "string"
	}
	],
	"lastMeasureSent": "2023-01-24T10:30:45.292Z",
	"evitaUserId": "string",
	"enabled": true
	}
		{
		"timestamp": "2023-01-20T17:33:39.710Z",
		"message": "string",
		"status": 0,
		"error": "string"
		}

Table 30: Retrieve User Device by ID API description

Retrieve User Device Historical Data by ID	
Request Type	GET
Request Description	This request allows to obtain all the information of the user corresponding to the identifier passed as a parameter to the request. The user information obtained in this case is about the device which has the ID specified as a parameter; it allows to obtain all the measurements collected by the specific device, belonging to the user. The request is made by an external application whose identity and authorization are defined by the JWT authorization token passed within the request header. The information obtained as a response is in string format containing the past measurements detected by the device and stored within the platform. The format of the string depends on the attributes that the specific device has, but in general it presents the list of device's attributes and the value they have assumed, in relation to the date on which the measurement was taken. It takes in input the following parameters: • userId: unique identifier of the user within e-VITA platform. • deviceId: unique identifier of the device within e-VITA platform. • startDate: it is a string in *DateTime* format which represents the starting date from which to show the measurements detected by the device. • endDate: it is a string in *DateTime* format which represents the end date until which to view the measurements taken by the device. Response code: • Successful operation uses 200 OK: indicates that the device's historical data were successfully obtained.

The project has received funding from the European Union H2020 Programme under grant agreement n° 101016453. The Japanese consortium received funding by the Japanese Ministry of Internal Affairs and Communication (MIC).

	• Errors use a non-2xx and (optionally) an error payload. See subsection on "Error Responses" for more details.	
Request Path	/clients/users/{userId}/devices/{deviceId}/historical{?startDate,endDate}	
Request Headers	Authorization: Bearer JWT	
Request Parameters	**userId** (required, string) - Unique identifier of the user within e-VITA platform. **deviceId** (required, string) - Unique identifier of the device within e-VITA platform. **startDate** (required, dateTime) - Starting date from which to show the measurements detected by the device. **endDate** (dateTime) - End date until which to view the measurements taken by the device	
Request Body	---	
Response Codes	200	400 401 403 404 500
Response Body	`string`	`{ "timestamp": "2023-01-20T17:33:39.710Z", "message": "string", "status": 0, "error": "string" }`

Table 31: Retrieve User Device Historical Data by ID API description

The project has received funding from the European Union H2020 Programme under grant agreement n° 101016453. The Japanese consortium received funding by the Japanese Ministry of Internal Affairs and Communication (MIC).

Retrieve User by Telegram ID		
Request Type	GET	
Request Description	This request allows to obtain all the information of the user corresponding to the identifier passed as a parameter to the request. The identifier passed as a parameter, in this case, is the ID of the user registered within the Telegram application. From it, the information of the corresponding user within the e-VITA platform is obtained. The request is made by an external application whose identity and authorization are defined by the JWT authorization token passed within the request header. The information obtained as a response is in JSON format; it is a JSON Object which contains user information. It takes in input one parameter: • **telegramId**: unique identifier of the user within Telegram application. Response code: • Successful operation uses 200 OK: indicates that the user's information was successfully obtained. • Errors use a non-2xx and (optionally) an error payload. See subsection on "Error Responses" for more details.	
Request Path	/clients/users/telegram{?telegramId}	
Request Headers	Authorization: Bearer JWT	
Request Parameters	**telegramId** (required, string) - Unique identifier of the user within Telegram application.	
Request Body	- - -	
Response Codes	200	400 401 403 404 500

Response Body	`{`
	`  "id": "string",`
	`  "country": "string",`
	`  "city": "string",`
	`  "language": "string",`
	`  "username": "string",`
	`  "gender": "string",`
	`  "telegramUsername":`
	`"string"`
	`}`
		`{`
		`  "timestamp": "2023-01-`
		`20T17:33:39.710Z",`
		`    "message": "string",`
		`    "status": 0,`
		`    "error": "string"`
		`  }`

Table 32: Retrieve User by Telegram ID API description

7 Conclusion

This document described in detail the e-VITA APIs, the main interfaces to access to the e-VITA platform functionalities. Thanks to these HTTP/Rest based interfaces, external system and applications can access to e-VITA functionalities and data, under specific authorisation in compliance to privacy regulation. This deliverable is a document version of the online specification of the e-VITA API (Apiary, 2022), that has to be considered the most relevant reference for the interface description, because it will be evolved during the next phase of the project to support new functionalities of the platform.

Ministry of Internal Affairs and Communications, JAPAN

The project has received funding from the European Union H2020 Programme under grant agreement n° 101016453. The Japanese consortium received funding by the Japanese Ministry of Internal Affairs and Communication (MIC).

References

Apiary, 2022. *e-VITA Manager APIs.* [Online]
Available at: https://evitaproject.docs.apiary.io/#

Engineering Ingegneria Informatica Spa, 2022. *Digital Enabler.* [Online]
Available at: (https://www.eng.it/our-platforms-solutions/digital-enabler) (Engineering Ingegneria Informatica Spa

e-VITA, 2022. *D7.4 - e-VITA Platform Architecture – Final Version ,* s.l.: s.n.

e-VITA, 2022. *D7.7 - e-VITA Data Privacy and Security Modules – First release,* s.l.: s.n.

OAuth, 2022. *OAuth 2.0 Client Credentials Grant.* [Online]
Available at: https://oauth.net/2/grant-types/client-credentials/

OAuth, 2022. *OAuth 2.0 Password Grant.* [Online]
Available at: https://oauth.net/2/grant-types/password/

OpenAPI Swagger, 2022. *e-VITA Manager API - EU instance.* [Online]
Available at: https://manager.evita.digital-enabler.eng.it/api/swagger-ui/index.html?configUrl=/api/v3/api-docs/swagger-config

OpenAPI Swagger, 2022. *e-VITA Manager API - JP instance.* [Online]
Available at: https://manager.apps.cluster.e-vita.site/api/swagger-ui/index.html?configUrl=/api/v3/api-docs/swagger-config

Ministry of Internal Affairs and Communications, JAPAN

The project has received funding from the European Union H2020 Programme under grant agreement n° 101016453. The Japanese consortium received funding by the Japanese Ministry of Internal Affairs and Communication (MIC).