

***API AGORA LABS***

**Developer guide**

**if you have any question send an email to:**

[**support@agoraplus.com**](mailto:support@agoraplus.com)

Document revision history

| *Version* | Date | Changes | Editor |
| --- | --- | --- | --- |
| *1.0* | 20/01/2019 | First version | DèLonghi |
| *1.1* | 29/06/2020 | Added chapter [4 Risorse](https://docs.google.com/document/d/13pk4kulgt4PPRhTzwxzDnHmqFyGaCIxGLgXSxFJ21KE/edit#heading=h.mbdft39ydhcu) | DèLonghi |
| *1.2* | 01/09/2020 | Updated nomenclature | DèLonghi |
| *1.3* | 14/09/2020 | Added exchange process | DèLonghi |
| *1.4* | 23/09/2020 | Removed “IRISDefault” field in the method “createRepair” | DèLonghi |
| *1.5* | 01/10/2020 | Added paragraphs:   * [1.1.5 Closing a Repair file without warranty intervention](#_myhcrufpebrn) * [1.1.6 Closing a Repair file created by mistake](#_l0we6c1yhp1g) * [1.2.4.1 Summary table of the Warranty report status returned by the API](#_iecyak5in4l3)   Updated paragraphs:   * [1.3 Replacement management](#_doe7umebyhom) * [1.5 Intervention status summary table](#_tgou7fiyy3q9)   - Updated description “Family” field in chapter [3.2.1.2 Body request](#_9582on166onx)  - Updated “searchproductbyref” method name in chapter [4.5 Bill of material - Step 1](#_e79gdlcevyg4) | DèLonghi |
| *1.6* | *18/03/2021* | Added mandatory “DefaultCode” field in the methods “createRepair” and “updateRepair” | DèLonghi |
| *1.7* | 22/04/2021 | Added paragraphs:   * [3.3.1.2 Note](#_v67pr9p7gqf8) * [3.3.2.2 Note](#_zi8t4wkcszhf) | DèLonghi |
| *1.8* | 19/11/2021 | Removed value "3 (Exchange request under warranty)" in the "InterventionType" field present in the "createRepair" method | DèLonghi |
| *1.9* | 01/07/2022 | Updated the reference to Manufacturer ID from constant to variable  Updated paragraphs:   * [1.1.5 Closing a Repair file without warranty intervention](#_myhcrufpebrn) * [1.1.6 Closing a Repair file created by mistake](#_l0we6c1yhp1g) * [1.3.4 Evaluating an Exchange request](#_rna6jfxvobvi) * [3.2.1.2 Body request](#_9582on166onx) * [3.2.2.1 Body request](#_txbk6lrgjiuz) * [3.3.1.1 Body request](#_qhqn5hj5b4dc) * [3.3.3 GET – Warranty reports list](#_o1awuld5qba9) * [4.1 Products](#_amqbjs5x89gb) * [4.2 Spare Parts](#_4iv9gc7wm5s4) * [4.3 Labor rates](#_uhi6kcbtfmo7) * [4.4 Defect codes](#_hcq3s2k9ssuw) * [4.5 Bill of material - Step 1](#_e79gdlcevyg4)   Added paragraphs:   * [4.8 Manufacturer’s ID list](#_t8hgpvl9u7lf) | DèLonghi |
| *2.0* | 24/10/2022 | Added the *Work order* data flow and its integration with the *Repair file.*  The meaning of the MissionOrder field has been updated  Added paragraphs:   * [1.0 Work order management (optional)](#_lwr5p5q43ea9)   + [1.0.1 Work order work cycle analysis](#_oeuyt5vjzmyo)     - [1.0.1.1 Created](#_x0sy6kt2xasa)     - [1.0.1.2 Assigned](#_obkdxal45nhm)     - [1.0.1.3 Accepted](#_2p5fd4h69mbj)     - [1.0.1.4 Refused](#_l06abac6na2o)   + [1.0.2 Work order cycle analysis in Agora](#_jboa4oxcms1r)     - [1.0.2.1 New](#_2p25wb2htnsq)     - [1.0.2.2 Assigned](#_u44un3n86ukz)     - [1.0.2.3 Accepted](#_z33p5640jctm)     - [1.0.2.4 Refused](#_jcu163m42xgx)   + [1.0.3 Accepting a work order](#_cvp6xecmiu64)   + [1.0.4 Refusing a work order](#_129am54cadyv)   + [3.2.3 GET – Downloading Repair file](#_d06z5gwhmdr)     - [3.2.3.1 ResponseData structure](#_l7yixvx4dybm) * [4.9 Language LCID (Locale) Code](#_fqk8e37u83tb)   Updated paragraphs:   * [1.1.3 Creating a Repair file](#_7dx55ohr1jv5) * [1.1.7 Downloading a Repair file](#_9qv1945vqi6q) * [3.3.1.1 Body request](#_qhqn5hj5b4dc) * [3.3.2.1 Body Request](#_l14mnqwhhkun) | DèLonghi |

Table of contents

[**1 Warranty repair intervention process**](#_3mo7zwphnogt) **10**

[1.0 Work order management (optional)](#_lwr5p5q43ea9) 11

[1.0.1 Work order work cycle analysis](#_oeuyt5vjzmyo) 11

[1.0.1.1 Created](#_x0sy6kt2xasa) 11

[1.0.1.2 Assigned](#_obkdxal45nhm) 11

[1.0.1.3 Accepted](#_2p5fd4h69mbj) 11

[1.0.1.4 Refused](#_l06abac6na2o) 11

[1.0.2 Work order cycle analysis in Agora](#_jboa4oxcms1r) 12

[1.0.2.1 New](#_2p25wb2htnsq) 12

[1.0.2.2 Assigned](#_u44un3n86ukz) 12

[1.0.2.3 Accepted](#_z33p5640jctm) 12

[1.0.2.4 Refused](#_jcu163m42xgx) 12

[1.0.3 Accepting a work order](#_cvp6xecmiu64) 12

[1.0.4 Refusing a work order](#_129am54cadyv) 13

[1.1 Repair management](#_7anrzvoo3gtt) 14

[1.1.1 Repair work cycle analysis](#_ofcqgr4zte4m) 14

[1.1.1.1 Opening](#_vecij7dcnca8) 14

[1.1.1.2 Acceptance](#_sqwubi2w1fpc) 15

[1.1.1.3 Inspection](#_rcxvwqxn6qmp) 15

[1.1.1.4 Spare parts ordered](#_aotaydee1ba8) 15

[1.1.1.5 Spare parts received](#_mmqhbers6h27) 15

[1.1.1.6 Completion](#_omn9byqyvguc) 15

[1.1.1.7 Product delivered](#_3ie42pbc8o8t) 15

[1.1.1.8 Replacement request](#_xrixgaggmo7b) 15

[1.1.1.9 Closure with replacement](#_1n5lipv9fyhg) 15

[1.1.2 Repair work cycle analysis in Agora](#_hhm4kpmzw7ud) 15

[1.1.2.1 Work order accepted (code 100)](#_d4162cqhi0xd) 16

[1.1.2.2 Product received (code 405)](#_6oiz9s8wioub) 16

[1.1.2.3 Diagnostic (code 406)](#_9pybykqqu37i) 16

[1.1.2.4 Spare parts ordered (code 205)](#_adgdh9uw6pcy) 16

[1.1.2.5 Spare parts received (code 201)](#_kww1dnra728x) 16

[1.1.2.6 Intervention completed (code 305)](#_q24ul8fdjg8t) 16

[1.1.2.7 Intervention closed with warranty intervention (code 601)](#_gnrgzgeku3o6) 16

[1.1.2.8 Product delivered (code 602)](#_e13ghhprqiir) 17

[1.1.2.9 Request for exchange or refund (code 302)](#_rpqobhkbzcjd) 17

[1.1.2.10 Intervention closed with exchange request (code 910)](#_6c59b0o6qg35) 17

[1.1.3 Creating a Repair file](#_7dx55ohr1jv5) 18

[1.1.4 Updating a Repair file](#_rtx296rxe91i) 18

[1.1.5 Closing a Repair file without warranty intervention](#_myhcrufpebrn) 18

[1.1.6 Closing a Repair file created by mistake](#_l0we6c1yhp1g) 19

[1.1.7 Downloading a Repair file](#_9qv1945vqi6q) 19

[1.2 Warranty intervention management](#_3k84wdmknmaa) 20

[1.2.1 Warranty report work cycle analysis](#_ir4b5fg9efug) 20

[1.2.1.1 Invalid (not transferred)](#_h53qa48yumc6) 20

[1.2.1.2 Valid (not transferred)](#_3ipjnzhk4ufx) 21

[1.2.1.3 Not conform](#_t6paw4axspfn) 21

[1.2.1.4 Closed and accepted](#_9ljcje6nzyqa) 21

[1.2.2 Creating a Warranty report](#_rlww5q36pyr5) 21

[1.2.3 Transferring a Warranty report to DèLonghi](#_7utjah8wktqk) 22

[1.2.4 Checking the status of a Warranty report transferred to DèLonghi](#_fjx4ezkdgul8) 22

[1.2.4.1 Summary table of the Warranty report status returned by the API](#_iecyak5in4l3) 22

[1.2.5 Updating a Warranty report](#_rgh3fzhgqta4) 23

[1.2.6 Delivery of a product to the consumer after recording a Warranty report](#_9ccd2v2c9d4v) 23

[1.3 Replacement management](#_doe7umebyhom) 24

[1.3.1 Exchange request work cycle analysis](#_g67l7dyytfp8) 24

[1.3.1.1 Invalid (not transmitted)](#_3tq9hyg2hnbs) 25

[1.3.1.2 Valid (not transmitted)](#_5dszgxmyt7p8) 25

[1.3.1.3 Not conform](#_4lqfoej0dcmj) 25

[1.3.1.4 Accepted](#_v4omp34ti0ev) 25

[1.3.1.5 Refused](#_13bkkl1a5gv1) 25

[1.3.2 Creating an Exchange request](#_b2blgd57d2jm) 25

[1.3.3 Transferring an Exchange request](#_z6wb8eydj0nn) 26

[1.3.4 Evaluating an Exchange request](#_rna6jfxvobvi) 26

[1.3.5 Updating an Exchange request](#_hj3ckkgehmvp) 27

[1.3.6 Delivery of the replaced product to the consumer](#_gug8mhe5akzl) 27

[1.4 Invoicing management](#_xdt2zmd3u6p6) 28

[1.4.1 Integration with Agora](#_uiq0f1ajo1a9) 28

[1.4.2 Sending reference of your invoice](#_40olg2d0l9kc) 28

[1.5 Intervention status summary table](#_tgou7fiyy3q9) 29

[**2 General Instructions**](#_q8pev4tln083) **30**

[2.1 Authentication](#_ce25t0jo1qb5) 30

[2.2 Environments](#_vziqxo1wzyb4) 30

[2.3 Service Request](#_9g6oq6he3ffs) 30

[2.3.1 API Key](#_g075kof0yw1k) 30

[2.3.2 Accept](#_cbob5xiitb19) 30

[2.3.3 Content type](#_25k42zlhq7cw) 30

[2.3.4 Example](#_bdth5ewm46bb) 30

[2.4 Service Response](#_p54yamxmnd62) 31

[2.4.1 Media Type](#_ucf44kwdukcg) 31

[2.4.2 Date / timestamp](#_4jzpj76jjrzh) 31

[2.4.3 HTTP Header - Status Codes](#_iu0m4h23sd2u) 31

[2.4.4 Body Message - Structure](#_bg28sa9bkerr) 31

[2.4.4.1 ResponseStatus](#_6shl5rq6w9xj) 31

[2.4.4.2 IsValid](#_inz8pbl7i3y9) 32

[2.4.4.3 ResponseMessage](#_amutlwl5qsfg) 32

[2.4.4.4 ResponseErrormessage](#_5ylmy8xj6ctq) 32

[2.4.4.5 ResponseData](#_gk9b6myqc66f) 32

[2.4.5 Example](#_v0h11yv3lk8p) 32

[**3 API field description**](#_vu92sh8h467n) **33**

[3.1 Technical documentation](#_c4e0236v5gz9) 33

[3.2 Repair file](#_lw34ghnzkl8f) 34

[3.2.1 POST – Creating Repair file](#_yki6zjn4fvmn) 34

[3.2.1.1 Body request](#_9582on166onx) 34

[3.2.2 POST – Updating Repair file](#_c0374okbscan) 36

[3.2.2.1 Body request](#_txbk6lrgjiuz) 36

[3.2.2.2 Note](#_25wxlm8xm55d) 37

[3.2.3 GET – Downloading Repair file](#_d06z5gwhmdr) 39

[3.2.3.1 ResponseData structure](#_l7yixvx4dybm) 39

[3.3 Warranty report](#_p4oxesn5fcgo) 42

[3.3.1 POST – Creating Warranty report](#_xayucf9wox31) 42

[3.3.1.1 Body request](#_qhqn5hj5b4dc) 42

[3.3.1.2 Note](#_v67pr9p7gqf8) 44

[3.3.2 POST – Updating Warranty report](#_4i7ojhp) 44

[3.3.2.1 Body Request](#_l14mnqwhhkun) 44

[3.3.2.2 Note](#_zi8t4wkcszhf) 44

[3.3.3 GET – Warranty reports list](#_o1awuld5qba9) 45

[3.4 Invoice](#_98o34owmq5nx) 45

[3.4.1 POST – Sending Invoice](#_2xcytpi) 45

[**4 Resources**](#_mbdft39ydhcu) **45**

[4.1 Products](#_amqbjs5x89gb) 46

[4.1.1 ResponseData - Structure](#_i3oaqyd1nhxv) 46

[4.1.2 Example](#_k8v1qz9fl7wu) 46

[4.2 Spare Parts](#_4iv9gc7wm5s4) 46

[4.2.1 ResponseData - Structure](#_140d05sogcc8) 47

[4.2.2 Example](#_wzt416y5m6qd) 47

[4.3 Labor rates](#_uhi6kcbtfmo7) 48

[4.3.1 ResponseData - Structure](#_48uuyp6v1h2r) 48

[4.3.2 Example](#_yfurzi5wt6ni) 48

[4.4 Defect codes](#_hcq3s2k9ssuw) 49

[4.4.1 ResponseData - Structure](#_7heijwupaqnl) 49

[4.4.2 Example](#_3nskwrmviqcp) 49

[4.5 Bill of material - Step 1](#_e79gdlcevyg4) 50

[4.5.1 ResponseData - Structure](#_vsm8bs8p040d) 50

[4.5.2 Example](#_jc5j8kshubd5) 50

[4.6 Bill of material - Step 2](#_8rlsrrxc3mr8) 51

[4.6.1 ResponseData - Structure](#_dew70hl2ieaw) 51

[4.6.2 Example](#_abik5r4j29we) 51

[4.7 CSV file (list of accepted warranty report)](#_w39qh8xpm0f3) 53

[4.8 Manufacturer’s ID list](#_t8hgpvl9u7lf) 55

[4.9 Language LCID (Locale) Code](#_fqk8e37u83tb) 55

## 1 Warranty repair intervention process

The warranty repair process involves the following entities:

* Contact center (optional)
* Consumer
* Service center
* Agora
* DèLonghi

and includes the following phases:

* Work order management ( ref [1.0 Work order management (optional)](#_lwr5p5q43ea9))
* Repair management ( ref. [1.1 Repair management](#_lwr5p5q43ea9) )
* Warranty intervention management ( ref. [1.2 Warranty intervention management](#_3k84wdmknmaa) ) /Replacement management ( ref. [1.3 Replacement management](#_doe7umebyhom) )
* Invoicing management ( ref. [1.4 Invoicing management](#_xdt2zmd3u6p6) )



### 1.0 Work order management (optional)

When the contact center agent receives a repair request for a product, he/she creates a work order through the CRM, specifying the consumer information and the product information. Then the contact center agent can either not assign or assign the work order to a service center which can accept or refuse it.



#### **1.0.1 Work order work cycle analysis**

The work order work cycle is composed by the following statuses:



##### 1.0.1.1 Created

This is the initial status of the work order and represents the creation of a new work order through the CRM.

##### 1.0.1.2 Assigned

This is the status following the creation of the work order, in which it is assigned to the service center.

##### 1.0.1.3 Accepted

This is the status following the assignation, in which the service center accepts the work order.

##### 1.0.1.4 Refused

This is the status following the assignation, in which the service center refuses the work order.

#### **1.0.2 Work order cycle analysis in Agora**

The *Work order* work cycle is composed by the following statuses:



##### 1.0.2.1 New

This is the initial status automatically assigned by Agora when the *Work order* is created. ( ref. [1.0.1.1 Created](#_x0sy6kt2xasa) )

##### 1.0.2.2 Assigned

This is the status assigned by Agora automatically when the *Work order* is assigned to a service center. ( ref. [1.0.1.2 Assigned](#_obkdxal45nhm) )

##### 1.0.2.3 Accepted

This is the status assigned by Agora automatically when the *Work order* is accepted by the service center. ( ref. [1.0.1.3 Accepted](#_2p5fd4h69mbj) )

##### 1.0.2.4 Refused

This is the status assigned by Agora automatically when the *Work order* is refused by the service center. ( ref. [1.0.1.4 Refused](#_l06abac6na2o) )

#### **1.0.3 Accepting a work order**

When the service center accepts to repair the product, you need to:

* accept a *Work order*
  + web API [ Not fulfilled, it must be manually accepted by the service center in Agora ]

NOTE

After the acceptance of the work order, Agora automatically creates the associated *Repair file* assigning the 100 *"Work order accepted"* status code. ( ref. [1.1.2.1 Work order accepted (code 100)](#_d4162cqhi0xd) )



MANDATORY

**When a *Repair file* is created by Agora automatically, you must not create it** ( ref. [1.1.3 Creating a Repair file](#_7dx55ohr1jv5) ) **but only update it.** ( ref. [1.1.4 Updating a Repair file](#_rtx296rxe91i) ) **and the repair created in the service center’s aftersale software MUST contain the *MissionOrder* which is the link with the Agora *Repair file***

SUGGESTION

You should download the *Repair file* from Agora in order to fill some repair fields automatically in the service center’s aftersale software. ( ref. [1.1.7 Downloading a Repair file](#_9qv1945vqi6q) ) Before downloading the *repair file* you must make the MissionOrder information typed by the service center manually in its afersale software.

#### **1.0.4 Refusing a work order**

When the service center refuses to repair the product, you need to:

* refuse a *Work order*
  + web API [ Not fulfilled, it must be manually refused by the service center in Agora ]

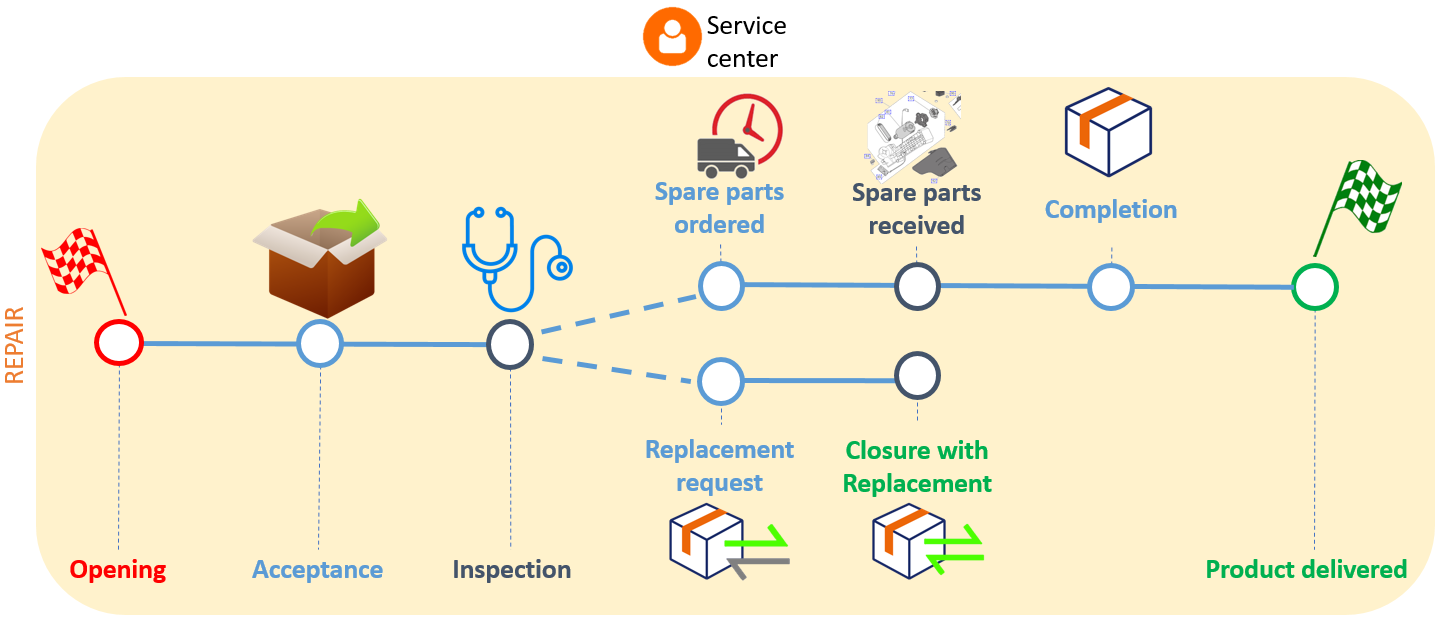
### 1.1 Repair management

When the service center receives a repair request for a product it must keep track of the related "repair status" changes in its aftersales software and transmits this information to Agora. In Agora the repair activity is represented by the entity called *Repair file* which could be created by Agora if the work order data flow has been implemented ( ref [1.0 Work order management (optional)](#_lwr5p5q43ea9) ) otherwise it must be created by the service center.



#### **1.1.1 Repair work cycle analysis**

The repair work cycle is composed by the following statuses:



##### 1.1.1.1 Opening

This is the initial status of the repair process and represents the creation of a new repair intervention in the aftersales software of the service center.

##### 1.1.1.2 Acceptance

This is the status following the opening of the repair, in which the service center receives the product to repair.

##### 1.1.1.3 Inspection

This is the status following the acceptance, in which the service center inspects the product to repair.

##### 1.1.1.4 Spare parts ordered

If spare parts are needed, this is the repair status in which the service center is waiting for the spare parts to proceed with the repair.

##### 1.1.1.5 Spare parts received

This is the status in which the service center - after receiving the spare parts ordered - starts the repair.

##### 1.1.1.6 Completion

This is the status following the repair activity and the service center has completed the repair.

##### 1.1.1.7 Product delivered

This is the final status of the process and represents the delivery of the repaired product to the consumer.

##### 1.1.1.8 Replacement request

If the repair is not possible, the service center requests a product replacement.

##### 1.1.1.9 Closure with replacement

This is the status following the replacement request in which the product replacement request has been accepted by the manufacturer.

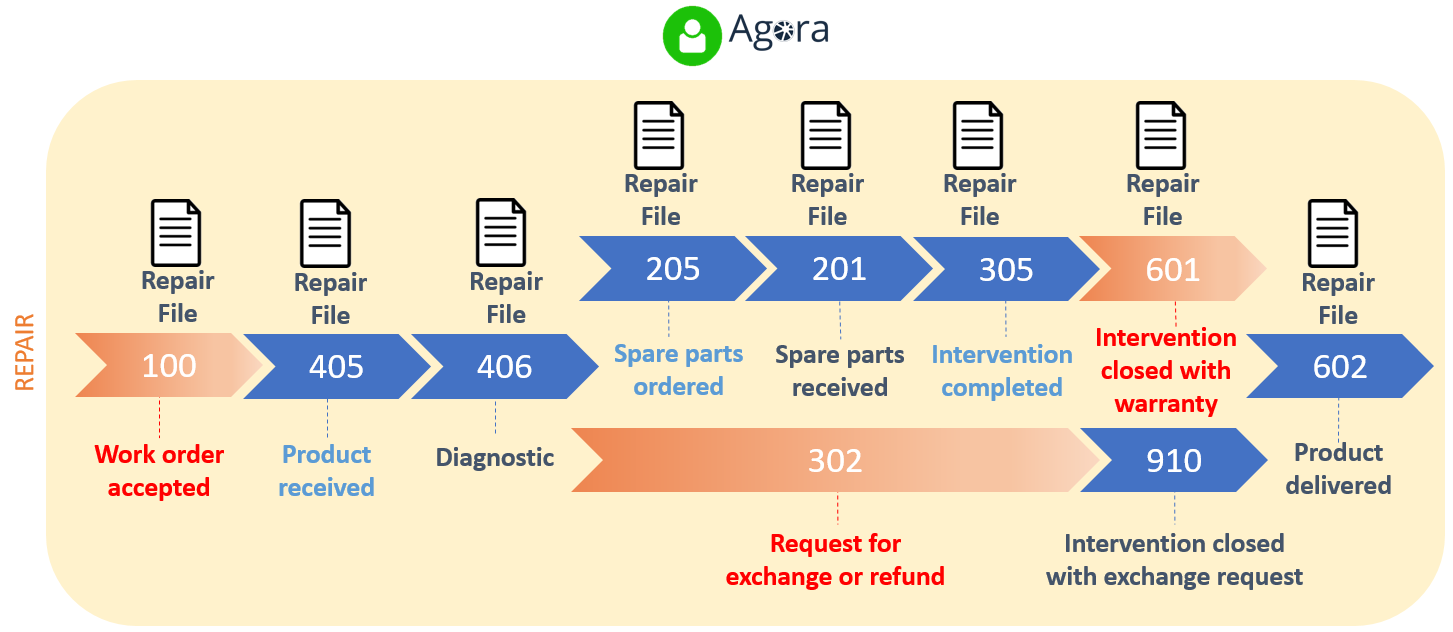
#### 

#### **1.1.2 Repair work cycle analysis in Agora**

A *Repair file* is characterized by:

* a **status** related to the repair work cycle
* a **step** that can be "In progress" - when the *Repair file* is editable - or "Closed" when the *Repair file* is no longer editable.

The *Repair file* work cycle is composed by the following statuses:



##### 1.1.2.1 Work order accepted (code 100)

This is the initial status automatically assigned by Agora when the *Repair file* is created. ( ref. [1.1.1.1 Opening](#_vecij7dcnca8) )

##### 1.1.2.2 Product received (code 405)

This is the status to send to Agora when the product is received from the consumer. ( ref. [1.1.1.2 Acceptance](#_sqwubi2w1fpc) )

##### 1.1.2.3 Diagnostic (code 406)

This is the status to send to Agora when an inspection is carried out. ( ref. [1.1.1.3 Inspection](#_rcxvwqxn6qmp) )

##### 1.1.2.4 Spare parts ordered (code 205)

This is the status to send to Agora when the service center is waiting for the spare parts. (ref. [1.1.1.4 Spare parts ordered](#_aotaydee1ba8) )

##### 1.1.2.5 Spare parts received (code 201)

This is the status to send to Agora when the service center - after receiving the spare parts ordered - can start the repair. ( ref. [1.1.1.5 Spare parts received](#_mmqhbers6h27) )

##### 1.1.2.6 Intervention completed (code 305)

This is the status to send to Agora when the repair is completed. ( ref. [1.1.1.6 Closure](#_omn9byqyvguc) )

##### 1.1.2.7 Intervention closed with warranty intervention (code 601)

This is the status assigned by Agora automatically when the *Warranty report* is created. This entity will be discussed in the paragraph related to warranty repairs. ( ref. [1.2.2 Creating a Warranty report](#_rlww5q36pyr5) )

Warning: When the repair shifts to the 601 "Intervention closed with warranty” status code, the *Repair file* is no longer editable and its step changes from "In Progress" to "Closed".

Note: The only envisaged exception allowing you to edit the *Repair file* - despite the step of the *Repair file* is "Closed" - is the change to the 602 "Product delivered” status code. The goal of this exception is for the Repair Center to be able to invoice the warranty intervention without having to wait for the product delivery and then successfully move on to the final status.

##### 1.1.2.8 Product delivered (code 602)

This is the status to send to Agora when the repaired (or replaced) product is delivered to the consumer. ( ref. [1.1.1.7 Product delivered](#_3ie42pbc8o8t) )

##### 1.1.2.9 Request for exchange or refund (code 302)

This is the status assigned by Agora automatically when the replacement request is created. This entity will be discussed in the paragraph related to the replacements. ( ref. [1.3.2 Creating an Exchange request](#_b2blgd57d2jm) )

##### 1.1.2.10 Intervention closed with exchange request (code 910)

This is the status to send when there is a replacement under warranty, in this case it is also necessary to send the closure of the repair. ( ref. [1.1.1.9 Closure with replacement](#_1n5lipv9fyhg) )

Warning: When the 910 "Intervention closed with exchange request” status code is sent, the *Repair file* is no longer editable and its step has to be changed from "In Progress" to "Closed". ( ref. [1.3.4 Evaluating an Exchange request](#_rna6jfxvobvi) )

Note: The only envisaged exception allowing you to edit the *Repair file* - despite the step of the *Repair file* is "Closed" - is the change to the 602 "Product delivered” status. The goal of this exception is for the Repair Center to be able to invoice the replacement without having to wait for the product delivery and then successfully move on to the final status.

#### **1.1.3 Creating a Repair file**

When the service center inserts a new repair into its aftersales software, you need to:

* create the *Repair file* in Agora
  + web API [ [/api/repair/createrepair](https://preprod-api.agoraplus.com/apidoc/ui/index#!/Repair/Repair_CreateRepair) ]

NOTE

When the *Repair file* is created, Agora automatically assigns the 100 "Work order accepted” status code to the repair file.

MANDATORY

**When a *Repair file* is created by Agora automatically** ( ref. [1.0 Work order management (optional)](#_lwr5p5q43ea9) )**, you must not create it but only update it.** ( ref. [1.1.4 Updating a Repair file](#_rtx296rxe91i) ) **and the repair created in the service center’s aftersale software MUST contain the *MissionOrder* which is the link with the Agora *Repair file***

SUGGESTION

You should download the *Repair file* from Agora in order to fill some Repair fields automatically in the service center’s aftersale software. ( ref. [1.1.7 Downloading a Repair file](#_9qv1945vqi6q) ) Before downloading the *repair file* you must make the MissionOrder information typed by the service center manually in its afersale software.

#### **1.1.4 Updating a Repair file**

When the repair work cycle status progress, you need to:

* update the *Repair file* in Agora by sending the desired status
  + web API [ [/api/repair/updaterepair](https://preprod-api.agoraplus.com/apidoc/ui/index#!/Repair/Repair_UpdateRepair) ]

#### **1.1.5 Closing a Repair file without warranty intervention**

When a repair must be closed without any warranty intervention, you need to:

* update the information about the Repair file in Agora by sending the 903 "Intervention out of warranty, directly invoiced to consumer” status code, adding a short note if needed, and filling in the "CLOSED" step (field StatusCode)
  + web API [ [/api/repair/updaterepair](https://preprod-api.agoraplus.com/apidoc/ui/index#!/Repair/Repair_UpdateRepair) ]

*Example*

{

"ManufacturerId": *{ManufacturerId}*,

"MissionOrder": *{MissionOrder}*,

"StateNatures": [

{

"Code": "903",

"Comment": *{Note}*

}

],

"StatusCode": "CLOSED"

}

#### **1.1.6 Closing a Repair file created by mistake**

When it is necessary to close a repair created by mistake, you need to:

* update the information of the *Repair file* in Agora by sending the 905 "Other” status code, adding a short note if needed, and filling in the "CLOSED" step (field StatusCode)
  + web API [ [/api/repair/updaterepair](https://preprod-api.agoraplus.com/apidoc/ui/index#!/Repair/Repair_UpdateRepair) ]

*Example*

{

"ManufacturerId": *{ManufacturerId}*,

"MissionOrder": *{MissionOrder}*,

"StateNatures": [

{

"Code": "905",

"Comment": *{Note}*

}

],

"StatusCode": "CLOSED"

}

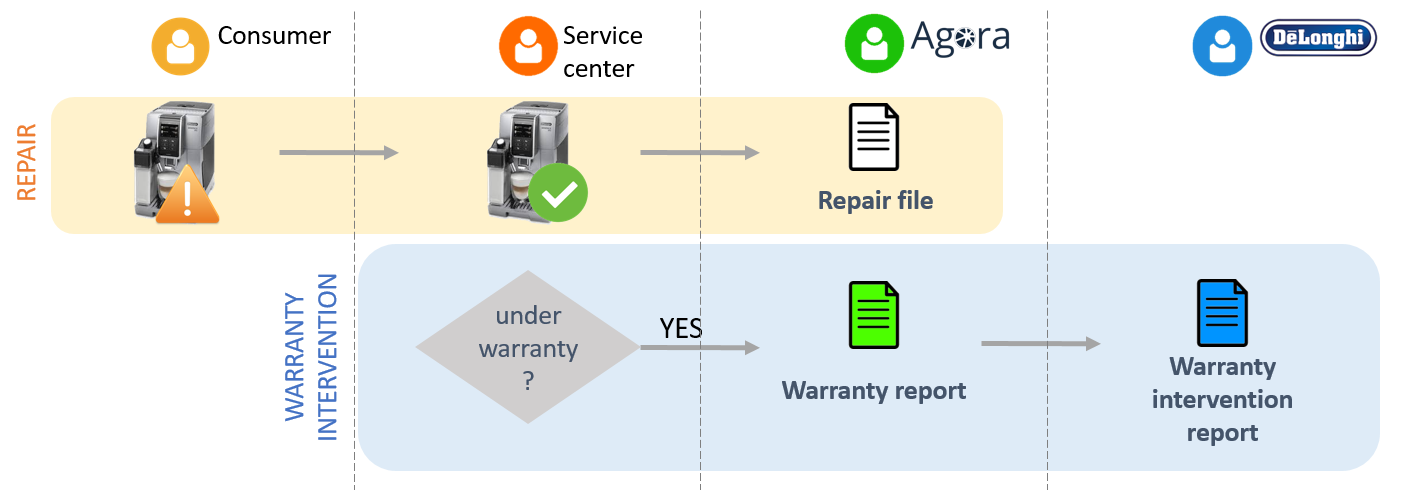
#### ***1.1.7 Downloading a Repair file***

When the service center accepts the *Work order*, you should to:

* download the *Repair file* data from Agora to fill all product and customer’s information and in the service centre’s aftersales software.
  + web API [ [/api/repair/printrepairfilehistory](https://preprod-api.agoraplus.com/apidoc/ui/index#!/Repair/Repair_PrintRepairFileHistory) ]

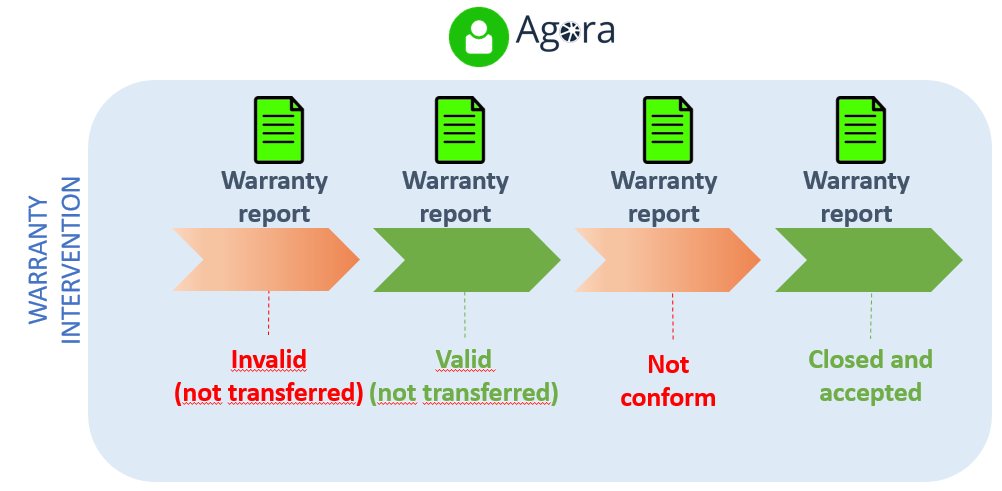
### 1.2 Warranty intervention management

At the end of the repair, if the product is under warranty, the service center transfers to DèLonghi through Agora API the Warranty intervention report containing the repair information, labor costs and spare parts used. In Agora, the warranty intervention report is represented by the entity called *Warranty report*.



#### **1.2.1 Warranty report work cycle analysis**

The *Warranty report* work cycle is composed by the following statuses:



##### 1.2.1.1 Invalid (not transferred)

This status means that the *Warranty report* has not been validated by Agora as one or more data are missing and / or incorrect. A *Warranty report* in this state must be updated to be valid for Agora and can be transferred from the service center to DèLonghi.

##### 1.2.1.2 Valid (not transferred)

This status means that the *Warranty report* has been validated by Agora. A *Warranty report* in this state can be transferred from the service center to DèLonghi.

##### 

##### 1.2.1.3 Not conform

This status means that the transferred *Warranty report* has been validated by DèLonghi as one or more data are missing and / or incorrect. Such *Warranty report* must be updated and then once more transferred to DèLonghi.

##### 1.2.1.4 Closed and accepted

This status means that the *Warranty report* has been transferred and validated by DèLonghi.

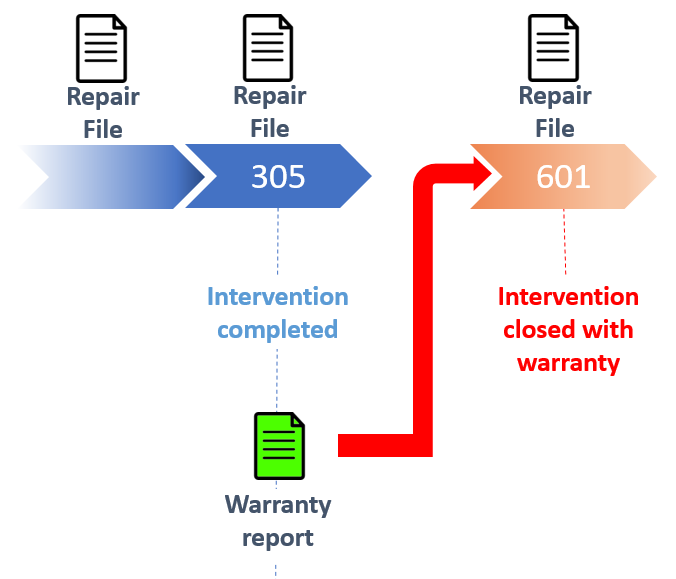
#### **1.2.2 Creating a Warranty report**

After repairing a product under warranty, the following steps must be performed in the following order:

* update the *Repair file* by sending the 305 "Intervention completed” status code
  + web API [ [/api/repair/updaterepair](https://preprod-api.agoraplus.com/apidoc/ui/index#!/Repair/Repair_UpdateRepair) ]
* create *the Warranty report*
  + web API [ [/api/repair/createwarrantyclaim](https://preprod-api.agoraplus.com/apidoc/ui/index#!/Repair/Repair_CreateWarrantyClaims) ]

NOTE

After the creation of the *Warranty report*, Agora automatically updates the status of the related *Repair file* to the 601 "Closed with warranty intervention" status code ( ref. [1.1.2.7 Intervention closed with warranty intervention (code 601)](#_gnrgzgeku3o6) )



##### 

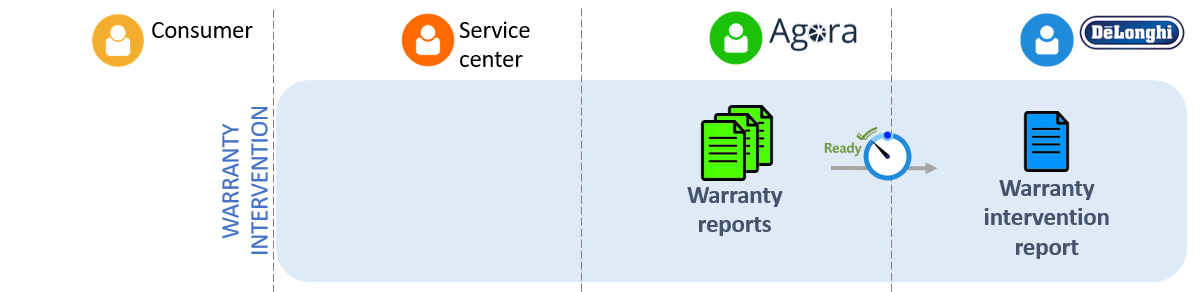
#### **1.2.3 Transferring a Warranty report to DèLonghi**

It is possible to transfer a *Warranty report* to DèLonghi, if validated by Agora, when it is created or later on. In both cases you need to:

* set to true the value of the *transferIfValid* parameter in the URL
  + web API [ [/api/repair/createwarrantyclaim](https://preprod-api.agoraplus.com/apidoc/ui/index#!/Repair/Repair_CreateWarrantyClaims) ]
  + web API [ [/api/repair/updatewarrantyclaim](https://preprod-api.agoraplus.com/apidoc/ui/index#!/Repair/Repair_UpdateWarrantyClaim) ]

NOTE

All Warranty *reports* that have this attribute set to true in Agora are sent from Agora to DèLonghi with a scheduled job, on a daily basis.



#### **1.2.4 Checking the status of a Warranty report transferred to DèLonghi**

To find out the result of the transfer to DèLonghi, you need to:

* check the status of the *Warranty report*
  + web API [ [/api/repair/getwarrantyclaims](https://preprod-api.agoraplus.com/apidoc/ui/index#!/Repair/Repair_GetWarrantyClaims) ]

##### 1.2.4.1 Summary table of the *Warranty report* status returned by the API

| State Code | State Description |
| --- | --- |
| *NOTVALID* | The *Warranty report* is not valid and consequently the transfer to Dèlonghi cannot be in progress |
| *VALID* | The *Warranty report* is valid and consequently the transfer to Dèlonghi may be in progress if the value of the transferIfValid parameter was set as true in the latest submission to Agora |
| *NOTCONFORM* | The *Warranty report* , transferred to DèLonghi, has been rejected as non-compliant |
| *CLOSEDACCEPTED* | The *Warranty report* , transferred to DèLonghi, is accepted and closed |

#### **1.2.5 Updating a Warranty report**

Updating the *Warranty report* is necessary when it has the following statuses:

* Invalid (not transmitted): there are errors in the creation phase (the message with the error is quoted in the Agora response)
* Not conform: the warranty intervention report has not been created by Dèlonghi due to incorrect or missing data

In such cases, the *Warranty report* must be corrected by the service center before it is forwarded to DèLonghi. To correct the error:

* update the *Warranty report*
  + web API [ [/api/repair/updatewarrantyclaim](https://preprod-api.agoraplus.com/apidoc/ui/index#!/Repair/Repair_UpdateWarrantyClaim) ]
* update the *Repair file* by sending the 601 "Intervention closed with warranty” status code
  + web API [ [/api/repair/updaterepair](https://preprod-api.agoraplus.com/apidoc/ui/index#!/Repair/Repair_UpdateRepair) ]

NOTE

After the creation of the *Warranty report* , Agora has automatically updated the status of its *Repair file* from the 305 "Intervention completed” status code - that is the last status sent by the service center - to the 601 "Intervention closed with warranty” status code.

#### **1.2.6 Delivery of a product to the consumer after recording a Warranty report**

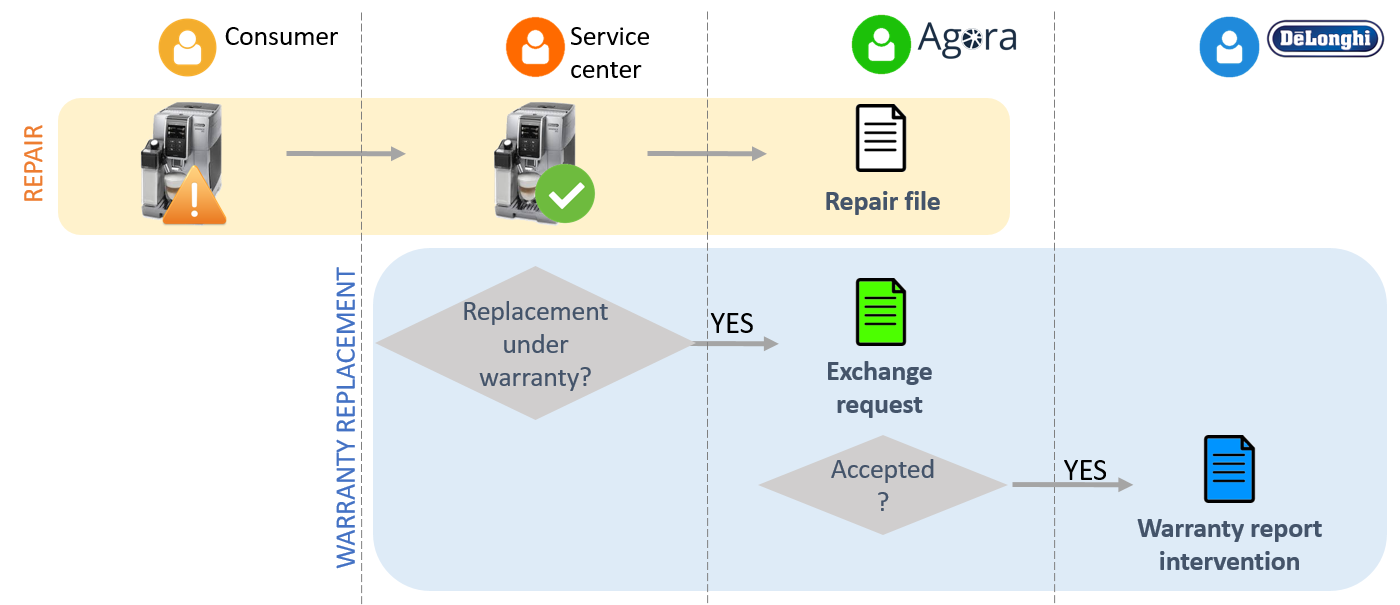
When the product is delivered to the consumer and the *Repair file* has the 601 "Intervention closed with warranty" status code, it is possible to upgrade the *Repair file* status. To do this:

* update the *Repair file* by sending the 602 "Product delivered” status code
  + web API [ [/api/repair/updaterepair](https://preprod-api.agoraplus.com/apidoc/ui/index#!/Repair/Repair_UpdateRepair) ]

### 

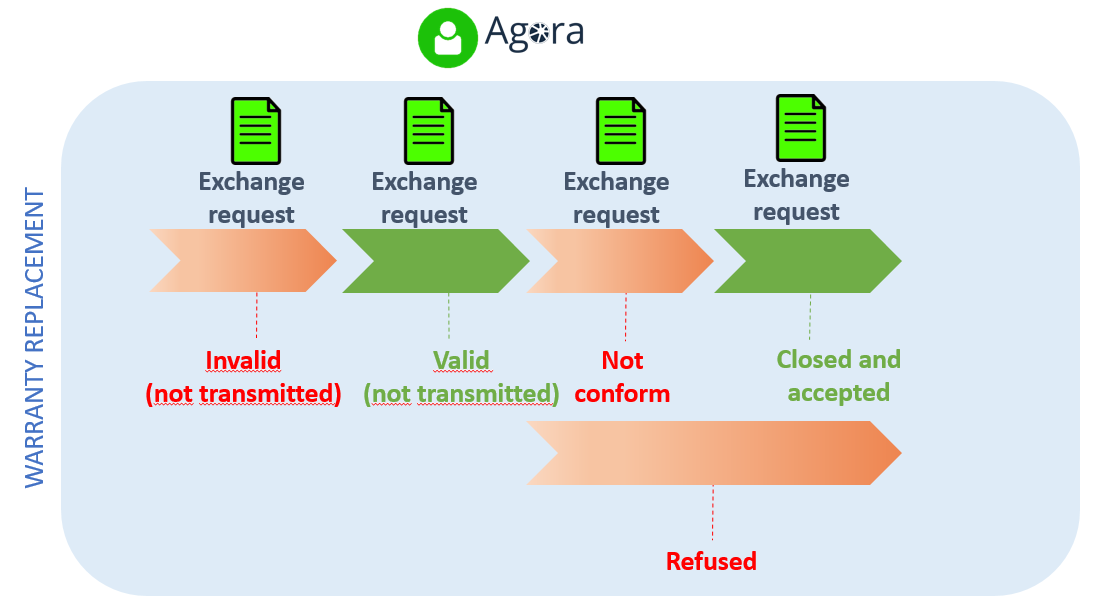
### 1.3 Replacement management

If the product is a non-repairable one, the service center sends the request to DèLonghi through Agora to have the product replaced under warranty. If the request is accepted, Agora sends the Warranty Intervention Report to DèLonghi containing the repair information and labor costs. In Agora, the replacement request is represented by the entity called Exchange request.



#### **1.3.1 Exchange request work cycle analysis**

The Exchange request work cycle is composed by the following statuses:



##### 1.3.1.1 Invalid (not transmitted)

This status means that the Exchange request has not been validated by Agora as one or more data are missing and / or incorrect. Such Exchange request in this state must be updated to be valid for Agora and can be transferred from the service center to DèLonghi.

##### 1.3.1.2 Valid (not transmitted)

This status means that the Exchange request has been validated by Agora. An Exchange request in this state can be transferred from the service center to DèLonghi

##### 1.3.1.3 Not conform

This status means that the transferred Exchange request has not been validated by DèLonghi as one or more data are missing and / or incorrect. Such Exchange request must be updated and then once more transferred to DèLonghi.

##### 

##### 1.3.1.4 Accepted

This status means that the transferred Exchange request is considered valid by DèLonghi.

##### 1.3.1.5 Refused

This status means that the Exchange request has been rejected by DèLonghi.

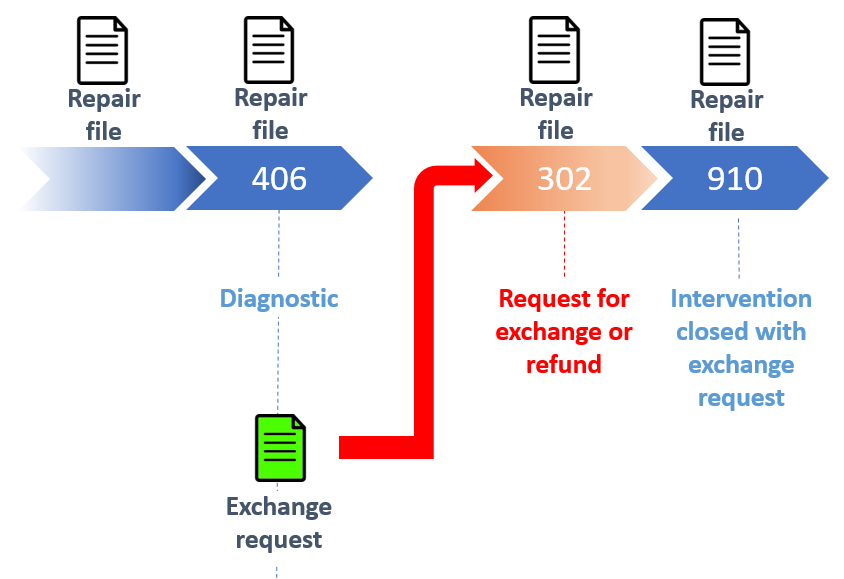
#### **1.3.2 Creating an Exchange request**

When a repair is not possible and a replacement must be requested, you need to:

* create an *Exchange request*
  + web API [ Not fulfilled, it must be manually created by the service center in Agora ]

NOTE

After the creation of an Exchange request, Agora automatically updates the related Repair file status to the 302 *"Request for exchange or refund"* status code ( ref. [1.1.2.9 Request for exchange or refund (code 302)](#_rpqobhkbzcjd) )



#### **1.3.3 Transferring an Exchange request**

When the request is complete and can be transferred to DèLonghi for acceptance, you need to:

* transfer the *Exchange request*
  + web API [ Not fulfilled, it must be manually created by the service center in Agora ]

#### **1.3.4 Evaluating an Exchange request**

When DéLonghi receives an exchange request, it can be evaluated as:

* **Accepted** ( ref. [1.3.1.4 Accepted](#_v4omp34ti0ev) ): the request is complete, accepted and the warranty intervention report is created
* **Not conform** ( ref. [1.3.1.3 Not conform](#_4lqfoej0dcmj) ): the request is incomplete and must be updated
* **Refused** ( ref. [1.3.1.5 Refused](#_13bkkl1a5gv1) ): the request is not accepted

NOTE

When an Exchange request is accepted, it is necessary to upgrade the work cycle status of the *Repair file* and update its general status. To do this:

* update the *Repair file* by both by sending the 910 "Intervention closed with exchange request" status code, adding a short note if needed, and filling in the "CLOSED" step (field StatusCode)
  + web API [ [/api/repair/updaterepair](https://preprod-api.agoraplus.com/apidoc/ui/index#!/Repair/Repair_UpdateRepair) ]

*Example*

{

"ManufacturerId": *{ManufacturerId}*,

"MissionOrder": *{MissionOrder}*,

"StateNatures": [

{

"Code": "910",

"Comment": *{Note}*

}

],

"StatusCode": "CLOSED"

}

#### **1.3.5 Updating an Exchange request**

Updating an Exchange request is necessary when it has the following statuses:

* Not valid (non transmitted) ( ref. [1.3.1.1 Invalid (not transmitted)](#_3tq9hyg2hnbs) ) : this is the case in which there are errors when creating an Exchange request. ( ref. [1.3.2 Creating an Exchange request](#_b2blgd57d2jm) )
* Not conform ( ref. [1.3.1.3 Not conform](#_4lqfoej0dcmj) ) : this is the case in which an Exchange request has not been accepted by Dèlonghi due to incorrect or missing data. ( ref. [1.3.4 Evaluating an Exchange request](#_rna6jfxvobvi) )

In such cases, the Exchange request must be corrected by the service center before forwarding to DèLonghi. To correct the error:

* update the *Exchange request*:
  + web API [ Not fulfilled, it must be manually created by the service center in Agora ]

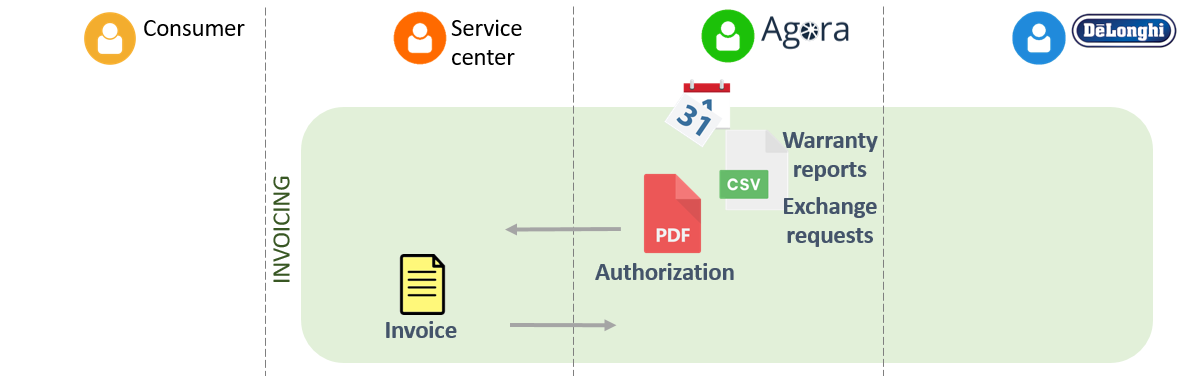
#### **1.3.6 Delivery of the replaced product to the consumer**

When the replaced product is delivered to the consumer, and the *Repair file* status is the 910 "Intervention closed with exchange request”, it is possible to upgrade the *Repair file* status. To do this:

* update the *Repair file* by sending the 602 "Product delivered” status code
  + web API [ [/api/repair/updaterepair](https://preprod-api.agoraplus.com/apidoc/ui/index#!/Repair/Repair_UpdateRepair) ]

### 1.4 Invoicing management

At the beginning of the month, the service center sends to Agora the invoice number related to the repairs under warranty and the replacements processed the previous month. Agora will send this information to DèLonghi for the reimbursement/refund.



#### **1.4.1 Integration with Agora**

The service center receives an email with the authorization, in PDF format, and the list, in CSV format, of all agreed replacements and warranty intervention reports by DèLonghi in the previous month. The service center, after creating the invoice in its aftersales software, sends the invoice reference to Agora.

#### **1.4.2 Sending reference of your invoice**

To communicate the details of the invoice to Agora, you need to:

* call the appropriate API by sending the invoice number and the authorization code
  + web API [ [/api/repair/createeinvoice](https://api.agoraplus.com/apidoc/ui/index#!/Repair/Repair_CreateEInvoice) ].

### 1.5 Intervention status summary table

| Repair (service center) | Repair file (Agora) | | | |
| --- | --- | --- | --- | --- |
| State | Intervention under warranty | | Replacement | |
| State code |  | State code |  |
| Opening | 100 | **set by Agora automatically** | 100 | **set by Agora automatically** |
| Acceptance | 405 | sent by the service center | 405 | sent by the service center |
| Inspection | 406 | sent by the service center | 406 | sent by the service center |
| Spare parts ordered | 205 | sent by the service center | - | - |
| Spare parts received | 201 | sent by the service center | - | - |
| Intervention completed | 305 | sent by the service center | - | - |
| N/A | 601 | **set by Agora automatically** | - | **-** |
| Product delivered | 602 | sent by the service center | 602 | sent by the service center |
| Exchange request | - | - | 302 | **set by Agora automatically** |
| Closure with replacement | - | - | 910 | sent by the service center |

## 2 General Instructions

AGORA APIs are a set of RESTful APIs designed to integrate AGORA functionalities with other aftersales softwares.

### 2.1 Authentication

Access to the AGORA API is allowed through authentication. To authenticate you need API keys entered as a parameter in the request to the web service. The service center must request the KEY API by email to support@agoraplus.com.

### 2.2 Environments

There are two different environments, one dedicated to development and testing called "pre-production" and the other one representing the "production" environment. The two environments have their own URL and API KEY to access the web services.

Pre-production environment url [ <https://preprod-api.agoraplus.com> ]

Production environment url [ <https://api.agoraplus.com> ]

### 2.3 Service Request

The parameters needed to make a request are described below.

#### **2.3.1 API Key**

This is the parameter used for authentication to the web service. The parameter name is *api\_key* and the value is different in the test and production environments for each service center.

#### **2.3.2 Accept**

This is the parameter used to indicate the MIME media type that the client is able to recognize and then accept. The parameter name is *Accept* and its value is *application/json*.

#### **2.3.3 Content type**

This is the parameter used to indicate the MIME media type used for the content of the request. Used for methods in POST and PUT. The parameter name is *Content-Type* and its value is *application/json*.

#### **2.3.4 Example**

GET /api/product/printmanufacturerproducts/19/01-01-2020/English HTTP/1.1

Host: api.agoraplus.com

api\_key: xxxxxxxxxxxxxxxxxxxxxxx

Accept: application/json

Content-Type: application/json

### 2.4 Service Response

#### **2.4.1 Media Type**

The MIME media type used for the response content is *application/json* and the encoding used is *UTF-8*.

#### **2.4.2 Date / timestamp**

The standard used to indicate dates and timestamps is *ISO 8601* and the format is *YYYY-MM-JJThh: mm: ssZ*

#### **2.4.3 HTTP Header - Status Codes**

| Code | Description | Note |
| --- | --- | --- |
| *200* | OK | The request is OK |
| *400* | Bad Request | Bad request: the request is incomplete or its structure is invalid. |
| *401* | Unauthorized | Not allowed: the authentication information that was sent is either missing or invalid. Authentication is to be fixed. |
| *403* | Forbidden | Forbidden: authentication is valid, but you do not have enough privileges. The user is not granted to access to the requested resources. This status is also returned for all requests from a locked user account. |
| *404* | Not found | No results match your request |
| *405* | Method Not Allowed | Method Not Allowed: The HTTP method is not supported for this resource. |
| *500* | Internal Server Error | Internal server error: used to indicate that there has been a server-side processing failure. |
| *503* | Service Unavailable | Service Not Available: try again later. The API sends this code when the systems on which it depends are stopped or react slowly. |

#### **2.4.4 Body Message - Structure**

All the services have the same body message. This is the json structure:

{

"ResponseStatus": {},

"IsValid": {},

"ResponseMessage": {},

"ResponseErrorMessage": {},

"ResponseData": {}

}

##### 

##### 2.4.4.1 ResponseStatus

This is an alphanumeric field that indicates the status of the response.

Expected values:

* ***S***: Success
* ***E***: Error
* ***W***: Warning

##### 2.4.4.2 IsValid

This is a boolean field that indicates the validity of the response.

Expected values:

* ***True***: the response doesn’t contain error
* ***False***: the response contains error

##### 2.4.4.3 ResponseMessage

This is string field that contains a summary message

##### 2.4.4.4 ResponseErrormessage

This is string field that contains an error message if the response is not valid

##### 2.4.4.5 ResponseData

This is an Array that contains the response objects. This is the real content of the response.

#### **2.4.5 Example**

{

"ResponseStatus": "S",

"IsValid": true,

"ResponseMessage": "19656 created or updated products found since 01/01/2020",

"ResponseErrorMessage": null,

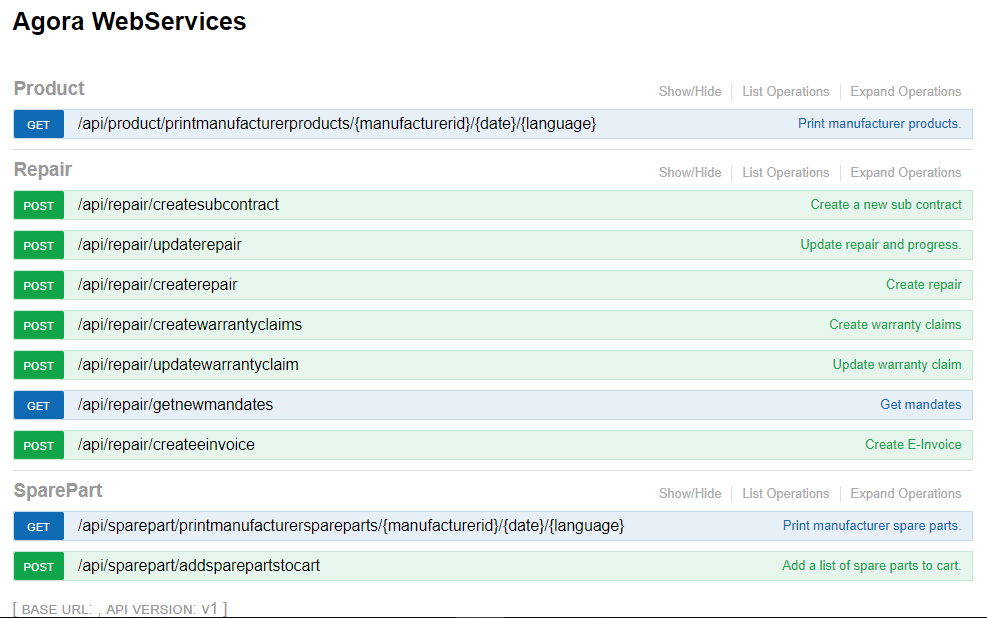
"ResponseData": [ {}, {}, {}, ...]

}

## 3 API field description

### 3.1 Technical documentation

The technical documentation relating to the REST API to be used is available through the Swagger interface.





For the pre-production environment the technical documentation is available at [<https://preprod-api.agoraplus.com>]. For the production environment the technical documentation is available at [<https://api.agoraplus.com>]

### 3.2 Repair file

#### **3.2.1 POST – Creating Repair file**

This is the service to create the *Repair file*

Resource:

**/api/repair/createrepair?language=**{language}

Parameters

* *language*: see the technical documentation for the allowed values

Technical documentation:

[https://preprod-api.agoraplus.com/apidoc/ui/index#!/Repair/Repair\_CreateRepair](about:blank)

##### 3.2.1.1 Body request

The red and blue fields are mandatory. The red fields are required by Agora and the blue ones are required by Delonghi

{

"**ActivationDate**": "string", *( Acceptance Date, UTC format required:*  YYYY-MM-DDTHH:mm:ss.sssZ )

"**Brand**": "string", *( Brand description , this value must be the BrandLabel available through the API* **printmanufacturerproducts***)*

"**CallDate**": "string", *( Acceptance Date, UTC format required:*  YYYY-MM-DDTHH:mm:ss.sssZ )

"**ClientNumber**": "string", *( Service center Agora code )*

"**CommercialRef**": "string", *( Product commercial code (TYPE))*

"**Consumer**": {

"**ConsumerInvoice**": "integer”, [values: **0** (*No*))]

"**Title**": "integer”, [values: **1** (*Mrs.*), **2** (*Ms.*), **3** (*Mr.*), **4** (*Company*)]

"**LastName**": "string",

"**FirstName**": "string",

"**Address1**": "string",

"Address2": "string",

"Address3": "string",

"**Zip**": "string",

"**Town**": "string",

"**Country**": "integer”, *( ISO Code 3 chars )*

"**Phone**": "string",

"PhoneType": "string",

"**SMSCode**": "string", *( Smartphone number)*

"ProPhone": "string",

"Email": "string",

"Remarks": "string"

},

"ConsumerContract": "string",

"**Family**": "string", *( Family Description, this value must be the FamilyLabel available through the API* **printmanufacturerproducts***)*

"GCPlusContract": "string",

"**Intervention**": {

"**InterventionType**": "integer” [values: **1** (*Intervention under warranty*)]

"**FailureDescription**": "string",

"IRISCondition": "string",

"IRISConditionEX": "string",

"IRISSymptom": "string",

"IRISSection": "string",

"IRISDefault": "string",

"**DefaultCode**": "string", *( Defect code )*

"ReplaceRef": "string"

},

"**ManufacturerId**": "integer”, ( *Manufacturer id* )

"**MissionOrder**": "string", *( The aftersales repair file number or the manufacturer workorder number if the workorder dataflow has been implemented )*

"OutOfWarAgreementNumber": "string",

"**PurchaseDate**": "string", *( UTC Format required:* YYYY-MM-DDTHH:mm:ss.sssZ )

"ResellerCountry": "integer”, *( ISO Code 3 chars )*

"ResellerName": "string",

"ResellerTown": "string",

"ResellerZip": "string",

"**SerialNumber**": "string", *( Product serial number )*

"**TechnicalRef**": "string", *( Product technical code (SKU))*

"**TimeZoneId**": "string" *( Time Zone ISO code of the service center country, example considering Italy "UTC+01:00" )*

}

#### 

#### **3.2.2 POST – Updating Repair file**

This is the service to update the *Repair file*

Resource:

**/api/repair/updaterepair**

Technical documentation:

[https://preprod-api.agoraplus.com/apidoc/ui/index#!/Repair/Repair\_UpdateRepair](about:blank)

##### 3.2.2.1 Body request

The red and blue fields are mandatory. The red fields are required by Agora and the blue ones are required by Delonghi

{

"**ManufacturerId**": "integer”, ( *Manufacturer id* )

"**MissionOrder**": "string", *( The aftersales repair file number or the manufacturer workorder number if the workorder dataflow has been implemented )*

"**NewConsumer**":

{

"BusinessPhoneNumber": "string",

"**Civility**": "string",

"Email": "string",

"**FirstName**": "string",

"**LastName**": "string",

"**PhoneNumber**": "string",

"PhoneType": "string",

"**PostalAddress**": {

"**AddressLines**": [ "string" ],

"**City**": "string",

"**CountryIsoCode**": "string",

"**Zip**": "string"

},

"**SmsCode**": "string" *( Smartphone number)*

},

"**NewProductReference**": {

"**CommercialReference**": "string", *( Product commercial code (TYPE))*

"**TechnicalReference**": "string" *( Product technical code (SKU))*

},

"**Intervention**": {

"**DefaultCode**": "string", *( Defect code )*

},

"**StateNatures**":

[

{

"**Code**": "string", ( *Agora status code )*

"Comment": "string",

"**AppointmentNumber**": "integer”, [values: **1**]

"**AppointmentDate**": "string", (Status Date*, UTC format required:*  YYYY-MM-DDTHH:mm:ss.sssZ )

"**AppointmentEndDate**": "string", (Status Date*, UTC format required:*  YYYY-MM-DDTHH:mm:ss.sssZ )

"**Parts**":  *( Spare parts related to this status )*

[

{

"Note": "string",

"**OrderNumber**": "string", *( Service center order number )*

"**Quantity**": "integer”,

"**Reference**": "string" *( Spare parts technical code (SKU))*

}

]

}

],

"StatusCode": "string",

"**Parts**": *(All spare parts involved in the repair )*

[

{

"Note": "string",

"**OrderNumber**": "string",

"**Quantity**": "integer”,

"**Reference**": "string" *( Spare parts technical code (SKU))*

}

]

}

##### 3.2.2.2 Note

The ***Parts*** field, which represents an array of spare parts, is present in two places in the data structure to send with the UpdateRepair method:

1. as an attribute of the MandateState -> StateNatures field
2. as an attribute of the MandateState field

In the first point, must be specified the spare parts relating to the state you are sending.

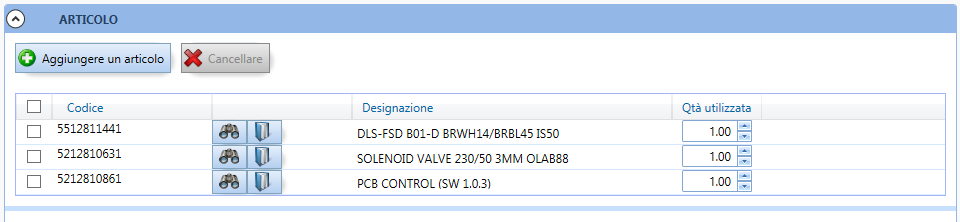
In the image below you can see how this information will be present and visible in the "History" section in Agora:

* 3 spare parts sent with status 205 (Spare parts ordered), i.e. spare parts ordered
* 3 spare parts sent with status 201 (Spare parts ordered and received), i.e. the spare parts received

### 

In the second point must be specified all the spare parts foreseen for the repair.

In the image below you can see how this information will be present and visible in the "Articles" section in Agora:



#### 

#### **3.2.3 GET – Downloading Repair file**

This is the service to download the *Repair file*

Resource:

**/api/repair/printrepairfilehistory?ordermission=**{ordermission}**&manufacturerId=**{manufacturerId}

Parameters:

* *ordermission* [string]: contains the aftersales repair file number or the manufacturer workorder number if the workorder dataflow has been implemented, it contains
* *manufacturerId* [Integer]: Manufacturer identifier

Technical documentation:

<https://preprod-api.agoraplus.com/apidoc/ui/index#!/Repair/Repair_PrintRepairFileHistory>

##### 3.2.3.1 ResponseData structure

The blue fields contain information about the product and consumer

{

"AggreementNumber": string,

"CRMAccount": string,

"**CallDate**": string,

"ClaimId": integer,

"ClientCode": string,

"Comment": string,

"**Consumer**": {

"BusinessPhoneNumber": string,

"**Civility**": integer, *(Consumer title: [values:* ***1*** *(Mrs.),* ***2*** *(Ms.),* ***3*** *(Mr.),* ***4*** *(Company)])*

"ConsumerId": string,

"**Email**": string,

"**FirstName**": string,

"**LastName**": string,

"Latitude": null,

"**Lng\_Id**": integer, *(Consumer language LCID code ( ref.* [*4.9 Language LCID (Locale) Code*](#_fqk8e37u83tb) *) )*

"Longitude": null,

"**PhoneNumber**": string,

"PhoneType": string,

"**PostalAddress**": {

"**Address1**": string,

"Address2": string,

"Address3": string,

"**City**": string,

"**Country**": string, *(Country ISO Code)*

"StateProvinceCode": string,

"**Zip**": string

},

"SMSCode": string

},

"DefaultCode": string,

"DefaultCodeDes": string,

"Diagnostic": string,

"EndDate": string,

"**FailureDescription**": string,

"InterventionID": interger,

"IrisCondition": string,

"IrisConditionEx": string,

"IrisDefault": string,

"IrisFamilyCode": string,

"IrisRepair": string,

"IrisSection": string,

"IrisSymptom": string,

"LastStateNatureCode": string,

"ManufacturerID": integer,

"ObjectID": integer,

"**OrderMission**": string, (*aftersales repair file number or the manufacturer workorder number if the workorder dataflow has been implemented*)

"Parts": [

{

"OrderNumber": string,

"Quantity": integer,

"Reference": string,

"Label": string,

"IsOutOfManufacturerStock": boolean,

"Refunding": boolean,

"ToBeOrdered": boolean

}

],

"**Product**": {

"**Brand**": string,

"CommercialReference": string,

"ContractNumber": string,

"**Family**": string,

"**PurchaseDate**": string,

"**SerialNumber**": string,

"**TechnicalReference**": string

},

"Reseller": {

"**Name**": string,

"PostalAddress": {

"Address1": string,

"Address2": string,

"Address3": string,

"City": string,

"Country": string,

"StateProvinceCode": string,

"Zip": string

}

},

"StartDate": string,

"StatesNatures": [

{

"AppointmentDate": string,

"AppointmentRank": string,

"AttachedFiles": [],

"Code": string,

"Comment": string,

"Label": string,

"Parts": [],

"StatusDate": string

}

],

"StatusCode": string,

"StatusLabel": string,

"**TimeZoneId**": string,

"VersionID": string,

"WarrantyClaimStatus": string,

"WarrantyClaimStatusLabel": string,

"WorkDescription": string

}

### 3.3 Warranty report

#### **3.3.1 POST – Creating Warranty report**

This is the service to create the *Warranty report*

Resource:

**/api/repair/createwarrantyclaims?language=**{language}&**transferIfValid**={true/false}

Parameters:

* *language*: see the technical documentation for the allowed values
* *transferIfValid*:specify to force the transfer of Warranty report if it is valid

Technical documentation:

<https://preprod-api.agoraplus.com/apidoc/ui/index#!/Repair/Repair_CreateWarrantyClaims>

##### 3.3.1.1 Body request

**The red and blue fields are mandatory**. The red fields are required by Agora and the blue ones are required by Delonghi.

{

"**ClaimType**":

{

"**ManufacturerId**": "integer", *( Manufacturer Id )*

"**CommercialRef**": "string", *( Product commercial code (TYPE))*

"**TechnicalRef**": "string", *( Product technical code (SKU))*

"**SerialNumber**": "string", *( Product serial number )*

"**PurchaseDate**": "string", *( UTC Format required:* YYYY-MM-DDTHH:mm:ss.sssZ )

"**Name**": "string", *(Technician name )*

"**ReportNumber**": "string", *( Service center repair number )*

"**ProductStatus**": "string", [values: **S** [under warranty]]

"OutOfWarAgreementNumber": "string",

"ConsumerContractNumber": "string",

"**MissionOrderNumber**": "string", *( Service center repair number )*

"ExchangeAgreementNumber": "string",

"AgoraRegistrationNumber": "string",

"CounterOne": "integer",

"CounterTwo": "integer"

},

"**ClaimDeliv**": *( Service center )*

{

"DeliveryMode": "integer",

"CustomerDeliveredCode": "string",

"**Company**": "string",

"Name": "string",

"**Address1**": "string",

"Address2": "string",

"Address3": "string",

"**Zip**": "string",

"**City**": "string",

"**Country**": "integer"

},

"ClaimResal":

{

"CompanyName": "string",

"Zip": "string",

"City": "string",

"Country": "integer",

"DistributorFileNumber": "string"

},

"**ClaimConso**": *( Consumer )*

{

"**Title**": "integer", [values: **1** (*Mrs.*), **2** (*Ms.*), **3** (*Mr.*), **4** (*Company*)]

"**LastName**": "string",

"**FirstName**": "string",

"**StreetNumber**": "string",

"**Address1**": "string",

"Address2": "string",

"Address3": "string",

"**Zip**": "string",

"**City**": "string",

"**Country**": "integer", *( ISO code 3 chars )*

"**Phone**": "string",

"Mobile": "string",

"Email": "string",

"**ConsumerKey**": "string" *( Consumer identification )*

},

"**ClaimInfos**": *( Repair )*

{

"IsInterventionAt48h": true,

"**CallDate**": "string", *( Acceptance Date, UTC format required:*  YYYY-MM-DDTHH:mm:ss.sssZ )

"**InterventionStartDateTime**": "string", *( UTC Format required:* YYYY-MM-DDTHH:mm:ss.sssZ )

"**InterventionEndDateTime**": "string", *( UTC Format required:* YYYY-MM-DDTHH:mm:ss.sssZ )

"DefaultPartRef": "string",

"IRISCondition": "string",

"IRISConditionEX": "string",

"IRISSymptom": "string",

"IRISSection": "string",

"IRISDefault": "string",

"IRISRepair": "string",

"**FailureDescription**": "string",

"**InterventionDescription**": "string",

"ActionCode": "string",

"ActionPattern": "integer",

"**DefaultCode**": "string" *( Defect code )*

},

"**ClaimLab**": *( Labor )*

{

"**OriginCode**": "string", *( Labor code - see )*

"LaborHours": "integer",

"LaborMinutes": "integer",

"LaborCost": "number",

"TravelKilometers": "integer",

"TravelCost": "number",

"OtherCosts": "number",

"OtherCostsDescription": "string",

"**Currency**": "string", [values: **EUR** (euro) ]

"Redirection": true

},

"**ClaimParts**": *(All spare parts involved in the repair )*

[

{

"**PartReference**": "string",  *( Spare parts technical code (SKU))*

"**Quantity**": "integer",

"Comments": "string",

"BillNumber": "string"

}

]

}

##### 3.3.1.2 Note

The ***OriginCode*** field, which represents the labor code, must be indicated without prefix “ARTFACQ” but only the numeric part.

*Example*

Labor code “ARTFACQ060”

"ClaimLab":

{

"OriginCode": "060",

….

}

#### **3.3.2 POST – Updating Warranty report**

This is the service to update the *Warranty report*

Resource:

**/api/repair/updatewarrantyclaim?warrantyClaimId=**{claimId}&**language=**{language}&**transferIfValid**={true/false}

Parameters:

* *warrantyClaimId*: must contain the AgoraID returned with API createWarrantyClaim
* *language*: see the technical documentation for the allowed values
* *transferIfValid*:specify to force the transfer of Warranty report if it is valid

Technical documentation: [https://preprod-api.agoraplus.com/apidoc/ui/index#!/Repair/Repair\_UpdateWarrantyClaim](about:blank)

##### 3.3.2.1 Body Request

See Section [*3.3.1.1 Body request*](#_qhqn5hj5b4dc)

##### 3.3.2.2 Note

See Section [*3.3.1.2 Note*](#_v67pr9p7gqf8)

#### **3.3.3 GET – Warranty reports list**

This is the service to get the *Warranty reports*

Resource: **/api/repair/getwarrantyclaims?claimId=**{claimId}**&claimCodeClient=**{claimCodeClient}**&manufacturerId=**{manufacturerId}**&modelTechnicalRef=**{modelTechnicalRef}**&modelCommercialRef=**{modelCommercialRef}**&searchDate=**{searchDate}**&fromDate=**{fromDate}**&toDate=**{toDate}

Parameters:

* *claimId* [integer]: identifier in Agora
* *claimCodeClient* [string]: identifier from client
* *manufacturerId* [Integer]: Manufacturer identifier
* *modelTechnicalRef* [string]: technical reference of product
* *modelCommercialRef* [string]: commercial reference of product
* *searchDate* [string]: search date filter for FromDate and ToDate fields. Possible values :
  + CreationDate : creation date of warranty claim
  + LastSentDate : last sent date to manufacturer
  + LastBackDate : last back date from manufacturer
  + LastValidDate : last valid date of warranty claim
* *fromDate* [date-time]: start date of the search
* *toDate* [date-time]: end date of the search

Technical documentation: <https://preprod-api.agoraplus.com/apidoc/ui/index#!/Repair/Repair_GetWarrantyClaims>

### 3.4 Invoice

#### **3.4.1 POST – Sending Invoice**

This is the service to send the pre authorization number and the service center invoice number

Resource: **/api/repair/createeinvoice?preInvoiceNumber=**{preInvoiceNumber}**&customerInvoiceNumber=**{customerInvoiceNumber}

Parameters:

* *preInvoiceNumber:* is the authorization number that service center receives by email
* *customerInvoiceNumber:* is the invoice number created by the service center

Technical documentation: [https://preprod-api.agoraplus.com/apidoc/ui/index#!/Repair/Repair\_CreateEInvoice](about:blank)

## 4 Resources

There are several REST API services to download the master data to use in the service center aftersales software.

### 4.1 Products

This is the service to get all the product

Resource:

**api/product/printmanufacturerproducts/**{manufactuerId}**/**{date}**/**{language}

Parameters:

* *manufacturerId*: contains the Manufactuer identifier
* *date*: contains the searching start date and the required format is MM-DD-YYYY
* *language*: see the technical documentation for the allowed values

Technical documentation: <https://preprod-api.agoraplus.com/apidoc/ui/index#!/Product/Product_PrintProductsByManufacturer>

#### **4.1.1 ResponseData - Structure**

ManufacturerProductResponseData

{

BrandCode,

BrandLabel,

CommercialReference,

Country,

Family,

FamilyLabel,

Label,

TechnicalReference

}

#### **4.1.2 Example**

ManufacturerProductResponseData

{

"BrandCode": "DELO",

"BrandLabel": "DELONGHI",

"CommercialReference": "XLN700",

"Country": null,

"Family": "HHFCSC",

"FamilyLabel": "STICK CLEANER",

"Label": "\*SCO INT DL XLN700 EX:1 GREY MET\*",

"TechnicalReference": "0-021012"

}

### 4.2 Spare Parts

This is the service to get all the manufacturer's spare parts

Resource:

**/api/sparepart/printmanufacturerspareparts/**{manufactuerId}**/**{date}**/**{language}

Parameters:

* *manufacturerId*: contains the Manufactuer identifier
* *date:* contains the searching start date and the required format is MM-DD-YYYY
* *language*: see the technical documentation for the allowed values

Technical documentation: <https://preprod-api.agoraplus.com/apidoc/ui/index#!/SparePart/SparePart_PrintSparePartsByManufacturer>

#### **4.2.1 ResponseData - Structure**

ManufacturerSparePartResponseData

{

CanBeOrdered, *(this field is not used and contains NULL value)*

Currency,

ExpiryDate,

Label,

PartReference,

PartType,

Price,

PriceGroupCode, *(this field is not used and contains NULL value)*

ReplaceReference, *(this field is not used and contains NULL value)*

Status *(this field is not used and contains NULL value)*

}

#### **4.2.2 Example**

ManufacturerSparePartResponseData

{

"CanBeOrdered": null,

"Currency": "EUR",

"ExpiryDate": "31/12/9999",

"Label": "TP INT1 DL CTLA2103.BK +BUN",

"PartReference": "0230020014",

"PartType": "Accessory",

"Price": 26.5800,

"PriceGroupCode": null,

"ReplaceReference": null,

"Status": null

}

### 4.3 Labor rates

This is the service to get all the labor rates

Resource:

**/api/repair/getwarrantylabourcost/**{manufactuerId}

Parameters:

* *manufacturerId*: contains the Manufactuer identifier

Technical documentation: <https://preprod-api.agoraplus.com/apidoc/ui/index#!/Repair/Repair_GetWarrantyLabourCost>

#### **4.3.1 ResponseData - Structure**

WarrantyLabourCostResponseData

{

Brand,

CommercialReference,

Cost,

Currency,

Family,

IsForProductExchange,

LaborCode

}

#### **4.3.2 Example**

WarrantyLabourCostResponseData

{

"Brand": "ARTE",

"CommercialReference": "002030",

"Cost": "5,50",

"Currency": "EUR",

"Family": "HHATKM",

"IsForProductExchange": false,

"LabourCode": "ARTFACQ541"

}

### 

### 4.4 Defect codes

This is the service to get all the defect codes

Resource:

**/api/repair/printdefectcodes?manufacturerID=**{manufactuerId}**&language=**{language}

Parameters:

* *manufacturerId*: contains the Manufactuer identifier
* *language*: see the technical documentation for the allowed values

Technical documentation:<http://preprod-api.agoraplus.com/apidoc/ui/index#!/Repair/Repair_PrintDefectCodes>

#### **4.4.1 ResponseData - Structure**

DefectCodesResponseData

{

DefectCode,

Description,

FamilyCode

}

#### **4.4.2 Example**

DefectCodesResponseData

{

"DefectCode": "104",

"Description": "Leakage water/liquid - Repair without replacement",

"FamilyCode": "921U"

}

### 

### 4.5 Bill of material - Step 1

You need to call two web services to get the bill of material. With this first web service you get the key to use in the second one.

Resource:

**api/product/searchproductbyref/**{manufacturerId}**/**{keyword}**/**{countrycode}

Parameters:

* *manufacturerId*: contains the Manufactuer identifier
* *keyword*: specify the technical reference, if you want to get a specific product. Specify the commercial reference if you want to get all the products belonging to that commercial reference.
* *countrycode*: contains the numeric ISO code (3 digits) of the country

Technical documentation:

[https://preprod-api.agoraplus.com/apidoc/ui/index#!/](https://preprod-api.agoraplus.com/apidoc/ui/index#!/SparePart/SparePart_PrintSparePartsByManufacturer)...

#### **4.5.1 ResponseData - Structure**

ListProductDTO

{

ManufacturerId,

BrandLabel,

Description,

Refcom,

Reftech,

AGORAProductKey  
}

### 

#### **4.5.2 Example**

ListProductDTO

{

"ManufacturerId": 19,

"BrandLabel": "DELONGHI",

"Description": "\*MC FR DL ECAM23.210.B +WELCOME PACK\*",

"Refcom": "ECAM23.210.B",

"Reftech": "R132213000",

"AGORAProductKey": "X3ITlzr2JpoAwCUve8P4sQ%3d%3d"

}

### 4.6 Bill of material - Step 2

In this second web service you have to use the key will be parameter to use in the second service to get the bill of material

Resource:

**api/product/printbomdata?AGORAProductKey=**{AGORAProductKey}

Parameters:

* *AGORAProductKey*: is the key obtained with the web service described in the previous paragraph ( ref. [4.5 Bill of material - Step 1](#_e79gdlcevyg4) )

Technical documentation:

[https://preprod-api.agoraplus.com/apidoc/ui/index#!/](https://preprod-api.agoraplus.com/apidoc/ui/index#!/SparePart/SparePart_PrintSparePartsByManufacturer)...

#### **4.6.1 ResponseData - Structure**

ListPartsDTO

{

{

PartRef,

Description,

Quantity

},

{

PartRef,

Description,

Quantity

},

{...}  
}

### 

#### **4.6.2 Example**

ListPartsDTO

{

{

"PartRef": "5313272551",

"Description": "BOILER SPECIMENT NATURAL (PA6) E",

"Quantity": 1.000

},

{

"PartRef": "7313260931",

"Description": "ASSY TANK WATER EC230",

"Quantity": 1.000

},

{…}

}

### 4.7 CSV file (list of accepted warranty report)

The array below is a description of the columns which are in the CSV file sent by email to the service partner once a month with the pre-invoice.

**Important:** the CSV is an extract of all warranty reports related to the pre-invoice. As one warranty report can contain several spare part references used for the repair please consider that for 1 warranty report, the CSV file will contain as many lines as spare part references used.

| **Column** | **Description** |
| --- | --- |
| index | Index of the line |
| ordernbagora | AGORA warranty report number |
| creationdate | Warranty report creation date |
| clientnumber | Manufacturer custormer number |
| type | ***Obsolete*** |
| username | AGORA user account |
| ordernbclient | Aftersales repair file number |
| assignmentnumber | Work order number |
| orderperson | Technician who's created the warranty report |
| technicalref | Product commercial refrence |
| commercialref | Product technical reference |
| manufacturingdate | Product Manufacturing date |
| Typestatus | ***Obsolete*** |
| number | ***Obsolete*** |
| Consotitle | Consumer civility |
| Consofirstname | Consumer firstname |
| ConsoLastName | Consumer lastname |
| ConsoStreetNumber | Consumer street number |
| ConsoAddress1 | Consumer address 1 |
| ConsoAddress2 | Consumer address 2 |
| ConsoAddress3 | Consumer address 3 |
| ConsoZip | Consumer postal code |
| ConsoCity | Consumer city |
| Consostate | Consumer state |
| Consoisocountrycode | Consumer ISO country code (3 digits) |
| Consophone | Consumer phone |
| Consomobile | Consumer mobile phone |
| Consoemail | Consumer email |
| Resisunknown | ***Obsolete*** |
| Resname | Dealer name |
| Reszipcode | Dealer postal code |
| Rescity | Dealer city |
| Resstate | Dealer state |
| Resisocountrycode | Dealer ISO country code (3 digits) |
| Resdistributorreference | Delear workorder number |
| purchasedate | Product purchase date |
| startdate | Repair start date |
| enddate | Repair end date |
| source | ***Obsolete*** |
| addcostsdescription | Additional cost description |
| addcostsamount | Additional cost amount |
| addcostscurrency | Additional cost currency |
| iriscode | Repair IRIS Code |
| irispart | Spare part at the origin of the failure |
| irisreplaceagreementnumber | Manufacturer exchange agreement number |
| breakdown characteristics | Breakdown description |
| partline | Index of the part in the file |
| partref | Spare part reference used for the repair |
| partquantity | Qauntity of spare part used |
| partinvoicenumber | Spare part invoice/order number |
| partcomment | Spare part comment |
| PriceCode | Spare part price code |
| deliverymode | ***Obsolete*** |
| deliveryclientcode | ***Obsolete*** |
| deliverycompany | ***Obsolete*** |
| deliveryname | ***Obsolete*** |
| deliveryaddress1 | ***Obsolete*** |
| deliveryaddress2 | ***Obsolete*** |
| deliveryaddress3 | ***Obsolete*** |
| deliveryzipcode | ***Obsolete*** |
| deliverycity | ***Obsolete*** |
| deliverystate | ***Obsolete*** |
| deliveryisocountrycode | ***Obsolete*** |
| deliveryphonee | ***Obsolete*** |
| deliveryemaile | ***Obsolete*** |
| pricenet | Spare part unit price |
| Pricecurrencycode | Spare part currency |
| warrantyfixedprice | Labor cost |
| warrantycurrencycode | Labor currency |

### 

### 

### 4.8 Manufacturer’s ID list

| Country | Manufacturer Id |
| --- | --- |
| European country | 19 |
| American country | 19 |
| African country | 19 |
| Asian country | 34 |
| Oceanic country | 34 |

### 4.9 Language LCID (Locale) Code

Please refer to the following page [ <https://learn.microsoft.com/en-us/openspecs/office_standards/ms-oe376/6c085406-a698-4e12-9d4d-c3b0ee3dbc4a> ]