

Orquestador Rocketbot: APIs Xperience

¿Qué es una API?

Una API (Interfaz de Programación de Aplicaciones) es un conjunto de reglas, protocolos y herramientas que permiten a diferentes sistemas de software comunicarse e interactuar entre sí, de manera estructurada. Las direcciones URL a través de las cuales se envían solicitudes y se recibe información de las APIs se denominan **endpoints**.

Requisitos Previos

- Acceso al Orquestador
- El usuario del Orquestador debe tener una [api Key generada](#).

Endpoints disponibles

Listar Proyectos

Devuelve un listado de los proyectos cargados en el Orquestador, con sus respectivos procesos y robots asignados.

Método URL

Parámetros

POST **https://www.myrb.io/dev/api/project/list**

Authorization: tipo Bearer Token; es el Token de acceso válido para autenticar al usuario (API Key asociada al perfil en el orquestador)

Avanzado

Ejemplos de requests

- cURL

```
curl --location --request POST 'https://www.myrb.io/dev/api/project/list' \
--header 'Authorization: Bearer <token>'
```

- Python

```
import requests
```

```
url = "https://www.myrb.io/dev/api/project/list"
```

```

payload = {}
headers = {
    'Authorization': 'Bearer <token>'
}

response = requests.request("POST", url, headers=headers, data=payload)

print(response.text)

```

Ejemplo de response

```

{
    "success": true,
    "data": [
        {
            "name": "test orquestador",
            "token": "JR0EYGENSXEMTQWUP",
            "created_at": "2024-08-15 08:42:16",
            "id": 69,
            "description": null,
            "process_count": 1,
            "process": [
                {
                    "id": 168,
                    "project_id": 69,
                    "token": "1KZQIJKTTR75CGWF",
                    "name": "bot orquestador",
                    "last_robot": {
                        "id": 143,
                        "name": "bot orquestador",
                        "user_id": 67,
                        "client_id": 3,
                        "cron": null,
                        "version": null,
                        "version_rocketbot": null,
                        "robot":
"myrb_clients/dev/project/69/process/168/robot/rocket.bot",
                        "md5": "7af04cc5dbbea2e6ff1f27c183ef4772",
                        "comment": null,
                        "type": 1,
                        "status": 0,
                        "process_id": 168,
                        "start_bot": "bot_orquestador",
                        "created_at": "2024-08-21 09:37:50",
                        "updated_at": "2024-08-21 09:37:50"
                    },
                    "backup": null,
                    "trigger_process": null,
                    "client": null,
                    "robots": [
                        {

```

```

        "id": 143,
        "name": "bot orquestador",
        "user_id": 67,
        "client_id": 3,
        "cron": null,
        "version": null,
        "version_rocketbot": null,
        "robot":
"myrb_clients/dev/project/69/process/168/robot/rocket.bot",
        "md5": "7af04cc5dbbea2e6ff1f27c183ef4772",
        "comment": null,
        "type": 1,
        "status": 0,
        "process_id": 168,
        "start_bot": "bot_orquestador",
        "created_at": "2024-08-21 09:37:50",
        "updated_at": "2024-08-21 09:37:50"
    }
}
    ]
}
    ]
}
}

```

Subir robot a un proceso existente

Permite asociar un robot a un proceso ya existente en el Orquestador.

Método URL	Formato	Parámetros
------------	---------	------------

Authorization:
 tipo Bearer Token;
 es el Token de
 acceso válido para
 autenticar al
 usuario (API Key
 asociada al perfil
 en el
 orquestador).
name: Nombre del
 proceso al que se
 asignará el robot.
startbot: Nombre
 del robot como
 aparece en Studio.
process_id: Token
 del proceso. Hace
 referencia al
token del proceso,
 se puede obtener
 desde el ROC ó
 utilizando el
 endpoint
/api/project/list.
project_id:
 Identificador del
 proyecto al que
 pertenece el
 proceso. Debe
 obtenerse
 previamente
 utilizando el
 endpoint
/api/project/list.
file: robot.bd

POST <https://www.myrb.io/dev/api/process/robot/update/data> multipart/form-data

Avanzado

Ejemplos de requests

- cURL

```
curl --location 'https://www.myrb.io/dev/api/process/robot/update/data' \
--header 'Authorization: Bearer <token>' \
--form 'name="process1"' \
--form 'startbot="navegador"' \
--form 'process_id="NDQGBJBSE4LDRSVA"' \
--form 'project_id="56"' \
--form 'file=@"/C:/Users/navegador.db"'
```

- Python

```
import requests
```

```
url = "https://www.myrb.io/dev/api/process/robot/update/data"
```

```
payload = {'name': 'process1',
'startbot': 'navegador',
'process_id': 'NDQGBJBSE4LDRSVA',
```

```

'project_id': '56'}
files=[
('file',('navegador.db',open('/C:/Users/navegador.db','rb'),'application/octet-stream'))
]
headers = {
    'Authorization': 'Bearer <token>'
}

response = requests.request("POST", url, headers=headers, data=payload,
files=files)

print(response.text)

```

Ejemplo de response

```

{
  "success": true,
  "data": {
    "robot": "myrb_clients/dev/project/56/process/152/robot/rocket.bot",
    "md5": "484871810fe6e7ed78b4bacb53a408b3",
    "start_bot": "navegador",
    "status": 0,
    "type": 1,
    "name": "process1",
    "user_id": 66,
    "process_id": 152,
    "client_id": 3,
    "updated_at": "2024-09-18 16:58:14",
    "created_at": "2024-09-18 16:58:14",
    "id": 145,
    "process": {
      "id": 152,
      "name": "process1",
      "token": "NDQGBJBSE4LDRSVA",
      "robot_id": 131,
      "config": null,
      "client_id": 3,
      "data": null,
      "status": 0,
      "minutes": null,
      "sent_email": null,
      "email": null,
      "trigger": null,
      "created_at": "2024-08-05 17:14:14",
      "updated_at": "2024-08-05 13:14:14",
      "project_id": 56,
      "form_id": null,
      "private": 0,
      "last_robot": {
        "id": 145,

```

```

        "name": "process1",
        "user_id": 66,
        "client_id": 3,
        "cron": null,
        "version": null,
        "version_rocketbot": null,
        "robot":
"myrb_clients/dev/project/56/process/152/robot/rocket.bot",
        "md5": "484871810fe6e7ed78b4bacb53a408b3",
        "comment": null,
        "type": 1,
        "status": 0,
        "process_id": 152,
        "start_bot": "navegador",
        "created_at": "2024-09-18 16:58:14",
        "updated_at": "2024-09-18 16:58:14"
    },
    "backup": null,
    "trigger_process": null,
    "client": {
        "id": 3,
        "name": "Academy",
        "owner": null,
        "parent": null,
        "type": 2,
        "status": 0,
        "created_at": "2022-09-05 11:07:44",
        "updated_at": "2023-01-03 11:04:21"
    },
    "robots": [
        {
            "id": 131,
            "name": "process1",
            "user_id": 66,
            "client_id": 3,
            "cron": null,
            "version": null,
            "version_rocketbot": null,
            "robot":
"myrb_clients/dev/project/56/process/152/robot/rocket.bot",
            "md5": "5594faf198e491c7868cf600204f39d2",
            "comment": null,
            "type": 1,
            "status": 0,
            "process_id": 152,
            "start_bot": "saludo",
            "created_at": "2024-08-05 17:14:14",
            "updated_at": "2024-08-06 10:38:46"
        },
        {
            "id": 145,

```

```

        "name": "process1",
        "user_id": 66,
        "client_id": 3,
        "cron": null,
        "version": null,
        "version_rocketbot": null,
        "robot":
"myrb_clients/dev/project/56/process/152/robot/rocket.bot",
        "md5": "484871810fe6e7ed78b4bacb53a408b3",
        "comment": null,
        "type": 1,
        "status": 0,
        "process_id": 152,
        "start_bot": "navegador",
        "created_at": "2024-09-18 16:58:14",
        "updated_at": "2024-09-18 16:58:14"
    }
}
]
}
}
}

```

Iniciar la ejecución de un robot

Permite iniciar la ejecución de un robot, sin necesidad de un formulario. Es ideal para integrar la ejecución de robots en sistemas externos.

Método URL	Formato	Parámetros
POST	https://www.myrb.io/dev/api/process/robots/start	Authorization: tipo Bearer Token; es el Token de acceso válido para autenticar al usuario (API Key asociada al perfil en el orquestador). process_id: Identificador del proceso. Hace referencia al token del proceso, se puede obtener desde el ROC o utilizando el endpoint <code>/api/project/list</code>.

Avanzado

Ejemplos de request

- cURL

```

curl --location 'https://www.myrb.io/dev/api/process/robots/start' \
--header 'Authorization: Bearer <token>' \

```

```
--data-urlencode 'process_id=CEDLU6FK79B0LPUM'
```

- **Python**

```
import requests

url = "https://www.myrb.io/dev/api/process/robots/start"

payload = 'process_id=CEDLU6FK79B0LPUM'
headers = {
    'Authorization': 'Bearer <token>',
    'Content-Type': 'application/x-www-form-urlencoded'
}

response = requests.request("POST", url, headers=headers, data=payload)

print(response.text)
```

Ejemplo de response

```
{
    "success": true
}
```

Obtener todas las queues

Devuelve un listado de las queues de todos los [formularios de Xperience](#) existentes en el Orquestador, pudiendo además filtrar por fecha de inicio y fin de las mismas.

Método URL	Formato	Parámetros
------------	---------	------------

Authorization:
tipo Bearer
Token; es el
Token de
acceso válido
para
autenticar al
usuario (API
Key asociada
al perfil en
el
orquestador).
from: Fecha de
inicio de la
búsqueda
(YYYY-MM-DD).
to: Fecha de
fin de la
búsqueda
(YYYY-MM-DD)

POST <https://www.myrb.io/dev/api/formData/all> application/json

Avanzado

Ejemplos de request

- cURL

```
curl --location 'https://www.myrb.io/dev/api/formData/all' \  
--header 'Authorization: Bearer <token>' \  
--header 'Content-Type: application/json' \  
--data '{  
    "from": "2024-06-9",  
    "to": "2024-07-16"  
'
```

- Python

```
import requests  
import json  
  
url = "https://www.myrb.io/dev/api/formData/all"  
  
payload = json.dumps({  
    "from": "2024-06-9",  
    "to": "2024-07-16"  
})  
headers = {  
    'Authorization': 'Bearer <token>',  
    'Content-Type': 'application/json'  
}
```

```
response = requests.request("POST", url, headers=headers, data=payload)

print(response.text)
```

Ejemplo de response

```
{
  "success": true,
  "data": [
    {
      "id": 8315,
      "form_name": "five9notas",
      "form_token": "6HNZNSYVG4W2MCS0",
      "created_at": "2024-07-10 11:04:55",
      "processed_at": "2024-07-10 11:06:26",
      "status": 1,
      "user_form_email": null,
      "locked_user": "Bot Integracion",
      "locked": 0,
      "form_id": 25
    },
    {
      "id": 8316,
      "form_name": "Clientify",
      "form_token": "F9J65RTNWWPMTYD8",
      "created_at": "2024-07-11 10:25:35",
      "processed_at": "2024-07-11 10:27:13",
      "status": 1,
      "user_form_email": null,
      "locked_user": "Robot Interno",
      "locked": 0,
      "form_id": 14
    }
  ]
}
```

Añadir datos a queue

Completa un formulario de Xperience y coloca los datos en cola para ser procesados (Debe tener activa la opción “send API” en el formulario).

Método URL

Formato

Parámetros

Authorization:
tipo Bearer
Token; es el
Token de
acceso válido
para
autenticar al
usuario (API
Key asociada
al perfil en
el
orquestador).
Dato1:
Dato2:
(Otros Datos):
Se pueden
incluir campos
adicionales
según el
diseño del
formulario.

https://www.myrb.io/dev/api/formData/addQueue/:token
POST **token:** Representa el token de formulario creado en multipart/form-data el
ROC Xperience.

Avanzado

Ejemplo de request

- **cURL**

```
curl --location 'https://www.myrb.io/dev/api/formData/addQueue/:token' \  
--header 'Authorization: Bearer <token>' \  
--form 'email="prueba@prueba.com"' \  
--form 'nombre="prueba1"'
```

- **Python**

```
import requests  
  
url = "https://www.myrb.io/dev/api/formData/addQueue/:token"  
  
payload = {'email': 'prueba@prueba.com',  
           'nombre': 'prueba1'}  
files=[  
  
]  
headers = {  
    'Authorization': 'Bearer <token>'  
}  
  
response = requests.request("POST", url, headers=headers, data=payload,  
                             files=files)  
  
print(response.text)
```

Ejemplo de response

```
{
  "success": true,
  "data": {
    "xperience":
"eyJpdjI6Inpid0RRVlV0bEczTW9DM1Z5SEFrV1E9PSIsInZhbHVlIjoi0VJCZHhHTDBwQ0p1Y2ps
VU9WZTYxbE5paFwvRTQ4cWlKSFZ3UUxGcVNUQXh4UTBx0FEwc01Ba0pvV1kwQklRdXByTE1JU2tWM
TMxZkx5Qk41SFNTNmdRPT0iLCJtYWMiOiI5YjFiNjAyMTRiODUwMjk3YjdjZGEwZjFmMjE4MjhiNG
I5MzA5MTViNmVhMTE4OTI0NmExNTgwYWE4YWJlYWJlIn0="
  }
}
```

Obtener todos los queues id

Devuelve un listado con los IDs de todas las queues con estado “Not Processed” de un formulario de Xperience existente.

Método URL

POST **https://www.myrb.io/dev/api/formData/get/:token:**
:token: Representa el token de formulario creado en ROC Xperience.

Parámetro

Authorization:
 tipo Bearer Token;
 es el Token de
 acceso válido para
 autenticar al
 usuario (API Key
 asociada al perfil
 en el
 orquestador).

Avanzado

Ejemplos de request

• cURL

```
curl --location --request POST
'https://www.myrb.io/dev/api/formData/get/:token' \
--header 'Authorization: Bearer <token>'
```

• Python

```
import requests

url = "https://www.myrb.io/dev/api/formData/get/:token"

payload = {}
headers = {
  'Authorization': 'Bearer <token>'
}
```

```
response = requests.request("POST", url, headers=headers, data=payload)

print(response.text)
```

Ejemplo de response

```
{
  "success": true,
  "data": [
    {
      "id": 4734,
      "form_name": null,
      "user_form_email": null,
      "locked_user": null,
      "form_token": null,
      "form": null,
      "user_form": null,
      "locker": null
    },
    {
      "id": 4735,
      "form_name": null,
      "user_form_email": null,
      "locked_user": null,
      "form_token": null,
      "form": null,
      "user_form": null,
      "locker": null
    },
  ]
}
```

Cambiar de estado una queue

Permite modificar el estado de un queue.

Método URL	Formato	Parámetro
------------	---------	-----------

Authorization:
tipo Bearer
Token; es el
Token de
acceso válido
para
autenticar al
usuario (API
Key asociada
al perfil en
el
orquestador).
status: Un
número entero
que indica el
nuevo estado
de la cola. 0:
No procesado
1: Procesado
lock: Un nú 0:
No bloqueada
1: Bloqueada

POST **https://www.myrb.io/dev/api/formData/setStatus/:id:**
:id: Representa el ID de la cola que se desea
modificar. Este ID debe obtenerse previamente
utilizando el endpoint **/formData/get/:token.** multipart/form-data

Avanzado

Ejemplos de request

- cURL

```
curl --location 'https://www.myrb.io/dev/api/formData/setStatus/:id' \  
--header 'Authorization: Bearer <token>' \  
--form 'status="1"' \  
--form 'locked="0"'
```

- Python

```
import requests  
  
url = "https://www.myrb.io/dev/api/formData/setStatus/:id"  
  
payload = {'status': '1'}  
files=[  
  
]  
headers = {  
    'Authorization': 'Bearer <token>'  
}  
  
response = requests.request("POST", url, headers=headers, data=payload,  
files=files)  
  
print(response.text)
```

Ejemplo de response

```
{
  "success": true
}
```

Obtener datos de un formulario

Permite recuperar los datos de un formulario específico que se encuentra dentro de una queue determinada. Es útil para procesar los datos de un formulario una vez que ha sido añadido a la queue.

Método URL

Parámetros

Authorization:
tipo Bearer
Token; es el
Token de
acceso válido
para
autenticar al
usuario (API
Key asociada
al perfil en
el
orquestrador).

POST **https://www.myrb.io/dev/api/formData/getQueue/:id/:token**
:id: Representa el ID de la cola. Este ID debe obtenerse previamente utilizando el endpoint **/formData/get/:token**.
:token: Representa el token de formulario creado en ROC Xperience.

Avanzado

Ejemplos de request

- **cURL**

```
curl --location --request POST
'https://www.myrb.io/dev/api/formData/getQueue/:id/:token' \
--header 'Authorization: Bearer <token>'
```

- **Python**

```
import requests

url = "https://www.myrb.io/dev/api/formData/getQueue/:id/:token"

payload = {}
headers = {
  'Authorization': 'Bearer <token>'
}

response = requests.request("POST", url, headers=headers, data=payload)

print(response.text)
```

Ejemplo de response

```
{
  "success": true,
  "data": {
    "id": 8728,
    "data": "{\"Dato1\":\"prueba\",\"Dato2\":\"prueba1\"}",
    "user_form_email": null,
    "xperience":
"eyJpdjI6InN0eXhRekhVUUJrWVptcGpzOUU1aXc9PSIsInZhbnVlIjoiY1NoMGtpcW5udk5vTmZx
VlVYZFg1V29CUGU4SDB1UVUwWGFuYVNmb2FVNkE2OE5MMnRwclorWGxtMU9VZ2pjTGJnSUNLaE5cL
1ArSklpPdk1BendRYVlvtlMyU0QwclN0E2VENmSDEyZTJRPSIsIm1hYyI6ImYyYjc2ZmM3OTAwMW
QzNDVlMmViNmM1NTIxNjc0OTc0Y2IwOTA2Yjg4YTMwODNlODUwNzU1NWQ4MmM2ZGRmMjcifQ=="
  }
}
```

Obtener Asset

Se utiliza para solicitar información de un asset almacenado en el orquestador. La búsqueda se realiza filtrando por nombre e instancia a la que pertenece.

Método URL	Formato	Parámetros
POST	https://www.myrb.io/dev/api/assets/get	Authorization: tipo Bearer Token; es el Token de acceso válido para autenticar al usuario (API Key asociada al perfil en el orquestador). name: El nombre del asset. instance: Key de la instancia, tal como se define en el archivo noc.ini.

Avanzado

Ejemplos de request

- cURL

```
curl --location 'https://www.myrb.io/dev/api/assets/get' \
--header 'Authorization: Bearer <token>' \
--data-urlencode 'name=toNotify' \
```

```
--data-urlencode 'instance=66b1418f93213'
```

- **Python**

```
import requests
```

```
url = "https://www.myrb.io/dev/api/assets/get"
```

```
payload = 'name=toNotify&instance=66b1418f93213'
```

```
headers = {  
    'Authorization': 'Bearer <token>',  
    'Content-Type': 'application/x-www-form-urlencoded'  
}
```

```
response = requests.request("POST", url, headers=headers, data=payload)
```

```
print(response.text)
```

Ejemplo de response

```
{  
    "success": true,  
    "data": {  
        "id": 31,  
        "name": "toNotify",  
        "type": "password",  
        "value": "prueba@prueba.com",  
        "client_id": 3,  
        "process_id": 0,  
        "instance_id": 0,  
        "created_at": "2023-03-21 17:12:35",  
        "updated_at": "2023-10-25 12:21:49"  
    }  
}
```