Integration & E2E test Plan wallbox-challenge-2022

**Contents**

[1. Purpose 3](#_Toc98522505)

[2. SCOPE 4](#_Toc98522506)

[3. Reference Documents 5](#_Toc98522507)

[4. Environmental needs 6](#_Toc98522508)

[4.2. ENV FILE 7](#_Toc98522509)

[4.3. Installation node and npm –lts version 8](#_Toc98522510)

[4.4. Install dependencies 9](#_Toc98522511)

[4.5. run application 9](#_Toc98522512)

[5. integration Test case specifications 11](#_Toc98522513)

[5.1. WB\_INT001 - Comprobar el estado de API Rest 11](#_Toc98522514)

[5.2. WB\_INT002-AutorizaTión signin 12](#_Toc98522515)

[5.3. WB\_INT003-Get Users 13](#_Toc98522516)

[5.4. WB\_INT004-POST chargers 15](#_Toc98522517)

[5.5. WB\_INT004 – GET Chargers 16](#_Toc98522518)

[5.6. WB\_INT005-PUT chargers 17](#_Toc98522519)

[6. end2end test plan 20](#_Toc98522520)

[6.1. WB\_E2E001-POST charger into Data Base 20](#_Toc98522521)

[7. Test REPORT 23](#_Toc98522522)

[7.1. INTEGRATION & E2E TEST REPORT 23](#_Toc98522523)

# Purpose

This document describes the plans for testing the integration between components, to ensure that they are compatible, using Postman. The tarjet is to ensure basic correctness for the REST API.

# SCOPE

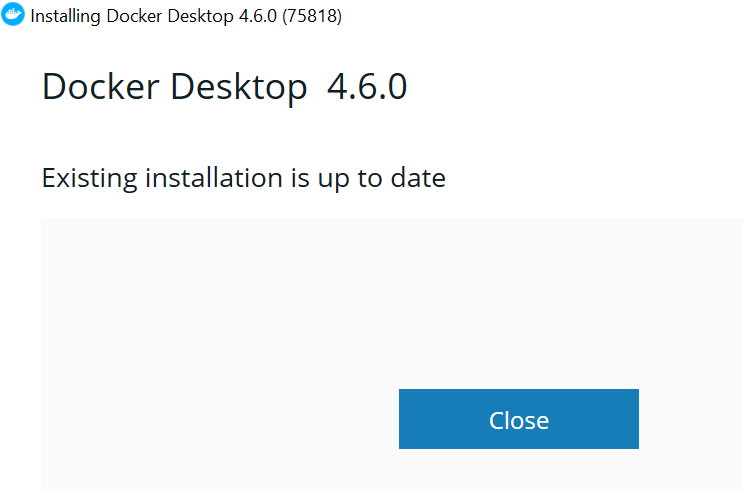
The software is a demo api Rest to users and chargers management. Our scope is to test this API to ensure Quality Software.

# Reference Documents

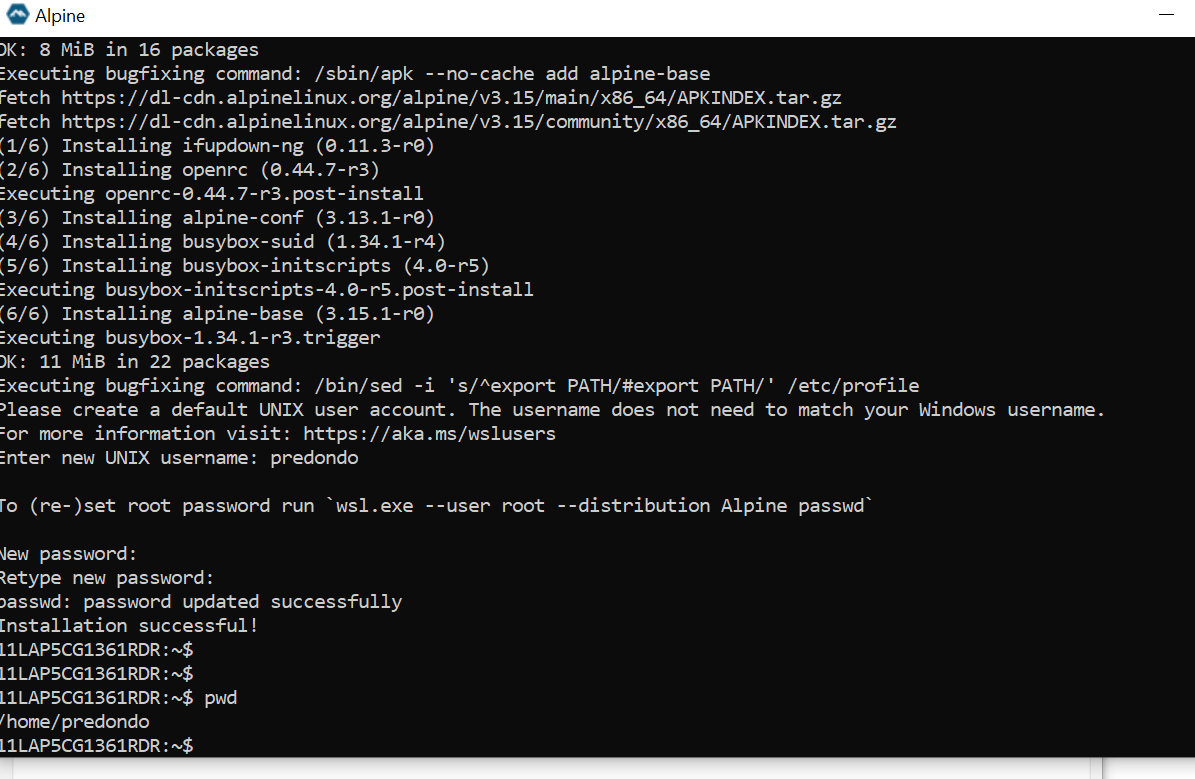
|  |  |
| --- | --- |
| [BQAWC] | Backend\_QA\_-\_Wallbox\_Challenge.pdf |
|  |  |

# Environmental needs

* 1. **INSTALACIÓN DOCKER**



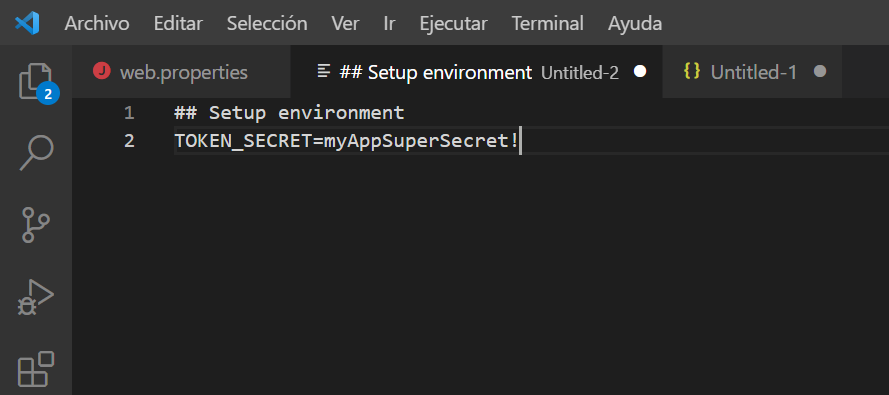
Installing Alpine Linux for WSL



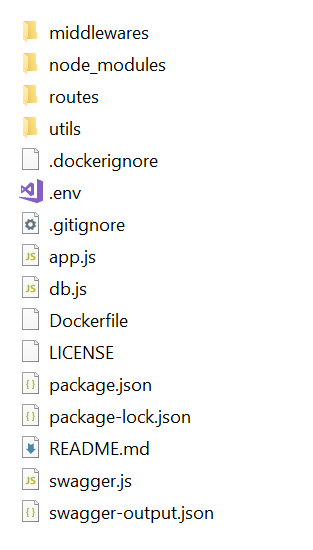
## ENV FILE

Contains de environment variables. Put .env file in the root path

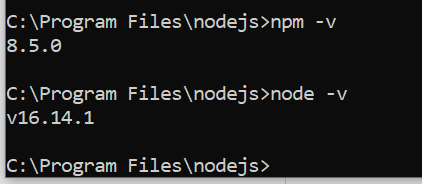
Using visual studio

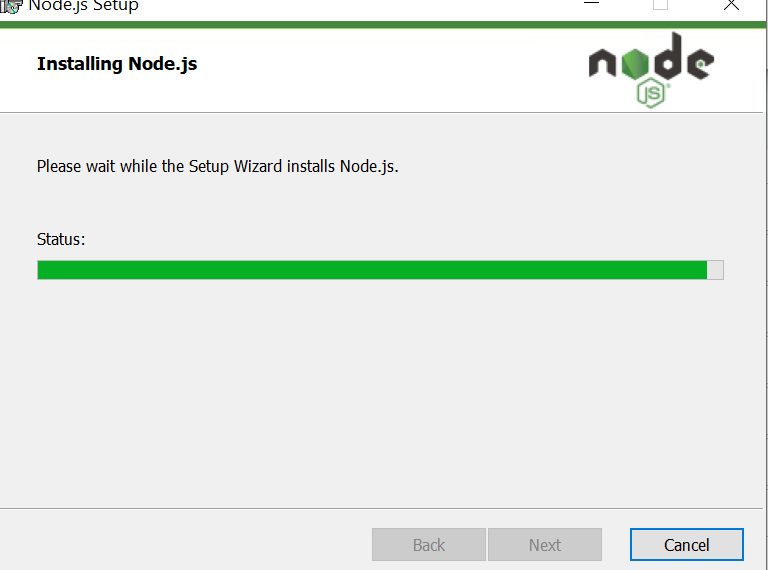


And save it

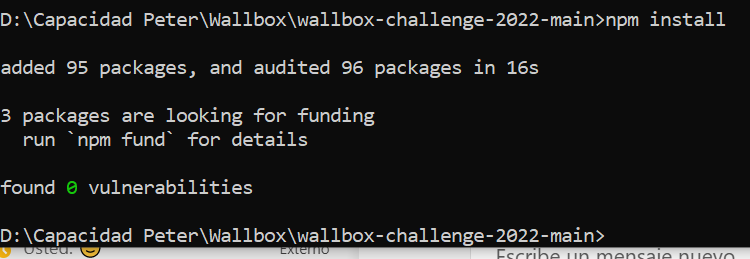


## Installation node and npm –lts version



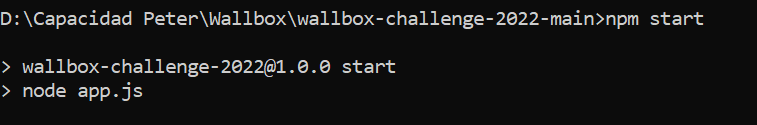


## Install dependencies



## run application

npm start



# integration Test case specifications

## WB\_INT001 - Comprobar el estado de API Rest

1. **Objetivo**

Comprobar que la API REST esta en ejecución

1. **Precondiciones**

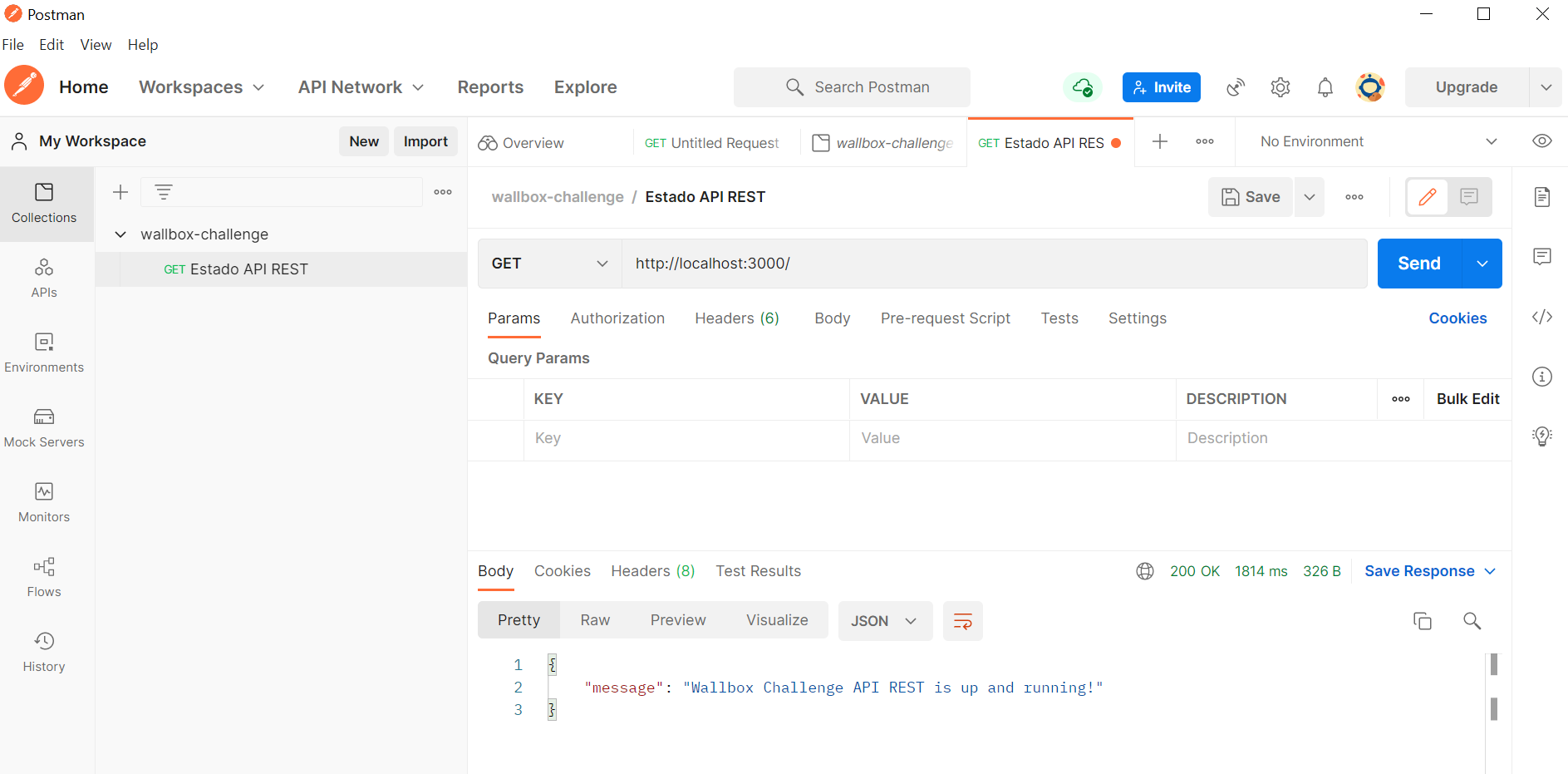
* El usuario tiene instalado npm y nod y configurado el proyecto
* Se ha decargado el proyecto a la carpeta wallbox-challenge-2022-main
* Se ha arrancado desde la línea de comandos la aplicación
* Postman arrancado

|  |  |
| --- | --- |
| **Paso** | **Resultado esperado** |
| 1. Operación GET – URL: localhost:3000/ | 200 - "message": "Wallbox Challenge API REST is up and running!" |
|  |  |
|  |  |

1. **Execution**

Base URL: localhost:3000/

GET <http://localhost:3000/>



Request in Python:

**import** requests

url = "http://localhost:3000/"

payload={}

headers = {}

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

**print**(response.text)

## WB\_INT002-AutorizaTión signin

1. **Objetivo**

Comprobar que nos logamos en la aplicación y obtenemos el token de session

1. **Precondiciones**

* El usuario tiene instalado npm y nod y configurado el proyecto
* Se ha decargado el proyecto a la carpeta wallbox-challenge-2022-main
* Se ha arrancado desde la línea de comandos la aplicación
* Postman arrancado

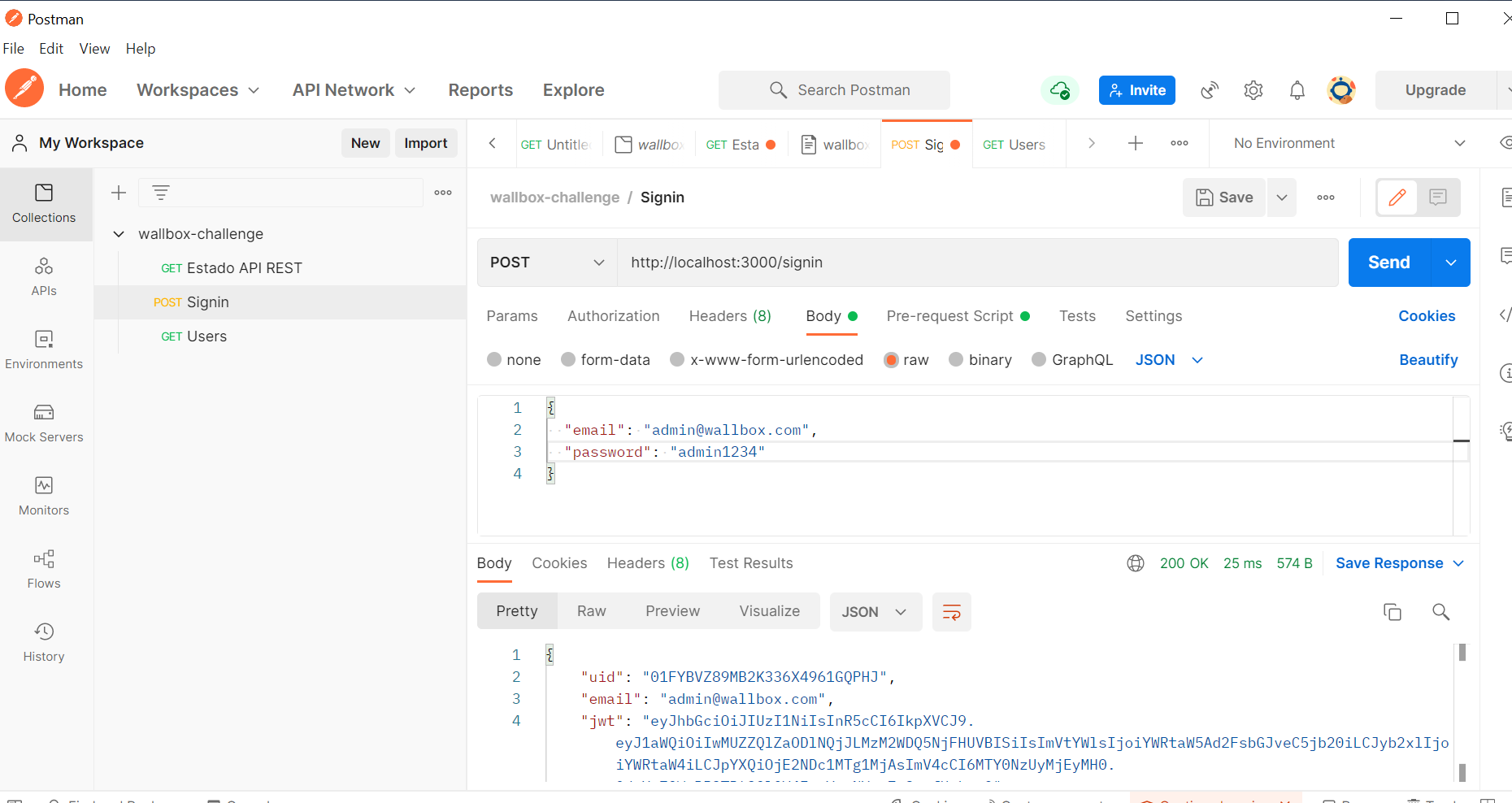
|  |  |  |
| --- | --- | --- |
| **Step** | **Params** | **Resultado esperado** |
| 1. POST – URL: localhost:3000/signin | email  password | 200 - uid, email, jwt |
|  |  |  |

1. **Execution**

Base URL: localhost:3000/

POST <http://localhost:3000/>**siggin**

Firs step: we execute POST



Body:

{

  "email": "admin@wallbox.com",

  "password": "admin1234"

}

Response:

{

    "uid": "01FYBVZ89MB2K336X4961GQPHJ",

    "email": "admin@wallbox.com",

    "jwt": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1aWQiOiIwMUZZQlZaODlNQjJLMzM2WDQ5NjFHUVBISiIsImVtYWlsIjoiYWRtaW5Ad2FsbGJveC5jb20iLCJyb2xlIjoiYWRtaW4iLCJpYXQiOjE2NDc1MTg1MjAsImV4cCI6MTY0NzUyMjEyMH0.OdpXo7QWyBR2TBt2QlQV4EpqYwnNVgg7x8qefUzLss0"

}

## WB\_INT003-Get Users

1. **Objetivo**

Comprobar que nos logamos en la aplicación y obtenemos el token de session

1. **Precondiciones**

* El usuario tiene instalado npm y nod y configurado el proyecto
* Se ha decargado el proyecto a la carpeta wallbox-challenge-2022-main
* Se ha arrancado desde la línea de comandos la aplicación
* Postman arrancado
* Sigin done

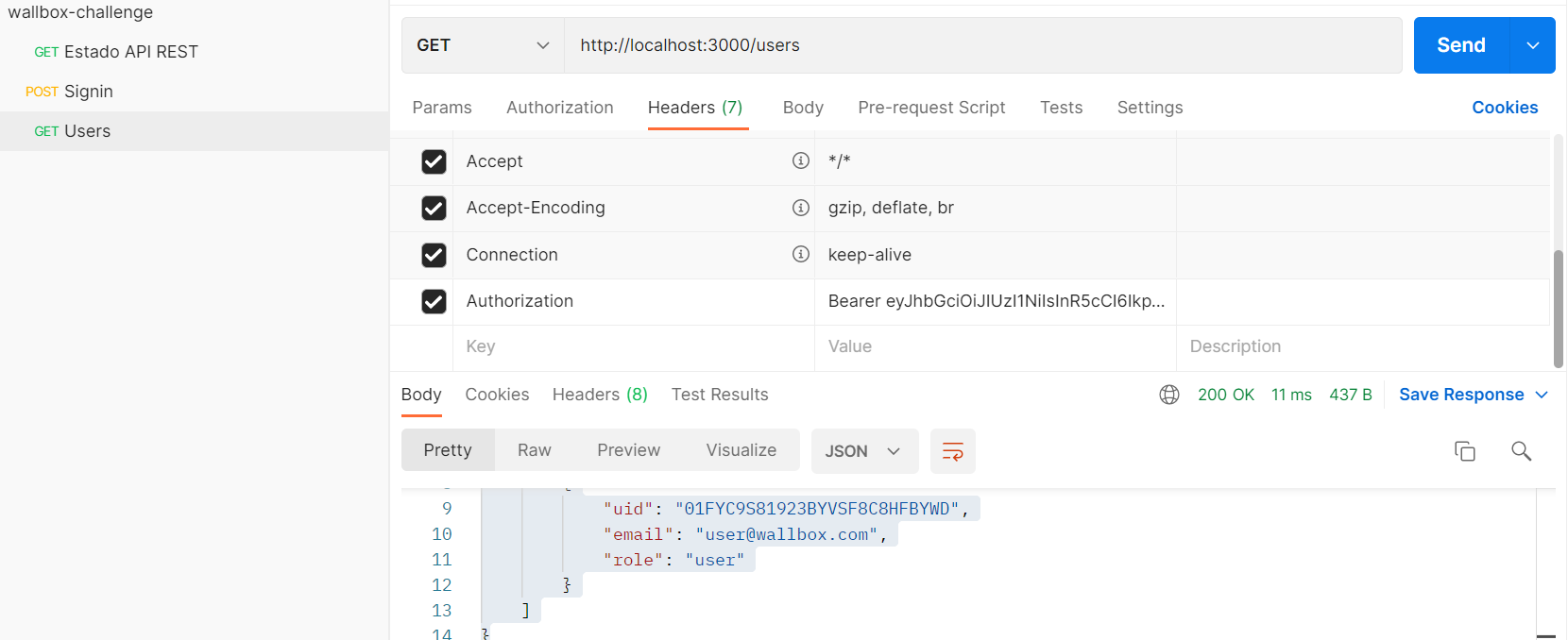
1. **Execution**

Base URL: localhost:3000/

Firs step: we execute GET

GET <http://localhost:3000/>**users**

|  |  |  |
| --- | --- | --- |
| **Step** | **Params** | **Resultado esperado** |
| 1. GET http://localhost:3000/users | -H 'Authorization: Bearer {adminaccesstoken} | Uid, email, role |
|  |  |  |
|  |  |  |



    "users": [

        {

            "uid": "01FYC9S818MJX7ZM1ZCGWJV4TC",

            "email": "admin@wallbox.com",

            "role": "admin"

        },

        {

            "uid": "01FYC9S81923BYVSF8C8HFBYWD",

            "email": "user@wallbox.com",

            "role": "user"

        }

    ]

}

## WB\_INT004-POST chargers

1. **Objetivo**

Comprobar que damos de alta en la sesión un nuevo charguer

1. **Precondiciones**

* El usuario tiene instalado npm y nod y configurado el proyecto
* Se ha decargado el proyecto a la carpeta wallbox-challenge-2022-main
* Se ha arrancado desde la línea de comandos la aplicación
* Postman arrancado
* Token obtenido

|  |  |  |
| --- | --- | --- |
| **Step** | **Params** | **Resultado esperado** |
| 1. POST – URL: localhost:3000/chargers | serialNumber  model | 200 – uid, serialNumber, model |
|  |  |  |
|  |  |  |

1. **Execution**

Base URL: localhost:3000/

POST http://**localhost:3000/chargers/**

Step: we execute POST

Body:

{

  "serialNumber": "12345678",

  "model": "Pulsar Plus"

}

Response:

{

    "message": "Charger registered",

    "charger": {

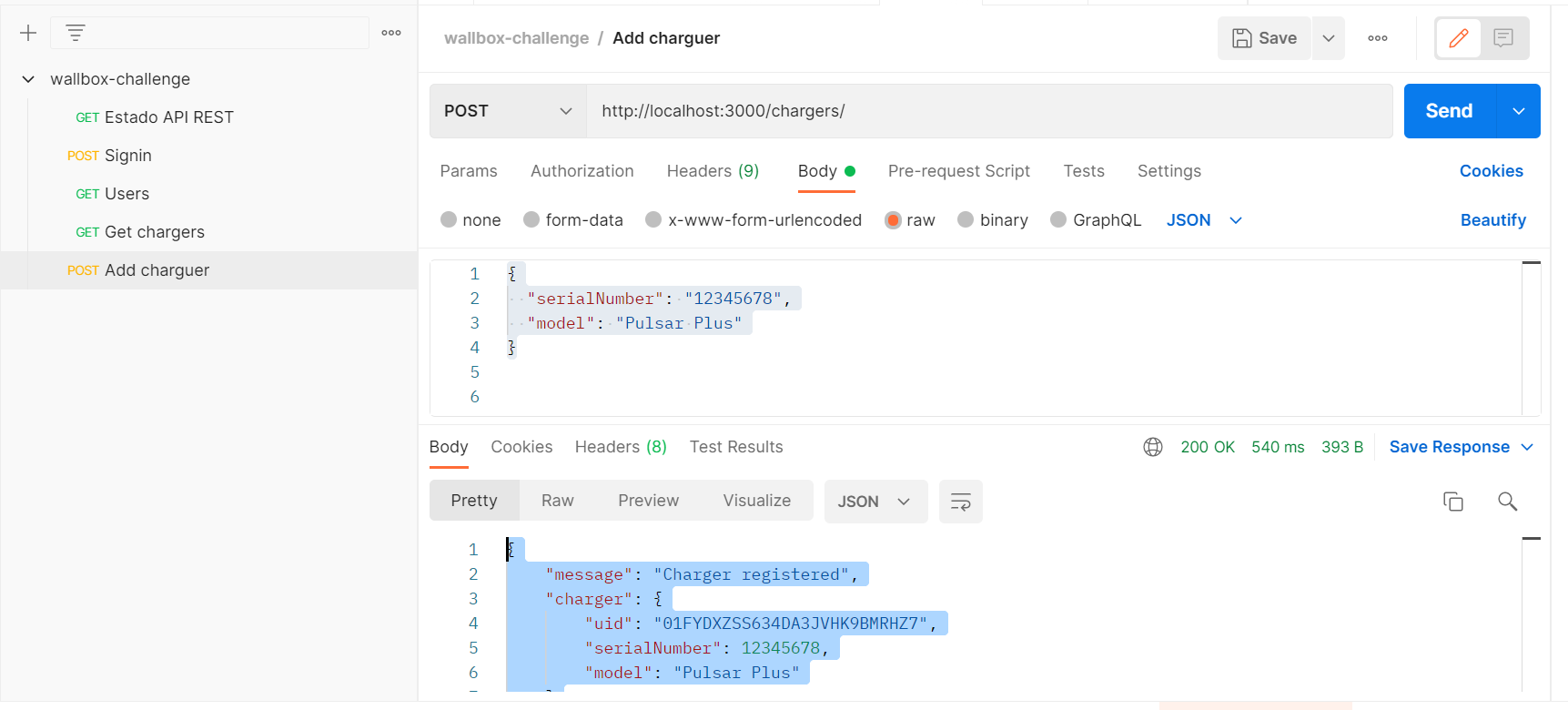
        "uid": "01FYDXZSS634DA3JVHK9BMRHZ7",

        "serialNumber": 12345678,

        "model": "Pulsar Plus"

    }

}



## WB\_INT004 – GET Chargers

1. **Objetivo**

Comprobar que se obtiene la lista de chargers

1. **Precondiciones**

* El usuario tiene instalado npm y nod y configurado el proyecto
* Se ha decargado el proyecto a la carpeta wallbox-challenge-2022-main
* Se ha arrancado desde la línea de comandos la aplicación
* Postman arrancado
* Ejecutado POST chargers

|  |  |
| --- | --- |
| **Paso** | **Resultado esperado** |
| 1. GET – URL: localhost:3000/chargers | 200 – uid, serialNumber, model |
|  |  |
|  |  |

1. **Execution**

Base URL: localhost:3000/

GET [http://localhost:3000/**chargers**](http://localhost:3000/chargers)

Body:

{

  "serialNumber": "1111111",

  "model": "Pulsar Plus"

}

Output:

{

    "chargers": [

        {

            "uid": "01FYDXZSS634DA3JVHK9BMRHZ7",

            "serialNumber": 1111111,

            "model": "Pulsar Plus"

        },

        {

            "uid": "01FYEZK4TEQTYD89NHG7MQ4F8C",

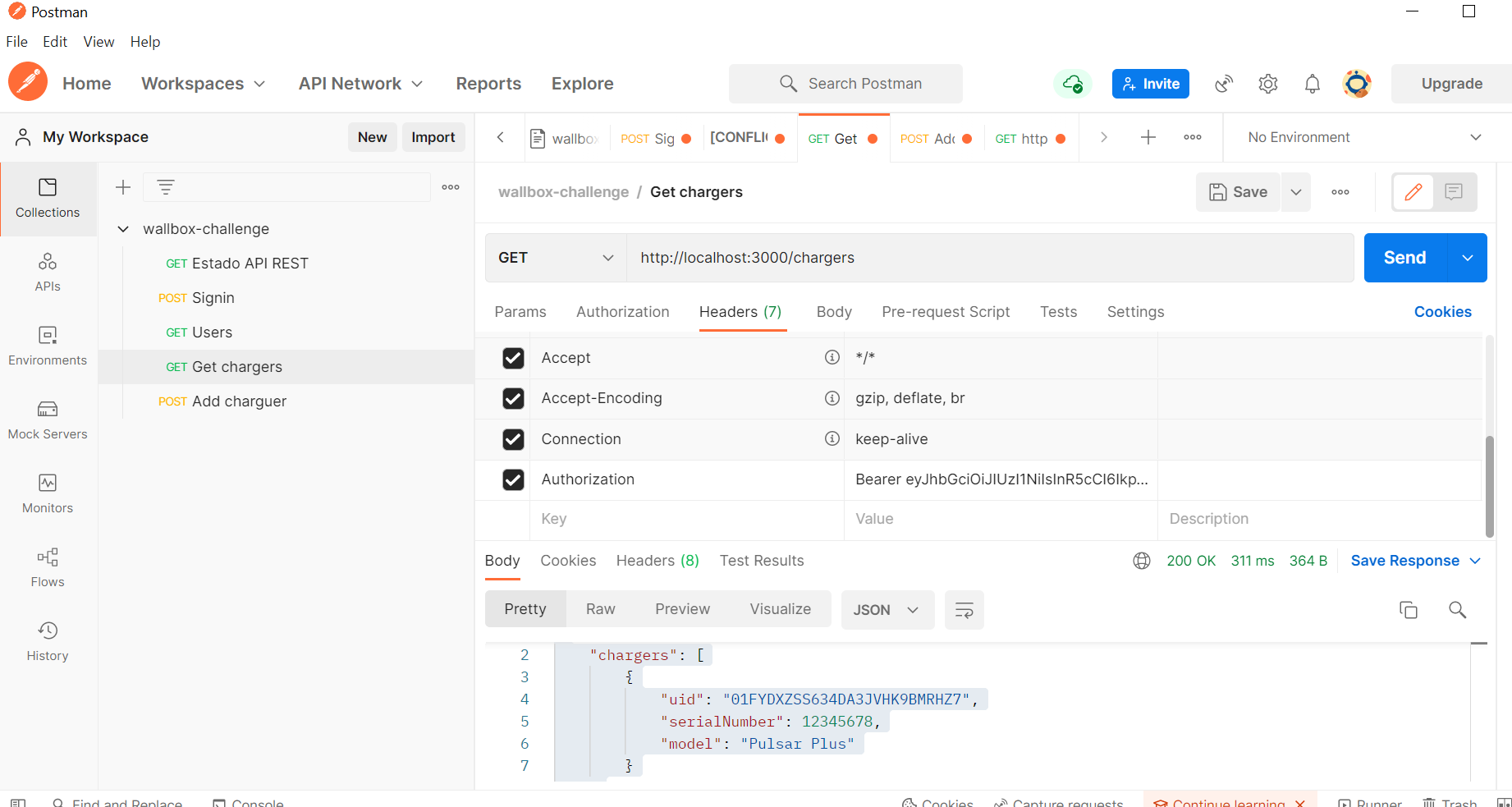
            "serialNumber": 2345678,

            "model": "Pulsar Plus"

        }

    ]

}



## WB\_INT005-PUT chargers

1. **Objetivo**

Comprobar que modificamos en la sesión un nuevo charguer

1. **Precondiciones**

* El usuario tiene instalado npm y nod y configurado el proyecto
* Se ha decargado el proyecto a la carpeta wallbox-challenge-2022-main
* Se ha arrancado desde la línea de comandos la aplicación
* Postman arrancado
* Token obtenido
* Chargers into BBDD

|  |  |  |
| --- | --- | --- |
| **Step** | **Params** | **Resultado esperado** |
| 1. PUT – URL: localhost:3000/chargers/{uidcharger} | serialNumber  model | 200 – uid, serialNumber, model |
|  |  |  |
|  |  |  |

1. **Execution**

Base URL: localhost:3000/

PUT http://**localhost:3000/chargers/01FYDXZSS634DA3JVHK9BMRHZ7**

Step1: we execute PUT

Body:

{

  "serialNumber": "1111111",

  "model": "Pulsar Plus"

}

Response:

{

    "message": "Charger updated",

    "charger": {

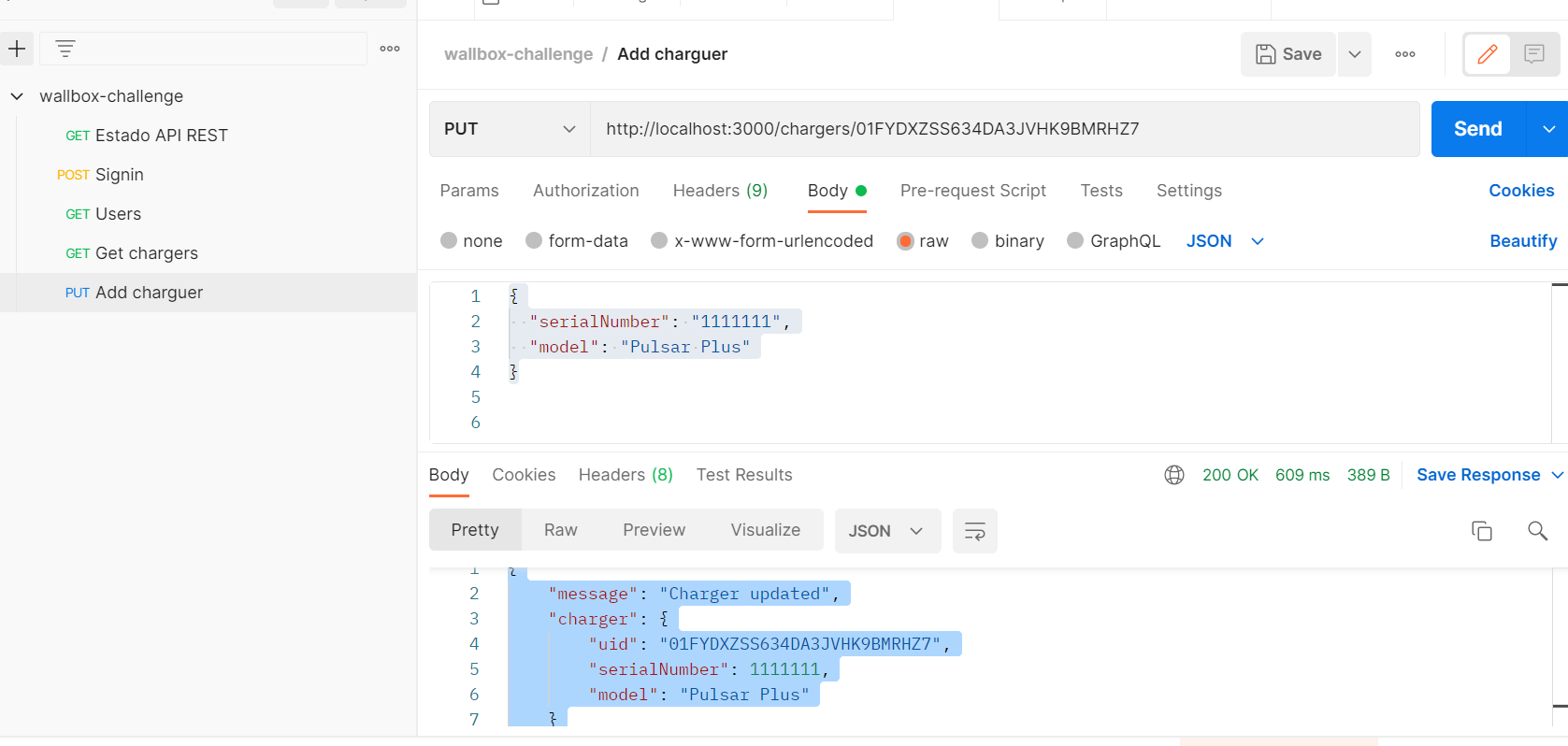
        "uid": "01FYDXZSS634DA3JVHK9BMRHZ7",

        "serialNumber": 1111111,

        "model": "Pulsar Plus"

    }

}



# end2end test plan

## WB\_E2E001-POST charger into Data Base

1. **Objetivo**

***Automation*** testint - The tarjet of this E2E test Plan is to execute all the rest API steps to post a new charger into the Data base from the begining to the end.

All off the steps will be executed using test tab into postman to automate Test

1. **Precondiciones**

* El usuario tiene instalado npm y nod y configurado el proyecto
* Se ha decargado el proyecto a la carpeta wallbox-challenge-2022-main
* Se ha arrancado desde la línea de comandos la aplicación
* Postman arrancado
* Token obtenido

|  |  |  |
| --- | --- | --- |
| **Step** | **Params** | **Resultado esperado** |
| 1. POST – URL: localhost:3000/siginn | Email, password | 200 - Uid, email, jwt  Tab Test: Status test |
| 1. POST – URL: localhost:3000/chargers | serialNumber  model | 200 – uid, serialNumber, model |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

1. **Execution**

Base URL: localhost:3000/

* **Step1**: we execute POST http://**localhost:3000/signin**

Automate test code:

pm.**test**("response must be valid and have a body", **function** () {

     pm.response.to.have.status(200);

     pm.response.to.be.withBody;

     pm.response.to.be.json;

});

Body:

{

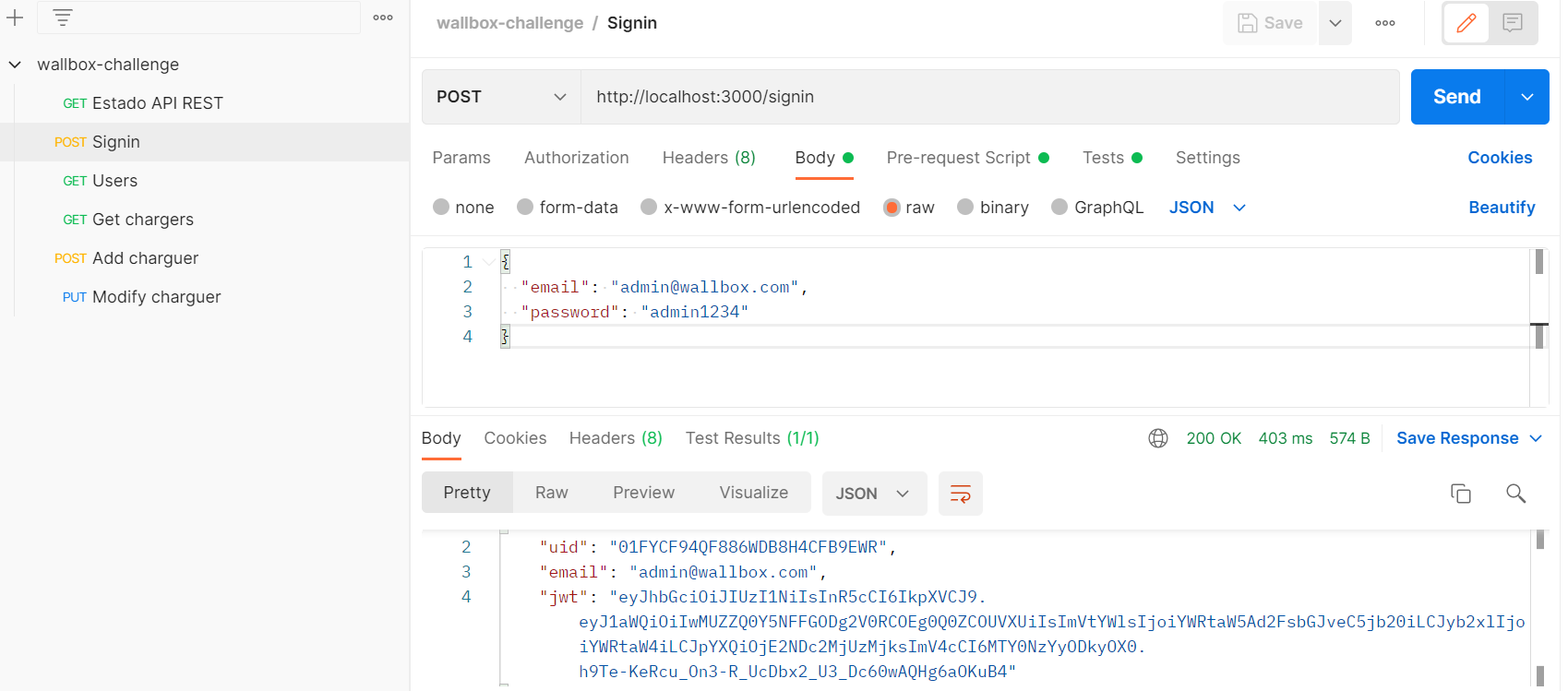
  "email": "admin@wallbox.com",

  "password": "admin1234"

}

Test Result: PASS response must be valid and have a body





* **Step2**: we execute POST – URL: http://localhost:3000/chargers

Automate test code:

pm.**test**("response should be okay to process", **function**  () {

pm.response.to.not.be.error;

pm.response.to.have.jsonBody("");

pm.response.to.not.have.jsonBody("error");

});

pm.**test**("response must be valid and have a body", **function** () {

     pm.response.to.have.status(200);

     pm.response.to.be.withBody;

     pm.response.to.be.json;

});

Body:

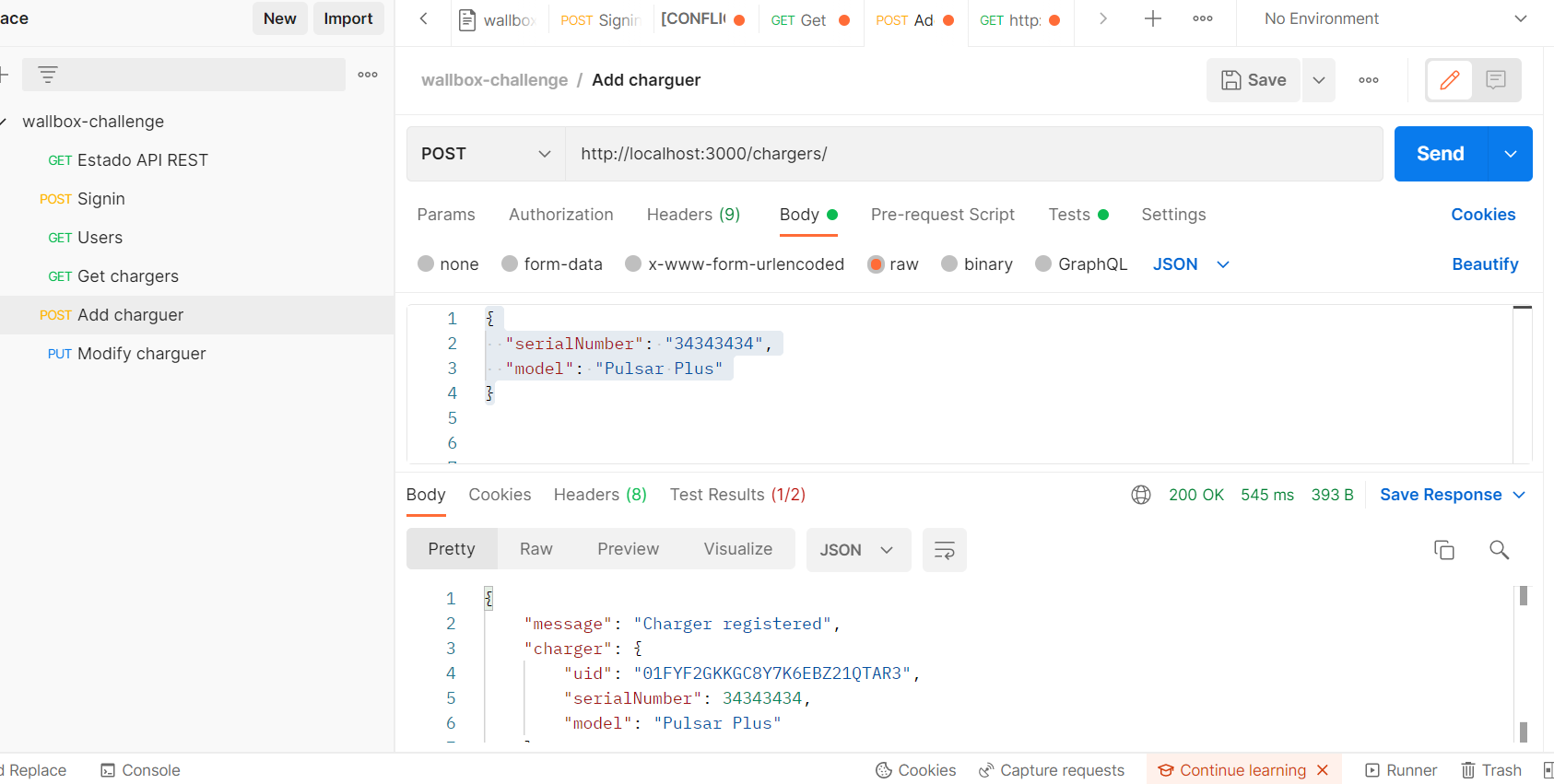
{

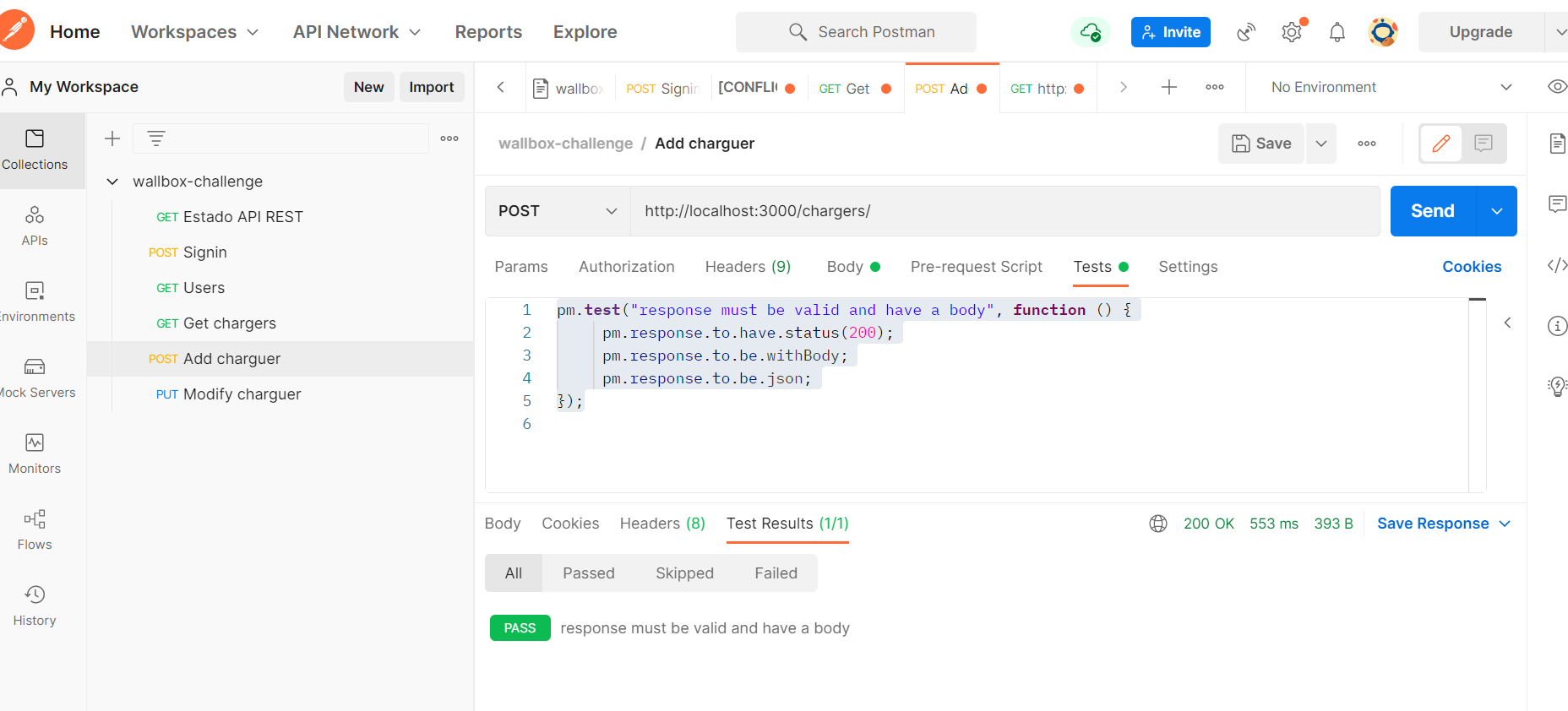
  "serialNumber": "34343434",

  "model": "Pulsar Plus"

}

Test Result: PASS response must be valid and have a body





# Test REPORT

## INTEGRATION & E2E TEST REPORT

WB\_INT001 successfully been executed by Pedro Redondo on March 18th, 2022.

WB\_INT002 successfully been executed by Pedro Redondo on March 18th, 2022.

WB\_INT003 successfully been executed by Pedro Redondo on March 18th, 2022.

WB\_INT004 successfully been executed by Pedro Redondo on March 18th, 2022.

WB\_INT005 successfully been executed by Pedro Redondo on March 18th, 2022.

WB\_E2E001 successfully been executed by Pedro Redondo on March 18th, 2022