**Pruebas API PERFDOG**

1. **Crear Usuario**:

**Curl**: **https://petstore.swagger.io/v2/user**

**Método:** POST

**BODY**:

{

  "id": 0,

  "username": "TestDos",

  "firstName": "UserTest",

  "lastName": "Test",

  "email": "emailfalso2@test.com",

  "password": "Prueba+1234",

  "phone": "3644123456",

  "userStatus": 0

}

Resultado:

{

    "code": 200,

    "type": "unknown",

    "message": "9223372036854758134"

}

Usuario creado correctamente, el mensaje devuelve un ID

Test Crear Usuario:

Post-Req:

pm.test("Usuario creado con éxito", *function* () {

    pm.response.to.have.status(200);

});

pm.collectionVariables.set("username", "testUser");

Resultado: PASSED Usuario creado con éxito

1. Loguear con el usuario recién creado:

**Curl**: [**https://petstore.swagger.io/v2/user/login**](https://petstore.swagger.io/v2/user/login)

**Método:** GET

**BODY**:

{

    "username": "TestDos",

    "password": "Prueba+1234"

}

Resultado:

{

    "code": 200,

    "type": "unknown",

    "message": "logged in user session:1742491823151"

}

**Test Login**:

Post Req:

// Verificar que el código de estado sea ok 200

pm.test("Status code is 200", *function* () {

    pm.response.to.have.status(200);

});

// Validar si devuelve un mensaje de confirmacion ok

pm.test("Logueo exitoso", *function* () {

*const* response = pm.response.json();

    pm.expect(response.message).to.include("logged in");

});

Resultado:

PASSED Status code is 200

PASSED Logueo exitoso

1. Listar Disponibles:

**Curl**: **https://petstore.swagger.io/v2/pet/findByStatus?status=availableMétodo:** POST

**Params**:

Key Value

Status available

Resultado:

[

    {

        "id": 9223372036854774401,

        "category": {

            "id": 0,

            "name": "string"

        },

        "name": "doggie",

        "photoUrls": [

            "string"

        ],

        "tags": [

            {

                "id": 0,

                "name": "string"

            }

        ],

        "status": "available"

    },

…… 200 OK

Test Listar Disponibles:

Post Req:

// ValidaR Operacion exitosa 200 ok

pm.test("Status code is 200", *function* () {

    pm.response.to.have.status(200);

});

// Validar que me devuelva al menos una mascota

pm.test("Mascotas disponibles", *function* () {

*const* response = pm.response.json();

    pm.expect(response).to.be.an("array").that.is.not.empty;

});

Resultado:

PASSED Status code is 200

PASSED Mascotas disponibles

1. Consultar Datos:

**Curl**: **https://petstore.swagger.io/v2/pet/702190001**

**Método:** GET

**By ID**

**Resultado:**

{

    "id": 702190001,

    "category": {

        "id": 1,

        "name": "dog"

    },

    "name": "lupin",

    "photoUrls": [

        "",

        ""

    ],

    "tags": [

        {

            "id": 1,

            "name": "vacinado"

        }

    ],

    "status": "available"

}

Test Consultar Datos:

Post Req:

// Validar 200 ok

pm.test("Status code is 200", *function* () {

    pm.response.to.have.status(200);

});

// Validar que mi mascota contenga el id del listado

pm.test("El ID de la mascota es correcto", *function* () {

*const* response = pm.response.json();

    pm.expect(response.id).to.eql(pm.variables.get("petId"));

});

Resultado:

PASSED Status code is 200

FAILED El ID de la mascota es correcto | AssertionError: expected 702190001 to deeply equal undefined

Obs: Test ok, resultado ok, No está configurado un mensaje de respuesta como “El ID de la mascota es correcto”, pero si trae los datos.

1. Crear Orden:

**Curl**: **https://petstore.swagger.io/v2/store/order**

**Método:** POST

**BODY**

**{**

**"id": 0,**

**"petId": 702190001,**

**"quantity": 0,**

**"shipDate": "2025-03-20T17:56:49.089Z",**

**"status": "placed",**

**"complete": true**

**}**

**Resultado:**

**{**

**"id": 9223372036854760337,**

**"petId": 702190001,**

**"quantity": 0,**

**"shipDate": "2025-03-20T17:56:49.089+0000",**

**"status": "placed",**

**"complete": true**

**}**

Test Crear Orden:

**Post Req:**

// validar el 200 ok

pm.test("Status code is 200", *function* () {

    pm.response.to.have.status(200);

});

// Verificar que la respuesta contenga un campo "status" con el valor "placed"

pm.test("Estado de la orden ok'", *function* () {

*const* response = pm.response.json();

    pm.expect(response.status).to.eql("placed");

});

Resultado:

PASSED Status code is 200

PASSED Estado de la orden ok'

1. Logout

**Curl**: **https://petstore.swagger.io/v2/user/logout**

**Método:** POST

**Resultado:**

**{**

**"code": 200,**

**"type": "unknown",**

**"message": "ok"**

**}**

Test Logout:

**Post Req:**

**// Validar el 200ok**

**pm.test("Status code is 200", *function* () {**

**pm.response.to.have.status(200);**

**});**

**// Validar el mensaje de desconexion exitosa**

**pm.test("Logout exitoso", *function* () {**

***const* response = pm.response.json();**

**pm.expect(response.message).to.eql("ok");**

**});**

**Resultado: PASSED Status code is 200**

**PASSED Logout exitoso**