**TEST CASES PETS STORE**

(PRUEBAS EXPLORATORIAS)

|  |  |
| --- | --- |
| Identificador | TC-P01 |
| Escenario | Agregar una mascota |
| Descripción | Como administrador de la tienda quiero agregar una nueva mascota para que se publique en la tienda \*que pretendemos hacer con este test\* |
| Precondiciones | - Tener postman instalado  - API disponible   * - Tener mascotas con estado sold |
| Pasos | 1. Ingresar a postman 2. Ingresar metodo 3. ingresar url 4. body (que datos usare en el body 5. enviar request |
| Resultado Esperado | - Al crear una nueva mascota debe arrojar status code 201 con la info de la mascota creada |
| Probador | Daniel |
| Estado | Exitoso |

|  |  |
| --- | --- |
| Identificador | TC-P02 |
| Escenario | Creación de lista de usuarios |
| Descripción | Creación de usuarios mediante la utilización del recurso  POST /user/createWithList |
| Precondiciones | * el API PetShop debe estar desplegado y accesible desde internet * tener instalado un cliente REST como Postman |
| Pasos | Consumir el recurso POST/ user/createWithList enviando información correcta en el body de la petición |
| Resultado Esperado | 200 ok |
| Probador | Daniel |
| Estado | Exitoso |

|  |  |
| --- | --- |
| Identificador | TC-P03 |
| Escenario | Creación de lista de usuarios |
| Descripción | Creación de usuarios mediante la utilización del recurso  POST /user/createWithList |
| Precondiciones | * el API PetShop debe estar desplegado y accesible desde internet * tener instalado un cliente REST como Postman |
| Pasos | Consumir el recurso POST/ user/createWithList enviando información errada en el id del body de la petición |
| Resultado Esperado | 500 Server Error |
| Probador | Daniel |
| Estado | Fallido |

|  |  |
| --- | --- |
| Identificador | TC-P04 |
| Escenario | Creación de lista de usuarios con matriz |
| Descripción | Creación de usuarios mediante la utilización del recurso  POST /user/createWithArray |
| Precondiciones | * el API PetShop debe estar desplegado y accesible desde internet * tener instalado un cliente REST como Postman |
| Pasos | Consumir el recurso POST /user/createWithArray enviando información valida en el body de la petición |
| Resultado Esperado | 200 ok |
| Probador | Daniel |
| Estado | Exitoso |

|  |  |
| --- | --- |
| Identificador | TC-P05 |
| Escenario | Creación de lista de usuarios con matriz |
| Descripción | Creación de usuarios mediante la utilización del recurso  POST /user/createWithArray |
| Precondiciones | * el API PetShop debe estar desplegado y accesible desde internet * tener instalado un cliente REST como Postman |
| Pasos | Consumir el recurso POST /user/createWithArray enviando información vacia en el campo ID del body de la petición |
| Resultado Esperado | 500 Server Error |
| Probador | Daniel |
| Estado | Fallido |

|  |  |
| --- | --- |
| Identificador | TC-P06 |
| Escenario | Creación de lista de usuarios con matriz |
| Descripción | Creación de usuarios mediante la utilización del recurso  POST /user |
| Precondiciones | * el API PetShop debe estar desplegado y accesible desde internet * tener instalado un cliente REST como Postman |
| Pasos | Consumir el recurso POST /user enviando información valida en el body de la petición |
| Resultado Esperado | 200 ok |
| Probador | Daniel |
| Estado | Exitoso |

|  |  |
| --- | --- |
| Identificador | TC-P07 |
| Escenario | Creación de lista de usuarios con matriz |
| Descripción | Creación de usuarios mediante la utilización del recurso  POST /user/createWithArray |
| Precondiciones | * El API PetShop debe estar desplegado y accesible desde internet * Tener instalado un cliente REST como Postman |
| Pasos | Consumir el recurso POST /user/createWithArray enviando información vacia en el campo ID del body de la petición |
| Resultado Esperado | 400 Bad Request |
| Probador | Daniel |
| Estado | Fallido |

|  |  |
| --- | --- |
| Identificador | TC-P08 |
| Escenario | Registra al usuario en el sistema |
| Descripción | Creación de usuarios mediante la utilización del recurso  GET /user/login?username=IDTEST&password=pass |
| Precondiciones | * El API PetShop debe estar desplegado y accesible desde internet * Tener instalado un cliente REST como Postman |
| Pasos | Consumir el recurso GET /user/login?username=IDTEST&password=pass enviando información correcta con el usuario y el password creados en los campos de Params |
| Resultado Esperado | 200 ok |
| Probador | Daniel |
| Estado | Exitoso |

|  |  |
| --- | --- |
| Identificador | TC-P09 |
| Escenario | Registra al usuario en el sistema |
| Descripción | Creación de usuarios mediante la utilización del recurso  GET /user/login?username=IDTEST&password=pass |
| Precondiciones | * El API PetShop debe estar desplegado y accesible desde internet * Tener instalado un cliente REST como Postman |
| Pasos | Consumir el recurso GET /user/createWithArray enviando información vacia en el campo ID del body de la petición |
| Resultado Esperado | 400 Bad Request |
| Probador | Daniel |
| Estado | Negativo |

|  |  |
| --- | --- |
| Identificador | TC-P10 |
| Escenario | Cierra la sesión actual del usuario conectado |
| Descripción | Cerrar Sesión mediante la utilización del recurso  GET /user/logout |
| Precondiciones | * El API PetShop debe estar desplegado y accesible desde internet * Tener instalado un cliente REST como Postman |
| Pasos | Consumir el recurso GET /user/logout sin parametros |
| Resultado Esperado | 200 ok |
| Probador | Daniel |
| Estado | Exitoso |

|  |  |
| --- | --- |
| Identificador | TC-P11 |
| Escenario | Obtener usuario por nombre de usuario |
| Descripción | Buscar usuarios por nombre de usuario mediante la utilización del recurso  GET /user/:username |
| Precondiciones | * El API PetShop debe estar desplegado y accesible desde internet * Tener instalado un cliente REST como Postman |
| Pasos | Consumir el recurso GET /user/:username con nombre de usuario creado y valido en Params |
| Resultado Esperado | 200 ok |
| Probador | Daniel |
| Estado | Exitoso |

|  |  |
| --- | --- |
| Identificador | TC-P12 |
| Escenario | Obtener usuario por nombre de usuario |
| Descripción | Buscar usuarios por nombre de usuario mediante la utilización del recurso  GET /user/:username |
| Precondiciones | * El API PetShop debe estar desplegado y accesible desde internet * Tener instalado un cliente REST como Postman |
| Pasos | Consumir el recurso GET /user/:username con nombre de usuario no creado e invalido en Params |
| Resultado Esperado | 404 Not Found |
| Probador | Daniel |
| Estado | Fallido |

|  |  |
| --- | --- |
| Identificador | TC-P13 |
| Escenario | Usuario actualizado |
| Descripción | Buscar usuarios por nombre de usuario mediante la utilización del recurso  PUT /user/:username |
| Precondiciones | * El API PetShop debe estar desplegado y accesible desde internet * Tener instalado un cliente REST como Postman |
| Pasos | Consumir el recurso PUT /user/:username con información valida a actualizar |
| Resultado Esperado | 200 ok |
| Probador | Daniel |
| Estado | Positiva |

|  |  |
| --- | --- |
| Identificador | TC-P14 |
| Escenario | Usuario actualizado |
| Descripción | Actualizar usuario por nombre de usuario mediante la utilización del recurso  PUT /user/:username |
| Precondiciones | * El API PetShop debe estar desplegado y accesible desde internet * Tener instalado un cliente REST como Postman |
| Pasos | Consumir el recurso PUT /user/:username con usuario invalido a actualizar |
| Resultado Esperado | 404 Not Found |
| Probador | Daniel |
| Estado | Negativo |

|  |  |
| --- | --- |
| Identificador | TC-P15 |
| Escenario | Borrar Usuario |
| Descripción | Borrar usuarios por nombre de usuario mediante la utilización del recurso  DELETE /user/:username |
| Precondiciones | * El API PetShop debe estar desplegado y accesible desde internet * Tener instalado un cliente REST como Postman |
| Pasos | Consumir el recurso DELETE /user/:username con usuario valido a borrar |
| Resultado Esperado | 200 ok |
| Probador | Daniel |
| Estado | Positiva |

|  |  |
| --- | --- |
| Identificador | TC-P16 |
| Escenario | Borrar Usuario |
| Descripción | Borrar usuarios por nombre de usuario mediante la utilización del recurso  DELETE /user/:username |
| Precondiciones | * El API PetShop debe estar desplegado y accesible desde internet * Tener instalado un cliente REST como Postman |
| Pasos | Consumir el recurso DELETE /user/:username con usuario invalido a borrar |
| Resultado Esperado | 404 Not Found |
| Probador | Daniel |
| Estado | Negativo |

|  |  |
| --- | --- |
| Identificador | TC-P17 |
| Escenario | Agregar una nueva mascota a la tienda |
| Descripción | Agregar nueva mascota mediante la utilización del recurso  POST /pet |
| Precondiciones | * El API PetShop debe estar desplegado y accesible desde internet * Tener instalado un cliente REST como Postman |
| Pasos | Consumir el recurso POST /pet con información valida en el body |
| Resultado Esperado | 200 ok |
| Probador | Daniel |
| Estado | Exitosa |

|  |  |
| --- | --- |
| Identificador | TC-P18 |
| Escenario | Agregar una nueva mascota a la tienda |
| Descripción | Agregar nueva mascota mediante la utilización del recurso  POST /pet |
| Precondiciones | * El API PetShop debe estar desplegado y accesible desde internet * Tener instalado un cliente REST como Postman |
| Pasos | Consumir el recurso POST /pet con id errado |
| Resultado Esperado | 400 Bad content |
| Probador | Daniel |
| Estado | Negativo |

|  |  |
| --- | --- |
| Identificador | TC-P19 |
| Escenario | Actualizar una mascota existente |
| Descripción | Actualizar una mascota existente mediante la utilización del recurso  PUT /pet |
| Precondiciones | * El API PetShop debe estar desplegado y accesible desde internet * Tener instalado un cliente REST como Postman |
| Pasos | Consumir el recurso PUT /pet con datos validos en el body |
| Resultado Esperado | 200 ok |
| Probador | Daniel |
| Estado | Positiva |

|  |  |
| --- | --- |
| Identificador | TC-P20 |
| Escenario | Actualizar una mascota existente |
| Descripción | Actualizar una mascota existente mediante la utilización del recurso  PUT /pet |
| Precondiciones | * El API PetShop debe estar desplegado y accesible desde internet * Tener instalado un cliente REST como Postman |
| Pasos | Consumir el recurso PUT /pet con datos no validos en el body |
| Resultado Esperado | 400 Bad Request |
| Probador | Daniel |
| Estado | Negativo |

|  |  |
| --- | --- |
| Identificador | TC-P21 |
| Escenario | Encontrar mascotas por estado |
| Descripción | Encontrar mascotas por estado existente mediante la utilización del recurso  GET /pet/findByStatus?status=available&status=available |
| Precondiciones | * El API PetShop debe estar desplegado y accesible desde internet * Tener instalado un cliente REST como Postman |
| Pasos | Consumir el recurso GET /pet/findByStatus?status=available&status=available con datos validos en el params |
| Resultado Esperado | 200 ok |
| Probador | Daniel |
| Estado | Positivo |

|  |  |
| --- | --- |
| Identificador | TC-P22 |
| Escenario | Encontrar mascotas por estado |
| Descripción | Encontrar mascotas por estado existente mediante la utilización del recurso  GET /pet/findByStatus?status=available&status=available |
| Precondiciones | * El API PetShop debe estar desplegado y accesible desde internet * Tener instalado un cliente REST como Postman |
| Pasos | Consumir el recurso GET /pet/findByStatus?status=available&status=available con datos no validos en el params |
| Resultado Esperado | 400 Bad Request |
| Probador | Daniel |
| Estado | Negativo |

|  |  |
| --- | --- |
| Identificador | TC-P23 |
| Escenario | Encontrar mascotas por etiquetas |
| Descripción | Encontrar mascotas por etiquetas existente mediante la utilización del recurso  GET /pet/findByTags?tags=PetTest |
| Precondiciones | * El API PetShop debe estar desplegado y accesible desde internet * Tener instalado un cliente REST como Postman |
| Pasos | Consumir el recurso GET /pet/findByStatus?status=available&status=available con datos validos en el params |
| Resultado Esperado | 200 ok |
| Probador | Daniel |
| Estado | Positiva |

|  |  |
| --- | --- |
| Identificador | TC-P24 |
| Escenario | Encontrar mascotas por etiquetas |
| Descripción | Encontrar mascotas por etiquetas existente mediante la utilización del recurso  GET /pet/:petId |
| Precondiciones | * El API PetShop debe estar desplegado y accesible desde internet * Tener instalado un cliente REST como Postman |
| Pasos | Consumir el recurso GET /pet/:petId con tags no validos en el params |
| Resultado Esperado | 404 Not Found |
| Probador | Daniel |
| Estado | Negativo |

|  |  |
| --- | --- |
| Identificador | TC-P25 |
| Escenario | Encontrar mascotas por ID |
| Descripción | Encontrar mascotas por ID existente mediante la utilización del recurso  GET /pet/:petId |
| Precondiciones | * El API PetShop debe estar desplegado y accesible desde internet * Tener instalado un cliente REST como Postman |
| Pasos | Consumir el recurso GET /pet/:petId con ID valido en el params |
| Resultado Esperado | 200 ok |
| Probador | Daniel |
| Estado | Positivo |

|  |  |
| --- | --- |
| Identificador | TC-P26 |
| Escenario | Encontrar mascotas por ID |
| Descripción | Encontrar mascotas por ID existente mediante la utilización del recurso  GET /pet/:petId |
| Precondiciones | * El API PetShop debe estar desplegado y accesible desde internet * Tener instalado un cliente REST como Postman |
| Pasos | Consumir el recurso GET /pet/:petId con ID no valido en el params |
| Resultado Esperado | 404 Not Found |
| Probador | Daniel |
| Estado | Negativo |

|  |  |
| --- | --- |
| Identificador | TC-P27 |
| Escenario | Encontrar mascotas por ID |
| Descripción | Encontrar mascotas por ID existente mediante la utilización del recurso  GET /pet/:petId |
| Precondiciones | * El API PetShop debe estar desplegado y accesible desde internet * Tener instalado un cliente REST como Postman |
| Pasos | Consumir el recurso GET /pet/:petId con ID no valido en el params |
| Resultado Esperado | 404 Not Found |
| Probador | Daniel |
| Estado | Negativo |