

	Checking	Status	Test data and Coments
	Check requests with Post /user		
1	Check that the request is sent with valid data in all fields	not run	{ "id": 0, "username": "user2", "firstName": "Bella", "lastName": "Chao", "email": "bellachao@post.com", "password": "1234", "phone": "123456", "userStatus": 0 }
2	Check data format (username in wrong format)	not run	{ "id": 0, "username": "user2", "firstName": "Bella", "lastName": "Chao", "email": "bellachao@post.com", "password": "1234", "phone": "123456", "userStatus": 0 }
3	Check data format (firstName in wrong format)	not run	{ "id": 0, "username": "user2", "firstName": null, "lastName": "Chao", "email": "bellachao@post.com", "password": "1234", "phone": "123456", "userStatus": 0 }
4	Check data format (lastName is in wrong format)	not run	{ "id": 0, "username": "user2", "firstName": "Bella", "@#12445: "Chao", "email": "bellachao@post.com", "password": "1234", "phone": "123456", "userStatus": 0 }
5	Check the data format (email is in the wrong format)	not run	{ "id": 0, "username": "user2", "firstName": "Bella", "lastName": "Chao", "email": " bellachaopost.com ", "password": "1234", "phone": "123456", "userStatus": 0 }
6	Check the data format (email is in the wrong format)	not run	{ "id": 0, "username": "user2", "firstName": "Bella", "lastName": "Chao", "email": "748726486348623", "password": "1234", "phone": "123456", "userStatus": 0 }
7	Check the data format (phone is in the wrong format)	not run	{ "id": 0, "username": "user2", "firstName": "Bella", "lastName": "Chao", "email": "bellachao@post.com", "password": "1234", "phone": "одиндватричотирип'ятьшість", "userStatus": 0 }
8	Check the data format (phone is in the wrong format)	not run	{ "id": 0, "username": "user2", "firstName": "Bella", "lastName": "Chao", "email": "bellachao@post.com", "password": "1234", "phone": true, "userStatus": 0 }
9	Check sending a request with an empty username value	not run	{ "id": 0, "": "user2", "firstName": "Bella", "lastName": "Chao", "email": "bellachao@post.com", "password": "1234", "phone": "123456", "userStatus": 0 }
10	Test sending a request with an empty firstName value	not run	{ "id": 0, "username": "user2", "firstName": "", "lastName": "Chao", "email": "bellachao@post.com", "password": "1234", "phone": "123456", "userStatus": 0 }
11	Check sending a request with an empty lastName value	not run	{ "id": 0, "username": "user2", "firstName": "Bella", "lastName": "", "email": "bellachao@post.com", "password": "1234", "phone": "123456", "userStatus": 0 }
12	Check sending a request with an empty email value	not run	{ "id": 0, "username": "user2", "firstName": "Bella", "lastName": "Chao", "email": "", "password": "1234", "phone": "123456", "userStatus": 0 }

13	Check sending a request with an empty password value	not run	{ "id": 0, "username": "user2", "firstName": "Bella", "lastName": "Chao", "email": "bellachao@post.com", "password": "", "phone": "123456", "userStatus": 0 }
14	Check sending a request with an empty phone value	not run	{ "id": 0, "username": "user2", "firstName": "Bella", "lastName": "Chao", "email": "bellachao@post.com", "password": "1234", "phone": "", "userStatus": 0 }
15	Send request with empty json	not run	
16	Rearranging fields in the request body	not run	{ "id": 0, "username": "user2", "phone": "123456", "firstName": "Bella", "lastName": "Chao", "password": "1234", "email": "bellachao@post.com", "userStatus": 0 }
17	Check the case sensitivity of the value	not run	{ "id": 0, "username": "USER2", "firstName": "BELLA", "lastName": "CHAO", "email": "bellachao@post.com", "password": "1234", "phone": "123456", "userStatus": 0 }
18	Check registry keys	not run	{ "id": 0, "USERNAME": "user2", "firstName": "Bella", "lastName": "Chao", "EMAIL": "bellachao@post.com", "password": "1234", "PHONE": "123456", "userStatus": 0 }
19	Check for mandatory presence of query keys: id	not run	{ "username": "user2", "firstName": "Bella", "lastName": "Chao", "email": "bellachao@post.com", "password": "1234", "phone": "123456", "userStatus": 0 }
20	Check for mandatory availability of query keys: username	not run	{ "id": 0, "firstName": "Bella", "lastName": "Chao", "email": "bellachao@post.com", "password": "1234", "phone": "123456", "userStatus": 0 }
21	Check for mandatory presence of query keys: firstname	not run	{ "id": 0, "username": "user2", "lastName": "Chao", "email": "bellachao@post.com", "password": "1234", "phone": "123456", "userStatus": 0 }
22	Check for mandatory availability of query keys: lastname	not run	{ "id": 0, "username": "user2", "firstName": "Bella", "email": "bellachao@post.com", "password": "1234", "phone": "123456", "userStatus": 0 }
23	Check for mandatory presence of key-requests: email	not run	{ "id": 0, "username": "user2", "firstName": "Bella", "lastName": "Chao", "password": "1234", "phone": "123456", "userStatus": 0 }
24	Check for mandatory presence of request keys: password	not run	{ "id": 0, "username": "user2", "firstName": "Bella", "lastName": "Chao", "email": "bellachao@post.com", "phone": "123456", "userStatus": 0 }
25	Check for mandatory presence of request keys: phone	not run	{ "id": 0, "username": "user2", "firstName": "Bella", "lastName": "Chao", "email": "bellachao@post.com", "password": "1234", "userStatus": 0 }

26	Checking for mandatory presence of request keys: userstatus		{ "id": 0, "username": "user2", "firstName": "Bella", "lastName": "Chao", "email": "bellachao@post.com", "password": "1234", "phone": "123456", }
27	Send malformed JSON and check the server's response	not run	{ "id": 0, "username": "user2" "firstName": "Bella" "lastName": "Chao" "email": "bellachao@post.com" "password": "1234" "phone": "123456" "userStatus": 0, },
28	Check the creation date of the object on successful submission	not run	{ "id": 0, "username": "user2", "firstName": "Bella", "lastName": "Chao", "email": "bellachao@post.com", "password": "1234", "phone": "123456", "userStatus": 0 }
29	Check the correctness of the status code response (for each request)	not run	{ "id": 0, "username": "user2", "firstName": "Bella", "lastName": "Chao", "email": "bellachao@post.com", "password": "1234", "phone": "123456", "userStatus": 0 }
30	Check whether data is sent to the database after a successful request	not run	{ "id": 0, "username": "user2", "firstName": "Bella", "lastName": "Chao", "email": "bellachao@post.com", "password": "1234", "phone": "123456", "userStatus": 0 }
31	Check for sending a request with an invalid URL	not run	https://petstore.swagger.io/v2/users / { "id": 0, "username": "user2", "firstName": "Bella", "lastName": "Chao", "email": "bellachao@post.com", "password": "1234", "phone": "123456", "userStatus": 0 }
Check requests with Get /user/{username}			
32	Check the get request with a valid username parameter	not run	https://petstore.swagger.io/v2/user/{username}
33	Check that the username parameter has the correct format	not run	https://petstore.swagger.io/v2/user/{username}
34	Check the server's response to an incorrect URL	not run	https://petstore.swagger.io/v2/user/{username}}e
35	Check the response code status with an incorrect URL	not run	https://petstore.swagger.io/v2/user/{username}}e
36	Check the correctness of the response (json)	not run	https://petstore.swagger.io/v2/user/{username}
37	Request data for a non-existent username	not run	https://petstore.swagger.io/v2/user/{username}
Check requests with Get /user/login			
38	Verification of sending a request with existing valid user data	not run	https://petstore.swagger.io/v2/user/login?username=user3&password=user2 /parameters: username: user3; password: user2
39	Verification of sending a request with invalid parameters (username)	not run	https://petstore.swagger.io/v2/user/login?username=user6&password=user2 /parameters: username: user6; password: user2
40	Verification of sending a request with invalid parameters (password)	not run	https://petstore.swagger.io/v2/user/login?username=user3&password=ussss /parameters: username: user3; password: ussss

41	Check the server's response to an incorrect URL	not run	https://petstore.swagger.io/v2/user/login?username=user3&password=user2 /parameters: username: user3; password: user2
42	Check the response code status with an incorrect URL	not run	https://petstore.swagger.io/v2/user/login?username=user3&password=user2 /parameters: username: user3; password: user2
43	Check if there is a message after successfully sending valid data	not run	https://petstore.swagger.io/v2/user/login?username=user3&password=user2 /parameters: username: user3; password: user2
44	Send only one parameter (password)	not run	https://petstore.swagger.io/v2/user/login?password=user2 /parameters: password: user2
45	Send only one parameter (username)	not run	https://petstore.swagger.io/v2/user/login?username=user2 /parameters: username: user3
46	Enter the parameter in upper case (username)	not run	https://petstore.swagger.io/v2/user/login?username=USER3&password=user2 /parameters: username: USER3; password: user2
47	Enter the parameter in upper case (password)	not run	https://petstore.swagger.io/v2/user/login?username=user3&password=USER2 /parameters: username: user3; password: USER2
48	Swap the username and password keys	not run	https://petstore.swagger.io/v2/user/login?password=user2&username=user3 /parameters: password: user2; username: user3
49	Enter in the value of the parameter username spec. symbols	not run	https://petstore.swagger.io/v2/user/login?username=user3N% % %&password=user2 /parameters: username: user3N% % %; password: user2
50	Enter the value of the password parameter spec. symbols	not run	https://petstore.swagger.io/v2/user/login?username=user3&password=user2%23\$\$\$ /parameters: username: user3; password: user23\$\$\$