wallarm

97 JSON Tests for for Authentication Endpoints

API Security Cheatsheet

1. Basic credentials:

```
{"login": "admin",
 "password": "admin"}
```

2. Empty credentials:

```
{"login": "",
 "password": ""}
```

3. Null values:

   ```
   {"login": null,
    "password": null}
   ```

4. Credentials as numbers:

   ```
   {"login": 123,
    "password": 456}
   ```

5. Credentials as booleans:

   ```
   {"login": true,
    "password": false}
   ```

6. Credentials as arrays:

   ```
   {"login": ["admin"],
    "password": ["password"]}
   ```

7. Credentials as objects:

   ```
   {"login": {"username": "admin",
    "password": {"password": "password"}}}
   ```

8. Special characters in credentials:

   ```
   {"login": "@dm!n",
    "password": "p@ssw0rd#"}
   ```

9. SQL Injection:

   ```
   {"login": "admin' --",
    "password": "password"}
   ```

10. HTML tags in credentials:

    ```
    {"login": "<h1>admin</h1>",
     "password": "ololo-HTML-XSS"}
    ```

11. Unicode in credentials:

    ```
    {"login": "\u0061\u0064\u006D\u0069\u006E",
     "password":
    "\u0070\u0061\u0073\u0073\u0077\u006F\u0072\u0064"}
    ```

12. Credentials with escape characters:

    ```
    {"login": "ad\\nmin",
     "password": "pa\\ssword"}
    ```

13. Credentials with white space:

    ```
    {"login": " ",
     "password": " "}
    ```

14. Overlong values:

    ```
    {"login": "a"*10000,
     "password": "b"*10000}
    ```

15. Malformed JSON (missing brace):

    ```
    {"login": "admin",
     "password": "admin"}
    ```

16. Malformed JSON (extra comma):

    ```
    {"login": "admin",
     "password": "admin",
    }
    ```

17. Missing login key:

    ```
    {"password": "admin"}
    ```

18. Missing password key:

    ```
    {"login": "admin"}
    ```

19. Swapped key values:

    ```
    {"admin": "login",
     "password": "password"}
    ```

20. Extra keys:

    ```
    {"login": "admin",
     "password": "admin",
     "extra": "extra"}
    ```

21. Missing colon:

    ```
    {"login" "admin",
     "password": "password"}
    ```

22. Invalid Boolean as credentials:

    ```
    {"login": yes,
     "password": no}
    ```

23. All keys, no values:

    ```
    {"": "",
     "": ""}
    ```

24. Nested objects:

    ```
    {"login": {"innerLogin": "admin",
     "password": {"innerPassword": "password"}}}
    ```

25. Case sensitivity testing:

    ```
    {"LOGIN": "admin",
     "PASSWORD": "password"}
    ```

26. Login as a number, password as a string:

    ```
    {"login": 1234,
     "password": "password"}
    ```

27. Login as a string, password as a number:

    ```
    {"login": "admin",
     "password": 1234}
    ```

28. Repeated keys:

```
{"login": "admin",
 "login": "user",
 "password": "password"}
```

29. Single quotes instead of double:

```
{'login': 'admin',
 'password': 'password'}
```

30. Login and password with only special characters:

```
{"login": "@#$%^&*",
 "password": "!@#$%^&*"}
```

31. Unicode escape sequence:

```
{"login": "\u0041\u0044\u004D\u0049\u004E",
 "password": "\u0050\u0041\u0053\u0053\u0057\u004F\u0052\u0044"}
```

32. Value as object instead of string:

```
{"login": {"$oid":
"507c7f79bcf86cd7994f6c0e"},
 "password": "password"}}
```

33. Nonexistent variables as values:

```
{"login": undefined,
 "password": undefined}
```

34. Extra nested objects:

```
{"login": "admin",
 "password": "password",
 "extra": {"key1": "value1",
 "key2": "value2"}}
```

35. Hexadecimal values:

```
{"login": "0x1234",
 "password": "0x5678"}
```

36. Extra symbols after valid JSON:

    ```
    {"login": "admin",
     "password": "password"}@@@@@@}
    ```

37. Only keys, without values:

    ```
    {"login":,
     "password":}
    ```

38. Insertion of control characters:

    ```
    {"login": "ad\u0000min",
     "password": "pass\u0000word"}
    ```

39. Long Unicode Strings:

    ```
    {"login": "\u0061"*10000,
     "password": "\u0061"*10000}
    ```

40. Newline Characters in Strings:

    ```
    {"login": "ad\nmin",
     "password": "pa\nssword"}
    ```

41. Tab Characters in Strings:

```
{"login": "ad\tmin",
 "password": "pa\tssword"}
```

42. Test with HTML content in Strings:

```
{"login": "<b>admin",
 "password": "password"}
```

43. JSON Injection in Strings:

```
{"login": "{\"injection\":\"value\"}",
 "password": "password"}
```

44. Test with XML content in Strings:

```
{"login": "admin",
 "password": "password"}
```

45. Combination of Number, Strings, and Special characters:

```
{"login": "ad123min!@",
 "password": "pa55w0rd!@"}
```

46. Floating numbers as Strings:

```
{"login": "123.456",
 "password": "789.123"}
```

47. Value as a combination of languages (Here, English and Hindi):

```
{"login": "adminà¤µà¥à¤¯à¤µà¤¸à¥à¤
¥à¤¾à¤ªà¤•",
 "password": "passwordà¤ªà¤¾à¤¸à¤µà¤°à¥à¤¡"}
```

48. Non-ASCII characters in Strings:

```
{"login": "âˆ†adminâˆ†",
 "password": "âˆ†passwordâˆ†"}
```

49. Single Character Keys and Values:

```
{"l": "a",
 "p": "p"}
```

50. Use of environment variables:

```
{"login": "${USER}",
 "password": "${PASS}"}
```

51. Backslashes in Strings:

```
{"login": "ad\\min",
 "password": "pa\\ssword"}
```

52. Long strings of special characters:

```
{"login": "!@#$%^&*()"*1000,
 "password": "!@#$%^&*()"*1000}
```

53. Empty Key in JSON:

```
{"": "admin",
 "password": "password"}
```

54. JSON Injection in Key:

```
{"

 {\"injection\":\"value\"}

": "admin",
 "password": "password"}
```

55. Quotation marks in strings:

    ```
    {"login": "\"admin\"",
     "password": "\"password\""}
    ```

56. Credentials as nested arrays:

    ```
    {"login": [["admin"]],
     "password": [["password"]]}
    ```

57. Credentials as nested objects:

    ```
    {"login": {"username": {"value": "admin",
     "password": {"password": {"value":
    "password"}
    ```

58. Keys as numbers:

    ```
    {123: "admin",
     456: "password"}
    ```

59. Testing with greater than and less than signs:

    ```
    {"login": "admin>1",
     "password": "<password"}
    ```

60. Testing with parentheses in credentials:

    ```
    {"login": "(admin)",
     "password": "(password)"}
    ```

61. Credentials containing slashes:

    ```
    {"login": "admin/user",
     "password": "pass/word"}
    ```

62. Credentials containing multiple data types:

    ```
    {"login": ["admin",
     123,
     true,
     null,
     {"username": ["admin"],
     "password": ["password",
     123,
     false,
     null,
     {"password": "password"]}}}
    ```

63. Using escape sequences:

    ```
    {"login": "admin\\r\\n\\t",
     "password": "password\\r\\n\\t"}
    ```

64. Using curly braces in strings:

    ```
    {"login": "{admin}",
     "password": "{password}"}
    ```

65. Using square brackets in strings:

    ```
    {"login": "[admin]",
     "password": "[password]"}
    ```

66. Strings with only special characters:

    ```
    {"login": "!@#$$%^&*()",
     "password": "!@#$$%^&*()"}
    ```

67. Strings with control characters:

    ```
    {"login": "admin\b\f\n\r\t\v\0",
     "password": "password\b\f\n\r\t\v\0"}
    ```

68. Null characters in strings:

    ```
    {"login": "admin\0",
     "password": "password\0"}
    ```

69. Exponential numbers as strings:

    ```
    {"login": "1e5",
     "password": "1e10"}
    ```

70. Hexadecimal numbers as strings:

    ```
    {"login": "0xabc",
     "password": "0x123"}
    ```

71. Leading zeros in numeric strings:

    ```
    {"login": "000123",
     "password": "000456"}
    ```

72. Multilingual input (here, English and Korean):

    ```
    {"login": "adminê´€ë¦¬ìž",
     "password": "passwordë¹„ë°€ë²ˆí˜¸"}
    ```

73. Extremely long keys:

```
{"a"*10000: "admin",
 "b"*10000: "password"}
```

74. Extremely long unicode strings:

```
{"login": "\u0061"*10000,
 "password": "\u0062"*10000}
```

75. JSON strings with semicolon:

```
{"login": "admin;",
 "password": "password;"}
```

76. JSON strings with backticks:

```
{"login": "`admin`",
 "password": "`password`"}
```

77. JSON strings with plus sign:

```
{"login": "admin+",
 "password": "password+"}
```

78. JSON strings with equal sign:

    ```
    {"login": "admin=",
     "password": "password="}
    ```

79. Strings with Asterisk (*) Symbol:

    ```
    {"login": "admin*",
     "password": "password*"}
    ```

80. JSON containing JavaScript code:

    ```
    {"login": "admin<script>alert('hi')
    </script>",
     "password": "password"}
    ```

81. Negative numbers as strings:

    ```
    {"login": "-123",
     "password": "-456"}
    ```

82. Values as URLs:

    ```
    {"login": "https://admin.com",
     "password": "https://password.com"}
    ```

83. Strings with email format:

```
{"login": "admin@admin.com",
 "password": "password@password.com"}
```

84. Strings with IP address format:

```
{"login": "192.0.2.0",
 "password": "203.0.113.0"}
```

85. Strings with date format:

```
{"login": "2023-08-03",
 "password": "2023-08-04"}
```

86. JSON with exponential values:

```
{"login": 1e+30,
 "password": 1e+30}
```

87. JSON with negative exponential values:

```
{"login": -1e+30,
 "password": -1e+30}
```

88. Using Zero Width Space (U+200B) in strings:

```
{"login": "adminâ€‹",
 "password": "passwordâ€‹"}
```

89. Using Zero Width Joiner (U+200D) in strings:

```
{"login": "adminâ€",
 "password": "passwordâ€"}
```

90. JSON with extremely large numbers:

```
{"login": 12345678901234567890,
 "password": 12345678901234567890}
```

91. Strings with backspace characters:

```
{"login": "admin\b",
 "password": "password\b"}
```

92. Test with emoji in strings:

```
{"login": "adminðŸ˜€",
 "password": "passwordðŸ˜€"}
```

93. JSON with comments, although they are not officially supported in JSON:

    ```
    {/*"login": "admin",
     "password": "password"*/}
    ```

94. JSON with base64 encoded values:

    ```
    {"login": "YWRtaW4=",
     "password": "cGFzc3dvcmQ="}
    ```

95. Including null byte character (may cause truncation):

    ```
    {"login": "admin\0",
     "password": "password\0"}
    ```

96. JSON with credentials in scientific notation:

    ```
    {"login": 1e100,
     "password": 1e100}
    ```

97. Strings with octal values:

    ```
    {"login": "\141\144\155\151\156",
     "password":
    "\160\141\163\163\167\157\162\144"}
    ```

Interesting in learning more?

Follow Wallarm on LinkedIn

wallarm