```
import pytest
from swarms.structs.agent import Agent
from swarms_cloud.utils.api_key_generator import generate_api_key


# Basic tests
def test_generate_api_key_default():
    api_key = generate_api_key()
    assert isinstance(api_key, str)
    assert api_key.startswith("sk-")
    assert len(api_key) == 53  # Prefix (3) + 50 random characters


def test_generate_api_key_custom_length():
    api_key = generate_api_key(length=10)
    assert len(api_key) == 13  # Prefix (3) + 10 random characters


def test_generate_api_key_custom_prefix():
    api_key = generate_api_key(prefix="custom-")
    assert api_key.startswith("custom-")
    assert len(api_key) == 57  # Custom prefix (7) + 50 random characters


# Exception tests
def test_generate_api_key_short_length():
```

```
    with pytest.raises(ValueError):
        generate_api_key(length=2)


# Parameterized tests
@pytest.mark.parametrize(
    "prefix, length",
    [
        ("pre-", 15),
        ("test-", 25),
        ("longprefix-", 100),
    ],
)
def test_generate_api_key_parameterized(prefix, length):
    api_key = generate_api_key(prefix=prefix, length=length)
    assert api_key.startswith(prefix)
    assert len(api_key) == len(prefix) + length


# Test performance (optional)
def test_generate_api_key_performance(benchmark):
    benchmark(generate_api_key)


# Additional tests
def test_generate_api_key_special_characters_prefix():
```

```
    api_key = generate_api_key(prefix="@#$")
    assert api_key.startswith("@#$")
    assert len(api_key) == 53  # Special prefix (3) + 50 random characters



def test_generate_api_key_zero_length():
    with pytest.raises(ValueError):
        generate_api_key(length=0)



class MockAgent(Agent):
    # The class to mock the tests
    def __init__(self, *args, **kwargs):
        super().__init__(*args, **kwargs)
        self.method_called = False

    def mock_method(self):
        self.method_called = True
        return "Hello World"
```