

```
def test_sign_in_success_soft_token(monkeypatch):
```

```
    """
```

```
    Test successful sign in process with SOFT TOKEN MFA method.
```

```
    - Mocks the oathtool response and simulates the full sign-in flow.
```

```
    """
```

```
    from unittest.mock import MagicMock
```

```
    import signIn
```

```
    # Mocks
```

```
    driver = MagicMock()
```

```
    driver.current_url = 'https://mint.com/overview'
```

```
    driver.get.return_value = None
```

```
    driver.implicitly_wait.return_value = None
```

```
    driver.find_element.return_value = MagicMock()
```

```
    monkeypatch.setattr(signIn, 'home_page', lambda d: None)
```

```
    monkeypatch.setattr(signIn, 'user_selection_page', lambda d: None)
```

```
    monkeypatch.setattr(signIn, 'handle_same_page_username_password', lambda d, e, p: None)
```

```
    monkeypatch.setattr(signIn, 'handle_login_failures', lambda d: None)
```

```
    monkeypatch.setattr(signIn, 'bypass_verified_user_page', lambda d: False)
```

```
    monkeypatch.setattr(signIn, 'bypass_passwordless_login_page', lambda d: None)
```

```
    monkeypatch.setattr(signIn, 'mfa_selection_page', lambda d, m: None)
```

```
    monkeypatch.setattr(signIn, 'account_selection_page', lambda d, a: None)
```

```
    monkeypatch.setattr(signIn, 'password_page', lambda d, p: None)
```

```
    monkeypatch.setattr(signIn, 'handle_wait_for_sync', lambda d, t, s: 'Sync Complete')
```

```
    # Force MFA method/inputs
```

```
    mfa_token_input = MagicMock()
```

```
    mfa_token_button = MagicMock()
```

```
    monkeypatch.setattr(signIn, 'set_mfa_method', lambda d, m: (mfa_token_input, mfa_token_button, signIn.constants.MFA_VIA_
```

```
    monkeypatch.setattr(signIn, 'search_mfa_method', lambda d: (mfa_token_input, mfa_token_button, signIn.constants.MFA_VIA_
```

```
    monkeypatch.setattr('oathtool.generate_otp', lambda t: '654321')
```

```
    result = signIn.sign_in(
        email='test@example.com',
        password='password123',
        driver=driver,
        mfa_method=signIn.constants.MFA_VIA_SOFT_TOKEN,
        mfa_token='base32token',
        mfa_input_callback=None,
        wait_for_sync=True,
        imap_account=None,
        imap_password=None,
        imap_server=None
    )
```

```
    assert result == 'Sync Complete'
```

```
---Extra Details---
```

```
Filename: signIn.py
```

```
Description: Tests a successful login flow using the soft token MFA method, ensuring the sign_in function integrates the MFA han
```

```
Score: 99%
```

```
Alignment: 98%
```

```
Validation Notes: Highly relevant to core logic; tests comprehensive path with actual flow up to post-login.
```