

Korbit API Tutorial - Chap. 2

지난번 튜토리얼을 통하여 Access Key를 얻는 방법까지 살펴보았다. 이번 챕터에서는 얻은 HTML Response에서 Access Key를 분리하고 이를 이용하여 거래하는 방법을 살펴본다.

1. Access Key Separation

HTML Response는 기본적으로 JSON(JavaScript Object Notation)에 의하여 구성되어 있다. Python에서는 JSON Parser를 기본적으로 내장하고 있으므로 이를 활용하여 원하는 값을 추출할 수 있다. 아래의 코드를 살펴보자.

- JSON Response

```
{
  "token_type": "Bearer",
  "access_token": "1t1LgTs1DrGznxP1hYz7R1dsNVIbnEK",
  "expires_in": 3600,
  "refresh_token": "vn5xoOf4Pzckgn4jQSL9Sb3KxWJvYtm"
}
```

- Python Code

```
resp = conn.getresponse()
if resp.status != 200:
    print "Connection Error"
    sys.exit()
data = resp.read()
js = json.loads(data)
AccessToken = js['access_token']
RefreshToken = js['refresh_token']
```

즉 Response를 `read()` 함수를 이용하여 string 형태로 저장한 후 `json.loads()` 함수로 읽어들인다. 이 경우 json 함수는 Dictionary 자료형으로 이를 보관하며 key를 통하여 해당 값을 읽을 수 있다. 자세한 사항은 Python 자료형 중 Dictionary를 참고한다.

2. Inquiry - Wallet

지갑과 관련된 정보는 아래의 API를 사용하여 가지고 올 수 있다.

```
GET https://api.korbit.co.kr/v1/user/wallet
```

이와 관련하여 별도의 파라미터 전달은 필요하지 않으나 모든 API는 nonce를 파라미터로 가져야 한다. nonce는 임의의 숫자로서 API는 이전의 nonce보다 작은 값이 입력되어 API가 올 경우 이에 응답하지 않는다. 그러므로 항상 이전보다 더 큰 nonce를 파라미터로 전달하여야 하며 이를 위하여 nonce는 시스템 시간(epoch time)을 입력하는 것이 일반적이다.. 이 API를 사용하는 Python 코드와 API의 응답은 아래와 같다.

- JSON Response

```
{
  "in": [
    {
      "currency": "krw",
      "address": {
        "bank": "우리은행",
        "account": "1234567890",
        "owner": "(주)코빗"
      }
    },
    {
      "currency": "btc",
      "address": {
        "address": "1anjg6B2XbxjHh8LFw8mXHATH54vrxs2F"
      }
    }
  ],
  "out": [
    {
      "currency": "krw",
      "status": "owner_mismatch",
      "registeredOwner": "김탁구",
      "address": {
        "bank": "교보증권",
        "account": "1234567890",
        "owner": "김강모"
      }
    }
  ]
  "balance" : [{"currency": "krw", "value": "10000000"}, {"currency": "btc", "value": "32.0"}],
  "pendingOut" : [{"currency": "krw", "value": "3000000"}, {"currency": "btc", "value": "13.0"}],
  "pendingOrders" : [{"currency": "krw", "value": "4000000"}, {"currency": "btc", "value": "7.0"}],
```

```
"available":[{"currency":"krw","value":"3000000"}, {"currency":"btc","value":"10.0"}],  
"fee":"0.006"  
}
```

- Python Code

```
conn.request("GET", "/v1/user/wallet?access_token="+ AccessToken +"&nonce=" + str(int(time.time())))  
resp = conn.getresponse()  
if resp.status != 200:  
    WalletData = WalletData + "\n" + str(resp.status) + str(resp.reason)  
else:  
    data = resp.read()  
    js = json.loads(data)  
    WalletData = WalletData + "\n" + "Balance"  
    WalletData = WalletData + "\n" + "  KRW : " + js['available'][0]['value']  
    WalletData = WalletData + "\n" + "  BTC : " + js['available'][1]['value']  
    WalletData = WalletData + "\n" + ""
```