

```
@app.route('/register-node', methods=['POST']) # 새로운 노드를 연결하는 요청 받은 노
def register_node():
    new_node_url = request.json['newNodeUrl']
    node_not_already_present = (new_node_url not in bitcoin.network_nodes) #채우시오
    not_current_node = (bitcoin.current_node_url != new_node_url) #채우시오 : current
    if node_not_already_present and not_current_node: #두 가지 조건을 모두 만족하면
        #새로운 노드 network_node에 추가
        bitcoin.network_nodes.append(new_node_url)
    return jsonify({'note': 'New node registered successfully.'})
```

현재 노드가 새로운 노드를 연결하라는 요청을 받았을 때 원래 연결되어 있던 노드에게 새로운 노드를 등록하는 역할을 하는 함수이다. 새로운 노드의 주소가 현재 노드와 연결된 노드들의 주소 목록 중에 없고 현재 노드의 주소와 새로운 노드의 주소가 다르면 새로운 노드가 현재 노드와 현재 노드와 연결되었다고 기록한다.

```
@app.route('/register-nodes-bulk', methods=['POST']) # 새로운 노드를 추가한 뒤 전체
def register_nodes_bulk():
    all_network_nodes = request.json['allNetworkNodes']
    for network_node_url in all_network_nodes:
        node_not_already_present = (network_node_url not in bitcoin.network_nodes)
        not_current_node = (network_node_url != bitcoin.current_node_url) #채우시오
        if node_not_already_present and not_current_node: #두 가지 조건을 모두 만족
            #새로운 노드 network_node에 추가
            bitcoin.network_nodes.append(network_node_url)
```

새로운 노드를 추가한 뒤에 현재 노드와 연결된 모든 노드에 대한 정보를 새로운 노드에게 전달 되도록 하는 함수이다. 이 함수는 register\_and\_broadcast\_node 함수의 끝부분에서 실행이 된다. register\_and\_broadcast\_node 함수의 아래의 부분에서 새로운 노드의 register\_nodes\_bulk를 실행 시켜 현재 연결된 모든 노드들에 대한 정보를 전달하게 된다.

```
for response in reg_nodes_promises:
    if response.status_code == 200:
        requests.post(f"{new_node_url}/register-nodes-bulk", json={'allNetworkNodes': bitcoin.network_nodes + [bitcoin.current_node_url]})
```