

# Othello Protocol 문서

## 홍성빈

## Othello Protocol

MessagePack .....	3
프로토콜 절차 .....	4
메소드 설명 .....	5
join .....	5
play .....	5
result .....	6
reset.....	6
end.....	7

# MessagePack

It's like JSON.  
but fast and small.

MessagePack is an efficient binary serialization format. It lets you exchange data among multiple languages like JSON. But it's faster and smaller. Small integers are encoded into a single byte, and typical short strings require only one extra byte in addition to the strings themselves.

JSON 27 bytes

```
{ "compact": true, "schema": 0 }
```

MessagePack 18 bytes

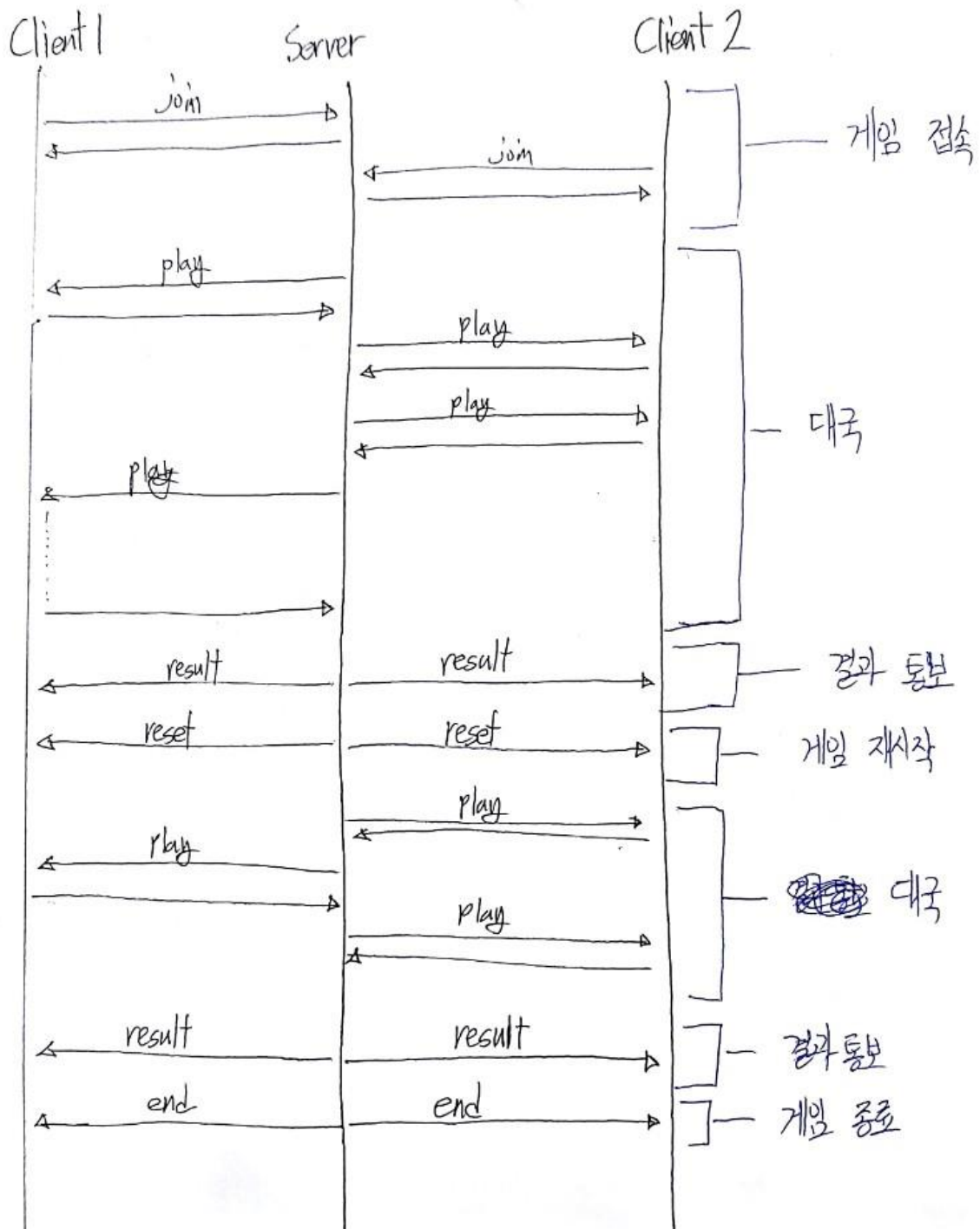
2-element map      7-byte string      true      6-byte string      integer 0

Next: MessagePack is supported by over 50 programming languages and environments.  
See [list of implementations](#).

바이트 형식으로 직렬화 시키는 포맷. JSON보다 직렬화 및 역직렬화가 빠르고 직렬화 된 데이터의 크기가 더 작다. 현재 50개의 언어가 지원중임. 실제 주고받는 프로토콜 데이터는 바이트 형식이지만, 프로토콜을 프로그램에서 처리할 때는 텍스트형식으로 처리할 것.

<https://msgpack.org/>

## 프로토콜 절차



## 메소드 설명

모든 메소드 패킷을 보낼 땐 [패킷 이름 + 패킷 내용]형태로 전송

ex) {"packet\_name":"join","request":{...}} -> 요청할 때

{"packet\_name":"join","response":{...}} -> 응답 받을 때

join

-요청 데이터 형식 :

```
{  
    "player_name": "madmaxAI"  
}
```

-반환 데이터 형식 :

```
{  
    "success": true|false,  
    "reason": "~ ~"  
}
```

-설명 : 게임 참가 패킷. join 실패시 reason에 그 이유를 반환

-방향 : 클라이언트 -> 서버

play

-요청 데이터 형식 :

```
{  
    "map":[0,0,0,0,0,0,0,0,  
           0,0,0,0,0,0,0,0,  
           0,0,0,0,0,0,0,0,  
           0,0,0,2,1,0,0,0,  
           0,0,0,1,2,0,0,0,  
           0,0,0,0,0,0,0,0,  
           0,0,0,0,0,0,0,0,  
           0,0,0,0,0,0,0,0],  
    "last_move":[-1,-1],  
    "white":1  
}
```

-반환 데이터 형식 :

```
{
    "move":[2,2]
}
```

-설명 : 판 위에 착수를 요청하는 패킷. 맵에서 1은 흑, 2는 백을 의미한다. 착수를 요청하는 패킷이지만, this\_turn 필드를 조절해서 단순히 클라이언트의 맵을 갱신시키는 용도로 사용할 수도 있다. 클라이언트의 응답이 오래 걸릴 시 타임아웃 패배처리.

-방향 : 서버 -> 클라이언트

result

-요청 데이터 형식 :

```
{
    "map":[1,1,1,1,1,1,1,1,
            1,1,1,1,1,1,1,1,
            1,1,1,1,1,1,1,1,
            1,1,1,2,1,1,1,2,
            1,1,1,1,2,2,1,2,
            1,1,1,1,2,2,1,2,
            1,1,1,1,2,2,2,2,
            1,1,1,1,2,2,2,2],
    "last_move":[-1,-1],
    "white":16,
    "black":48,
    "winner":"black",
    "reason":""
}
```

-반환 데이터 형식 :

```
{
    "move":[2,2]
}
```

-설명 : 판 위에 착수를 요청하는 패킷. 맵에서 1은 흑, 2는 백을 의미한다. 착수를 요청하는 패킷이지만, this\_turn 필드를 조절해서 단순히 클라이언트의 맵을 갱신시키는 용도로 사용할 수도 있다. 클라이언트의 응답이 오래 걸릴 시 타임아웃 패배처리.

-방향 : 서버 -> 클라이언트

reset

-요청 데이터 형식 : 없음

-반환 데이터 형식 : 없음.

-설명 : 클라이언트에게 진행된 게임을 모두 초기화 하고 이어서 새로운 게임을 할 수 있는 상태로 변경 명령.

-방향 : 서버 -> 클라이언트

end

-요청 데이터 형식 : 없음

-반환 데이터 형식 : 없음.

-설명 : 게임이 끝났음을 알림.

-방향 : 서버 -> 클라이언트