

Communication Protocol

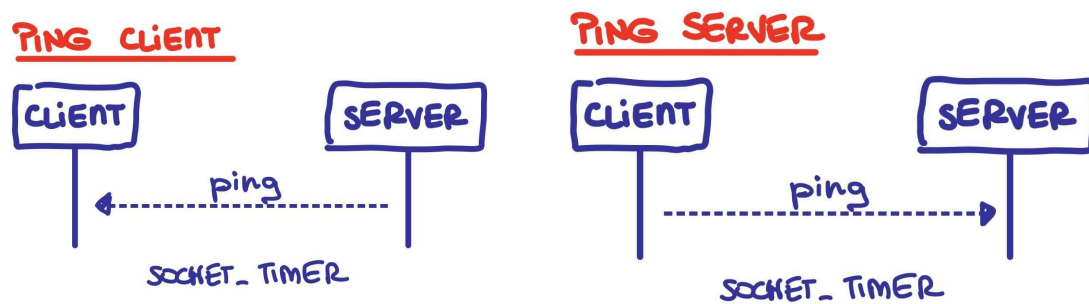
The protocol used is **JSON based**

Checking nodes connection

When connection is established both server and client set a socket timeout.

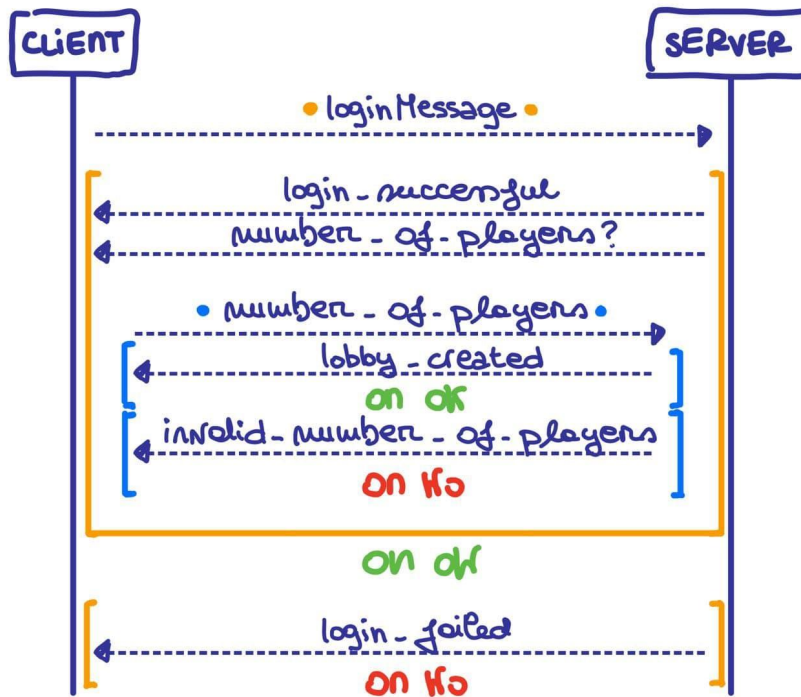
Then client and server keep sending each other periodical messages to keep the timeout from expiring.

Note: this isn't a request - reply protocol. Every message is standAlone



Other Communication Examples:

FIRST PLAYER



//-----login-----

```
{
  "username": "pippo",
  "messageType": "LOGIN",
  "payload": null
}
```

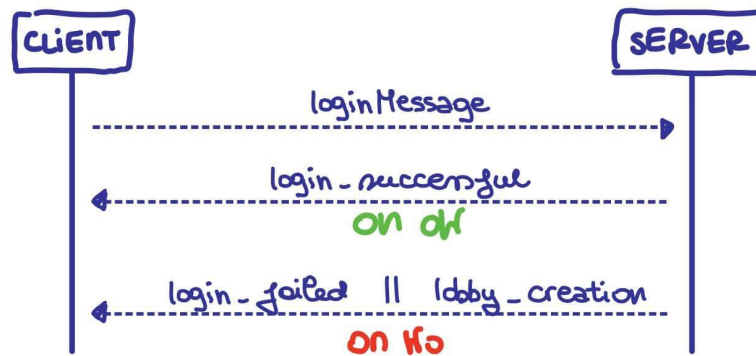
//-----Number of players--- Server -> Client -----

```
{
  "username": null,
  "messageType": "NUMBER_OF_PLAYERS",
  "payload": null
}
```

//-----Number of players--- Client -> Server -----

```
{
  "username": "pippo",
  "messageType": "NUMBER_OF_PLAYERS",
  "payload": "3" //number between 1 and 4
}
```

OTHER PLAYERS



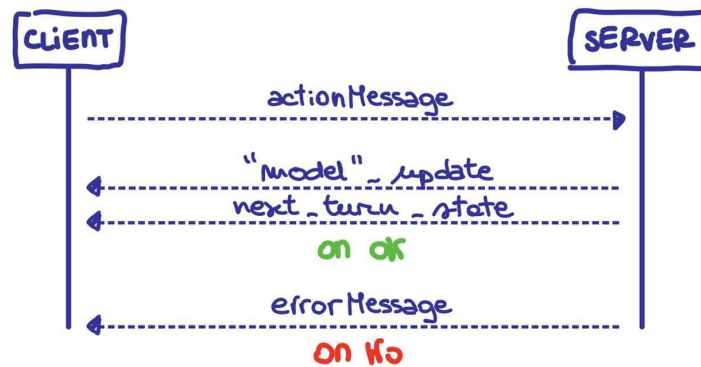
//-----login-----

```

{
  "username": "pluto",
  "messageType": "LOGIN",
  "payload": null
}

```

ACTION



//----- Action example: BUY_CARD -----

```

{
  "username": "pippo",
  "messageType": "ACTION",
  "payload": {
    "type": "BUY_CARD",
    "row": "2",
    "column": "0",
    "cardSlotIndex": "1"
  }
}

```

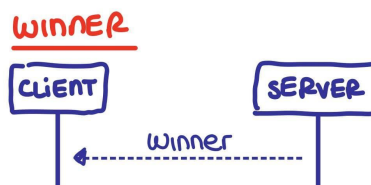
//----- model update example: after Action BUY_CARD -----

```
1)
{
  "username": "pippo",
  "messageType": "DEVELOP_CARD_DECK_UPDATED",
  "payload":
  {
    "row": "2",
    "column": "0"
  }
}
```

```
2)
{
  "username": "pippo",
  "messageType": "CARD_SLOT_UPDATE",
  "payload":
  {
    "SlotNumber": "2",
    "DevelopCardID": "D47"
  }
}
```

//----- model Error message example -----

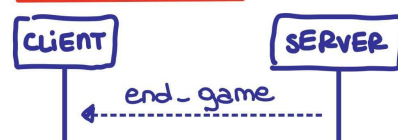
```
{
  "username": "pippo",
  "messageType": "ERROR",
  "payload": "NOT_ENOUGH_SPACE" //this is an error code established in the protocol
}
```



WHEN GAME STARTS



GAME ENDED



Client Side Model

Client contains:

- whole DevelopCard Deck
- Market
- Faith track position (int)
- Chest and Warehouse
- Card Slots
- Leader Cards
- For other clients:
 - Faith track position (int)
 - Card Slots
 - Leader Cards