

# ネットワークClass解説

---

## Classリスト

1. NetworkManager.cs) ネットワーク Main Class
2. JsonOBJ.cs) データ送受信用 Class
3. SendTest.cs) メッセージ送信テスト Class

## eventType定義

```
//NetworkManager.cs
public enum eventType
{
    init,
    match,
    chat,
    Message,
    disconnect,
    client_Reload,
    //=====
    //ここに必要な処理タイプ追加
    //customType,
}
```

- eventHandlerに使う、処理タイプを定義

## サーバーから受けたデータを処理

```
//NetworkManager.cs
private void eventHandler(JsonOBJ data)
{
    eventType state = (eventType)Enum.Parse(typeof(eventType), data.type);

    switch (state)
    {
        case eventType.init:

            uuids.Add(data.uuid);

            var playerObj = Instantiate(playerPrifab);
            players.Add(data.uuid, playerObj);
            if (players.Count == 1)
                information.uuid = data.uuid;
            break;

        case eventType.match:
            break;
    }
}
```

```

        case eventType.Message:
            break;

        case eventType.client_Reload:

            players.Remove(data.uuid);
            break;

        //=====
        //ここに必要な処理を追加
        //case eventType.customType:
        //    break;

        default:
            break;
    }
}

```

## データ送受信用（カスタマイズ可能）

```

//JsonOBJ.cs
[Serializable]
public class inform
{
    public string uuid;
    public string address;
    public int port;
    public string name;
}

[Serializable]
public class JsonOBJ
{
    //=====
    //カスタマイズ可能
    public string type;
    public string name;
    public string uuid;
    public string msg;
    // public float pos_x;
    // public float pos_y;

    public inform ribal;
    //=====

    public JsonOBJ(eventType state)
    {
        this.type = state.ToString();
    }
}

```

```
}

public static JsonOBJ CreateFromJSON(string data)
{
    try
    {
        return JsonUtility.FromJson<JsonOBJ>(data);
    }
    catch (Exception e)
    {
        Debug.Log("<color=green> Client: </color> JSONデータで変換できませんでした。");
        return null;
    }
}

public static string CreateToJSON(JsonOBJ data)
{
    return JsonUtility.ToJson(data);
}
```