

# シグナリングサーバ上の読み取りと書き出しのルール

## 読み取り

自分の名前上の場所を常時監視し、何かしらのデータが書き込まれたらリアルタイムにそのデータを読み取るようにする。

## 書き出す

通常、相手の名前上の場所書き出すものとする。

# シグナリングサーバへ送信するデータの構造

## 1. offer

以下の構造のデータを P2P 通信相手がリスンしている場所書き込む。

```
const sampleOffer = {
  sender: 'データの送信者',
  sessionDescription: {
    sdp: 'Session Description Protocol',
    type: 'offer',
  },
  type: 'offer',
};
```

プロパティ	説明
sender	誰が書き込んだのかを書き込む。例) Taro、たろう、等ユニークな文字列。
sessionDescription	RTCPeerConnection.createOffer() によって作成する。P2P 通信に必要な情報諸々、例えば、オーディオやビデオのコーデック情報等をまとめたもの。
type	offer の type は offer とする。

## 2. answer

以下の構造のデータを P2P 通信相手がリスンしている場所書き込む。

```
const sampleAnswer = {
  sender: 'データの送信者',
  sessionDescription: {
    sdp: 'Session Description Protocol',
    type: 'answer',
  },
  type: 'answer',
};
```

プロパティ	説明
sender	誰が書き込んだのかを書き込む。
sessionDescription	RTCPeerConnection.createAnswer() によって作成する。 データ構造は、RTCPeerConnection.createOffer() によって作られるものと似ている。
type	answer の type は answer とする。

### 3. candidate

以下の構造のデータを P2P 通信相手がリスンしている場所へ書き込む。

```
const sampleCandidate = {
  candidate: {
    candidate: 'candidate',
    sdpMLineIndex: 0,
    sdpMid: '0',
  },
  sender: 'データの送信者',
  type: 'candidate',
};
```

プロパティ	説明
candidate	RTCPeerConnection.onicecandidate によって取得できる経路情報。
sender	誰が書き込んだのかを書き込む。
type	candidate の type は candidate とする。