



# T5 - Web frameworks

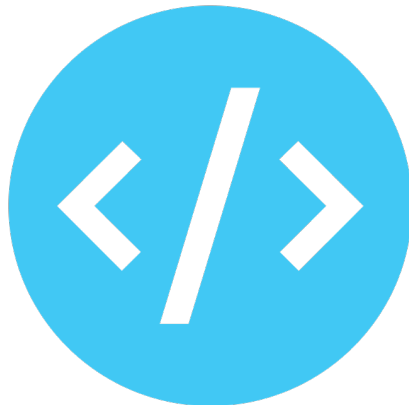
---

T-WEB-510

## P2PApp

---

Bootstrap





# P2PApp

language: JS, REACT



- Your repository must contain the totality of your source files, but no useless files (binary, temp files, obj files,...).

- Documentation
  - [RTCPeerConnection](#)
  - [RTCIceCandidate](#)
  - [RTCSessionDescription](#)
  - [RTCView](#)
  - [getUserMedia](#)
- Instructions

## DOCUMENTATION

### + RTCPeerConnection

```
Terminal
const pc = new RTCPeerConnection(configuration);
```

Allows to create a connection with a particular configuration and to retrieve the candidate. "RTCPeerConnection" is the basis of the connection and allows the continuation of interactions.

### + RTCIceCandidate

```
Terminal
pc.addIceCandidate(new RTCIceCandidate(data.candidate));
```

Allows the creation and identification of a new "client".



## + RTCSESSIONDESCRIPTION

Allows you to create a description for the current customer with the necessary information.

## + RTCVIEW

Basic component allowing the use of functionalities.

## + GETUSERMEDIA

```
Terminal
getUserMedia({
  audio: true,
  video: {
    mandatory: {
      minWidth: 640, // Provide your own width, height and frame rate here
      minHeight: 360,
      minFrameRate: 30,
    },
    facingMode: (isFront ? "user" : "environment"),
    optional: (videoSourceId ? [{sourceId: videoSourceId}] : []),
  },
  function (stream) {
    console.log('getUserMedia success', stream);
    callback(stream);
  }, logError);
```

Allows to retrieve the user's media, and to determine if the required characteristics are respected.

## INSTRUCTIONS

Based on the objects described in the documentation, create a native REACT application that allows you to send a message to another instance of your application.