// permet d'écouter jusqu'à que le server envoi "open" et lance la fonction TestOpen()
**socket.On("open", TestOpen())**

e = jsonObject => { name : "", data: ""}
public void TestOpen(SocketIOEvent e)
  {

	Debug.Log("[SocketIO] Open received: " + e.name + " " + e.data);

};

// permet  d'envoyer  "beep" au server
**socket.Emit("beep", jsonobject);**

# Contrat :

## Join

### Send

socket.Emit("join");
{
	name : "name player"
}

### Receive

socket.on("join");
{
	id : "id player",
	name : "name player"
}

## GetQuestions

### Receive

socket.Emit("getQuestions");
{
	"0" :  {
		title: "blabla ?"
		"answer"  : {
			"0" : "blouyblou",
			"1" : "blibsqdqs"

```
                        “2” : “dsqdqs”,
                        “3” : “dqsdqs”
                    }
            }
            “1” : {

            ….
            }
    }
```

## getCurrentQuestion

**Receive**

```
socket.on("getCurrentQuestion");
{
        question : “0”
}
```

## setReponse

**Send**

```
socket.Emit("setReponse");
{
        id: “player Id”,
        answer: “ reponse en int”
}
```

**Receive**

```
socket.on("setReponse");
{
        id : “id player”,
        answer : “bool ”
}
```

## getScore

**Receive**

```
socket.on("getScore");
{
        0: {
                name : “player name”
                score : “score en int”
        }
        1 : {
                name : “player name”
                score : score en int”
        }
        ….
}
```