NPM пакет

Начиная со сборки 1.0.1.29, SFU SDK доступен в виде NPM пакета @flashphoner/sfusdk

```
npm i @flashphoner/sfusdk
```

Пример использования в приложении на ReactJS: создание и удаление комнаты

```
import React, {useState} from "react";
import {Sfu, SfuEvent} from "@flashphoner/sfusdk";

export default function TestApp() {
    const [url, setUrl] = useState("ws://localhost:8080");
    const [established, setEstablished] = useState("Not connected");
    const [buttonText, setButtonText] = useState("Connect");
    const [session, setSession] = useState(null);

    function createRoom(options: {
        url: string,
        roomName: string,
        pin: string,
        nickname: string,
        pc: RTCPeerConnection
    }) {
        const sfu = new Sfu();
        sfu.connect({
            url: options.url,
            nickname: options.nickname,
            logGroup: options.roomName
        });
        const room = sfu.createRoom({
            name: options.roomName,
            pin: options.pin,
            pc: options.pc
        });
        return sfu;
    }

    function onClick() {
        if (session == null) {
            console.log("Trying to connect");
            const pc = new RTCPeerConnection();
            const s = createRoom({
                url: url,
                roomName: "ROOM",
                pin: "1234",
                nickname: "Test",
                pc: pc
            }).on(SfuEvent.CONNECTED, (room) => {
                console.log("Connected!");
                setEstablished("Connected!");
```

```
                    setButtonText("Disconnect");
                    setSession(s);
                    room.join();
                }).on(SfuEvent.DISCONNECTED, () => {
                    console.log("Disconnected!");
                    setEstablished("Not connected!");
                    setButtonText("Connect");
                    setSession(null);
                });
            } else {
                let room = session.room();
                if(room) {
                    room.destroyRoom()
                }
                session.disconnect();
            }
        }

        return(
            <div>
                <div>
                    <input type="text" value={url} placeholder="Enter server
websocket URL" onChange={(e) => setUrl(e.target.value)}/>
                    <button onClick={() => onClick()}>{buttonText}</button>
                </div>
                <div>{established}</div>
            </div>

        )
    }
```