

Websocket server connection control

Since SFU SDK build [2.0.194](#) it is possible to check a websocket server connection consistency. WCS server sends a special ping packets periodically (every 5 seconds by default). The ping packets are received and counted by SFU SDK. If more than a certain number of consequent packets are missed, the websocket connection is supposed to be lost, and the client receives `SfuEvent.CONNECTION_FAILED` event.

Connection control parameters setup in `config.json` file

If the example reads the connection parameters from `config.json` file, the maximum number of pings missing and pings checking interval may be set in the file:

[code](#)

```
{
  "room": {
    ...,
    "failedProbesThreshold": 5,
    "pingInterval": 5000
  },
  ...
}
```

Where:

- `failedProbesThreshold` - the maximum number of pings missing
- `pingInterval` - pings checking interval in ms

Connection control parameters setup in a source code

1. `roomConfig` object creation

[code](#)

```
let roomConfig = {
  url: config.room.url || "ws://127.0.0.1:8080",
  roomName: config.room.name || "ROOM1",
  pin: config.room.pin || "1234",
  nickname: config.room.nickName || "User1"
};
if (config.room.failedProbesThreshold !== undefined) {
  roomConfig.failedProbesThreshold = config.room.failedProbesThreshold;
}
if (config.room.pingInterval !== undefined) {
  roomConfig.pingInterval = config.room.pingInterval;
}
```

2. Connection establishing and the room creation

[code](#)

```
const session = await sfu.createRoom(roomConfig);
```

3. `SfuEvent.CONNECTION_FAILED` event handling

[code](#)

```
session.on(constants.SFU_EVENT.DISCONNECTED, function() {
    ...
}).on(constants.SFU_EVENT.FAILED, function(e) {
    ...
    setStatus(state.statusId(), "FAILED", "red");
    if (e.status && e.statusText) {
        ...
    } else if (e.type && e.info) {
        setStatus(state.errInfoId(), e.type + ": " + e.info, "red");
    }
});
```

Disabling connection control at client side

Sometimes pings sending may be disabled at server side

```
keep_alive.algorithm=NONE
```

In this case, pings checking at client side should also be disabled

```
{
  "room": {
    ...,
    "failedProbesThreshold": 0,
    "pingInterval": 0
  },
  ...
}
```