

Контроль целостности websocket соединения с сервером

В сборке 2.0.194 добавлена возможность контролировать целостность websocket соединения с сервером. WCS сервер периодически (по умолчанию каждые 5 секунд) присылает ping пакеты. На стороне SFU SDK эти пакеты принимаются и подсчитываются. Если больше заданного количества пакетов подряд было пропущено, предполагается, что соединение с сервером потеряно, и клиенту возвращается событие `SfuEvent.CONNECTION_FAILED`.

Настройка параметров контроля в конфигурационном файле `config.json`

Если пример поддерживает загрузку параметров из конфигурационного файла `config.json`, максимальное количество пропущенных ping пакетов и интервал проверки могут быть настроены в файле:

[code](#)

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

Здесь:

- `failedProbesThreshold` - максимальное число пропущенных подряд ping пакетов от сервера
- `pingInterval` - интервал проверки числа пропущенных ping пакетов в мс

Настройка параметров контроля на уровне исходного кода

1.Формирование объекта `roomConfig`

[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. Установка соединения и создание комнаты

[code](#)

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

3. Обработка события `SfuEvent.CONNECTION_FAILED`

[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");
  }
});
```

Отключение контроля на стороне клиента

В некоторых случаях отправка ring пакетов может быть отключена на стороне сервера

```
keep_alive.algorithm=NONE
```

На стороне клиента при этом проверка также должна быть отключена

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