config.js

В модуль `config.js` вынесены функции для конфигурации комнаты и доступа к локальным медиа устройствам в соответствии с файлом конфигурации. Пример файла конфигурации:

```
{
  "room": {
    "url": "ws://127.0.0.1:8080",
    "name": "ROOM1",
    "pin": "1234",
    "nickName": "User1",
    "failedProbesThreshold": 5,
    "pingInterval": 5000
  },
  "media": {
    "audio": {
      "tracks": [{
        "source": "mic",
        "channels": 2,
        "type": "mic1"
      }]
    },
    "video": {
      "tracks": [
        {
          "source": "camera",
          "width": 1280,
          "height": 720,
          "codec": "H264",
          "constraints": {
            "frameRate": 25
          },
          "encodings": [
            { "rid": "180p", "active": true, "maxBitrate": 200000,
"scaleResolutionDownBy": 4 },
            { "rid": "360p", "active": true, "maxBitrate": 500000,
"scaleResolutionDownBy": 2 },
            { "rid": "720p", "active": true, "maxBitrate": 900000 }
          ],
          "type": "cam1"
        }
      ]
    }
  }
}
```

Исходный код модуля

Для анализа исходного кода возьмем версию модуля `config.js`, которая находится здесь

1. Получение конфигурации комнаты

`getRoomConfig()` code

```
const getRoomConfig = function(config) {
    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;
    }
    return roomConfig;
}
```

2. Получение списка видео потоков с доступом к локальным медиа устройствам

`getVideoStreams()` code

```
const getVideoStreams = async function(config) {
    let streams = [];
    if (config.media && config.media.video && config.media.video.tracks) {
        streams = await getStreams(config.media.video.tracks);
    }
    return streams;
}
```

3. Получение списка аудио потоков с доступом к локальным медиа устройствам

`getAudioStreams()` code

```
const getAudioStreams = async function(config) {
    let streams = [];
    if (config.media && config.media.audio && config.media.audio.tracks) {
        streams = await getStreams(config.media.audio.tracks);
    }
    return streams;
}
```

4. Получение доступа к медиа устройствам и добавление потока в список

`getStreams()` code

```
const getStreams = async function(tracks) {
    let streams = [];
    for (let track of tracks) {
        let stream = await getMedia(track);
        if (stream) {
            streams.push({
                stream: stream,
                encodings: track.encodings,
                source: track.source,
                type: track.type
            });
        }
    }
    return streams;
}
```

5. Получение доступа к медиа устройствам в соответствии с заданными ограничениями

5.1. Настройка ограничений аудио

`getMedia()` code

```
const getMedia = async function(track) {
    //convert to constraints
    let screen = false;
    const constraints= {};
    if (track.source === "mic") {
        //audio
        constraints.audio = {};
        if (track.constraints) {
            constraints.audio = track.constraints;
        }
        constraints.audio.stereo = track.channels !== 1
        if (track.channels && track.channels === 2) {
            constraints.audio.echoCancellation = false;
            constraints.audio.googEchoCancellation = false;
        }
    } else if (track.source === "camera") {
        ...
    } else if (track.source === "screen") {
        ...
    }
    ...
    return stream;
}
```

5.2. Настройка ограничений видео

`getMedia()` code

```
const getMedia = async function(track) {
    //convert to constraints
    let screen = false;
    const constraints= {};
    if (track.source === "mic") {
        ...
    } else if (track.source === "camera") {
        constraints.video = {};
        if (track.constraints) {
            constraints.video = track.constraints;
        }
        constraints.video.width = track.width;
        constraints.video.height = track.height;
    } else if (track.source === "screen") {
        ...
    }
    ...
    return stream;
}
```

5.3. Настройка ограничений для захвата экрана

`getMedia()` code

```
const getMedia = async function(track) {
    //convert to constraints
    let screen = false;
    const constraints= {};
    if (track.source === "mic") {
        ...
    } else if (track.source === "camera") {
        ...
    } else if (track.source === "screen") {
        constraints.video = {};
        if (track.constraints) {
            constraints.video = track.constraints;
        }
        constraints.video.width = track.width;
        constraints.video.height = track.height;
        screen = true;
    }
    ...
    return stream;
}
```

5.4. Получение доступа к локальным медиа устройствам

`getMedia()` code

```
const getMedia = async function(track) {
    //convert to constraints
    let screen = false;
    const constraints= {};
    if (track.source === "mic") {
        ...
    } else if (track.source === "camera") {
        ...
    } else if (track.source === "screen") {
        ...
    }

    //get access to a/v
    let stream;
    if (screen) {
        stream = await navigator.mediaDevices.getDisplayMedia(constraints);
    } else {
        stream = await navigator.mediaDevices.getUserMedia(constraints);
    }
    return stream;
}
```