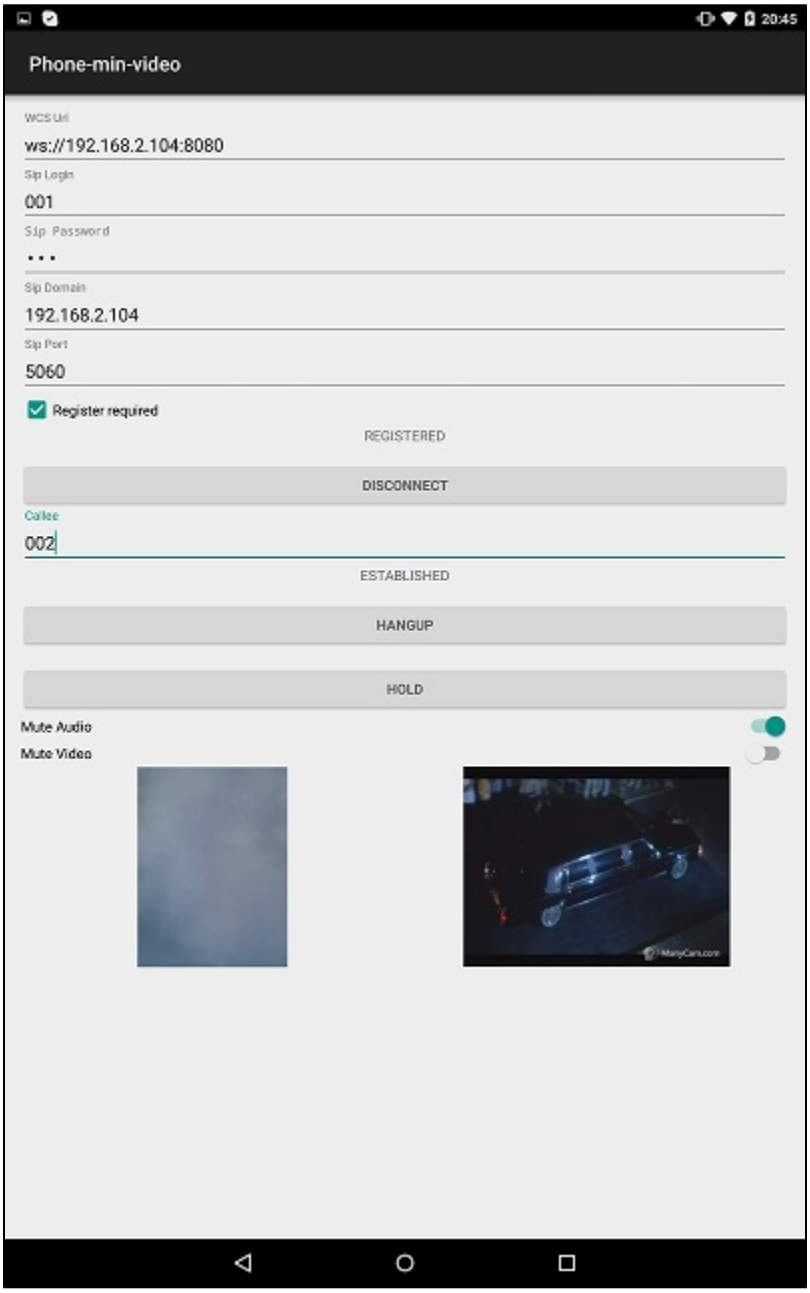
Android Phone Video

Пример Android-приложения для видеозвонков

На скриншоте ниже представлен пример во время звонка.

Интерфейс приложения такой же, как в примере Phone, за исключением того, что добавлены контролы для отключения/включения аудио и видео и воспроизводятся два видео

- слева отображается видео с камеры данного пользователя
- справа воспроизводится видео от другой стороны



Работа с кодом примера

Для разбора кода возьмем класс PhoneMinVideoActivity.java примера phone-min-video, который доступен для скачивания в соответствующей сборке 1.0.1.38.

Для инициализации, исходящего звонка и ответа на входящий звонок используются те же методы, что в примере Phone.

Отличия от примера Phone:

1. Создание сессии.

Flashphoner.createSession() код

Методу передается объект SessionOptions с параметрами:

- URL WCS-сервера
- SurfaceViewRenderer, который будет использоваться для отображения видео с камеры
- SurfaceViewRenderer, который будет использоваться для воспроизведения видео от другой стороны

```
SessionOptions sessionOptions = new SessionOptions(mWcsUrlView.getText().toString());
sessionOptions.setLocalRenderer(localRender);
sessionOptions.setRemoteRenderer(remoteRender);
session = Flashphoner.createSession(sessionOptions);
```

2. Исходящий звонок.

Session.createCall(), Call.call() код

При создании в метод session.createCall() передается объект CallOptions с параметрами:

- SIP логин вызываемого аккаунта
- настройки видео

```
case CALL_REQUEST_CODE: {
    if (grantResults.length == 0 ||
          grantResults[0] != PackageManager.PERMISSION_GRANTED ||
          grantResults[1] != PackageManager.PERMISSION_GRANTED ) {
        Log.i(TAG, "Permission has been denied by user");
    } else {
        mCallButton.setEnabled(false);
        /**
          * Get call options from the callee text field
          */
        CallOptions callOptions = new CallOptions(mCalleeView.getText().toString());
        callOptions.getConstraints().updateVideo(true);
        call = session.createCall(callOptions);
        call.on(callStatusEvent);
        /**
          * Make a new outgoing call
          */
        call.call();
        Log.i(TAG, "Permission has been granted by user");
    }
    break;
}
```

3. Ответ на входящий звонок.

Call.answer() код

```
case INCOMING_CALL_REQUEST_CODE: {
    if (grantResults.length == 0 ||
           grantResults[0] != PackageManager.PERMISSION_GRANTED ||
           grantResults[1] != PackageManager.PERMISSION_GRANTED ) {
        call.hangup();
        incomingCallAlert = null;
        Log.i(TAG, "Permission has been denied by user");
    } else {
        mCallButton.setText(R.string.action_hangup);
        mCallButton.setTag(R.string.action_hangup);
        mCallButton.setEnabled(true);
        mCallStatus.setText(call.getStatus());
        call.getCallOptions().getConstraints().updateVideo(true);
        call.getCallObject().setHasVideo(true);
        call.answer();
        incomingCallAlert = null;
        Log.i(TAG, "Permission has been granted by user");
    }
}
```

4. Включение/выключение аудио и видео.

Call.unmuteAudio(), Call.muteAudio(), Call.unmuteVideo(), Call.muteVideo() код

```
mMuteAudio = (Switch) findViewById(R.id.mute_audio);
/**
  * Mute or Unmute audio for the SIP call
  * Mute if it is not muted.
  * Unmute if it is muted.
  */
mMuteAudio.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {
    public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
        if (call != null) {
            if (isChecked) {
                call.muteAudio();
            } else {
                call.unmuteAudio();
            }
        }
    }
});
mMuteVideo = (Switch) findViewById(R.id.mute_video);
/**
  * Mute or Unmute video for the SIP call
  * Mute if it is not muted.
  * Unmute if it is muted.
  */
mMuteVideo.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {
    public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
        if (call != null) {
            if (isChecked) {
                 call.muteVideo();
            } else {
                 call.unmuteVideo();
            }
        }
    }
});
```