

# Подключение к существующей сессии

Начиная со сборки Android SDK [1.1.0.55](#), поддерживается подключение к существующей websocket сессии на сервере для приема входящего SIP звонка при получении push-уведомления:

1. При создании сессии, установите опцию `keepAlive`

```
Connection connection = new Connection();
connection.setSipLogin(mSipLoginView.getText().toString());
connection.setSipPassword(mSipPasswordView.getText().toString());
...
connection.setKeepAlive(true);
session.connect(connection);
```

2. После успешной установки соединения, сохраните токен сессии

```
@Override
public void onConnected(final Connection connection) {
    runOnUiThread(new Runnable() {
        @Override
        public void run() {
            ...
            String token = connection.getAuthToken();
            if (token != null && !token.isEmpty()) {
                mAuthTokenView.setText(token);
                mConnectTokenButton.setEnabled(true);
            }
        }
    });
}
```

После этого сессия может быть разорвана на мобильном устройстве при уходе приложения в фон, но на сервере сохранится, по умолчанию, в течение 1 часа.

3. При получении push уведомления, подключитесь к существующей сессии по токenu

```
createSession();
Connection connection = new Connection();
connection.setAuthToken(authToken);
connection.setKeepAlive(true);
session.connect(connection);
```

4. Получите сообщение о входящем звонке и создайте диалог для приема/отклонения вызова

```

@Override
public void onCall(final Call call) {
    call.on(callStatusEvent);
    /**
     * Display UI alert for the new incoming call
     */
    runOnUiThread(new Runnable() {
        @Override
        public void run() {
            AlertDialog.Builder builder = new
AlertDialog.Builder(PhoneMinActivity.this);

            builder.setTitle("Incoming call");

            builder.setMessage("Incoming call from '" + call.getCaller() +
""");
            builder.setPositiveButton("Answer", new
DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialogInterface, int
i) {
                    PhoneMinActivity.this.call = call;

                    ActivityCompat.requestPermissions(PhoneMinActivity.this,
new String[]{Manifest.permission.RECORD_AUDIO},
INCOMING_CALL_REQUEST_CODE);
                }
            });
            builder.setNegativeButton("Hangup", new
DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialogInterface, int
i) {
                    call.hangup();
                    incomingCallAlert = null;
                }
            });
            incomingCallAlert = builder.show();
        }
    });
}
}

```

## 5. Примите входящий звонок

```

case INCOMING_CALL_REQUEST_CODE: {
    if (grantResults.length == 0 ||
grantResults[0] != 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.answer();
    }
}

```

```
        incomingCallAlert = null;  
        Log.i(TAG, "Permission has been granted by user");  
    }  
}
```