



Лекция 6

<https://github.com/IFMO-Android-2016/lesson6>

Интеграция с социальными сетями

Одноклассики

ВКонтакте

Зачем?

- Приложение выполняет авторизацию в какой-то социальной сети и использует данные пользователя из социальной сети для персонализации сервисов приложения: кнопка «Войти через...».
- Приложение использует данные и сервисы социальной сети для своей работы: альтернативные клиенты сервисов социальных сетей.
- Использование социального фактора для привлечения аудитории: «Поделиться в...»

Авторизация

Нельзя просто взять и попросить пользователя прямо в вашем приложении ввести логин и пароль от ВКонтакте (это называется фишинг).

Надо предложить пользователю самому зайти в его социальную сеть и оттуда разрешить нашему приложению доступ к его персональным данным.

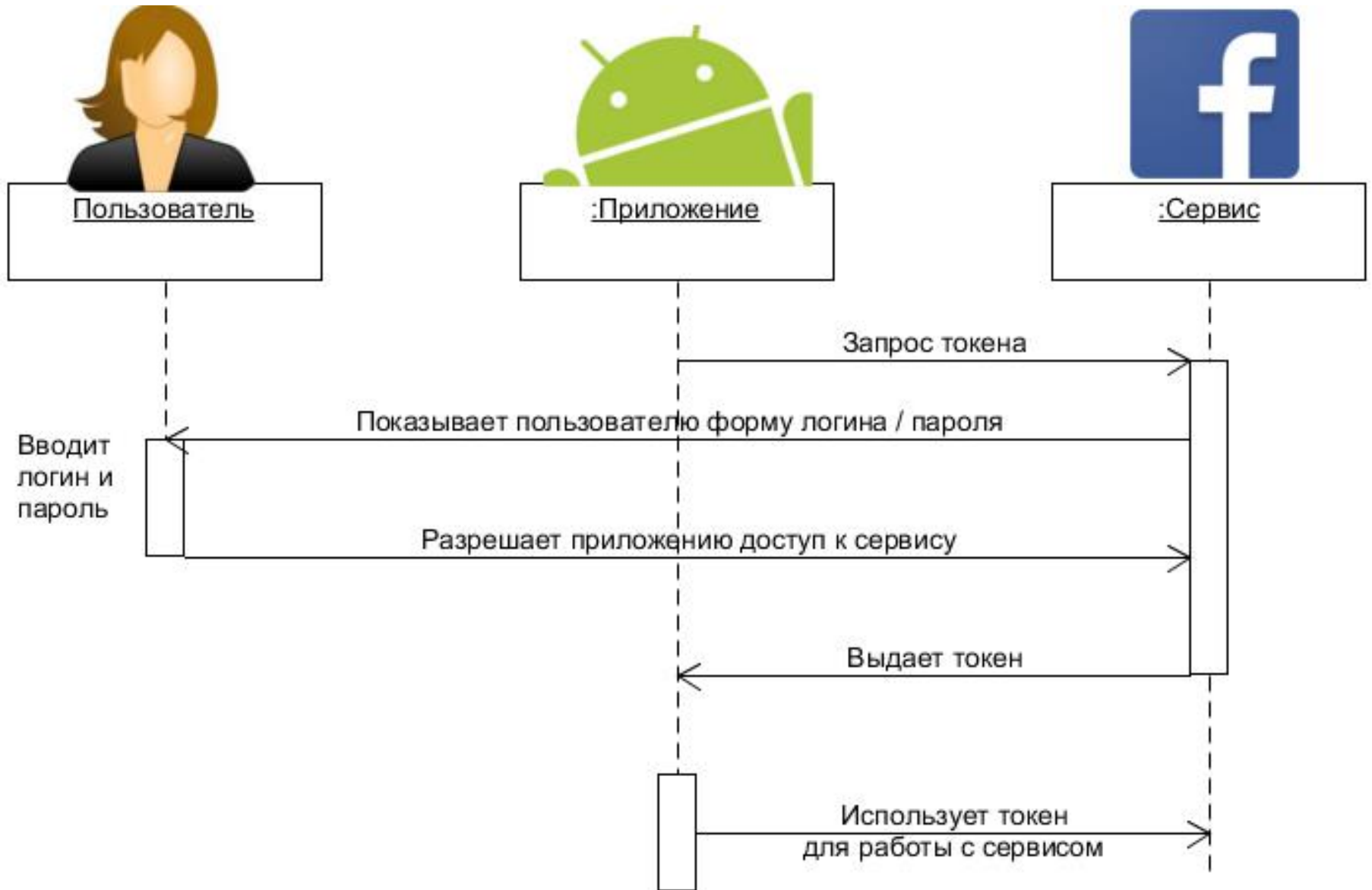
Авторизация

OAuth - открытый протокол авторизации, который позволяет предоставить приложению ограниченный доступ к защищённым ресурсам пользователя (данные и сервисы социальной сети) без необходимости передавать приложению логин и пароль.

OAuth 2.0

1. Приложение обращается к сервису с просьбой дать доступ
2. Сервис взаимодействует с пользователем в *защищенной среде** и получает его одобрение на предоставление доступа
3. Сервис передает приложению авторизационный токен
4. Приложение использует токен для выполнения запросов к API сервиса

OAuth 2.0



*Защищенная среда

- Окно браузера, не подконтрольное приложению
- WebView внутри приложения, показывающее страницу авторизации сервиса
- Собственное приложение сервиса, установленное на устройстве
- Телефонный звонок из службы поддержки пользователю

OAuth в приложениях

Есть два основных способа реализовать OAuth авторизацию в стороннем сервисе:

- Использовать SDK, предоставленный сервисом – просто, но пользователь привязывается к приложению сервиса
- Сделать самим в WebView – немного сложнее, зато авторизация в нашем приложении не связана с авторизацией в приложении сервиса.

Демо №1

Выполнение авторизации OAuth при помощи
WebView на примере сервиса Одноклассники

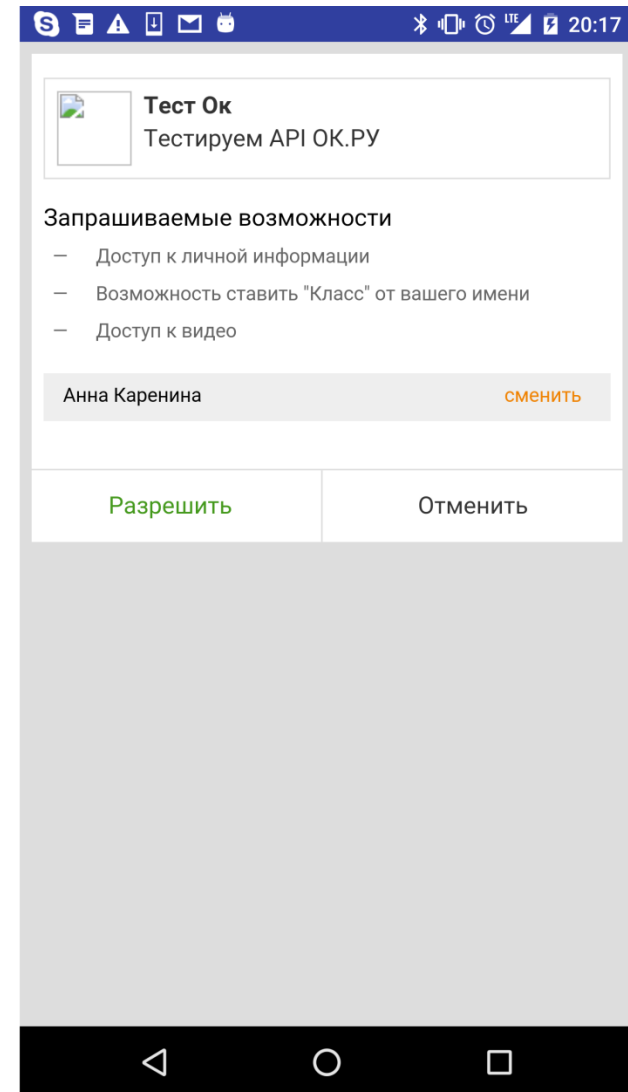
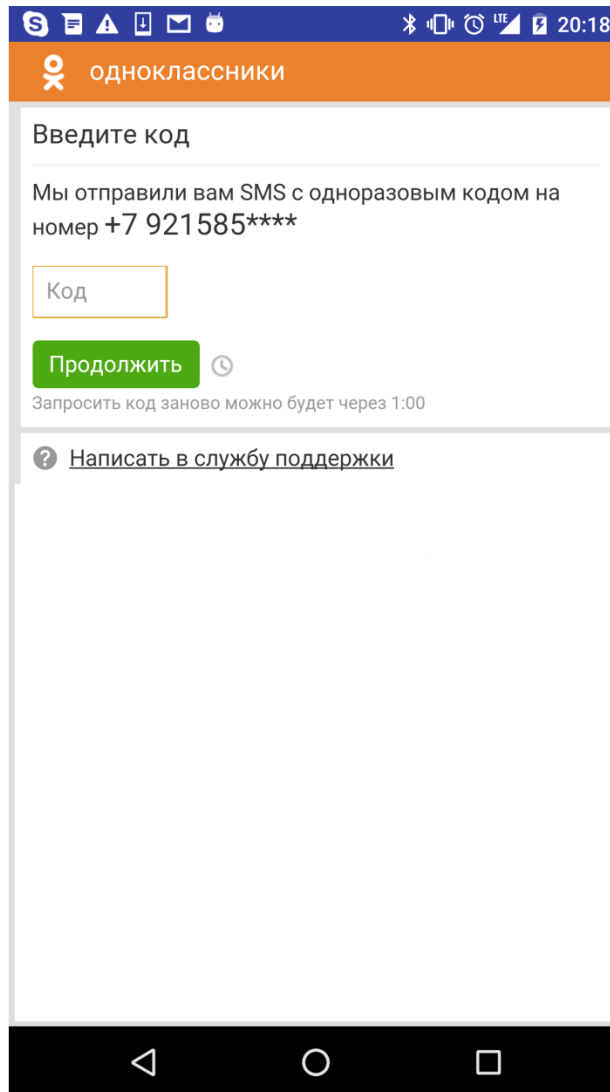
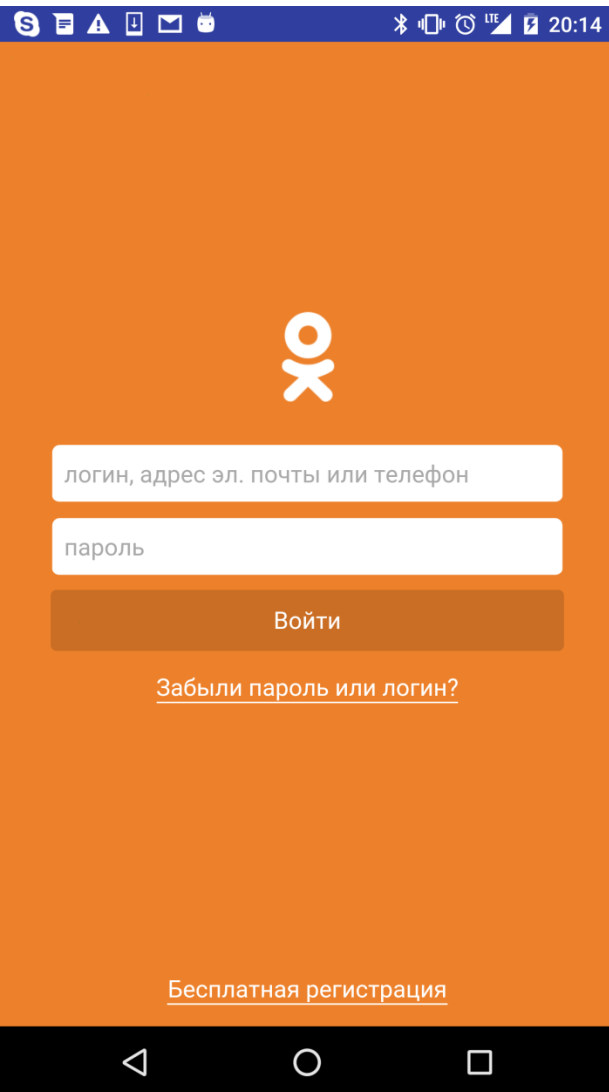
<https://github.com/IFMO-Android-2016/lesson6>

<https://apiok.ru/ext/oauth/client>

OAuth в WebView

1. Показываем начальную страницу OAuth в WebView, передав информацию о приложении в параметрах URL страницы.
2. Пользователь взаимодействует со страницей сервиса, авторизует приложение.
3. Перехватываем редирект на специальный URL, в параметрах которого нам передается токен для доступа к API.

OAuth в WebView



URL страницы OAuth

```
https://www.ok.ru/oauth/authorize ?  
  client_id = <ID приложения> &  
  scope = <Запрашиваемые права> &  
  redirect_uri = <URL для передачи токена> &  
  response_type = token &  
  layout = a // верстка для приложения
```

Нам нужно указать три параметра:

- ID приложения
- Права, нужные нам для работы с API
- URL, по которому мы получим результат

ID приложения

Приложение регистрируется в сервисе (<http://ok.ru/devaccess>) и получает уникальный ID.

Информация о приложении хранится в сервисе:

- Название
- Описание
- Иконка

Запрашиваемые права

Пользователь будет уведомлен о том, какие права запрашивает приложение:

VALUABLE_ACCESS – персональные данные

PHOTO_CONTENT – просмотр фотографий

VIDEO_CONTENT – просмотр видео

MESSAGING – переписка



Тест Ок

Тестируем API OK.RU

Запрашиваемые возможности

- Доступ к личной информации
- Возможность ставить "Класс" от вашего имени
- Доступ к видео

Анна Каренина

[сменить](#)

Разрешить

Отменить

OAuth URL из демо

```
http://www.ok.ru/oauth/authorize ?  
  client_id=396601344 &  
  scope=VALUABLE_ACCESS;VIDEO_CONTENT &  
  response_type=token &  
  redirect_uri=okauth%3A%2F%2Fok396601344 &  
  layout=a
```

WebView

Используется как обычный элемент верстки для отображения веб-страниц

```
<WebView  
    android:id="@+id/web_view_login"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"/>
```

WebView

Открыть страницу в WebView по URL:

```
String oauthUrl = ... // Вычисляем OAuth URL
WebView webView =
    (WebView) findViewById(R.id.web_view_login);
webView.loadUrl(oauthUrl);
```

WebView

Перехватить редирект на URL с результатом:

```
webView.setWebViewClient(new WebViewClient() {  
    public boolean shouldOverrideUrlLoading(  
        WebView view, String url) {  
        if (url.startsWith(redirectUrl)) {  
            parseResultAndFinish(url);  
            return true;  
        }  
        return false;  
    }  
});
```

Метод **shouldOverrideUrlLoading** вызывается при переходе на следующую страницу в WebView

Авторизационные данные

Передаются редиректом на URL,
начинающийся на

`oauth://ok<ID приложения>`

```
oauth://ok396601344 #  
  access_token =  
    1659089be31809...d47b997c04.d6d5 &  
  session_secret_key =  
    0b0bfcba88743d...77beb4344092914
```

Выполнение запросов к API

access_token – передается одним из параметров запроса, служит для идентификации пользователя.

session_secret_key – не передается в запросе, а используется для создания цифровой подписи запроса, служит секретом для предотвращения простых атак.

Цифровая подпись запроса

Подпись = значение хэш-функции MD5 от конкатенации всех пар ключ=значение, отсортированных по алфавиту, и секретного ключа сессии, в шестнадцатеричной записи.

```
MD5("param1=value1param2=value2param3=value3param4=value4param5=value5param6=value6...paramN=valueNsession_secret_key");
```