

Отчет для третьей контрольной точки

Студент: Пугачев Александр

Группа: БПМИ156

Тема проекта: Android-приложение для образовательной платформы "Мой Универ"

В рамках данного проекта мне было необходимо написать мобильное приложение под ОС Android. В основном, для написания приложений под Android необходимо знания языка программирования Java. На первом курсе я изучал объектно-ориентированный язык программирования C++. Знание языка C++ помогло мне без труда освоить синтаксис и логику языка программирования Java. Также, необходимо знания языка разметки XML. Этот язык я освоил достаточно быстро благодаря опыту работы с CSS.

Структура любого Android приложения

Каждое приложение состоит из окон, которые называются *Activity*. В конкретный момент времени обычно отображается одно *Activity*, которое занимает весь экран, а приложение переключается между ними. Одно приложение может содержать как одно *Activity*, так и несколько. Содержимое *Activity* формируется из различных компонентов, называемых *View*. Это может быть кнопка, поле ввода текста, *RadioButton* и многое другое. Файлы *Activity* имеют расширение .java.

Для описания разметки элементов в *Activity* служат файлы с расширением .xml. В них содержится информация о всех элементах, содержащихся в конкретном *Activity*, также разметка и макет (layout).

Также существуют xml файлы, содержащие информацию о названии приложения, шрифтах, цветах и стилях.

Подробное описание моего приложения

В первой версии мое приложение имело 2 *Activity* - главный экран и экран информации о том, куда пойдут студенты. Впоследствии я решил улучшить функциональность своего приложения и освоил работу с *Navigation Drawer* и *Fragment*. *Navigation Drawer* - шаблон, главная особенность которого - наличие бокового меню (Drawer). Боковое меню выдвигается с левой части приложения и содержит несколько пунктов меню. По нажатию на каждый из пунктов, запускается отдельный для каждого пункта *Fragment*. Мое приложение содержит 3 пункта меню: "Главная", "Список" и "О приложении".

В разделе "Главная" находится *Fragment* *fragment_main* с формой для записи. В этом *Fragment* находятся 2 заголовка *TextView*, поле ввода имени *EditText* а также *RadioGroup* с тремя *RadioButton*. В зависимости от того, куда пойдет студент, активизируется соответствующий *RadioButton*. Пользователь вводит информацию в поля ввода *EditText*, затем нажимает кнопку сохранить. По нажатию кнопки выводится Toast с текстом "Записано". Введенная информация сохраняется на сервере.

В разделе "Список" находится *Fragment* *fragment_list* с двумя *TextView*, отображающим какой студент куда пойдет. Информация берется с сервера и выводится в двух *TextView*.

В разделе "О приложении" находится *Fragment* *fragment_about* с информацией о разработчике и руководителе проекта. Также здесь находятся 2 кнопки *Button* для связи с разработчиком.

Реализация обращений к серверу

Для обращения (записи на сервер и чтения с сервера) были использованы методы клиента *OkHttp* для Android и Java.

Запись на сервер

Запись на сервер выполняется следующим образом. Формируется строка с данными, которые требуется записать.

Пример запроса:

http://dev.moyuniver.ru/api/php/v03/_places/api_set_place.php?memberid=28665485147fa7133b44cb&appid=306&appsgn=d8629af695839ba5481757a519e57fb1&appcode=&os=&ver=&width=&height=&place=home, где

memberid=28665485147fa7133b44c - идентификатор студента в системе
place=home - место, куда пойдет студент.

Для формирования запроса был создан шаблон запроса:

```
String finalurl;  
private String mainurl = "http://dev.moyuniver.ru/api/php/v03/_places/api_set_place.php?memberid=";  
private String addurl = "&appid=306&appsgn=d8629af695839ba5481757a519e57fb1&  
appcode=&os=&ver=&width=&height=&place=";
```

Далее студент вводит свой идентификатор и место, куда он пойдет. Эти данные сохраняются в переменные stringName и stringWhere соответственно. Далее по нажатию на кнопку отправления запускается метод, с помощью которого введенная информация записывается на сервер. Реализация следующая:

```
View.OnClickListener btnclick = new View.OnClickListener() {  
    private OkHttpClient OkHttpClient;  
    private Request request;  
  
    public void onClick(View v) {  
        final String stringName = nameText.getText().toString();  
        String stringWhere = edittextWhere.getText().toString();  
        finalurl = mainurl + stringName + addurl + stringWhere; // формируем строку - запрос  
        edittextWhere.setText(""); // очищаем поля  
        nameText.setText("");  
        Toast toast = Toast.makeText(getActivity().getApplicationContext(), // выводим Toast  
            "Записано!", Toast.LENGTH_SHORT);  
        toast.show();  
  
        OkHttpClient = new OkHttpClient();  
        request = new Request.Builder().url(finalurl).build(); // посылаем запрос  
        OkHttpClient.newCall(request).enqueue(new Callback() {  
  
            @Override  
            public void onFailure(Call call, IOException e) {}  
  
            @Override  
            public void onResponse(Call call, Response response) throws IOException {}  
        });  
    }  
};
```

Таким образом формируется строка - запрос finalurl и информация записывается на сервер.

Чтение с сервера

На сервере по запросу

http://dev.moyuniver.ru/api/php/v03/_places/api_get_places.php?memberid=28665485147fa7133b44cb&appid=306&appsgn=d8629af695839ba5481757a519e57fb1&appcode=&os=&ver=&width=&height=

хранится строка с подстроками вида <memberid>#<place>, где

memberid - идентификатор студента в системе

place - место, куда пойдет студент.

Строка, записанная на сервере сохраняется, парсится и каждая подстрока (идентификатор либо место) выводятся в соответствующий *TextView*. Для этого был написан следующий метод:

```
private String url = "http://dev.moyuniver.ru/api/php/v03/_places/api_get_places.php?
memberid=28665485147fa7133b44cb&appid=306&appsgn=d8629af695839ba5481757a519e57fb1&
appcode=&os=&ver=&width=&height=";

...

OkHttpClient = new OkHttpClient();
request = new Request.Builder().url(url).build();

OkHttpClient.newCall(request).enqueue(new Callback() {
    @Override
    public void onFailure(Call call, IOException e) {
    }

    @Override
    public void onResponse(Call call, Response response) throws IOException {
        mainstring = response.body().string(); // сохраняем считанную строку
        int i = 0; // запускаем собственный парсер
        while (i < mainstring.length()) {
            if (mainstring.charAt(i) != '#' && mainstring.charAt(i) != '\n') {
                part = part + mainstring.charAt(i);
                ++i;
            } else if (mainstring.charAt(i) == '#') {
                part = part.replace("null", "");
                final String FirstPart = part; // сохраняем идентификатор
                Handler handler = new Handler(getActivity().getMainLooper());
                handler.post(new Runnable() {
                    @Override
                    public void run() {
                        name = (TextView) getActivity().findViewById(R.id.name);
                        String text = name.getText().toString();
                        String newtext = text + "\n" + FirstPart;
                        name.setText(newtext); // записываем в TextView идентификатор
                    }
                });
                part = "";
                ++i;
            } else if (mainstring.charAt(i) == '\n') {
                ++i;
                final String SecondPart = part; // сохраняем место
                Handler handler = new Handler(getActivity().getMainLooper());
                handler.post(new Runnable() {
                    @Override
                    public void run() {
                        place = (TextView) getActivity().findViewById(R.id.place);
                        String text = place.getText().toString();
                        String newtext = text + "\n" + SecondPart;
                    }
                });
            }
        }
    }
});
```

```

        place.setText(newtext); // записываем в TextView место
    }
} );
part = "";
}
}
});

```

Таким образом, данные с сервера считываются и выводятся в *TextView*.

Инструменты

Весь код был написан при помощи IDE Android Studio 2.3. Минимальная версия API приложения - API 21, Android 5.0 Lollipop (согласно данным на май 2017 года, под управлением Android 5.0 и выше работает более 70% устройств). Приложение тестировалось на следующих смартфонах:

Samsung Galaxy S7, Android 7.0 Nougat, разрешение 2560 x 1440
 HTC One M7, Android 5.0 Lollipop, разрешение 1920 x 1080
 Xiaomi Redmi Note 3 Pro, Android 6.0.1 Marshmallow, разрешение 1920 x 1080

Инструкция по установке

1. Удостовериться, что устройство работает под управлением Android 5.0 или новее
2. В настройках включить опцию "Установка приложений с неизвестных источников"
3. Скачать файл .apk по ссылке <http://bit.ly/2qyTSAJ>
4. Установить приложение
5. Запустить

Источники

Официальная документация от Google: <https://developer.android.com/develop/index.html>
 Сайт <http://www.fandroid.info/>
 Клиент OkHttp: <http://square.github.io/okhttp/>
 Плейлист на YouTube: <https://www.youtube.com/playlist?list=PLyfVjOYzujugap6Rf3ETNKkx4v9ePllNK>
 Поиск ошибок и помощь: <http://stackoverflow.com/>

Скриншоты приложения (Стр. 5)

