

Создание мобильного приложения с помощью сайта

Ульянов Егор Андреевич

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В данной статье рассматривается и описывается создание мобильного приложения на основе сайта. Практическим результатом является приложение для android-устройств.

Ключевые слова: сайт, приложение, андроид

Creating a mobile application using the site

Ulianov Egor Andreevich

Sholom-Aleichem Priamursky State University

Student

Abstract

This article discusses and describes how to create a mobile application based on a site. The practical result is an app for android devices.

Keywords: Site, app, android

На сегодняшний день практически каждый имеет смартфон. Каждый ежедневно пользуется различными мобильными приложениями. Соответственно, если кто-то пользуется приложениями, то должны быть те кто их создает. Имеются очень сложные в плане разработки приложения, но и существуют так же и очень простые.

Цель данной статьи создать мобильное приложение, где основой будет служить сайт.

А.А. Скворода и Д.Ю. Гамаюнов рассмотрели множество мобильных приложений с вредоносной базой и предложили методы для автоматической классификации приложений [1]. В своей работе К.В.Рубинов протестировал большое множество интерфейсов в мобильных приложениях и разработал свой подход к тестированию [2]. В.Г.Орлов и С.С. Шаврин рассмотрели мобильные приложения для использования в системах мониторинга и диспетчеризации технологических служб [3]. С.В. Глазков и А.Л. Ронжин рассмотрели способы формализации и анализа контекста приложений, исполняемых на мобильных гетерогенных устройствах [4].

Для начала создадим новый проект и выберем в предложенных шаблонах «Empty Activity», дадим имя проекту и выберем папку для сохранения (рис.1).

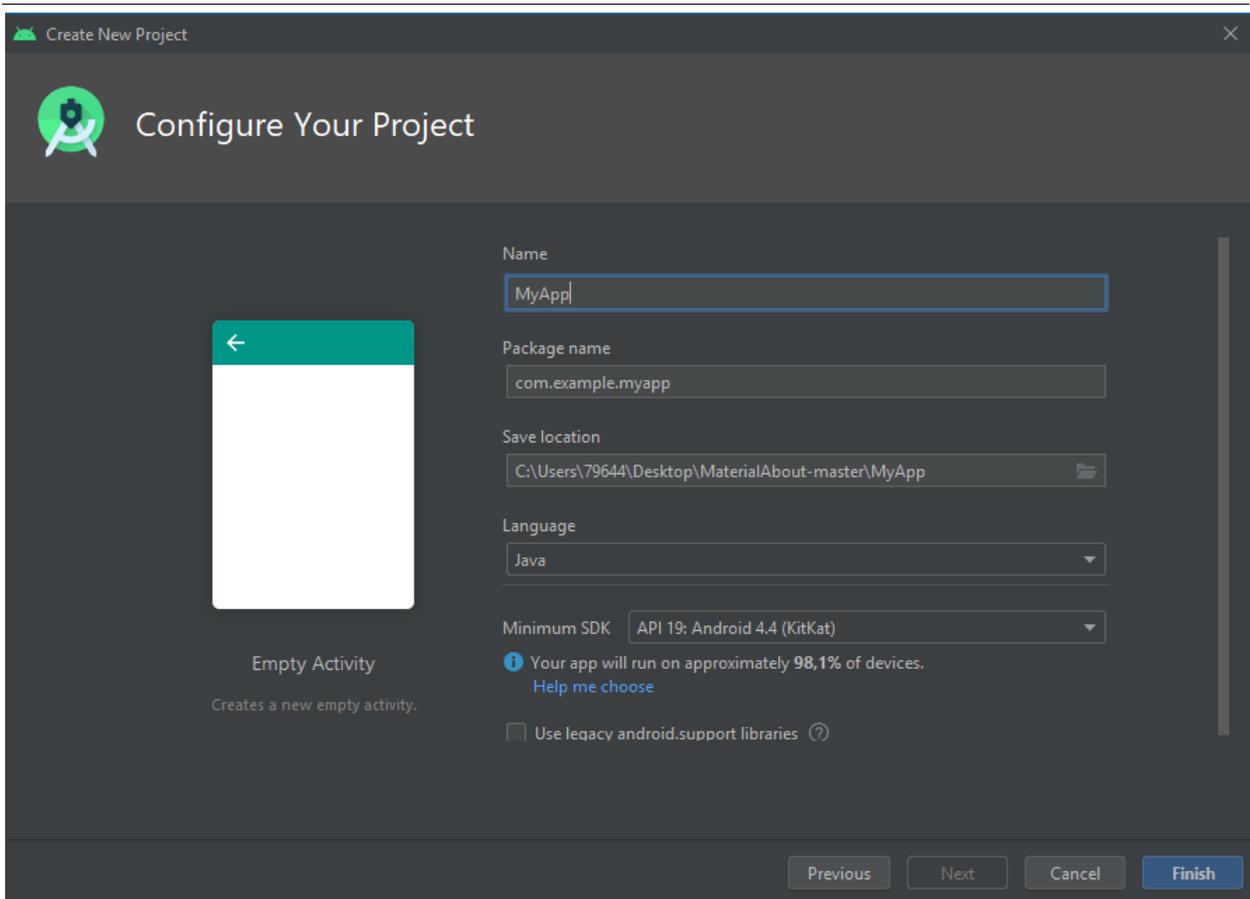


Рисунок 1 – Создание проекта

После создания зайдём в «AndroidManifest» и добавим разрешение для использования интернета, ведь без доступа к интернету ни одно приложение не заработает (рис.2).

```
<uses-permission android:name="android.permission.INTERNET"></uses-permission>
```

Рисунок 2 – Разрешение на использование

Далее зайдём в папку «res» и там откроем «activity_main», в нём необходимо будет удалить часть кода, которая отвечает за «TextView» и напишем новую часть с использованием «WebView» (рис.3).

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res-auto"
3   xmlns:app="http://schemas.android.com/apk/res-auto"
4   xmlns:tools="http://schemas.android.com/tools"
5   android:layout_width="match_parent"
6   android:layout_height="match_parent"
7   tools:context=".MainActivity">
8
9   <WebView
10     android:id="@+id/webview_id"
11     android:layout_width="match_parent"
12     android:layout_height="match_parent" />
13
14 </androidx.constraintlayout.widget.ConstraintLayout>
```

Рисунок 3 – Добавление WebView

И последнее, что осталось сделать, это зайти в «MainActivity» и удалить все строки кода за исключением первой строчки, она дает возможность использовать данный «java» класс во всем проекте (рис.4).

```
14 public class MainActivity extends AppCompatActivity {
15
16     WebView webView;
17
18     @Override
19     protected void onCreate(Bundle savedInstanceState) {
20         super.onCreate(savedInstanceState);
21
22         webView = new WebView(this);
23         webView.getSettings().setJavaScriptEnabled(true);
24
25         final Activity activity = this;
26
27         webView.setWebViewClient(new WebViewClient() {
28             @Override
29             public void onReceivedError(WebView view, int errorCode, String description, String failingUrl) {
30                 Toast.makeText(activity, description, Toast.LENGTH_SHORT).show();
31             }
32             @TargetApi(android.os.Build.VERSION_CODES.M)
33             @Override
34             public void onReceivedError(WebView view, WebResourceRequest req, WebResourceError rerr) {
35                 onReceivedError(view, rerr.getErrorCode(), rerr.getDescription().toString(), req.getUrl().toString());
36             }
37         });
38
39         webView.loadUrl("https://wikipedia.com/");
40
41         setContentView(webView);
42     }
```

Рисунок 4 – MainActivity

В данном классе описаны методы расстановки всех элементов сайта под мобильные устройства, а так же подключение javascript. В последней строчке должен вводиться сайт, который будет портироваться в приложение. В нашем случае, это использование сайта википедия.

Теперь запустим проект и посмотрим на созданное приложение (рис.5).

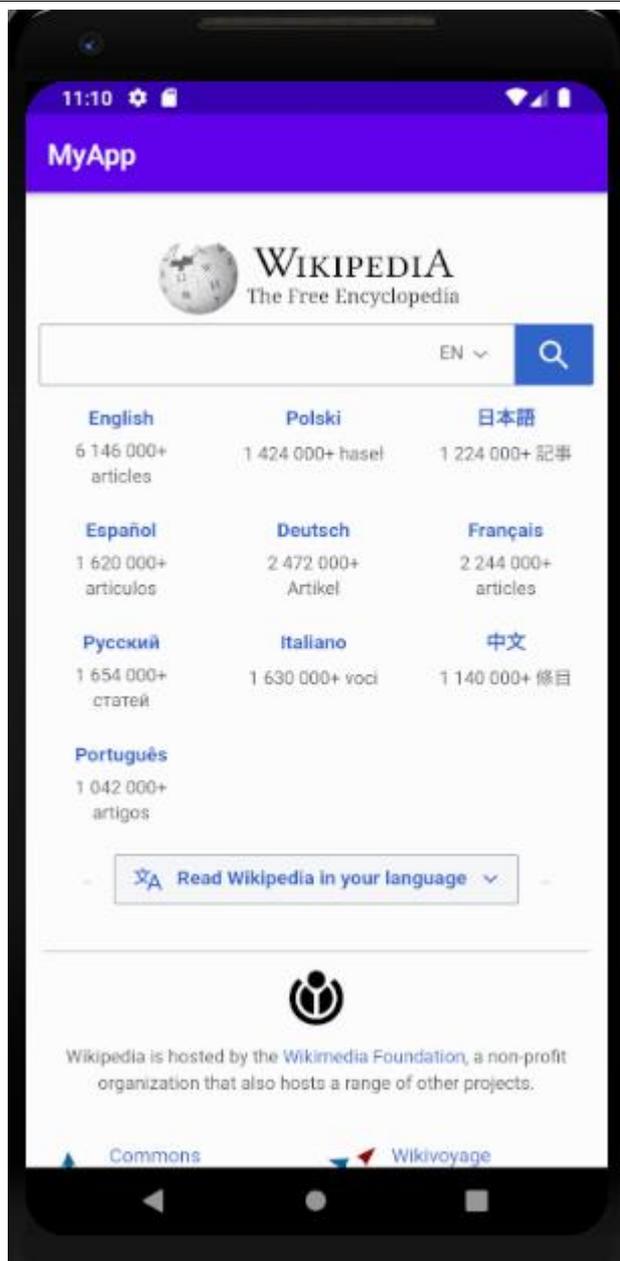


Рисунок 5 – Рабочее приложение

Приложение работает и спокойно отображает весь материал, находящийся на сайте. Главное, что можно портировать таким образом из сайта в мобильное приложение, только сайты имеющих защиту HTTPS.

В данной статье была рассмотрена возможность создания мобильного приложения с использованием «WebView». Практическим результатом является рабочее мобильно приложение.

Библиографический список

1. Сковорода А.А., Гамаюнов Д.Ю. Анализ мобильных приложений с использованием моделей привилегий и api-вызовов вредоносных приложений// Современные научные исследования и инновации. 2018. № 3-5 (63). С. 42-44.

2. Рубинов К.В. Подход к тестированию программных интерфейсов приложений мобильных устройств // Труды Международного симпозиума «Надежность и качество». 2019. №5. С. 16-22.
3. Орлов В.Г., Шаврин С.С. Беспроводные мобильные приложения в системах мониторинга и диспетчеризации технологических служб // Автоматика. Вычислительная техника. 2019. №1. С. 67-71
4. Глазков С.В., Ронжин А.Л. Методы анализа контекста приложений в мобильных гетерогенных устройствах// Автоматика. Вычислительная техника. 2018. №3. С. 34-39