

Alfred  
Samman

Your First  
**ANDROID**  
Application



# Your First Android Application

Alfred Samman

This book is for sale at <http://leanpub.com/androidjavascriptapp>

This version was published on 2020-04-08



Leanpub

This is a [Leanpub](#) book. Leanpub empowers authors and publishers with the Lean Publishing process. [Lean Publishing](#) is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.

© 2014 - 2020 Alfred Samman

# Contents

<b>Preface</b> . . . . .	<b>i</b>
How this Book is Organized . . . . .	i
Who Should Read this Book . . . . .	i
What is Covered in this Book . . . . .	i
Source Code . . . . .	ii
<b>Introduction</b> . . . . .	<b>1</b>
HTML5 . . . . .	1
DOM . . . . .	1
Native Apps versus Web Apps . . . . .	2
<b>Detecting Swipe Actions</b> . . . . .	<b>3</b>
Detecting the start position of your touch . . . . .	3
Transitioning a swiped element . . . . .	3
Detecting when the swipe ends . . . . .	4
Summary . . . . .	8
<b>Running your App in the Android Emulator</b> . . . . .	<b>9</b>

# Preface

With the recent proliferation of Apps, I felt that there was a need for a book to teach the average Jane how to develop their own Apps. The web has been with us for a while now and many of the technologies and terminologies are familiar to us. We are familiar with what a webpage is, what a link is, what a browser is etc. Building on this familiarity, my aim is to show that with relative ease anyone can start developing their own App.

## How this Book is Organized

This book is divided into self contained chapters of increasing difficulty. The best way to read this book is to read it in order, from the first chapter to the last. However, you can also jump between chapters and pick topics that interest you. Each chapter is has accompanying code that can be downloaded and run, or read from within the text.

## Who Should Read this Book

This book can be read by a complete novice. It can also be read by anyone with basic web development knowledge. You do not have to be an expert developer to gain the full advantage of this book.

## What is Covered in this Book

In this book we will learn how to create what is known as a WebView Android App. The WebView is an Android container that is capable of rendering a webpage. So essentially we will be learning how to develop an Android App that opens up a webpage.

The book begins with a gentle introduction to the main web technologies, HTML, CSS and Javascript. We then move on to learn how to set up a development environment for developing our Apps. You will then create your first simple App. Subsequent chapters touch on the DOM, touch events, saving data, swipe action, AJAX and finally how to publish your App.

## Source Code

The companion code and assets for each chapter can be downloaded from Github.

<https://github.com/budu3/the-book>

You can also use git to download them by issuing the following command. <sup>1</sup>

```
git clone https://github.com/budu3/the-book.git
```

Alternatively, you can use your Android studio to download them by doing the following,

1. VCS -> Checkout from Version Control -> Git
2. Enter the URL: <https://github.com/budu3/the-book.git>
3. Click the **Clone** button
4. VCS -> Git -> Branches
5. Select the branch corresponding to the example code that you want to download
6. Click **Checkout As** then click the **OK** button

---

<sup>1</sup>I will not go into Git details here. You can use your search engine to find suitable tutorials on Git.

# Introduction

The Apple App Store made its debut on July on the iPhone 2 and has not looked back since<sup>2</sup>. To say that the App Store has been a success is an understatement. As of the date of writing this book the Apple App Store has a record 800 000 apps available for download. The concept of an “App” and an “App Store” has now been adopted by a plethora of platforms including, Google (Google Play), Amazon (Amazon Appstore for Android), Microsoft (Windows Store), Firefox (Firefox Marketplace), Chrome (Chrome Web Store) and a host of other platforms. It seems like every device today has an app Store and for every task out there, there’s an App for that.

In this book we will learn how to create Android WebView Apps. The aim is to show a novice how to develop simple Web apps for mobile devices. Before you can develop Apps you’ll need to know some basics web technologies. Your App will consist of 3 main components, HTML5, CSS and Javascript. HTML5 is used for presentation, CSS is used for styling and Javascript is used to code the actual functionality of your app.

## HTML5

HTML5 (HyperText Markup Language 5) will be used to create all the visual elements of your app.

## DOM

The DOM (Document Object Model) is an object representation of the elements that make up your App. It is organized as a tree. Every Web App consists of a DOM and the functionality of your App usually involves manipulating the DOM.

We will discuss the aforementioned technologies later on in this book.

---

<sup>2</sup><http://9to5mac.com/2011/10/21/jobs-original-vision-for-the-iphone-no-third-party-native-apps/>

## **Native Apps versus Web Apps**

Native Apps are written specifically for the platform on which they run on. They take advantage of the platform's native user interface elements. Web Apps on the other hand are developed using web technologies, can be viewed in a browser and can usually work on any platform with a compliant browser.

# Detecting Swipe Actions

In this chapter we will add a swipe effect to our app so that when you swipe an item it disappears from the screen.

Performing a swipe action involves three major tasks. Detecting the point at which the user begins swiping, detecting the point at which the user stops swiping and then transitioning or moving the HTML element between these two positions.

## Detecting the start position of your touch

To detect the start position of a touch, we attach a function to the touchstart event. We then use `changeTouches[0].pageX` to get the starting point of the touch as shown below.

```
1 element.addEventListener('touchstart',function(e){
2     startX = parseInt(e.changedTouches[0].pageX);
3 });
```



Note that `TouchEvent.changedTouches` will return a list of touch points. `Touch.pageX` will return the x coordinate of a particular touch point

## Transitioning a swiped element

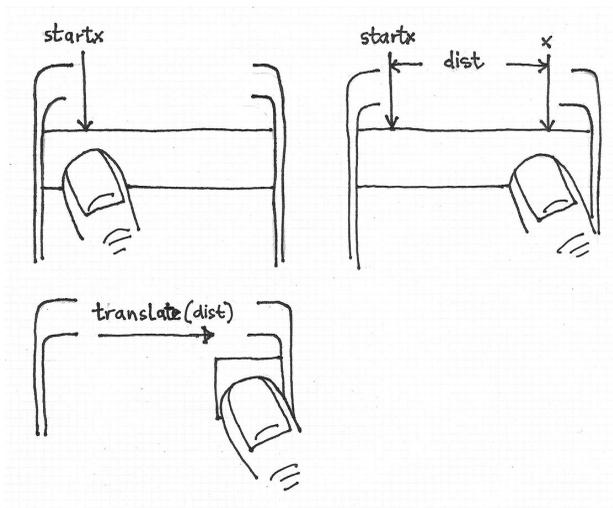
Now, to produce the swiping effect we translate/move the touched list element the distance that the finger has moved horizontally along the x axis. This is done by manipulating the list element's `webkitTransform` style attribute as shown below.

```
1 element.style.webkitTransform = 'translate(' + dist + 'px, 0)';
```

## Detecting when the swipe ends

Finally, we need to detect when the swipe action is over. We do this by simply listening to the touchend event. Whatever action that needs to be performed when the touch ends can be attached to the event as shown.

```
1 element.addEventListener('touchend', function(e){  
2     ...  
3 })
```



image

The code listing below adds swipe functionality to your app. Type, save and open the code below in your Browser.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <title>todo list</title>
5  <link href="css/main.css" rel="stylesheet" type="text/css">
6  </head>
7  <body>
8
9  <div class="container">
10   <ul>
11
12   </ul>
13   <input type="text" id="add-item"><input type="submit" id="submit-btn"></input>
14 </div>
15 <script>
16
17   //retrieve items from local storage
18   retrieveFromStorage();
19
20   //dynamically add an <li> element when the submit button is clicked
21
22   var submit_btn = document.getElementById('submit-btn'); //get the submit button
23   //attach a function to the submit button
24   submit_btn.onclick = function additem(){
25     var count = new Date().getTime();
26     var textInput = document.getElementById('add-item');
27     var item = textInput.value;
28     var ul = document.getElementsByTagName('ul');
29     var li;
30     var img;
31
32     createListElement(count,item);
33     localStorage.setItem(count, item);
34   }
35
36   function retrieveFromStorage(){
37     var ul_item = document.getElementsByTagName('ul');
38
39     for(index in localStorage){
40
41       var itemValue = localStorage[index];
42       var li;
43
44       li = createListElement(index, itemValue);
45     }
46   }
47
48
49   function createListElement(id, value){
50     var textInput = document.getElementById('add-item');
51     var ul = document.getElementsByTagName('ul');
```

```
52     var li;
53     var img;
54
55     ul[0].insertAdjacentHTML('beforeend', '<li id="' + id + '" class="item">' + value + '</li>');
56
57     li = document.getElementById(id);
58
59     li.addEventListener("touchstart", handleStart);
60     li.addEventListener("touchmove", handleMove);
61     li.addEventListener("touchend", handleEnd);
62
63     swipe(li);
64
65     li.onclick = strikeThrough;
66     li.addEventListener("touchstart", strikeThrough);
67 }
68
69
70 function handleStart(e){
71     e.preventDefault();
72     this.style.backgroundColor = "#FFCC66";
73 }
74
75 function handleMove(e){
76     e.preventDefault();
77     this.style.backgroundColor = "#FFCC66";
78 }
79
80 function handleEnd(e){
81     e.preventDefault();
82     this.style.backgroundColor = "#FFFFFF";
83 }
84
85 function strikeThrough(e){
86     e.preventDefault();
87
88     if (this.style.textDecoration == "none"){
89         this.style.textDecoration = "line-through";
90     }else{
91         this.style.textDecoration = "none";
92     }
93 }
94
95 function swipe(element){
96     var startx;
97     var dist;
98     var delta;
99
100    element.addEventListener('touchstart', function(e){
101        e.preventDefault();
102        startx = parseInt(e.changedTouches[0].pageX);
```

```
103     });
104
105     element.addEventListener('touchmove', function(e){
106         e.preventDefault();
107         var x = parseInt(e.changedTouches[0].pageX);
108         dist = x - startX;
109
110         move(element,dist,0);
111     });
112
113     element.addEventListener('touchend', function(e){
114         if (dist > 100){
115             localStorage.removeItem(this.id);
116             animate(element, 4);
117         }else{
118             delta = (dist * -1)/2;
119             move(element,dist * -1,50);
120             move(element,0,0);
121         }
122     })
123 }
124
125 function move(element, dist, speed){
126     element.style.transitionDuration = speed + 'ms';
127     element.style.webkitTransform = 'translate(' + dist + 'px,0)';
128 }
129
130 function slide(e) {
131
132     var speed = 10;
133     var dist = 1000;
134
135     this.style.transform = 'translateX(10px)';
136     this.style.transitionDuration = speed + 'ms';
137     this.style.webkitTransform = 'translate(' + dist + 'px,0)' + 'translateZ(0)';
138 }
139
140
141 function animate(element, speed){
142     var children = element.childNodes;
143
144     element.style.setProperty('-webkit-transition', 'all 0.5s');
145     element.style.setProperty('height', '0px');
146     element.style.setProperty('padding', '0px');
147     element.style.setProperty('border', '0px');
148     element.removeChild(children[0]); //remove text node
149 }
150
151 </script>
152 </body>
153 </html>
```

Look at the *Running your Simple App* chapter for steps on how to run this example in Android Studio.

## Summary

In this chapter we learnt how to detect that a user swipe action has taken place and how to retrieve touch points.

# Running your App in the Android Emulator

The Android Emulator is essentially a virtual Android device that runs on your computer. You can use it as an alternative to running your App on a physical device.

The steps below will guide you on how to set up and run the emulator.

## 1. Click Run

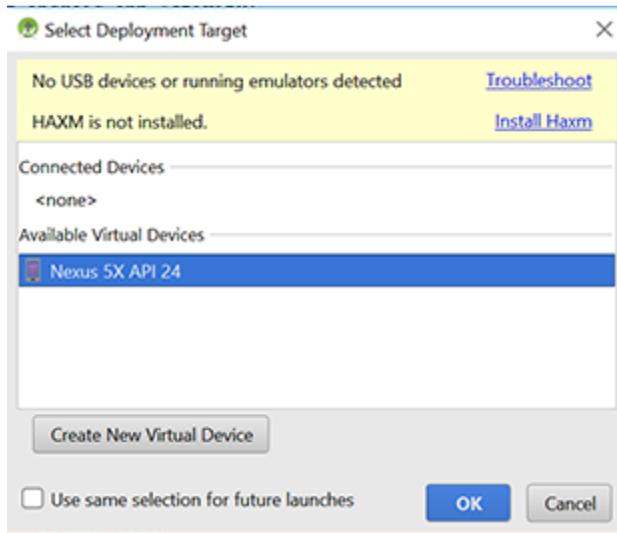
- In Android Studio to run an App in the emulator you can simply click the Run icon in the toolbar or you can go to Run -> Run 'app' in the menu bar.
- The *Select Deployment Target* dialog box will appear as shown in the figure.

## 2. Create New Virtual Device

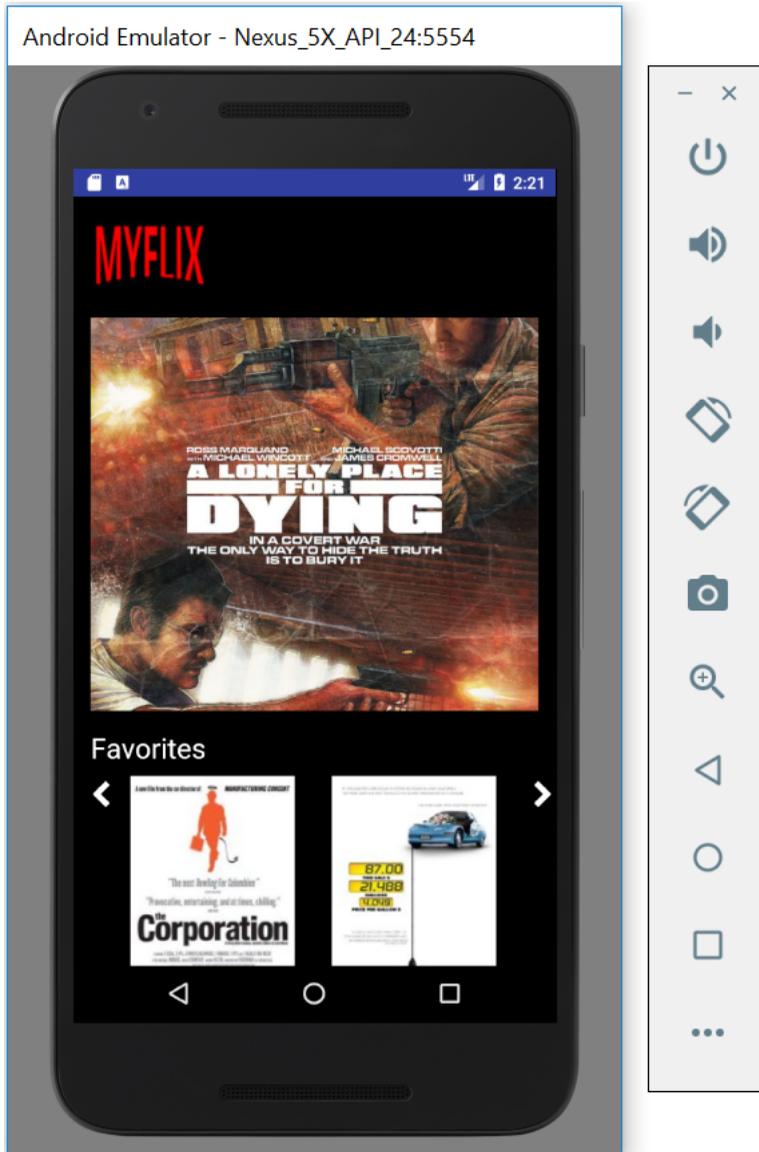
- Click on the "Create New Virtual Device" button in the *Select Deployment Target* dialog box.
- Download a system image. Choose the image that you prefer from the *Select a system image* screen. I downloaded the *Nougat* system image.
- Follow the prompts that will appear.

## 3. Select Deployment Target

- Select a device in the *Select Deployment Target* dialog box and click *OK*. Your app will then open up in the emulator as shown in the figure.



Select Deployment Target dialog box



Android Emulator



During the process of setting up the emulator, Android Studio might prompt you to download certain recommended programs. HAXM (Intel Hardware Accelerated Execution Manager) or KVM (Kernel-based Virtual Machine) are examples of this. HAXM and KVM use hardware acceleration to improve the performance of your emulator.