

WebView is a view that display web pages inside your application. You can also specify HTML string and can show it inside your application using WebView. WebView makes turns your application to a web application.

In order to add WebView to your application, you have to add <WebView> element to your xml layout file. Its syntax is as follows –

```
<WebView xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/webview"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
/>
```

In order to use it, you have to get a reference of this view in Java file. To get a reference, create an object of the class WebView. Its syntax is –

```
WebView browser = (WebView) findViewById(R.id.webview);
```

In order to load a web url into the WebView, you need to call a method loadUrl(String url) of the WebView class, specifying the required url. Its syntax is:

```
browser.loadUrl("http://www.tutorialspoint.com");
```

Apart from just loading url, you can have more control over your WebView by using the methods defined in WebView class. They are listed as follows –

Sr.No	Method & Description
1	<p>canGoBack()</p> <p>This method specifies the WebView has a back history item.</p>

2	<code>canGoForward()</code> This method specifies the WebView has a forward history item.
3	<code>clearHistory()</code> This method will clear the WebView forward and backward history.
4	<code>destroy()</code> This method destroy the internal state of WebView.
5	<code>findAllAsync(String find)</code> This method find all instances of string and highlight them.
6	<code>getProgress()</code> This method gets the progress of the current page.
7	<code>getTitle()</code> This method return the title of the current page.
8	<code>getUrl()</code> This method return the url of the current page.

If you click on any link inside the webpage of the WebView, that page will not be loaded inside your WebView. In order to do that you need to extend your class from WebViewClient and override its method. Its syntax is –

```
private class MyBrowser extends WebViewClient {  
    @Override  
    public boolean shouldOverrideUrlLoading(WebView view, String url) {  
        view.loadUrl(url);  
        return true;  
    }  
}
```

Example

Here is an example demonstrating the use of WebView Layout. It creates a basic web application that will ask you to specify a url and will load this url website in the WebView.

To experiment with this example, you need to run this on an actual device on which internet is running.

Step s	Description
1	You will use Android studio to create an Android application under a package com.example.sairamkrishna.myapplication.
2	Modify src/MainActivity.java file to add WebView code.
3	Modify the res/layout/activity_main to add respective XML components
4	Modify the AndroidManifest.xml to add the necessary permissions

5

Run the application and choose a running android device and install the application on it and verify the results.