

Getting Your Hooks In Overriding WebView Behaviour

Noah Adams

Software Engineer at Mobify

noah@mobify.com



The WebView Client

`android.webkit.WebViewClient`

Subclass this, and supply it to your `webView` for all sorts of fun callbacks:

```
public class WebViewClient {
    public boolean shouldOverrideUrlLoading(WebView view, String url)
    public void onPageStarted(WebView view, String url, Bitmap favicon)
    public void onPageFinished(WebView view, String url)
    public void onLoadResource(WebView view, String url)
    public WebResourceResponse shouldInterceptRequest(WebView view, String url)
    public void onTooManyRedirects(WebView view, Message cancelMsg, Message continueMsg)
    ...
}
```

What Kinds of Callbacks?

- **Web browsing lifecycle:** onPageStarted, onPageFinished, onLoadResource, doUpdateVisitedHistory...
- **Security related:** onReceivedClientCertRequest, onReceivedHttpAuthRequest...

What Kinds of Callbacks?

- **Errors:** `onReceivedError`, `onTooManyRedirects...`
- **Opportunities to override behaviour:**
`shouldOverrideUrlLoading`, `shouldInterceptRequest`

Web Browsing Lifecycle

- Update UI based on state of navigation
- Callbacks drive states of Transition managers, etc.
- Gotcha:
 - This is asynchronous, not part of UI event loop
 - Use callbacks to send messages through a message/event bus
 - You will probably have to model the state of the WebView
- Can drive some of this through JavaScript bridge as well

Overriding behaviour:

```
public boolean shouldOverrideUrlLoading(Webview view, String url) { }
```

- Lets you know where the WebView is going, before its navigated
- An opportunity to send the user into a different App using intents:
 - e-mail for `mailto:` links
 - A browser or other app for links to a different domain
- A different part of your app (e.g. not a WebView) for certain paths

How do you integrate it?

```
public class MyWebViewClient extends WebViewClient {  
    private static final String TAG = WebViewClient.class.getName();  
    public boolean shouldOverrideUrlLoading(WebView view, String url) {  
        Log.d(TAG, "Loaded: " + url);  
        return false;  
    }  
}
```


How do you integrate it?

Then, when you initialize your webView:

```
webView.setWebViewClient(new MyWebViewClient(applicationContext));
```

