Debugging webviews on Android and iOS

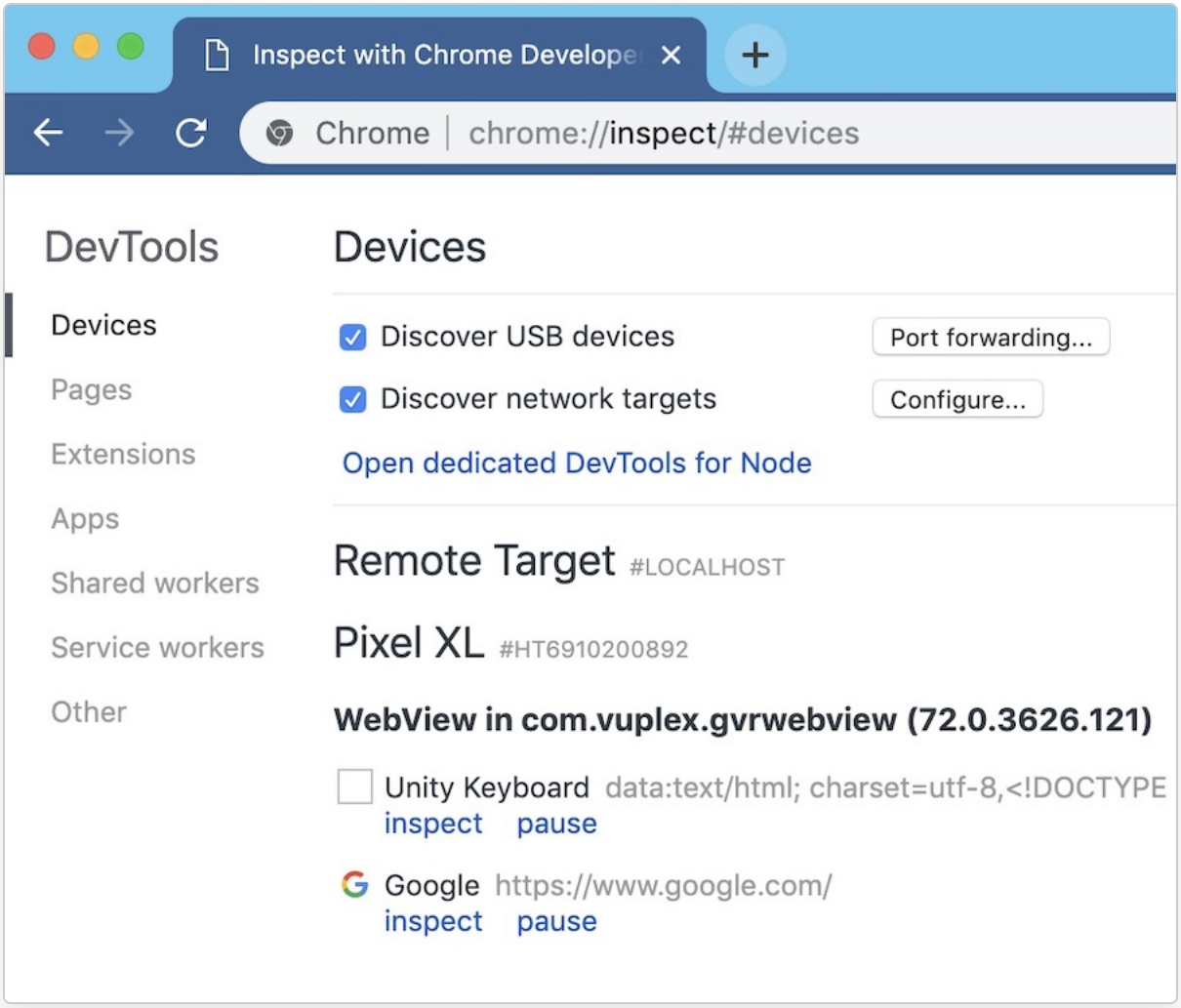
It's sometimes helpful to debug a webpage rendered in a webview in order to inspect its layout or to troubleshoot JavaScript. Fortunately, it's easy to remotely inspect webviews that are running in your Android or iOS application. This entails connecting your Android or iOS device to your dev computer and then using your browser's developer tools to inspect your app's webviews.

Debugging Android webviews

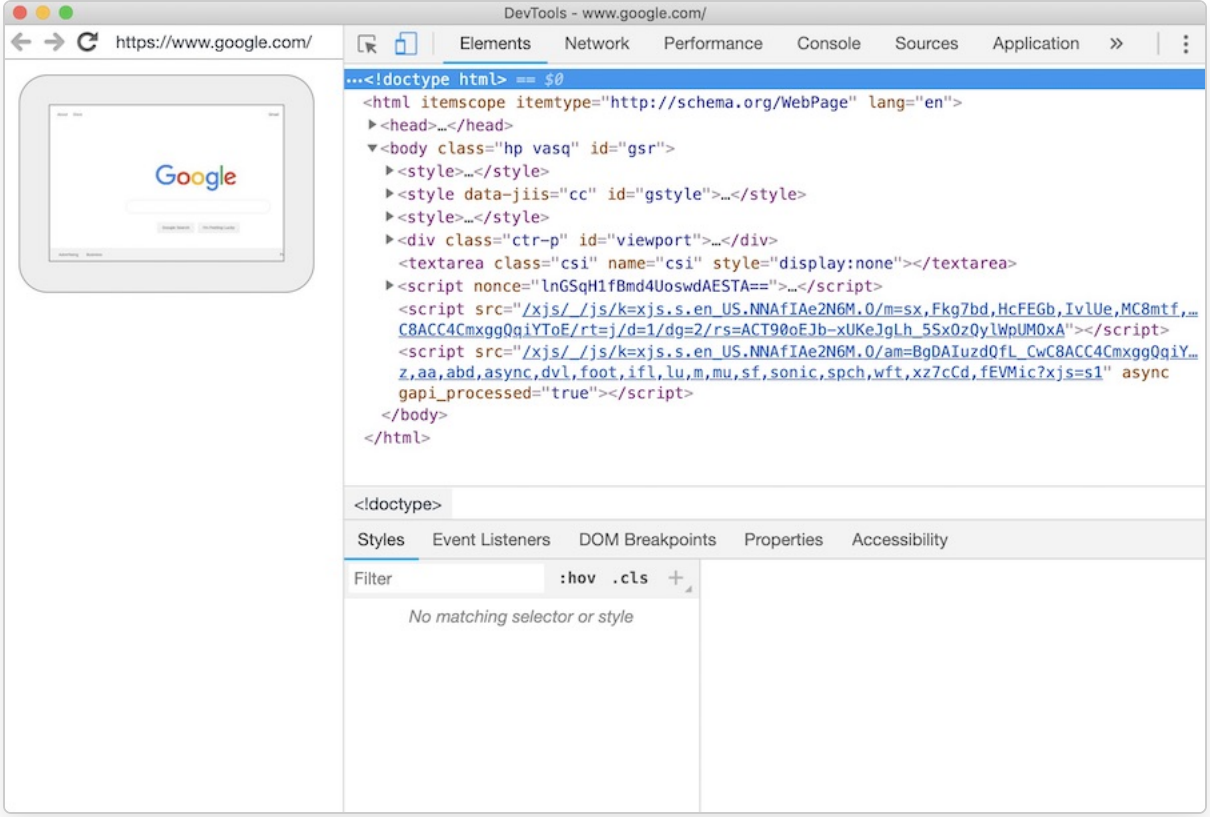
One way to troubleshoot JavaScript on Android is to insert `console.log()` statements into your code, since those messages are printed to the Logcat logs that are viewable in Android Studio. However, it's often much more helpful to inspect the page with Chrome DevTools, because that allows you to use its JavaScript debugger and other essential tools. To do that, just follow these steps:

1. First, enable USB debugging on your Android device. If you use the device for development, then you've probably done this already.
2. Install Chrome on your dev computer if you haven't yet.
3. Connect your Android device to your dev computer and launch your application.

4. Launch Chrome on your dev computer and navigate to the url `chrome://inspect`. This page shows you a list of webview instances running on your connected Android device.



5. Click the *inspect* link next to the webview that you wish to debug. This will open a new Chrome DevTools window for inspecting the webview.



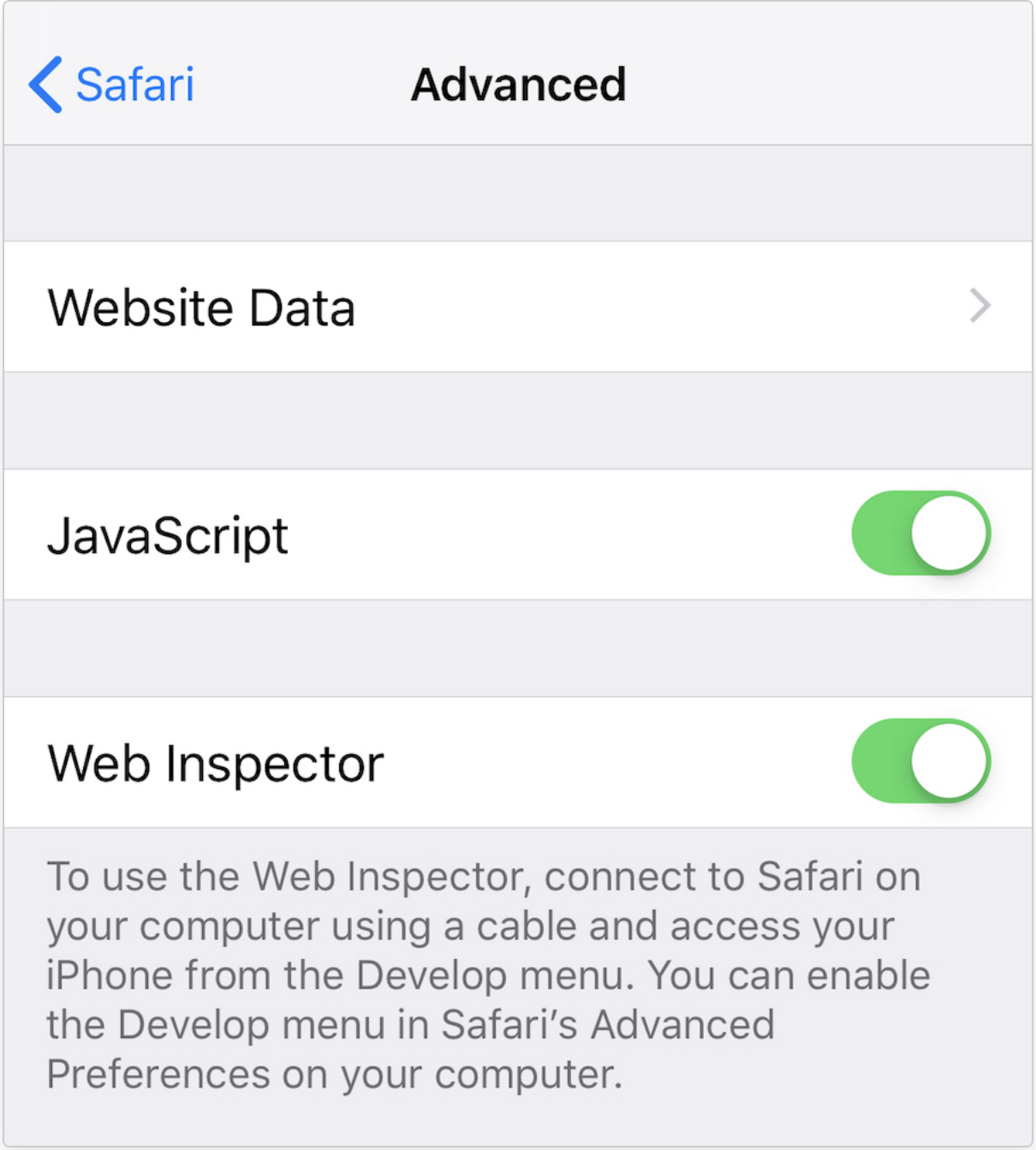
Debugging iOS webviews

In iOS webviews, messages passed to `console.log()` are *not* printed to the Xcode logs. It's still relatively easy to debug web content with Safari's developer tools, although there are a couple of limitations:

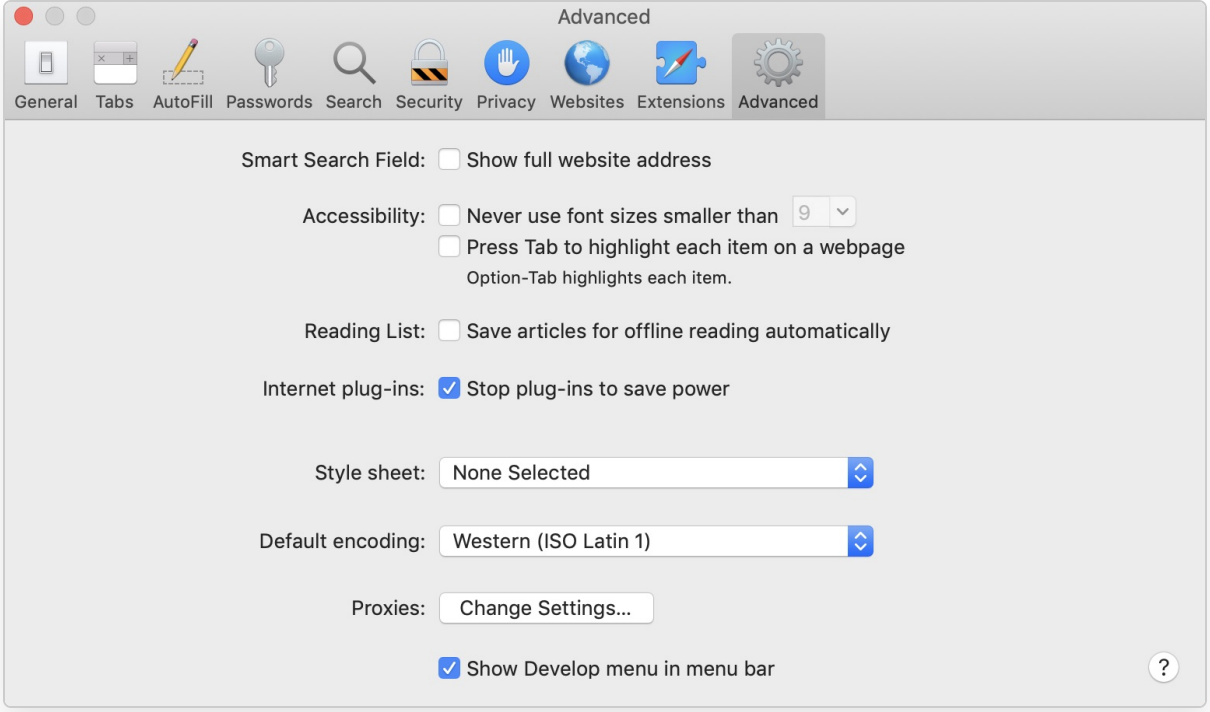
- Debugging iOS webviews requires Safari, so your dev computer must be running macOS.
- You can only debug webviews in applications loaded onto your device through Xcode. You can't debug webviews in apps installed through the App Store or Apple Configurator.

With those limitations in mind, here are the steps to remotely debug a webview in iOS:

1. First, enable the Safari Web Inspector on your iOS device by opening the iOS *Settings* app, navigating to **Settings > Safari > Advanced**, and toggling the *Web Inspector* option on.

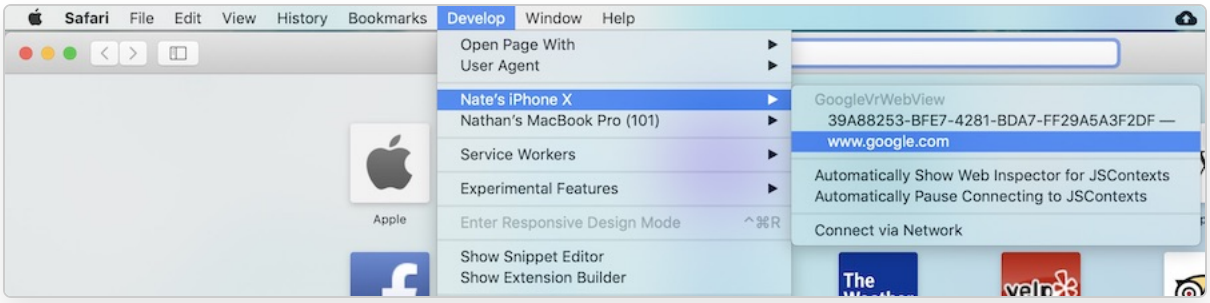


2. Next, you must also enable developer tools in Safari on your dev computer. Launch Safari on your dev machine and navigate to **Safari > Preferences** in the menu bar. In the preferences pane that appears, click on the *Advanced* tab and then enable the *Show Develop menu* option at the bottom. After you do that, you can close the preferences pane.

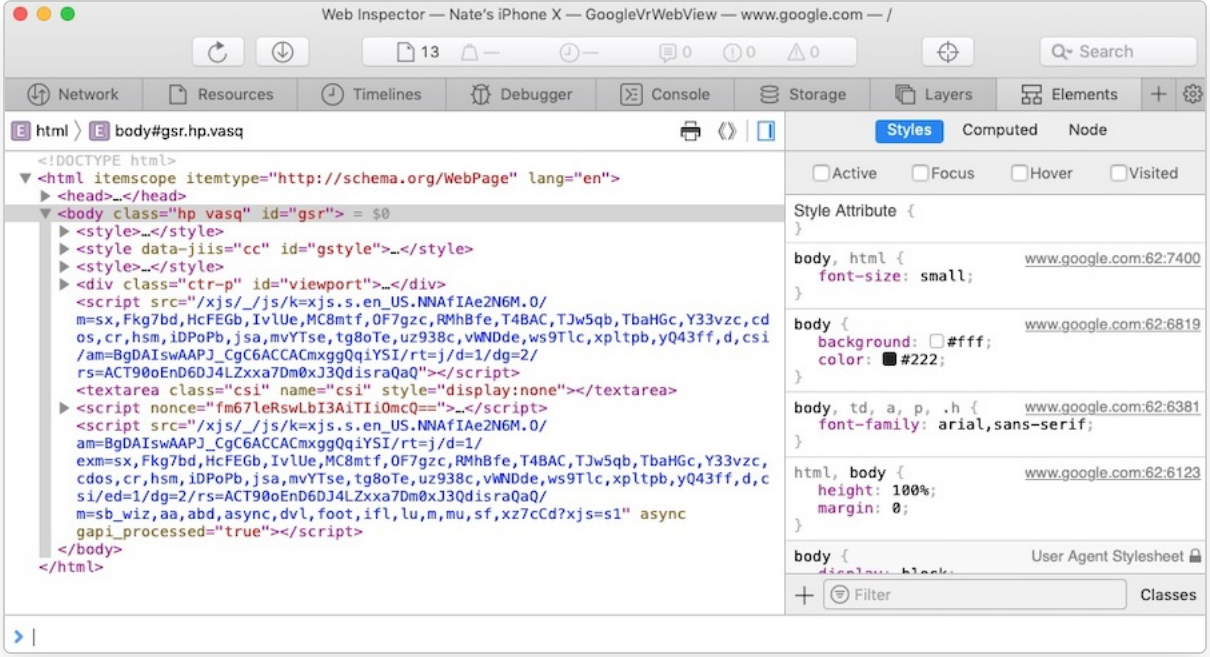


3. Connect your iOS device to your dev computer and launch your app.

4. In Safari on your dev computer, click on *Develop* in the menu bar and hover over the dropdown option that is your iOS device's name to show a list of webview instances running on your iOS device.



5. Click the dropdown option for the webview that you wish to debug. This will open a new Safari Web Inspector window for inspecting the webview.


Web Inspector — Nate's iPhone X — GoogleVrWebView — www.google.com — /
Search
Network
Resources
Timelines
Debugger
Console
Storage
Layers
Elements
html
body#gsr.hp.vasq
<!DOCTYPE html>
<html itemscope itemtype="http://schema.org/WebPage" lang="en">
<head>…</head>
<body class="hp vasq" id="gsr"> = $0
<style>…</style>
<style data-jiis="cc" id="gstyle">…</style>
<style>…</style>
…
<script src="/xjs/_/js/k=xjs.s.en_US.NNAfIAe2N6M.O/
m=sx,Fkg7bd,HcFEGb,IvlUe,MC8mtf,OF7gzc,RMhBfe,T4BAC,TJw5qb,TbaHGc,Y33vzc,cd
os,cr,hsm,iDPoPb,jsa,mvYTse,tg8oTe,uz938c,vWNDde,ws9Tlc,xpltpb,yQ43ff,d,csi
/am=BgDAIswAAPJ_CgC6ACCACmxggQqiYSI/rt=j/d=1/dg=2/
rs=ACT90oEnD6DJ4LZxxa7Dm0xJ3QdisraQaQ"></script>
<textarea class="csi" name="csi" style="display:none"></textarea>
<script nonce="fm67leRswLbI3AiTIiOmcQ==">…</script>
<script src="/xjs/_/js/k=xjs.s.en_US.NNAfIAe2N6M.O/
am=BgDAIswAAPJ_CgC6ACCACmxggQqiYSI/rt=j/d=1/
exm=sx,Fkg7bd,HcFEGb,IvlUe,MC8mtf,OF7gzc,RMhBfe,T4BAC,TJw5qb,TbaHGc,Y33vzc,
cdos,cr,hsm,iDPoPb,jsa,mvYTse,tg8oTe,uz938c,vWNDde,ws9Tlc,xpltpb,yQ43ff,d,c
si/ed=1/dg=2/rs=ACT90oEnD6DJ4LZxxa7Dm0xJ3QdisraQaQ/
m=sb_wiz,aa,abd,async,dvl,foot,ifl,lu,m,mu,sf,xz7cCd?xjs=s1" async
gapi_processed="true"></script>
</body>
</html>
Styles
Computed
Node
Active
Focus
Hover
Visited
Style Attribute {
}
body, html {
font-size: small;
}
www.google.com:62:7400
body {
background: #fff;
color: #222;
}
www.google.com:62:6819
body, td, a, p, .h {
font-family: arial,sans-serif;
}
www.google.com:62:6381
html, body {
height: 100%;
margin: 0;
}
www.google.com:62:6123
body {
User Agent Stylesheet
Filter
Classes