

Silk Test 15.0

モバイル Web アプリ
ケーションのテスト

Micro Focus
575 Anton Blvd., Suite 510
Costa Mesa, CA 92626

Copyright © Micro Focus 2014. All rights reserved. Silk Test は Borland Software Corporation に由来する成果物を含んでいます, Copyright © 2014 Borland Software Corporation (a Micro Focus company).

MICRO FOCUS, Micro Focus ロゴ、及びその他は Micro Focus IP Development Limited またはその米国、英国、その他の国に存在する子会社・関連会社の商標または登録商標です。

その他、記載の各名称は、各所有社の知的所有財産です。

2013-12-02

目次

モバイル Web アプリケーションのテスト	4
モバイル Web テストのクライアント	4
Android 上のモバイル Web アプリケーションのテスト	4
物理 Android デバイス上のモバイル Web アプリケーションのテスト	4
Android エミュレータ上のモバイル Web アプリケーションのテスト	5
USB ドライバをインストールする	5
USB デバッグの有効化	6
Android デバイスまたはエミュレータのプロキシとして Open Agent を手動で設定する	6
Silk Test 用に Android エミュレータを設定する	7
モバイル Web アプリケーションの記録	9
モバイル デバイスの操作	9
モバイル Web アプリケーションのテスト時のトラブルシューティング	9
セキュアな Web アプリケーションをテストするためにルート証明書を手動で追加する	11
モバイル Web アプリケーションのテストにおける制限事項	12

モバイル Web アプリケーションのテスト

Silk Test では、モバイル Web アプリケーションを自動的にテストすることができます。Silk Test を使用した自動テストには、次のメリットがあります。

- モバイル Web アプリケーションのテスト時間を大幅に減少させることができます。
- テストを一旦作成すれば、数多くの異なるデバイスやプラットフォーム上でモバイル Web アプリケーションをテストできます。
- エンタープライズ モバイル Web アプリケーションに要求される信頼性とパフォーマンスを確保できます。
- QA チームのメンバーおよびモバイル Web アプリケーションの開発者の効率を向上できます。
- モバイル Web アプリケーションは、多くのモバイル デバイスとプラットフォームで動作することを要求されるため、アジャイルにフォーカスした開発環境にとって手動テストは十分効率的とは言えない場合があります。

モバイル Web テストのクライアント

以下の Silk Test クライアントは、モバイル Web テストをサポートします。

- Silk Test Workbench
- Silk4J
- Silk4NET

Android 上のモバイル Web アプリケーションのテスト

Silk Test では、Android デバイスまたは Android エミュレータ上のモバイル Web アプリケーションをテストすることができます。

物理 Android デバイス上のモバイル Web アプリケーションのテスト

物理 Android デバイス上のモバイル Web アプリケーションをテストするには、次のタスクを実行します。

1. Silk Test をインストールしたマシンにデバイスを接続します。
2. このマシンで、この Android デバイスをはじめてテストしている場合、適切な Android USB ドライバをマシンにインストールします。
詳細については、「[USB ドライバをインストールする](#)」を参照してください。
3. USB デバッグを Android デバイスで有効化します。
詳細については、「[USB デバッグの有効化](#)」を参照してください。
4. Android デバイスが接続しているマシン上で Open Agent が実行していることを確認します。

モバイル Web アプリケーションをテストする場合、Open Agent は Android デバイスに対するプロキシとして自動的に使用されます。



注: Silk Test がモバイル Web アプリケーションをフックしようとするときに、アクティブな WiFi 接続がない場合、Open Agent を Android デバイスのプロキシとして手動で設定します。詳細については、「[Android デバイスまたはエミュレータのプロキシとして Open Agent を手動で設定する](#)」を参照してください。

5. Android デバイスで **Silk Test Web Tunneler** アプリを開きます。

Open Agent とデバイス間の USB 接続を有効化するため、Silk Test は、Android デバイスに **Silk Test Web Tunneler** アプリをインストールします。

6. HTTPS を使用したセキュアなモバイル Web アプリケーションをテストするために、Silk Test はフック中にルート証明書をデバイスにコピーします。証明書がインストールされていない場合、Silk Test Web Tunneler アプリはルート証明書がインストールされていないことを示すメッセージボックスを表示します。メッセージをクリックして、証明書をインストールします。



注: 証明書がフック中に自動的にインストールされない場合、「セキュアな Web アプリケーションをテストするためにルート証明書を手動で追加する」を参照してください。

7. モバイル Web アプリケーション用の Silk Test プロジェクトを作成します。

8. モバイル Web アプリケーション用のテストを作成します。

9. **モバイルの記録** 機能を使用して、モバイル Web アプリケーションに対するテストを記録します。

10. テストを再生します。

11. テスト結果を分析します。

Android エミュレータ上のモバイル Web アプリケーションのテスト

Android エミュレータ上のモバイル Web アプリケーションをテストするには、次のタスクを実行します。

1. Silk Test のエミュレータ設定を構成します。

詳細については、「*Silk Test* 用に *Android* エミュレータを設定する」を参照してください。

2. Android エミュレータを開始します。

3. モバイル Web アプリケーションをテストするには、Open Agent を Android エミュレータのプロキシとして設定します。



注: モバイルデバイスが接続しているマシン上で Open Agent が実行していることを確認します。

詳細については、「*Android* デバイスまたはエミュレータのプロキシとして *Open Agent* を手動で設定する」を参照してください。

4. HTTPS を使用したセキュアなモバイル Web アプリケーションをテストするために、Silk Test はフック中にルート証明書をデバイスにコピーします。証明書がインストールされていない場合、Silk Test Web Tunneler アプリはルート証明書がインストールされていないことを示すメッセージボックスを表示します。メッセージをクリックして、証明書をインストールします。



注: 証明書がフック中に自動的にインストールされない場合、「セキュアな Web アプリケーションをテストするためにルート証明書を手動で追加する」を参照してください。

5. モバイル Web アプリケーション用の Silk Test プロジェクトを作成します。

6. モバイル Web アプリケーション用のテストを作成します。

7. **モバイルの記録** 機能を使用して、モバイル Web アプリケーションに対するテストを記録します。

8. テストを再生します。

9. テスト結果を分析します。

USB ドライバをインストールする

モバイル Web アプリケーションをテストするために、ローカルマシンに最初に Android デバイスに接続するには、適切な USB ドライバをインストールする必要があります。

デバイスの製造元は、そのデバイスに必要なすべてのドライバをもった EXE を提供している可能性があります。この場合、ローカルマシンにその EXE をインストールするだけです。製造元がこのような EXE を提供していない場合、マシン上にデバイスに対する単一の USB ドライバをインストールできます。

Microsoft Windows 7 上に Android USB ドライバをインストールするには：

1. デバイス用の適切なドライバを探します。
USB ドライバを探してインストールする方法についての詳細は、『<http://developer.android.com/tools/extras/oem-usb.html>』を参照してください。
2. Android デバイスをローカル マシンの USB ポートに接続します。
3. デスクトップ、または **Windows Explorer** から、**コンピュータ** を右クリックし、**管理** を選択します。
4. 左側のペインで、**デバイス マネージャ** を選択します。
5. 右側のペインで、**その他のデバイス** を探して展開します。
6. デバイス名 (*Nexus S* など) を右クリックして、**ドライバ ソフトウェアの更新** を選択します。**ハードウェアの更新ウィザード** が開きます。
7. **コンピュータを参照してドライバ ソフトウェアを検索します** を選択して、**次へ** をクリックします。
8. **参照** をクリックして、USB ドライバ フォルダを探します。
デフォルトでは、Google USB ドライバは、<sdk>%extras%google%usb_driver% にあります。
9. **次へ** をクリックしてドライバをインストールします。

既存の USB ドライバのアップグレード、または他のオペレーティング システムに USB ドライバをインストールする方法については、『<http://developer.android.com/tools/extras/oem-usb.html>』を参照してください。


USB デバッグの有効化

Android Debug Bridge (adb) 上での Android デバイスと通信するために USB デバッグを有効化します。

1. Android デバイスで設定を開きます。
2. **Dev Settings** をタップします。
Dev Settings (開発者向けオプション) がデバイスの設定メニューに含まれていない場合：
 - a) 画面を下にスクロールさせて **端末情報** をタップします。
 - b) 再度画面を下にスクロールさせて **ビルド番号** を 7 回タップします。
3. **開発者向けオプション** ウィンドウで、**USB デバッグ** をオンにします。

Android デバイスまたはエミュレータのプロキシとして Open Agent を手動で設定する

Android デバイスまたは Android エミュレータのプロキシとして Open Agent を設定するには、デバイスまたはエミュレータのテストを行いたいマシン上に Open Agent をインストールし、デバイスまたはエミュレータ上で [USB デバッグ] を有効化します。

 **注:** 現実の Android デバイスをテストする場合、Open Agent は Android デバイスに対するプロキシとして自動的に設定されます。

1. Android デバイスまたは Android エミュレータで設定を開きます。
2. ワイヤレス ネットワーク接続を使用するには、次のように設定します。
 - a) **Wi-Fi 設定** をクリックします。
 - b) アクティブな接続を長押しします。
 - c) **ネットワークを変更** をクリックします。
 - d) **詳細オプションを表示** をオンにします。
 - e) **プロキシ設定** をクリックします。
 - f) **手動** を選択します。
3. 3G 接続を使用するには、次のように設定します。

- a) **その他** をクリックします。
 - b) **モバイル ネットワーク** をクリックします。
 - c) **アクセス ポイント名** をクリックします。
 - d) アクティブなアクセス ポイントを選択します。
 - e) **プロキシ** をクリックします。
4. Open Agent がインストールされているマシンの IP アドレスを **プロキシ** または **プロキシ ホスト名** フィールドに入力します。
 5. 3G 接続を設定している場合は、**OK** をクリックします。Open Agent がインストールされているマシンが Android デバイスまたは Android エミュレータのプロキシとして一覧表示されます。
 6. **ポート** をクリックします。
 7. **ポート** フィールドに Open Agent のポート番号を入力します。デフォルトのポート番号はランダムです。エミュレータ上の AUT をテストしたり、Android デバイス上のワイヤレス ネットワーク接続を構成したりするには、構成設定 `ext.http.proxy.port` (AppData¥Roaming¥Silk¥SilkTest¥conf ¥silkproxy.properties ファイル) を使用して、不変のポート番号を設定します。たとえば、ポート番号を 9999 に設定するには、`ext.http.proxy.port=9999` を設定します。そして、ポート番号を **ポート** フィールドに入力します。
 8. **OK** をクリックします。

以上で Open Agent が Android デバイスまたは Android エミュレータのプロキシとして設定されました。Android デバイスまたは Android エミュレータのプロキシの構成についての詳細は、デバイスまたはエミュレータのドキュメントを参照してください。



注: Open Agent が実行している限り、Open Agent をプロキシとして使用してモバイル デバイス上のインターネット接続を使用できます。Open Agent が実行していない場合、接続は機能しないため、モバイル デバイスからインターネットに接続するために他の接続を使用する必要があります。デバイスまたはエミュレータが実行している間にワイヤレス ネットワーク接続が削除されると、Open Agent との接続はデバイスまたはエミュレータをシャットダウンするまで開放されません。

Silk Test 用に Android エミュレータを設定する

Silk Test を使用して Android エミュレータ上でモバイル Web アプリケーションをテストする場合、テスト用にエミュレータを設定する必要があります。

1. Android エミュレータをインストールします。
Android エミュレータのインストールと設定についての詳細は、「[Get the Android SDK](#)」を参照してください。
2. Eclipse 上で、**ウィンドウ > Android SDK Manager** をクリックして、**Android SDK Manager** を起動します。
3. エミュレータを使ってテストするすべての Android のバージョンに対して、バージョン ノードを展開し、**Intel x86 Atom System Image** の隣のチェック ボックスをオンにします。
4. **Install** をクリックして、選択したパッケージをインストールします。
5. **Extras** ノードを展開し、**Intel x86 Emulator Accelerator (HAXM)** の隣にあるチェック ボックスをオンにします。
6. **Install** をクリックして、選択したパッケージをインストールします。
7. *Intel Corporation license agreement* を確認します。条項に同意できる場合は、**Accept** を選択して、**Install** をクリックします。**Android SDK Manager** は、メイン SDK ディレクトリの下 extras ディレクトリにインストーラをダウンロードします。**Android SDK Manager** は、Installed というステータスを表示しますが、これは Intel HAXM 実行可能ファイルがダウンロードされたことを意味します。extras ディレクトリにあるインストーラを実行してインストールする必要があります。
8. extras ディレクトリにあるインストーラを実行し、プラットフォームごとのインストール手順に従います。
9. Eclipse 上で、**ウィンドウ > Android Virtual Device Manager** をクリックし、新しい Android Virtual Device (AVD) を追加します。

10 **Android Virtual Devices** タブを選択します。

11 **New** をクリックします。

12 要件に従って仮想デバイスを設定します。

13 エミュレータが使用する RAM サイズを対象のマシンで管理可能な量に設定します。

たとえば、エミュレータの RAM サイズを 512 に設定します。

14 SD カードのサイズを設定します。



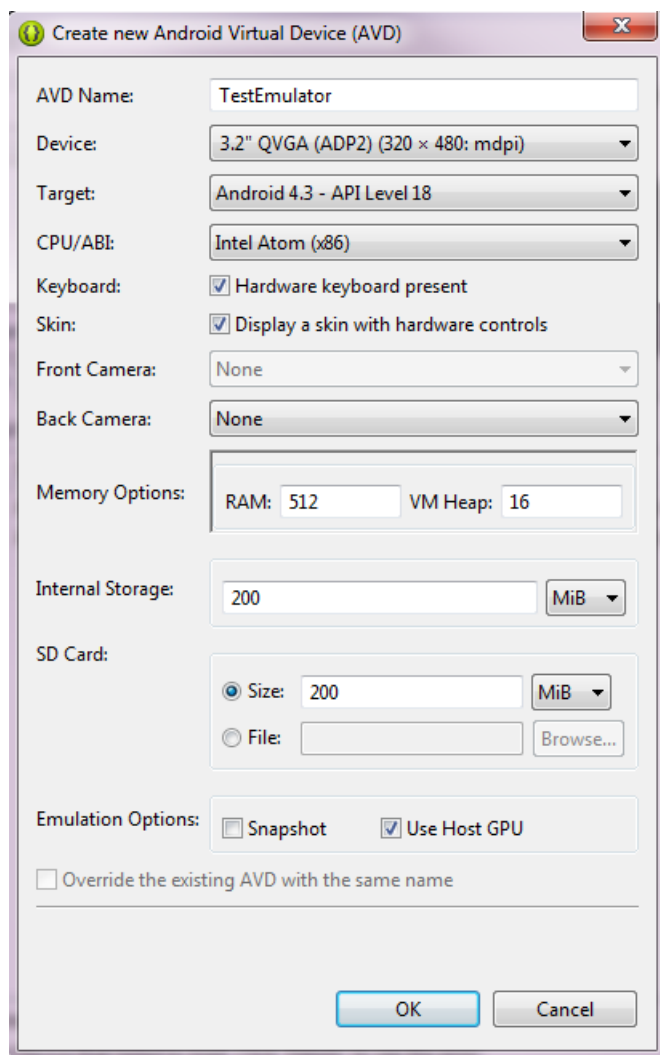
注: SD カードのサイズを設定しないと場合は、内部ストレージ (Internal Storage) の値を 50 MB 以上に設定する必要があります。そうしないと、証明書ファイルをエミュレータにコピーできません。

15 エミュレータのトランザクション速度を向上させるには、**CPU/ABI** フィールドの **Intel Atom (x86)** CPU を選択します。

16 省略可能 : エミュレータのトランザクション速度を向上させるには、エミュレーション オプション (Emulation Options) の **Use Host GPU** をオンにすることもできます。




注: Android では、**Use Host GPU** オプション使用時に問題があります。詳細については、<https://code.google.com/p/android/issues/detail?id=58724> を参照してください。




17 **OK** をクリックします。

モバイル Web アプリケーションの記録

 **注:** 一部の低レベル メソッドとクラスは、モバイル Web アプリケーションではサポートされません。モバイル Web アプリケーションに対して記録したテストを正しく再生できるようにするためには、モバイル Web アプリケーションに対して記録を行う前に、Silk Test のブラウザ オプションで、**ネイティブなユーザー入力を記録する** オプションをオフにします。詳細については、モバイル Web アプリケーションのテストの制限事項 を参照してください。

Silk Test とモバイル デバイスまたはエミュレータとの間の接続が一旦確立すると、デバイス上のモバイル ブラウザで実行する操作を記録してテストを作成できます。モバイル Web アプリケーションを記録するには、Silk Test は **モバイルの記録** 機能を使用します。この機能は、標準アプリケーションや Web アプリケーションに対して使われる記録よりもさらに多くの機能を提供します。

モバイルの記録 機能は、テストするモバイル デバイスまたはエミュレータの画面を表示します。

 **注:** モバイル デバイスがマシンに接続してなかったり、エミュレータが開始されていなかった場合は、**モバイルの記録** 機能はエラー メッセージを表示します。モバイル デバイスをマシンに接続するか、エミュレータを起動してから、**モバイルの記録** ウィンドウの **更新** をクリックします。

モバイルの記録 機能で操作を実行すると、モバイル デバイス上でも同じ操作が実行されます。

画面上のコントロールを操作すると、**モバイルの記録** 機能はデフォルトの操作を事前に選択します。コントロールに対して有効なすべての操作がリストで表示されるので、実行したい操作を選択するか、単に **OK** をクリックして事前に選択された操作を受け入れます。選択した操作のパラメータの値をパラメータ フィールドに入力することができます。Silk Test は自動的にパラメータを検証します。

コントロールを直接扱うことができない場合 (たとえば、コントロールが他のコントロールで隠されている場合)、**モバイルの記録** ウィンドウの **階層ビューの切り替え** をクリックして、コントロール階層ツリーからコントロールを選択できます。

記録を一時停止すると、画面上での操作は記録されないため、デバイスを記録を続けたい状態に変更することができます。

記録を停止すると、記録した操作でスクリプトが生成されるため、続いてテストの再生を行うことができます。

モバイル デバイスの操作

モバイル デバイスを操作したり、テスト対象アプリケーションでスワイプのような操作を実行するには、次の手順を実行します。

1. **モバイルの記録** ウィンドウで、**モバイル デバイス操作の表示** をクリックします。モバイル デバイスに対して実行できるすべての操作がリストされます。
2. リストからリストから実行したい操作を選択します。
3. テストの記録を続行します。

モバイル Web アプリケーションのテスト時のトラブルシューティング

[アプリケーションの選択] ダイアログ ボックスにモバイル ブラウザーが表示されない理由

Silk Test が、次の何れかの理由でモバイル デバイスまたはエミュレータを認識していない可能性があります。

理由	解決策
モバイル デバイスがローカル マシンに接続されていない。	モバイル デバイスをローカル マシンに接続します。
エミュレータが実行されていない。	エミュレータを開始します。
Android Debug Bridge (adb) がモバイル デバイスを認識しない。	モバイル デバイスが adb によって認識されているかどうか確認するには： <ol style="list-style-type: none"> 1. C:\Program Files (x86)\Silk\SilkTest\ngagent\plugins\com.microfocus.silktest.adb_15.0.0.6733\bin に移動します。 2. Shift を押しながら、ファイル エクスプローラ ウィンドウで右クリックします。 3. コマンド ウィンドウをここで開く を選択します。 4. コマンド ウィンドウで、adb devices を入力して、アタッチしたすべてのデバイスのリストを得ます。 5. デバイスがリストされない場合、USB デバッグがデバイスで有効化されていることを確認します。
デバイスのオペレーティング システムのバージョンを Silk Test がサポートしていない。	サポートするモバイル オペレーティング システムのバージョンについては、 リリース ノート を参照してください。
デバイスの USB ドライバがローカル マシンにインストールされていない。	デバイスの USB ドライバをローカル マシンにインストールしてください。詳細については、「 USB ドライバをインストールする 」を参照してください。
USB デバッグがデバイスで有効化されていない。	USDB デバッグをデバイスで有効化してください。詳細については、「 USB デバッグの有効化 」を参照してください。

モバイル デバイスまたはエミュレータがインターネットに接続できない理由

Open Agent をモバイル デバイスまたはエミュレータのすべてのネットワーク接続のプロキシとして構成し、現在どんなテストも記録または再生していない場合、モバイル デバイスまたはエミュレータはインターネットに接続できません。物理モバイル デバイスの場合、**Silk Test Web Tunneler** アプリケーションで接続ステータスを確認できます。

モバイル デバイスが接続され、Open Agent が実行中にも関わらず、モバイル デバイスがまだインターネットに接続できない場合には、プロキシ設定が正しいかどうか確認してください。

Open Agent が実行していない状態でインターネットに接続できるようにするためには、プロキシを一時的に無効化してください。

URL に移動せずに Silk Test が Chrome for Android で URL を検索する理由

アドレス バーに入力された URL を、Chrome for Android が検索として解釈する場合があります。回避策として、URL に移動するコマンドをスクリプトに手動で追加できます。

Android 4.3 の Android エミュレータで記録できない理由

Android バージョン 4.3 の Android エミュレータで記録するには、エミュレータ設定 (Emulator Settings) の **Use Host GPU** チェック ボックスをオフにします。

プロキシを構成したときにモバイル アプリケーションが機能しなくなる理由

WiFi 接続に対して設定できるグローバル プロキシを使用しないモバイル アプリケーションがあります。ブラウザや Gmail のようなアプリケーションは、プロキシ設定を使用しますが、多くのほかのモバイル

アプリケーションは、プロキシ設定を無視するため、プロキシが設定されている間、インターネットに接続できません。

adb サーバーが正しく起動しない場合にすべきこと

Android Debug Bridge (adb) サーバーが開始するとき、ローカル TCP ポート 5037 にバインドし、adb クライアントから送信されてくるコマンドをリッスンします。すべての adb クライアントは、ポート 5037 を使用して、adb サーバーと通信します。adb サーバーは、5555 から 5585 の範囲 (エミュレータやデバイスで使用される範囲) で奇数のポートをスキャンしてエミュレータやデバイス インスタンスを探します。adb はこれらのポートの変更を許しません。adb 開始中に問題が発生した場合、これらの範囲のポートの 1 つが、他のプログラムによって既に使用されているかどうか確認します。

詳細については、<http://developer.android.com/tools/help/adb.html> を参照してください。

エラーが発生する理由メモリの割り当てに失敗しました : :8?

エミュレータを開始しているときに、システムが十分なメモリを割り当てることができない場合に、このエラーが表示されます。以下を行ってみてください。

1. エミュレータのメモリ オプションの RAM サイズを下げる
2. Intel HAXM の RAM サイズを下げる RAM サイズを下げるには、IntelHaxm.exe を再度実行して、**Change** を選択します。
3. **タスク マネージャ** を開き、十分なフリー メモリが利用可能かどうかを確認します。不足している場合、プログラムを閉じてメモリを開放してください。

セキュア Web サイトに対して動作しない理由

物理モバイル デバイス上でセキュア Web サイト (HTTPS) をテストできない場合、以下を行ってみてください。

1. **Silk Test Web Tunneler** アプリケーションをモバイル デバイスで開き、セキュア Web サイトに対する証明書がインストールされているかどうか確認します。証明書がインストールされていない場合、警告が表示されます。
2. 警告をクリックして、**OK** を選択し、証明書をインストールします。
3. 証明書がデバイス上に見つからない場合、ファイル root.crt が sdcard/silk/certs/ の下に存在するかどうか確認します。
4. ファイル root.crt が存在しない場合、**ファイル エクスプローラ** を使用して、ファイルを手動でコピーします。モバイル デバイス上に書き込み権を持たない場合、証明書が見つからない可能性があります。
5. 証明書をデバイスにコピーした後に、**Silk Test Web Tunneler** アプリケーションを使うか、ファイル システムで証明書をクリックして、証明書をインストールできます。

エミュレータ上でセキュア Web サイト (HTTPS) をテストできない場合、Web サイトのルート証明書を手動で追加します。詳細については、「セキュアな Web アプリケーションをテストするためにルート証明書を手動で追加する」を参照してください。

セキュアな Web アプリケーションをテストするためにルート証明書を手動で追加する




注: このトピックで述べるステップを実行するには、Open Agent を Android デバイスまたは Android エミュレータのプロキシとして設定する必要があります。

Android デバイスや Android エミュレータ上で HTTPS を使ったモバイル Web アプリケーションをテストする場合、特定のサイトを開くリクエストごとに Open Agent がインストールされているマシンでこのサイトに対する証明書が自動的に生成されます。この新しい証明書は元の証明書と同じドメインに対して発行され、元の証明書を置き換えることで SSL 接続によるテストを可能にします。

生成される最初の証明書は、モバイル Web アプリケーションに対するルート証明書です。

Silk Test を使用してアプリケーションをテストできるようにするためには、このルート証明書を Android デバイスまたは Android エミュレータにインストールしなければなりません。デフォルトでは、ルート証

明書はフック中にデバイスにコピーされます。ただし、ルート証明書が自動的にインストールされない場合、テストしたいモバイル Web アプリケーションそれぞれに対して一度、ルート証明書をインストールする必要があります。

1. テストしたいモバイル Web アプリケーションを開きます。初めてモバイル Web アプリケーションを開くときに、Open Agent はアプリケーション用の修正したルート証明書を生成します。
2. Open Agent がインストールされているマシン上で、ルート証明書が生成されたフォルダに移動します。
デフォルトでは、フォルダは %Appdata%\%Silk%\SilkTest%\certs\%authority になります。
3. ルート証明書ファイル root.crt をコピーします。
4. Android デバイスのストレージのルート フォルダにルート証明書ファイルを貼り付けます。
Android エミュレータ上でテストする場合、Open Agent は、エミュレータのルート ディレクトリに証明書を自動的にコピーします。
 **注:** エミュレータへの証明書のコピーを Open Agent で有効にするには、エミュレータの設定で SD カードのサイズを設定します。
5. 物理 Android デバイス上でテストする場合、ストレージから Android デバイスに証明書をインストールします。
ストレージから証明書をインストールする方法に関する詳細については、Android デバイスまたは Android エミュレータのドキュメントを参照してください。
6. Android エミュレータ上でテストする場合：
 - a) エミュレータ上で **設定 > セキュリティ > SD カードからインストール** に移動します。
 - b) **OK** をクリックして証明書をインストールします。
 - c) 省略可能：**設定 > セキュリティ > 信頼できる認証情報 > ユーザー** に移動して、認証情報がエミュレータにインストールされていることを確認します。

モバイル Web アプリケーションのテストにおける制限事項

モバイル ブラウザ上でのテストの再生とロケータの記録のサポートは、サポートされている他のブラウザやネイティブ モバイル アプリケーションほど完全なものではありません。以下のリストに、モバイル ブラウザ上でのテストの再生とロケータの記録の既知の制限事項をリストします。

- モバイル Web アプリケーション上のオブジェクトをクリックするには、DomElement クラスの DomClick メソッドを使用します。モバイル Web アプリケーション上のオブジェクトをクリックするために、IMobileClickable インターフェイスの Click メソッドを使用することはできません。
- 次のクラス、インターフェイス、メソッド、プロパティは、モバイル Web アプリケーションでは現時点ではサポートされません。
 - BrowserApplication クラス
 - CloseOtherTabs メソッド
 - CloseTab メソッド
 - ExistsTab メソッド
 - GetActiveTab メソッド
 - GetSelectedTab メソッド
 - GetSelectedTabIndex メソッド
 - GetSelectedTabName メソッド
 - GetTabCount メソッド
 - OpenTab メソッド
 - SelectTab メソッド
 - DomElement クラス
 - DomDoubleClick メソッド

- DomMouseMove メソッド
- GetDomAttributeList メソッド
- DomForm クラスこのクラスのすべてのメソッドとプロパティは、モバイル Web アプリケーションではサポートされません。
- DomRadioButton クラス
 - RadioListItemCount プロパティ
 - RadioListItems プロパティ
 - RadioListSelectedIndex プロパティ
 - RadioListSelectedItem プロパティ
- DomTable クラスこのクラスのすべてのメソッドとプロパティは、モバイル Web アプリケーションではサポートされません。
- DomTableRow クラスこのクラスのすべてのメソッドとプロパティは、モバイル Web アプリケーションではサポートされません。
- IClickable インターフェイス
 - Click メソッド
 - DoubleClick メソッド
 - PressMouse メソッド
 - ReleaseMouse メソッド
- IKeyable インターフェイスこのインターフェイスのすべてのメソッドとプロパティは、モバイル Web アプリケーションではサポートされません。
- XPath 論理演算子は、すべての属性に対してサポートされません。たとえば、textContent 属性と innerText 属性に対して論理演算子はサポートされません。

索引

A

Android

- USB デバッグの有効化 6
- USB ドライバをインストールする 5
- エミュレータ上でのテスト 5
- エミュレータを設定する 7
- 前提条件 11
- テスト 4
- トラブルシューティング 9
- 物理デバイス上でのテスト 4
- プロキシの設定 6

U

- USB ドライバをインストールする
Android 5

き

- 記録する
モバイル アプリケーション 9

て

- デバイスが接続されていません
モバイル 9

と

- トラブルシューティング
モバイル 9

ふ

- プロキシ サーバー

Android の設定 6

も

- モバイル
 - トラブルシューティング 9
- モバイル Web アプリケーション
 - 制限事項 12
 - テスト 4
- モバイル Web テスト
 - Android 4
 - Android エミュレータ 5
 - 概要 4
 - クライアント 4
 - 物理 Android デバイス 4
- モバイル アプリケーション
 - 記録する 9
- モバイルデバイス
 - 操作する 9
 - に対して操作を実行する 9
- モバイルの記録
 - について 9
- モバイル ブラウザ
 - 制限事項 12

る

- ルート証明書
 - 生成する 11
 - 追加する 11
- ルート証明書を追加する
Android 11