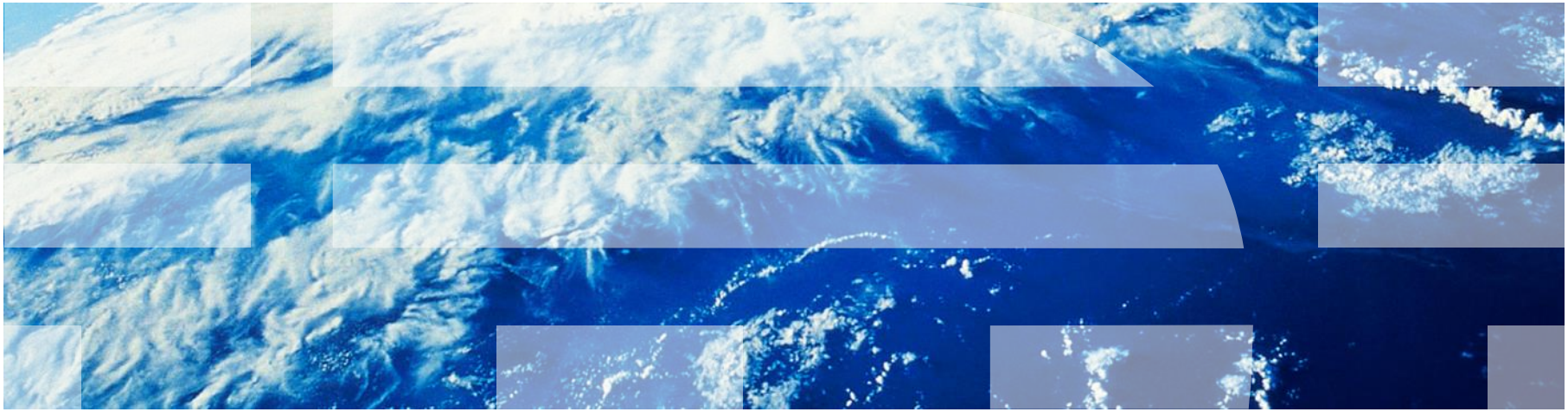
IBM Worklight Foundation V6.2.0 入門

認証の概念



2014 年 6 月 20 日

商標

- IBM、IBM ロゴ、ibm.com、WebSphere および Worklight は、世界の多くの国で登録された International Business Machines Corporation の商標です。他の製品名およびサービス名等は、それぞれ IBM または各社の商標である場合があります。現時点での IBM の商標リストについては、http://www.ibm.com/legal/copytrade.shtml をご覧ください。
- Java およびすべての Java 関連の商標およびロゴは Oracle やその関連会社の米国およびその他の国における商標または登録商標です。
- この資料は、事前に IBM の書面による許可を得ずにその一部または全部を複製することは禁じられています。

IBM® について

- http://www.ibm.com/ibm/us/en/ を参照してください。

アジェンダ

- 認証の概念とエンティティー
- レルム、オーセンティケーター、ログイン・モジュールの定義
- セキュリティー・テストの定義
- アプリケーションの保護
- アダプターの保護
- 静的リソースの保護
- 次のステップ
- 確認テスト

認証の概念とエンティティー (1/10)

- IBM Worklight Foundation ® のエンティティー (例えば、アプリケーション、アダプター・プロシージャー、静的リソースなど) を、無許可アクセスから保護できます。
- エンティティー保護ルールは**セキュリティー・テスト**によって定義され、その中には 1 つ以上の *認証レルム* が含まれます。
- **認証レルム** は、ユーザーの認証に使用するプロセスを定義したものです。
- 各認証レルムは、**オーセンティケーター**と**ログイン・モジュール**で構成されます。これらはサーバー・サイド・コンポーネントです。
- 同じ認証レルムを使用して、複数のリソースを保護できます。
- それぞれの認証レルムは、クライアント・サイドに**チャレンジ・ハンドラー**・コンポーネントを必要とします。
- すべての認証コンポーネントの詳細な定義は、これ以降のスライドで紹介します。

認証の概念とエンティティー (2/10)
オーセンティケーター

- オーセンティケーターは、クライアント・アプリケーションから資格情報を収集する役割を持つサーバー・サイドのエンティティーです。
- オーセンティケーターは、HTTP 要求オブジェクトからアクセス可能なあらゆるタイプの情報 (Cookie、ヘッダー、本文、またはその他の任意のプロパティー) を収集できます。
- Worklight Server には、以下のような一連のオーセンティケーターが事前定義されています。
  - **フォーム・ベース**のオーセンティケーター: HTML ログイン・フォームの形式でチャレンジを返し、Web 環境やモバイル・アプリケーションで使用できるようにします。
  - **アダプター・ベース**のオーセンティケーター: Worklight アダプター・プロシージャーを使用して、クライアント・アプリケーションから資格情報を収集し、それを検証します。
  - **ヘッダー・ベース**のオーセンティケーター: 対話式による資格情報の収集は必要とせず、代わりに特定の HTTP ヘッダーを検査します。
- 事前定義されたオーセンティケーターに加え、Java™ コードを使用して独自の**カスタム**・オーセンティケーターを作成することもできます。

認証の概念とエンティティー (3/10)
ログイン・モジュール

- ログイン・モジュールは、ユーザー資格情報を検証し、*ユーザー ID* オブジェクトを作成する役割を持つサーバー・サイドのエンティティーです。ユーザー ID オブジェクトには、セッションの残りの期間中、ユーザー・プロパティーが保持されます。
- 例えば、以下のいずれかの方法で資格情報を検証できます。
  - Web サービスを使用する。
  - データベース内のユーザー・テーブルでユーザーを検索する。
  - WebSphere® LTPA トークンを使用する。
- 企業のニーズに応じて、カスタム・ユーザー・プロパティーを追加できます。
- ログイン・モジュールは、認証されたセッションが (ログアウトまたはタイムアウトによって) 終了するときに、ユーザー ID オブジェクトを破棄します。
- 監査用に、ログイン試行を自動的に記録するようにログイン・モジュールを構成できます。
- 事前定義されたログイン・モジュールに加え、Java コードを使用して独自のカスタム・ログイン・モジュールを作成することもできます。

認証の概念とエンティティー (4/10)
認証レルム

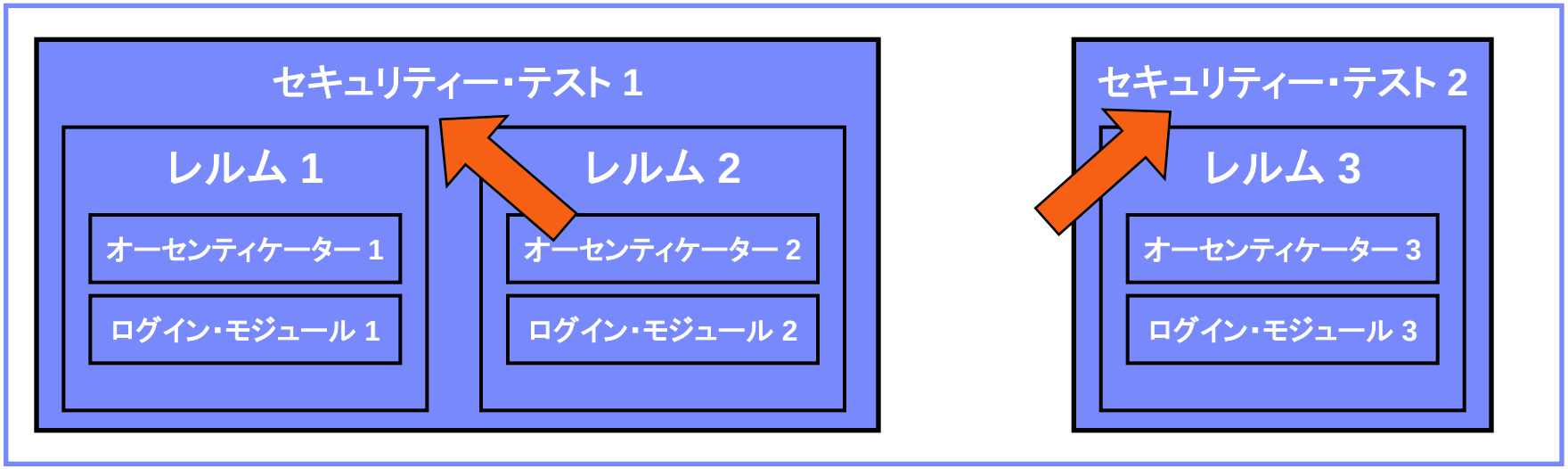
- 認証レルムは、1 つのオーセンティケーターと 1 つのログイン・モジュールを組み合わせたものです。
- 各認証レルムの認証フローは、以下のように定義されます。
  - 認証プロセスがトリガーされた後に何が起きるか
  - クライアント・アプリケーションに送信すべきチャレンジの形式
  - 収集すべき資格情報
  - 資格情報を収集する方法とその時期
  - 資格情報をサーバーに送信する方法
  - サーバーが資格情報を検証する方法
  - 資格情報の検証結果
  - ユーザー ID オブジェクトのプロパティー
- Worklight には、セキュリティー機能 (リモート・アプリケーションの無効化、アプリケーション認証性など) 用にいくつかの認証レルムが事前定義されています。
- サーバー認証構成で定義された認証レルムごとに、対応するチャレンジ・ハンドラーがクライアント・アプリケーション内に必要です。

認証の概念とエンティティー (5/10)
セキュリティー・テスト

- セキュリティー・テストは、認証レルムの順序付きセットであり、アダプター・プロシージャー、アプリケーション、静的 URL などのリソースを保護するために使用します。
- セキュリティー・テストでは、ユーザーが保護リソースへのアクセス権限を得るために認証しなければならない対象レルムが定義されます。
- 開発者は、認証の実行順序を定義できます。例えば、レルム 1 の認証が成功した後にのみ、レルム 2 で認証を要求するように定義できます。
- IBM Worklight Foundation フレームワークには、モバイル環境および Web 環境用にデフォルトのセキュリティー・テストの定義が用意されており、さらにカスタム・セキュリティー・テストを作成する機能も用意されています。
  - 以降のスライドで詳しく説明します。

認証の概念とエンティティー (6/10)

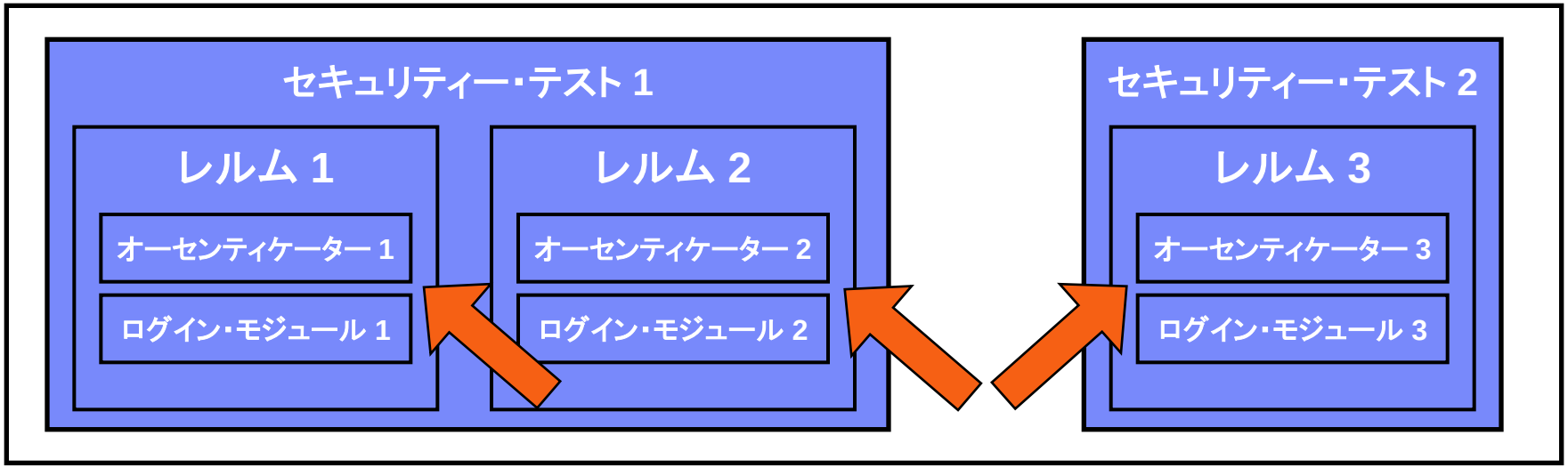
- セキュリティー構成の例



- リソース (例えば、アプリケーションやアダプター・プロシージャーなど) は、2 つのセキュリティー・テストのいずれかで保護できます。
- セキュリティー・テスト 1 を使用する場合、ユーザーはレルム 1 とレルム 2 の両方で認証を行う必要があります。各レルムには独自のルール・セットがあります。
- セキュリティー・テスト 2 を使用する場合、ユーザーはレルム 3 のみで認証を行う必要があります。

認証の概念とエンティティー (7/10)

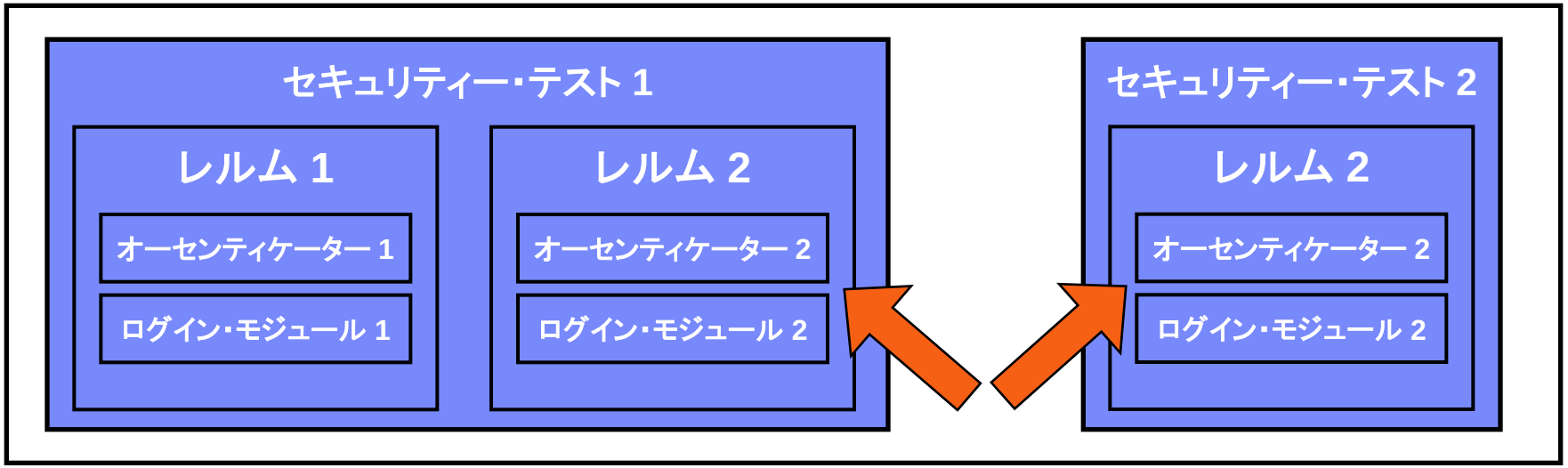
- セキュリティー構成の例



- 各レルムでは、オーセンティケーターとログイン・モジュールからなる独自のセットが定義されます。つまり、各レルムには、資格情報を収集してそれを検証するための独自のルールがあります。

認証の概念とエンティティー (8/10)

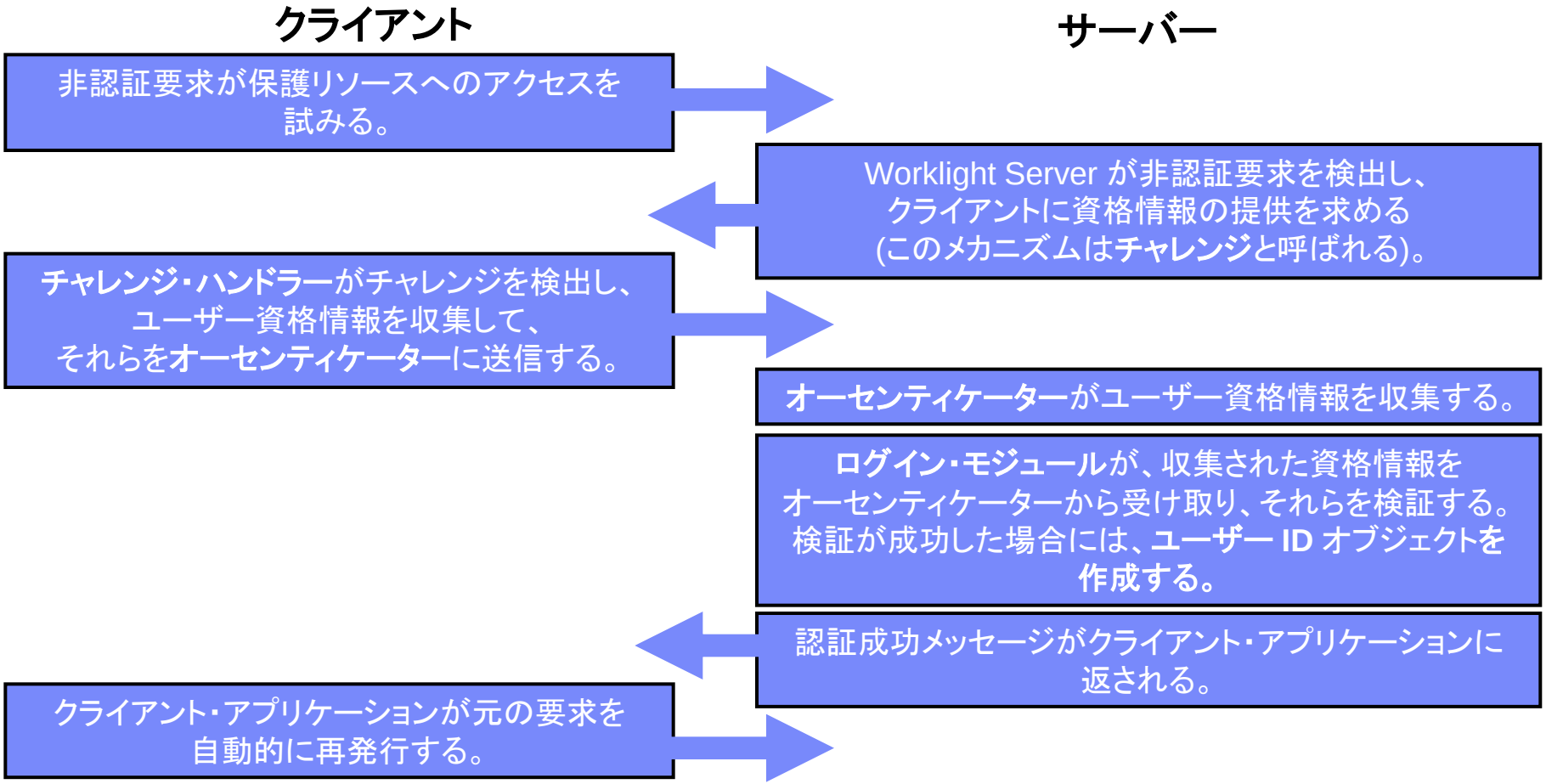
- セキュリティー構成の例



- レルム、オーセンティケーター、ログイン・モジュールは再使用できます。
- 上記の更新された構成では、レルム 2 が再使用されています。
- セキュリティー・テスト 1 でリソースを保護する場合は、レルム 1 とレルム 2 の両方で認証を行う必要があります。
- セキュリティー・テスト 2 でリソースを保護する場合は、レルム 2 のみで認証を行う必要があります。

認証の概念とエンティティー (9/10)

- 保護エンティティーに対して要求が出されると、IBM Worklight はセッションが認証済みかどうかを検査します。認証済みでない場合は、IBM Worklight によってユーザー ID の検証プロセスが自動的にトリガーされます。



認証の概念とエンティティー (10/10)
チャレンジ・ハンドラー

- チャレンジ・ハンドラーは、認証プロセスを制御するクライアント・サイドのエンティティーです。この使用目的は、サーバー応答で認証チャレンジを検出し、それらを処理することです。
- チャレンジ・ハンドラー・インスタンスは、アプリケーションが認証を行う必要のあるレルムごとに、個別に作成する必要があります。
- チャレンジ・ハンドラーを使用すると、Worklight 関連の認証チャレンジと外部認証チャレンジ (認証プロキシーやゲートウェイなど) の両方を検出して処理できます。
- チャレンジ・ハンドラーは、サーバーから返された認証チャレンジを検出した後で、必要な資格情報を収集し、それらをサーバーに返送します。
- 認証フローが完了したら、チャレンジ・ハンドラーは認証の成功または失敗に関する通知を Worklight フレームワークに返送できます。
- チャレンジ・ハンドラーはカスタマイズ可能ですが、いくつかのメソッドが事前設定された状態で作成されます。それらのメソッドを使用すると、Worklight Server の組み込みユーザー認証タイプに資格情報を送信できます。

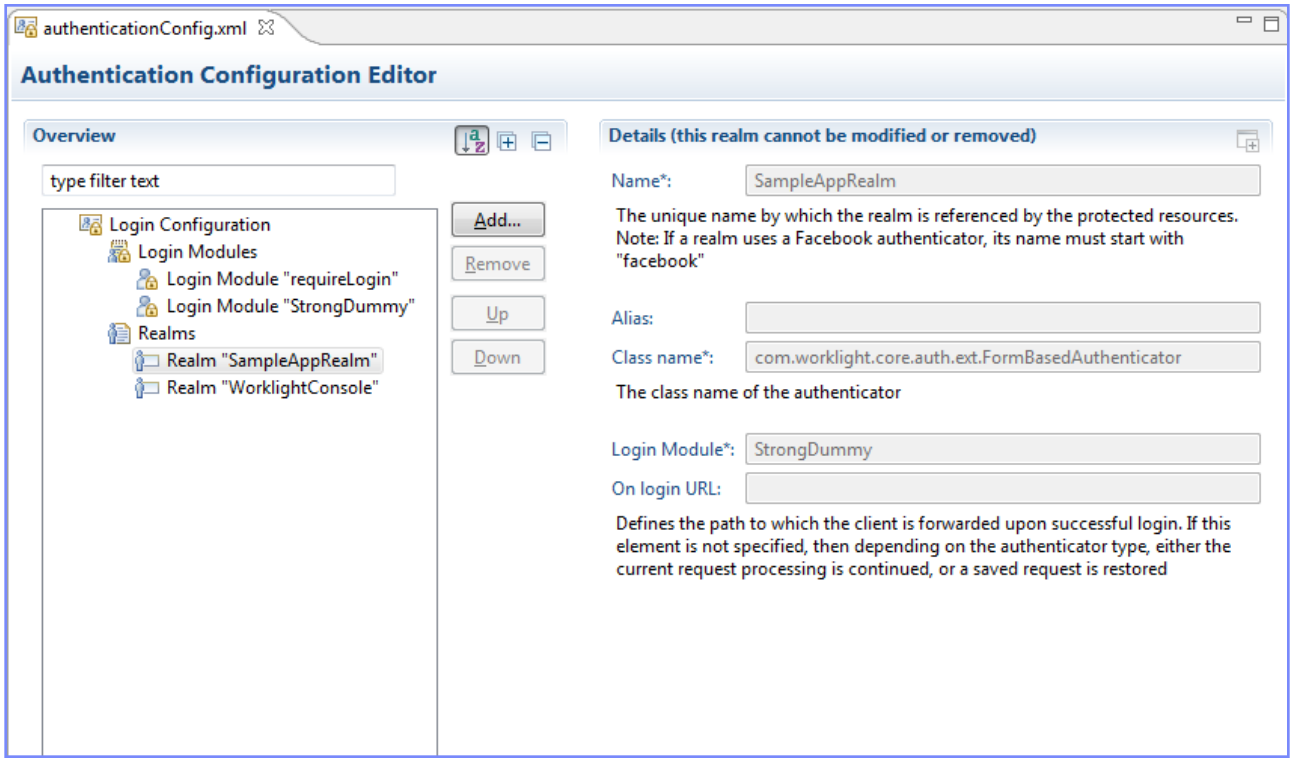
チャレンジ・ハンドラーを作成して、カスタマイズされた認証フローを定義します。認証フローと関係のない変更をユーザー・インターフェースに対して行うコードは、チャレンジ・ハンドラーに追加しないでください。

アジェンダ

- 認証の概念とエンティティー
- レルム、オーセンティケーター、ログイン・モジュールの定義
- セキュリティー・テストの定義
- アプリケーションの保護
- アダプターの保護
- 静的リソースの保護
- 次のステップ
- 確認テスト

レルム、オーセンティケーター、ログイン・モジュールの定義 (1/3)

- 認証設定は、**server¥conf¥authenticationConfig.xml** ファイル内の Worklight プロジェクトで構成されます。
- 認証構成エディターを使用して、これらの設定を変更できます。



レルム、オーセンティケーター、ログイン・モジュールの定義 (2/3)

- 認証設定は、**server¥conf¥authenticationConfig.xml** ファイル内の Worklight プロジェクトで構成されます。
- 認証構成エディターを使用して、これらの設定を変更できます。

```
<realms>
    <realm loginModule="StrongDummy" name="SampleAppRealm">
        <className>com.worklight.core.auth.ext.FormBasedAuthenticator</className>
    </realm>
    <realm loginModule="requireLogin" name="WorklightConsole">
        <className>com.worklight.core.auth.ext.FormBasedAuthenticator</className>
        <onLoginUrl>/console</onLoginUrl>
    </realm>
</realms>

<loginModules>
    <loginModule name="StrongDummy">
        <className>com.worklight.core.auth.ext.NonVali
    </loginModule>

    <loginModule name="requireLogin">
        <className>com.worklight.core.auth.ext.SingleI
    </loginModule>
</loginModules>
```

各レルムには、名前、**loginModule** の指定、オーセンティケーター実装の **className**、**オプション・パラメーター**があります。

レルム、オーセンティケーター、ログイン・モジュールの定義 (3/3)

- 認証設定は、**server¥conf¥authenticationConfig.xml** ファイル内の Worklight プロジェクトで構成されます。
- 認証構成エディターを使用して、これらの設定を変更できます。

```
<realms>
    <realm loginModule="StrongDummy" name="SampleAppRea
        <className>com.worklight.core.auth.ext.FormBase
    </realm>
    <realm loginModule="requireLogin" name="WorklightCo
        <className>com.worklight.core.auth.ext.FormBase
        <onLoginUrl>/console</onLoginUrl>
    </realm>
</realms>

<loginModules>
    <loginModule name="StrongDummy">
        <className>com.worklight.core.auth.ext.NonValidatingLoginModule</className>
    </loginModule>

    <loginModule name="requireLogin">
        <className>com.worklight.core.auth.ext.SingleIdentityLoginModule</className>
    </loginModule>
</loginModules>
```

各**ログイン・モジュール**には、名前、実装の **className**、および**オプション・パラメーター**があります。

アジェンダ

- 認証の概念とエンティティー
- レルム、オーセンティケーター、ログイン・モジュールの定義
- セキュリティー・テストの定義
- アプリケーションの保護
- アダプターの保護
- 静的リソースの保護
- 次のステップ
- 確認テスト

セキュリティー・テストの定義 (1/5)

- IBM Worklight Foundation では、1 つのセキュリティー・テストに対して複数のレルムを設定できます。
- セキュリティー・テストのセットアップの一部として、「**ユーザー・レルム**」と見なされるレルムと、「**デバイス・レルム**」と見なされるレルムを設定する必要があります。
- **ユーザー・レルム**として定義されているレルムから得たすべての ID は、IBM Worklight Foundation によって、**ユーザー ID** を必要とするフィーチャー (例えば、プッシュ通知やアプリケーション使用レポートなど) のユーザー ID として使用されます。
- **デバイス・レルム**として定義されているレルムから得たすべての ID は、IBM Worklight Foundation によって、**デバイス ID** を必要とするフィーチャー (例えば、デバイス・プロビジョニングやプッシュ通知、SMS 通知など) のデバイス ID として使用されます。

セキュリティー・テストの定義 (2/5)

- 認証レルムをセットアップした後で、アプリケーション、アダプター・プロシージャー、および静的リソースの保護に使用するセキュリティー・テストを定義する必要があります。
- **authenticationConfig.xml** ファイルに、以下の 3 タイプのセキュリティー・テストを定義できます。
  - **webSecurityTest** – デフォルトの Web セキュリティー関連レルムを有効にするテスト。
  - **mobileSecurityTest** – デフォルトのモバイル・セキュリティー関連レルムを有効にするテスト。
  - **customSecurityTest** – カスタム・セキュリティー・テスト。デフォルトのレルムは含まれません。

セキュリティー・テストの定義 (3/5)
webSecurityTest

- **webSecurityTest** を使用して Web アプリケーションを保護します。
- デフォルトで、**webSecurityTest** には XSRF 攻撃に対する保護が含まれます。この保護について詳しくは、IBM Worklight Foundation のユーザー文書を参照してください。
- 各 **webSecurityTest** には、1 つの **<testUser>** エレメントとレルム定義を含める必要があります。
- このレルムは、**ユーザー・レルム**と見なされます。

```
<webSecurityTest name="SampleWebSecurityTest">
    <testUser realm="SampleRealm"/>
</webSecurityTest>
```

セキュリティー・テストの定義 (4/5)
mobileSecurityTest

- **mobileSecurityTest** を使用してモバイル・アプリケーションを保護します。
- デフォルトで、**mobileSecurityTest** には以下の機能が含まれます。
  - XSRF 攻撃に対する保護
  - アプリケーション認証性テスト。詳しくは、ユーザー文書を参照してください。
  - モバイル・アプリケーションを Worklight Console からリモート操作で使用不可にする機能。
- 各 **mobileSecurityTest** には、1 つの **<testUser>** エレメントとレルム定義を含める必要があります。
- このレルムは、**ユーザー・レルム**と見なされます。

```
<mobileSecurityTest name="SampleMobileSecurityTest">
    <testUser realm="SampleRealm"/>
</mobileSecurityTest>
```

セキュリティー・テストの定義 (5/5)
customSecurityTest

- **customSecurityTest** を使用して独自のセキュリティー設定を指定します。
- モバイル・セキュリティー・テストおよび Web セキュリティー・テストとは異なり、**customSecurityTest** には事前定義された認証レルムが含まれません。開発者によって定義されているテストのみが含まれます。
- **customSecurityTest** 内には、テストをいくつでも定義できます。
- **isInternalUserId="true"** というプロパティーを追加することによって、どのレルムをユーザー・レルムとして使用するかを定義できます。
- ユーザーが認証を行うレルムの順序を定義できます。

```
<customSecurityTest name="SampleCustomSecurityTest">
    <test realm="SampleRealm1" step="1" />
    <test realm="SampleRealm2" step="2"/>
    <test realm="SampleRealm2" isInternalUserID="true" step="3"/>
</customSecurityTest>
```

アジェンダ

- 認証の概念とエンティティー
- レルム、オーセンティケーター、ログイン・モジュールの定義
- セキュリティー・テストの定義
- アプリケーションの保護
- アダプターの保護
- 静的リソースの保護
- 次のステップ
- 確認テスト

アプリケーションの保護

- アプリケーションの保護とは、アプリケーションが Worklight Server への接続を試みるとすぐに認証が要求されることを意味します。
- アプリケーション環境ごとに個別の **securityTest** を **application-descriptor.xml** ファイル内に定義できます。

```
<common securityTest="SampleWebSecurityTest"/>

<android version="1.0" securityTest="SampleMobileSecurityTest">
    <worklightSettings include="true"/>
    <pushSender key="a" senderId="b"/>
            <security>
        <encryptWebResources enabled="true"/>
        <testWebResourcesChecksum enabled="true"/>
    </security>
</android>
```

- 特定の環境に対して **securityTest** が定義されていない場合は、デフォルトのプラットフォーム・テストからなる最小限のセットのみが実行されます。

アジェンダ

- 認証の概念とエンティティー
- レルム、オーセンティケーター、ログイン・モジュールの定義
- セキュリティー・テストの定義
- アプリケーションの保護
- アダプターの保護
- 静的リソースの保護
- 次のステップ
- 確認テスト

アダプターの保護

- アダプター・プロシージャーの保護とは、クライアント・アプリケーションによってこのアダプター・プロシージャーが呼び出されると、認証が要求されることを意味します。
- アダプター・プロシージャーごとに個別の **securityTest** をアダプター XML ファイル内に定義できます。

```
<wl:adapter xmlns:wl="http://www.worklight.com/integration" xmlns:http="http://w

    <displayName>DummyAdapter</displayName>
    <description>DummyAdapter</description>
    <connectivity>
        <connectionPolicy xsi:type="http:HTTPConnectionPolicyType">
            <protocol>http</protocol>
            <domain>rss.cnn.com</domain>
            <port>80</port>
        </connectionPolicy>
        <loadConstraints maxConcurrentConnectionsPerNode="2"/>
    </connectivity>

    <procedure name="getSecretData" securityTest="DummyAdapter-securityTest"/>

</wl:adapter>
```

アジェンダ

- 認証の概念とエンティティー
- レルム、オーセンティケーター、ログイン・モジュールの定義
- セキュリティー・テストの定義
- アプリケーションの保護
- アダプターの保護
- 静的リソースの保護
- 次のステップ
- 確認テスト

静的リソースの保護

- 静的リソースは、Worklight Server からロードされる URL です。
  - 例: Worklight Console、モバイル Web アプリケーション
- 静的リソースの保護とは、指定された URL へのブラウズを試みるときに、必ず Worklight Server が認証を要求することを意味します。
- 静的リソースとその保護は、**authenticationConfig.xml** ファイル内で定義できます。

```
<staticResources>
    <resource id="worklightConsole" securityTest="WorklightConsoleSecurityTest">
        <urlPatterns>/console*</urlPatterns>
    </resource>
</staticResources>
```

アジェンダ

- 認証の概念とエンティティー
- レルム、オーセンティケーター、ログイン・モジュールの定義
- セキュリティー・テストの定義
- アプリケーションの保護
- アダプターの保護
- 静的リソースの保護
- 次のステップ
- 確認テスト

次のステップ

- 以下のモジュールにおいて、いくつかの認証タイプを実装します。
  - フォーム・ベースの認証
  - アダプター・ベースの認証
  - カスタム Java オーセンティケーターとログイン・モジュール
  - LDAP ログイン・モジュール
  - WebSphere Application Server 内の LPTA トークン
- 認証について詳しくは、ユーザー文書を参照してください。

アジェンダ

- 認証の概念とエンティティー
- レルム、オーセンティケーター、ログイン・モジュールの定義
- セキュリティー・テストの定義
- アプリケーションの保護
- アダプターの保護
- 静的リソースの保護
- 次のステップ
- 確認テスト

設問 (1/3)

このモジュールで学習した内容を確認します。
答えは確認テストのスライド 3 にあります。

- オーセンティケーターとログイン・モジュールの違いとして、正しい記述は次のどれですか。
  - オーセンティケーターは、資格情報の収集と検証に使用するサーバー・サイドのエンティティーである。ログイン・モジュールは、ユーザー ID の作成に使用するサーバー・サイドのエンティティーである。
  - オーセンティケーターは、資格情報の収集とユーザー ID の作成に使用するサーバー・サイドのエンティティーである。ログイン・モジュールは、資格情報の検証に使用するサーバー・サイドのエンティティーである。
  - オーセンティケーターは、資格情報の収集に使用するサーバー・サイドのエンティティーである。ログイン・モジュールは、資格情報の検証とユーザー ID の作成に使用するサーバー・サイドのエンティティーである。
  - オーセンティケーターは、資格情報の基本的な検証を実行するクライアント・サイドのエンティティーである。ログイン・モジュールは、資格情報の詳しい検証を実行するサーバー・サイドのエンティティーである。

- 開発者が 2 つのアダプター・プロシージャーを作成しました。各プロシージャーは、それぞれ異なるレルムで、独自のセキュリティー・テストによって保護されます。この方法の結果として、正しい記述は次のどれですか。
  - ユーザーが一方のレルムで認証を行うと、2 番目のレルムではそのユーザーが自動的に認証される。
  - ユーザーは、これらのプロシージャーを同じアプリケーション内で一緒に使用することができない。
  - ユーザーは各レルムに個別にログインする必要がある。
  - ユーザーは、一方のレルムからログアウトした後でなければ、もう一方のレルムによって保護されているプロシージャーを使用できない。

設問 (2/3)

このモジュールで学習した内容を確認します。
答えは確認テストのスライド 3 にあります。

- authenticationConfig.xml ファイル内でのレルム、オーセンティケーター、ログイン・モジュールの間の依存関係として、正しい記述は次のどれですか。
  - 各オーセンティケーター・エレメントで、その className、レルム、loginModule を指定する必要がある。
  - 各レルム・エレメントで、そのオーセンティケーターの className と loginModule 名を指定する必要がある。
  - 各 loginModule エレメントで、そのレルムの className とオーセンティケーター名を指定する必要がある。
  - 各オーセンティケーター・エレメントで、そのレルムと loginModule を指定する必要がある。

設問 (3/3)

- オーセンティケーターとログイン・モジュールの違いの正しい記述は、以下のとおりです。
  - オーセンティケーターは、資格情報の収集と検証に使用するサーバー・サイドのエンティティーである。ログイン・モジュールは、ユーザー ID の作成に使用するサーバー・サイドのエンティティーである。
  - オーセンティケーターは、資格情報の収集とユーザー ID の作成に使用するサーバー・サイドのエンティティーである。ログイン・モジュールは、資格情報の検証に使用するサーバー・サイドのエンティティーである。
  - オーセンティケーターは、資格情報の収集に使用するサーバー・サイドのエンティティーである。ログイン・モジュールは、資格情報の検証とユーザー ID の作成に使用するサーバー・サイドのエンティティーである。
  - オーセンティケーターは、資格情報の基本的な検証を実行するクライアント・サイドのエンティティーである。ログイン・モジュールは、資格情報の詳しい検証を実行するサーバー・サイドのエンティティーである。
- 開発者が 2 つのアダプター・プロシージャーを作成しました。各プロシージャーは、それぞれ異なるレルムで、独自のセキュリティー・テストによって保護されます。この方法の結果として、正しい記述は次のどれですか。
  - ユーザーが一方のレルムで認証を行うと、2 番目のレルムではそのユーザーが自動的に認証される。
  - ユーザーは、これらのプロシージャーを同じアプリケーション内で一緒に使用することができない。
  - ユーザーは各レルムに個別にログインする必要がある。
  - ユーザーは、一方のレルムからログアウトした後でなければ、もう一方のレルムによって保護されているプロシージャーを使用できない。
- authenticationConfig.xml ファイル内でのレルム、オーセンティケーター、ログイン・モジュールの依存関係として、正しい記述は次のどれですか。
  - 各オーセンティケーター・エレメントで、その className、レルム、loginModule を指定する必要がある。
  - 各レルム・エレメントで、そのオーセンティケーターの className と loginModule 名を指定する必要がある。
  - 各 loginModule エレメントで、そのレルムの className とオーセンティケーター名を指定する必要がある。
  - 各オーセンティケーター・エレメントで、そのレルムと loginModule を指定する必要がある。
  -

特記事項

- これらの資料は、以下のご使用条件に同意していただける場合に限りご使用いただけます。
- 本書は米国 IBM が提供する製品およびサービスについて作成したものです。
- 本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、または サービスのみが使用可能であることを意味するものではありません。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の 製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作または サービスの 評価および検証は、お客様の責任で行っていただきます。
- IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。
  - 〒103-8510
    東京都中央区日本橋箱崎町19番21号
    日本アイ・ビー・エム株式会社
    法務・知的財産
    知的財産権ライセンス渉外
- **以下の保証は、国または地域の法律に沿わない場合は、適用されません。IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態で提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含む すべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が 禁じられる場合、強行規定の制限を受けるものとします。**
- この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。
- 本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部では ありません。それらの Web サイトは、お客様の責任でご使用ください。
- IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。
- 本プログラムのライセンス保持者で、(i) 独自に作成したプログラムと その他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。
  - IBM Corporation
    Dept F6, Bldg 1
    294 Route 100
    Somers NY 10589-3216
    USA
- 本プログラムに関する上記の情報は、適切な使用条件の下で使用すること ができますが、有償の場合もあります。
- 本書で説明されているライセンス・プログラムまたはその他のライセンス資 料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。
- IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っておりません。したがって、 他社製品に関する実行性、互換性、またはその他の要求については確証できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者にお願いします。

著作権使用許諾:

- 本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに 準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。従って IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性が あることをほのめかしたり、保証することはできません。
- それぞれの複製物、サンプル・プログラムのいかなる部分、またはすべての派生的創作物にも、次の ように、著作権表示を入れていただく必要があります。
  - © (お客様の会社名) (西暦年).このコードの一部は、IBM Corp. のサンプル・プログラムから取られています。© Copyright IBM Corp. _年を入れる_. All rights reserved.

プライバシー・ポリシーの考慮事項

- サービス・ソリューションとしてのソフトウェアも含めた IBM ソフトウェア製品 (「ソフトウェア・オファリング」) では、製品の使用に関する情報の収集、エンド・ユーザーの使用感の向上、エンド・ユーザーとの対話またはその他の目的のために、Cookie はじめさまざまなテクノロジーを使用することがあります。多くの場合、ソフトウェア・オファリングにより個人情報が収集されることはありません。IBM の「ソフトウェア・オファリング」の一部には、個人情報を収集できる機能を持つものがあります。ご使用の「ソフトウェア・オファリング」が、これらの Cookie およびそれに類するテクノロジーを通じてお客様による個人情報の収集を可能にする場合、以下の具体的事項を確認ください。
- このソフトウェア・オファリングは、展開される構成に応じて、(アプリケーション・サーバーが生成する) セッション情報を収集するセッションごとの Cookie を使用する場合があります。これらの Cookie は個人情報を含まず、セッション管理のために要求されるものです。加えて、匿名ユーザーの認識および管理のために持続的な Cookie が無作為に生成される場合があります。これらの Cookie も個人情報を含まず、要求されるものです。
- この「ソフトウェア・オファリング」が Cookie およびさまざまなテクノロジーを使用してエンド・ユーザーから個人を特定できる情報を収集する機能を提供する場合、お客様は、このような情報を収集するにあたって適用される法律、ガイドライン等を遵守する必要があります。これには、エンドユーザーへの通知や同意の要求も含まれますがそれらには限られません。このような目的での Cookie を含む様々なテクノロジーの使用の詳細については、IBM の『IBM オンラインでのプライバシー・ステートメント』(http://www.ibm.com/privacy/details/jp/ja/) の『クッキー、ウェブ・ビーコン、その他のテクノロジー』および『IBM Software Products and Software-as-a-Service Privacy Statement』(http://www.ibm.com/software/info/product-privacy) を参照してください。

サポートおよびコメント

- IBM Worklight の一連の文書、トレーニング資料、および質問をポストできるオンライン・フォーラムはすべて、次の IBM Web サイトからご覧になれます。
  - http://www.ibm.com/mobile-docs
- **サポート**
  - ソフトウェア・サブスクリプション & サポート (ソフトウェア・メンテナンスと呼ばれる場合もあります) は、パスポート・アドバンテージおよびパスポート・アドバンテージ・エクスプレスから購入されたライセンスに含まれています。International Passport Advantage Agreement および IBM International Passport Advantage Express Agreement の追加情報については、次のパスポート・アドバンテージ Web サイトを参照してください。
    - http://www.ibm.com/software/passportadvantage
  - ソフトウェア・サブスクリプション & サポートが有効になっている場合、IBM は、インストールおよび使用法 (ハウツー) に関する短期間の FAQ に対するサポートや、コード関連の質問に対するサポートを提供します。詳しくは、次の IBM ソフトウェア・サポート・ハンドブックを参照してください。
    - http://www.ibm.com/support/handbook
- **ご意見**
  - 本資料に関するご意見をお寄せください。本資料の具体的な誤りや欠落、正確性、編成、題材、または完成度に関するご意見をお寄せください。お寄せいただくご意見は、本マニュアルまたは製品の情報、およびその情報の提示方法に関するもののみとしてください。
  - 製品の技術的な質問および情報、および価格については、担当の IBM 営業所、IBM ビジネス・パートナー、または認定リマーケターにお問い合わせください。
  - IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。IBM またはいかなる組織も、お客様から提示された問題についてご連絡を差し上げる場合にのみ、お客様が提供する個人情報を使用するものとします。
  - どうぞよろしくお願いいたします。
  - 次の IBM Worklight Developer Edition サポート・コミュニティーにご意見をお寄せください。
    - https://www.ibm.com/developerworks/mobile/worklight/connect.html
  - IBM からの回答を希望される場合は、以下の情報をご連絡ください。
    - 氏名
    - 住所
    - 企業または組織
    - 電話番号
    - E メール・アドレス

ありがとうございました