# BlackBerry Spark SDK

## Development Guide

0.6 beta

# Contents

# What is the BlackBerry Spark SDK?

The BlackBerry Spark SDK is a development tool that allows you to integrate advanced security features with your Android and iOS apps. The SDK gives any mobile app the ability to leverage BlackBerry security services that detect, evaluate, and respond to environmental risks and a wide range of cyber threats in real time. The BlackBerry Spark SDK enables you to build apps that are resistant to sophisticated mobile attacks while offering the highest level of protection for your organization's users and data.

The BlackBerry Spark SDK provides APIs that perform device security checks to ensure protection against security vulnerabilities, as well as APIs that initiate calls to the BlackBerry Infrastructure and dedicated cloud services to assess and respond to threats. For example, the mobile threat detection capabilities of the SDK initiate calls to the CylanceINFINITY cloud service, which uses sophisticated AI and machine-learning to provide a real-time evaluation of whether an Android app is safe or potentially malicious.

When you integrate the BlackBerry Spark SDK, you can decide which device checks and security services you want to implement and how you want your app's functionality, user experience, and UI to respond to the analysis and evaluation of security risks.

Any Android or iOS app can integrate the BlackBerry Spark SDK. The features and services offered by the SDK do not require the installation of any BlackBerry software or product. The SDK does not provide management capabilities for apps or user accounts, or any level of device control or administration. If you are interested in secure mobile app development in combination with the advanced controls offered by BlackBerry UEM, visit BlackBerry Docs to learn more about the BlackBerry Dynamics SDK.

**Note:** The BlackBerry Spark SDK is currently available as a public beta release that is subject to further testing and changes by BlackBerry. Some SDK features might not yet be available or may require further development. The SDK has been made available for early testing and evaluation purposes, with a full release to follow in the near future. Before you use the SDK, review and agree with terms and conditions of the beta release.

# Key features of the BlackBerry Spark SDK

The following features are available in the current beta release of the BlackBerry Spark SDK:

| Feature | Platform | Description |
|---|---|---|
| **Device security checks** | | |
| Jailbreak detection | iOS | Detect whether the device is jailbroken. |
| Root detection | Android | Detect whether the device is rooted. |
| Debugging detection | iOS<br>Android | Detect whether debug mode is enabled on the device. |
| Inline hooking detection | iOS<br>Android | Detect inline hooking, a method used by malicious software to intercept calls to target functions. |
| Emulation detection | Android | Detect whether the app is running on an emulator. |
| Screen lock check | iOS<br>Android | Detect whether a screen lock is enabled on the device (for example, a password or PIN). |

| Feature | Platform | Description |
|---|---|---|
| Developer mode check | Android | Detect whether developer mode is enabled on the device. |
| Disk encryption check | Android | Detect whether disk encryption is enabled on the device. |
| App authorization | iOS<br>Android | Require users to set a password or PIN to access the app. Optionally, enable a user to unlock the app using biometrics. |
| **Software security** | | |
| Minimum OS check | iOS<br>Android | Check whether the device satisfies a minimum OS requirement that you can configure. |
| Minimum security patch level check | Android | Check whether the device satisfies a minimum security patch level that you can configure. |
| Malicious app detection | Android | Use the local machine learning models that are built into the SDK or send the app files to the CylanceINFINITY cloud service to determine whether an app is safe or potentially malicious. |
| Sideloaded app detection | Android | Detect whether the app is installed from a trusted source (for example, Google Play or the Samsung Galaxy Store); apps from an untrusted source are considered sideloaded. |
| **User identity** | | |
| Malicious URL detection | iOS<br>Android | Send URLs, including URLs in text messages (if access is permitted), to the CylanceINFINITY cloud service to determine whether the URLs are safe or potentially malicious. |
| **Data security** | | |
| Secure app file system and storage | iOS<br>Android | Use secure data storage, allowing your app to store encrypted data that can be read by your app only. |
| App data backup to public cloud services | iOS<br>Android | Block app data backup to public cloud services such as iCloud and Google Cloud. |
| Application Authorization | iOS<br>Android | Require users to set a password or PIN to access the app. |

The following features are implemented in the sample apps that are included in the beta version of the SDK:

- Safe browsing (iOS, Android)
- Root detection (Android)
- Debugging detection (Android)
- Screen lock check (iOS, Android)
- Developer mode check (Android)

- Disk encryption check (Android)
- Minimum OS check (iOS, Android)
- Minimum security patch level check  (Android)
- Malicious app detection (Android)
- PIN creation and entry (Android)
- Biometric authentication (iOS, Android)

# Sharing data and feedback with BlackBerry

Your data and feedback are valuable to deliver a production version of the SDK that secures and protects your users and data as effectively as possible. We encourage you to activate the data collection API (see the DataCollectionRules class reference) that will allow BlackBerry to receive information about the environments, risks, and threats that you encounter. This API does not provide BlackBerry with any information that can be used to identify users or organizations and meets all privacy-related requirements. BlackBerry will not use the information that it receives for any purpose other than the improvement of the BlackBerry Spark SDK.

To submit feedback, visit BlackBerry Developer Support and access the BlackBerry Beta Community.

If you encounter any issues while using the SDK, you can share your log files with BlackBerry Support. Visit BlackBerry Developer Support to access the BlackBerry Beta Community, and see the Diagnostics Class in the API reference.

# Getting started with the BlackBerry Spark SDK

Before you use the SDK, review and agree with terms and conditions of the beta release.

| Step | Description |
| --- | --- |
| 1 | Review the Software requirements. |
| 2 | Register the app with BlackBerry. |
| 3 | Register the identity provider for your app. |
| 4 | Add the App Client ID to your app.<br>• Add the BlackBerry App Client ID to your Android app<br>• Add the BlackBerry App Client ID to your iOS app |
| 5 | Integrate the BlackBerry Spark SDK into your app.<br>• Integrate the BlackBerry Spark SDK into your Android app<br>• Integrate the BlackBerry Spark SDK into your iOS app |
| 6 | Initialize the BlackBerry Spark SDK.<br>• Initialize the BlackBerry Spark SDK in your Android app<br>• Initialize the BlackBerry Spark SDK in your iOS app |

## Software requirements

| Platform | Requirements |
| --- | --- |
| Android | • Android Studio 3.6.3 or later<br>• Gradle 3.6.3 or later<br>• Android SDK API level 24 or higher |
| iOS | • Swift 5 or later<br>• Xcode 11.3 or later<br>• CocoaPods 1.7 or later |

## Register the app with BlackBerry

You must register your app with BlackBerry through your BlackBerry Online Account. If you don't have an account, you can create one.

1. Browse to the following URL: https://account.blackberry.com/a/organization//applications/add?capability=mtd
2. Log in using your BlackBerry Online Account (*my*Account) credentials.
3. Enter the following information:
   - **Application Name**: The name of your app (for example, MyApp).
   - **Entitlement ID**: It is recommended that you use the package name of your app (for example, com.company.myapp).
   - **Version**: 1.0.0.0

     **Note:** The version number does not need to be updated when you upgrade your app and does not need to match your native app version.
   - **Management**: Clear the **Application will be managed by BlackBerry UEM** option. You must remove this option so that you can use your own identity provider for authentication.
   - **Capabilities**: Select **BlackBerry Protect**. This enables your application to utilize the BlackBerry Protect threat models.
4. Click **Add application**.

**After you finish:** Register the identity provider for your app.

# Register the identity provider for your app

The BlackBerry Spark SDK reuses the existing user identity within your application to facilitate getting the latest security threat information from the BlackBerry Cloud. The library works with your user identity and management systems to provide strong authentication and authorization.

In practice, an OpenID Connect Identity Token belonging to the user that is currently logged in is provided to the BlackBerry Spark SDK runtime. BlackBerry validates this token against your identity provider's token introspection endpoint. This process avoids the need to rely on an application-specific API key.

You can use any identity provider as long as it is compliant with OpenID Connect (https://openid.net/connect/). For more information, see Information about compliant IDPs.

If you don't have an identity provider, you can use Firebase as your identity provider (IDP).

When the IDP is registered, you are provided a BlackBerry App Client ID which you add to your app.

**Before you begin:**

- Register the app with BlackBerry.
- Verify that you have the following information:
  - The discovery URL of your IDP
  - The Authorized Client ID for your app

1. In your organization's BlackBerry Online Account, on the navigation menu, click **Applications**.
2. Click your app.
3. On the **IDP** tab, in the **Identity Provider** section, do the following:.
   a) In the **Discovery URL** field, type the discovery URL of the identity provider.
   b) In the **Client ID** field, type the Authorized Client ID.

   No other fields are required.
4. Click **Register IDP**.
   A BlackBerry App Client ID is created.

**After you finish:**

- Add the BlackBerry App Client ID to your Android app
- Add the BlackBerry App Client ID to your iOS app

### Information about compliant IDPs

You can integrate the BlackBerry Spark SDK into your app using any identity provider (IDP) over the internet as long as it is compliant with OpenID Connect (https://openid.net/connect/).

The following table lists a few examples of IDPs that are compatible and how to determine the discovery URLs and authorized client IDs:

| Identity provider | Discovery URL | Authorized Client IDs |
|---|---|---|
| Firebase | https://securetoken.google.com/*${Project-ID}*/.well-known/openid-configuration | ${Project-ID}<br>The Project ID in Firebase. |
| Okta | https://*${yourOktaOrg}*/.well-known/openid-configuration | One of your app's OAuth 2.0 client IDs registered with Okta. |
| Google | https://account.google.com/.well-known/openid-configuration | In your app's Google-Services.json file, use the value at 'client > oauth_client > client_id'. |

If you don't have access to your IDP to determine the discovery URL or authorized client ID, but you do have a JWT Identity token, you can use a third-party token inspection tool to examine the token (for example, https://jwt.io).

- 'iss' is the token issuer which you can use to determine the discovery URL by adding `/.well-known/openid-configuration`
- 'aud' is the intended audience of the token and is the Authorized Client ID.

# Integrating the IDP and BlackBerry Spark SDK into your Android app

This section describes how to add the IDP and integrate and initialize the BlackBerry Spark SDK with an Android app.

### Add the BlackBerry App Client ID to your Android app

**Before you begin:** Register the identity provider for your app and copy the BlackBerry App Client ID.

In Android Studio, in the **AndroidManifest.xml** file, include the App Client ID.
For example:

```
<application>
         <meta-data
            android:name="com.blackberry.security.ClientID"
            android:value="abcdefgh-1234-1234-1234-abcdefgh" />
 </application>
```

**After you finish:** Integrate the BlackBerry Spark SDK into your Android app.

### Integrate the BlackBerry Spark SDK into your Android app

Use Gradle to integrate BlackBerry Spark SDK into your Android Studio project.

**Before you begin:** Add the BlackBerry App Client ID to your Android app.

1. In your root-level (project-level) Gradle file (`build.gradle`), add a rule to include the BlackBerry Maven repository.

```
allprojects {
    repositories {
        google()
        jcenter()
        maven {
            url "https://software.download.blackberry.com/repository/maven/"
        }
    }
}
```

2. In the app-level module of your Gradle file (usually `app/build.gradle`), declare a dependency on the BlackBerry Spark SDK for Android.

```
# BlackBerry Spark SDK
implementation 'com.blackberry.security:appsecure:0.1+'
```

3. Sync your app to ensure that all dependencies are downloaded.

**After you finish:** Initialize the BlackBerry Spark SDK in your Android app.

### Initialize the BlackBerry Spark SDK in your Android app

**Before you begin:** Integrate the BlackBerry Spark SDK into your Android app.

1. Import the BlackBerry Spark SDK into an activity.

```
import com.blackberry.security.core.SecurityControl;
```

2. Call `enableSecurity`.

```
private SecurityControl mSecurity;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // Initialize BlackBerry Security Library
        mSecurity = new SecurityControl(this.getApplicationContext());
        mSecurity.enableSecurity();
```

3. Retrieve the identity token of your authenticated user from your IDP.

   The ID token is a JSON Web Token (JWT), which is a cryptographically-signed, Base64-encoded JSON object. To retrieve the ID token from your IDP, you must have already authenticated the user.

   If you are using Firebase, the ID token can be retrieved by following the Firebase instructions to retrieve ID tokens on clients. Other IDPs that are compliant with OpenID Connect typically provide an endpoint and client library which returns the ID token.

4. Provide the identity token to the BlackBerry Spark SDK runtime.

```
mSecurity.provideToken(idtoken)
```

5. Confirm that the `InitializationState` of the runtime is 'active'.

**After you finish:** Using the BlackBerry Spark SDK API reference, configure your application to be notified when a threat is detected.

# Integrating the IDP and BlackBerry Spark SDK into your iOS app

This section describes how to add the IDP and integrate and initialize the BlackBerry Spark SDK with an iOS app.

## Add the BlackBerry App Client ID to your iOS app

**Before you begin:** Register the identity provider for your app and copy the BlackBerry App Client ID.

In Xcode, add the App Client ID to your application's 'info.plist'.
For example:

```
<dict>
   <key>BlackBerrySecuritySettings</key>
   <dict>
      <key>ClientID</key>
      <string>abcdefgh-1234-1234-1234-abcdefgh</string>
</dict>
```

**After you finish:** Integrate the BlackBerry Spark SDK into your iOS app.

## Integrate the BlackBerry Spark SDK into your iOS app

In Xcode, you can add the BlackBerry Spark SDK as a dependency using CocoaPods.

**Before you begin:** Add the BlackBerry App Client ID to your iOS app.

In Xcode, do one of the following to integrate the BlackBerry Spark SDK into the project:

| Task | Steps |
|---|---|
| Use CocoaPods | **a.** Create a pod file (if you don't have one already).<br>`cd 'your project directory'`<br>`pod init`<br>**b.** Add a reference to the BlackBerry Spark SDK pod within your pod file.<br>`pod 'BlackBerrySecurity', :podspec => 'https://software.download.blackberry.com/repository/framework/appsecure/ios/BlackBerrySecurity-latest.podspec'`<br>**c.** Install the pod.<br>`pod install` |

**After you finish:** Initialize the BlackBerry Spark SDK in your iOS app.

### Initialize the BlackBerry Spark SDK in your iOS app

**Before you begin:** Integrate the BlackBerry Spark SDK into your iOS app.

1. Import the **BlackBerrySecurity** module into your class.

   ```
   import BlackBerrySecurity
   ```

2. Initialize the **BlackBerrySecurity** framework and invoke `enableSecurity`.

   ```
   SecurityControl.shared.enableSecurity()
   ```

3. Retrieve the identity token of your authenticated user from your IDP.

   The ID token is a JSON Web Token (JWT), which is a cryptographically-signed, Base64-encoded JSON object. To retrieve the ID token from your IDP, you must have already authenticated the user.

   If you are using Firebase, the ID token can be retrieved by following the Firebase instructions to retrieve ID tokens on clients. Other IDPs that are compliant with OpenID Connect typically provide an endpoint and client library which returns the ID token.

4. Provide the identity token to the BlackBerry Spark SDK runtime.

   ```
   SecurityControl.shared.provideToken(token: idtoken)
   ```

5. Confirm that the `InitializationState` of the runtime is 'active'.

**After you finish:** Using the BlackBerry Spark SDK API reference, configure your application to be notified when a threat is detected.

# Using the BlackBerry Spark SDK API reference

The BlackBerry Spark SDK API reference describes how to use the principal interfaces, packages, and classes of the SDK:

- BlackBerry Spark SDK for Android API reference
- BlackBerry Spark SDK for iOS API reference

The following table highlights key sections of the API reference:

| Item | Description |
|---|---|
| SecurityControl Class Reference | Initializes the BlackBerry Spark SDK library within your app so that threats can be detected and alerts can be provided. |
| AppAuthentication Class Reference | Methods to set, change and enter an application password and manage biometric authentication. |
| AppIdentity Class Reference | Provides a various app identifiers that can be used to determine if the user's session is originating from the same app instance and device when authenticating with the application server. |
| ThreatStatus Class Reference | Provides details about security threats related to the device, app, network, and user. |
| ContentChecker Class Reference | Detect potentially malicious URLs or IP addresses to protect users from malicious websites, phishing attempts, malware, adware, and other web sources that pose a threat to your data. |
| ContentCheckerRules Class Reference | Configure rules that change how the SDK detects malicious URLs and IP addresses. |
| DeviceChecker Class Reference | Perform security checks on the device to identify potential security risks. |
| DeviceSecurityRules Class Reference | Control which device security checks are evaluated when enableSecurity or checkDeviceSecurity are called. |
| DeviceSoftwareRules Class Reference | Configure a check for a minimum Android security patch level and OS version. If the device does not meet these requirements it is considered unsafe. |
| MalwareScanRules Class Reference | Configure rules that control how the SDK detects malware on an Android device. |
| ManageFeatures Class Reference | Retrieve the status of a security feature and enable or disable features. |
| ManageRules Class Reference | Configure and manage security rules. |
| Package com.blackberry.security.file | Store app data in the BlackBerry secure file system. |
| PasswordUtility Class Reference | Check the strength of passwords. |

| Item | Description |
| --- | --- |
| Preferences Class Reference | Manage shared preferences in the BlackBerry secure data store. |
| DataCollectionRules Class Reference | Enable anonymous data collection to help BlackBerry improve the features of the BlackBerry Spark SDK. |
| Diagnostics Class Reference > void uploadLogs (LogsUploadFinishedListener listener) | Send recent logs to BlackBerry support. |

# Troubleshooting IDP configuration issues

| Problem | Possible cause | Possible solution |
|---|---|---|
| After you initialize the BlackBerry Spark SDK with `enableSecurity`, the app does not run. | The BlackBerry App Client ID is missing from AndroidManifest.xml or from the info.plist of the Xcode project. | See:<br>• Add the BlackBerry App Client ID to your Android app<br>• Add the BlackBerry App Client ID to your iOS app |
| After calling `provideToken`, the following are returned:<br>• ErrorDomain: AppConfig<br>• ErrorType: ErrorTypeTokenInvalidClientID | The BlackBerry App Client ID is incorrect, possibly because the value was not copied correctly or the client has been deleted. | See Register the identity provider for your app. |
| After calling `provideToken`, the following are returned:<br>• ErrorDomain: IDPConfig<br>• ErrorType: ErrorTypeNoBearerPolicyForClient | The discovery URL for your identity provider in *my*Account does not match the issuer (iss) in your JWT Bearer token. | Update the discovery URL to match the issuer of the IDP.<br>See Register the identity provider for your app. |
| After calling `provideToken`, the following are returned:<br>• ErrorDomain: IDPConfig<br>• ErrorType: ErrorTypeAzpClaimMismatch | The Authorized Client IDs configured for your IDP in *my*Account do not match with the Audience (aud) or Authorized Party (azp) fields in your JWT Bearer token. | Update the Authorized Client ID.<br>See Register the identity provider for your app. |

# I don't have an identity provider

If you don't have an identity provider, you can create one using Firebase. The BlackBerry Spark SDK sample app 'Pyrite Financial' integrates Firebase as the identity provider and is available for Android and iOS.

You can use the Project ID from the Firebase project to determine the discovery URL and Authorized Client ID. See Information about compliant IDPs.

**Before you begin:** Register the app with BlackBerry

1. Create a Firebase project and register your application.
   - For Android, see https://firebase.google.com/docs/android/setup.
   - For iOS, see https://firebase.google.com/docs/ios/setup.
2. Determine the Google authentication mechanism that you want to integrate with. For example, you can easily use password authentication (Email/Password) as the sign-in method and add a test user. The Pyrite Financial sample application demonstrates password authentication.
3. To configure your Firebase IDP with BlackBerry, you need to retrieve the Project ID from the Firebase console.
   a) On the left menu, beside **Project Overview**, click the gear icon to view the **Project Settings**.

b) Copy the Project ID value.

**After you finish:** Register the identity provider for your app

# Legal notice

WERE FORESEEN OR UNFORESEEN, AND EVEN IF BLACKBERRY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW IN YOUR JURISDICTION, BLACKBERRY SHALL HAVE NO OTHER OBLIGATION, DUTY, OR LIABILITY WHATSOEVER IN CONTRACT, TORT, OR OTHERWISE TO YOU INCLUDING ANY LIABILITY FOR NEGLIGENCE OR STRICT LIABILITY.

THE LIMITATIONS, EXCLUSIONS, AND DISCLAIMERS HEREIN SHALL APPLY: (A) IRRESPECTIVE OF THE NATURE OF THE CAUSE OF ACTION, DEMAND, OR ACTION BY YOU INCLUDING BUT NOT LIMITED TO BREACH OF CONTRACT, NEGLIGENCE, TORT, STRICT LIABILITY OR ANY OTHER LEGAL THEORY AND SHALL SURVIVE A FUNDAMENTAL BREACH OR BREACHES OR THE FAILURE OF THE ESSENTIAL PURPOSE OF THIS AGREEMENT OR OF ANY REMEDY CONTAINED HEREIN; AND (B) TO BLACKBERRY AND ITS AFFILIATED COMPANIES, THEIR SUCCESSORS, ASSIGNS, AGENTS, SUPPLIERS (INCLUDING AIRTIME SERVICE PROVIDERS), AUTHORIZED BLACKBERRY DISTRIBUTORS (ALSO INCLUDING AIRTIME SERVICE PROVIDERS) AND THEIR RESPECTIVE DIRECTORS, EMPLOYEES, AND INDEPENDENT CONTRACTORS.

IN ADDITION TO THE LIMITATIONS AND EXCLUSIONS SET OUT ABOVE, IN NO EVENT SHALL ANY DIRECTOR, EMPLOYEE, AGENT, DISTRIBUTOR, SUPPLIER, INDEPENDENT CONTRACTOR OF BLACKBERRY OR ANY AFFILIATES OF BLACKBERRY HAVE ANY LIABILITY ARISING FROM OR RELATED TO THE DOCUMENTATION.

Prior to subscribing for, installing, or using any Third Party Products and Services, it is your responsibility to ensure that your airtime service provider has agreed to support all of their features. Some airtime service providers might not offer Internet browsing functionality with a subscription to the BlackBerry® Internet Service. Check with your service provider for availability, roaming arrangements, service plans and features. Installation or use of Third Party Products and Services with BlackBerry's products and services may require one or more patent, trademark, copyright, or other licenses in order to avoid infringement or violation of third party rights. You are solely responsible for determining whether to use Third Party Products and Services and if any third party licenses are required to do so. If required you are responsible for acquiring them. You should not install or use Third Party Products and Services until all necessary licenses have been acquired. Any Third Party Products and Services that are provided with BlackBerry's products and services are provided as a convenience to you and are provided "AS IS" with no express or implied conditions, endorsements, guarantees, representations, or warranties of any kind by BlackBerry and BlackBerry assumes no liability whatsoever, in relation thereto. Your use of Third Party Products and Services shall be governed by and subject to you agreeing to the terms of separate licenses and other agreements applicable thereto with third parties, except to the extent expressly covered by a license or other agreement with BlackBerry.

The terms of use of any BlackBerry product or service are set out in a separate license or other agreement with BlackBerry applicable thereto. NOTHING IN THIS DOCUMENTATION IS INTENDED TO SUPERSEDE ANY EXPRESS WRITTEN AGREEMENTS OR WARRANTIES PROVIDED BY BLACKBERRY FOR PORTIONS OF ANY BLACKBERRY PRODUCT OR SERVICE OTHER THAN THIS DOCUMENTATION.

BlackBerry Enterprise Software incorporates certain third-party software. The license and copyright information associated with this software is available at http://worldwide.blackberry.com/legal/thirdpartysoftware.jsp.

BlackBerry Limited
2200 University Avenue East
Waterloo, Ontario
Canada N2K 0A7

BlackBerry UK Limited
Ground Floor, The Pearce Building, West Street,
Maidenhead, Berkshire SL6 1RL
United Kingdom

Published in Canada