
ID Card Toolkit



الهيئة الاتحادية للهوية والجنسية

FEDERAL AUTHORITY FOR IDENTITY & CITIZENSHIP

**ID Card Toolkit - Java & Android Developer Guide –
v1.26**



Revision History

Version	Date	Description
1.0	22 th Feb, 2017	Beta Release 1
1.1	17 th Apr, 2017	Beta Release 2
1.2	09 th Aug ,2017	Beta Release 3
1.3	10 th Dec 2017	Release Version 1.0.0
1.4	30 th Mar, 2018	Release Version 1.0.3
1.5	28 th May, 2018	Release Version 1.0.5
1.6	12 th Jun, 2018	Release Version 1.0.6
1.7	02 nd Aug, 2018	Release Version 1.0.8
1.8	19 th Sep, 2018	Release Version 1.0.8
1.9	12 th Dec, 2018	Release Version 1.0.9
1.10	16 th Feb, 2019	Release Version 1.0.13
1.11	19 th Mar, 2019	Release Version 1.1.0
1.12	03 rd Apr, 2019	Release Version 1.2.0
1.13	13 th May, 2019	Release Version 1.3.0
1.14	11 th Jun, 2019	Release Version 1.4.0
1.15	17 th Jun, 2019	Release Version 1.4.1
1.16	31 st Jul, 2019	Release Version 1.4.3; Included APIs for Card Update functionality.
1.17	04 th Mar, 2020	Release Version 1.4.5
1.18	01 st Apr, 2020	Release Version 1.4.5-r2
1.19	05 th Apr, 2020	Release Version 1.4.6
1.20	02 nd Nov, 2020	Release Version 1.4.8
1.21	24 th May, 2021	Release Version 1.4.9
1.22	10 th Oct, 2021	Release Version 2.0.1-r2
1.23	15 th Nov, 2021	Release Version 2.0.3-r1
1.24	10 th Mar, 2022	Release Version 2.0.4
1.25	12 th Jun, 2022	Release Version 2.0.5-r1
1.26	29 th Oct, 2022	Release Version 2.0.6-r1



Contents

1	Introduction.....	6
1.1	Document Purpose.....	6
2	Development Prerequisites& Environment Setup	6
2.1	Java Environment Prerequisites	6
2.2	Java Environment Setup	6
2.3	Android Environment Prerequisites	6
2.4	Android Environment Setup	6
2.4.1	Mandatory Permissions	6
2.4.2	Permission Required to Use NFC Interface.....	7
2.4.3	Permission Required to Use Bluetooth Readers.....	7
2.4.4	Setting Required to Use USB Connected Readers	7
3	Developing Your First Program	7
4	Development Best Practices.....	9
4.1	Android Development Recommendations.....	9
4.1.1	Asynchronous Tasks.....	9
5	Toolkit API Reference.....	10
5.1	Class: Toolkit.....	10
5.1.1	Constructor: Toolkit	10
5.1.2	Method: getToolkitVersion.....	11
5.1.3	Method: setNfcMode	11
5.1.4	Method: cleanup.....	12
5.1.5	Method: listReaders	12
5.1.6	Method: getReaderWithEmiratesId	13
5.1.7	Method: prepareRequest	13
5.1.8	Method: registerDevice	14
5.1.9	Method: getDeviceId	15
5.1.10	Method: getDataProtectionKey	15
5.1.11	Method: parseMRZ.....	16
5.1.12	Method: getConfigCertificateExpiryDate	16
5.1.13	Method: getLicenseExpiryDate	17
5.1.14	Method: CaptureFace.....	17
5.1.15	Method: verifyFaceOnServerUsingID	18
5.1.16	Method: verifyFaceOnServerUsingPassport	19
5.2	Class: CardReader.....	20
5.2.1	Method: connect	20
5.2.2	Method: disconnect.....	21
5.2.3	Method: getReaderName	21
5.2.4	Method: getReaderSerialNumber	21
5.2.5	Method: getCardVersion	22
5.2.6	Method: prepareRequest	22
5.2.7	Method: setNfcAutenticationParameters	23



5.2.8	Method: setNfcAuthenticationParameters	24
5.2.9	Method: checkCardStatus	24
5.2.10	Method: getCardSerialNumber	25
5.2.11	Method: readPublicData	25
5.2.12	Method: readPublicDataEF	27
5.2.13	Method: parseEFData.....	28
5.2.14	Method: getPkiCertificates.....	29
5.2.15	Method: authenticatePki	29
5.2.16	Method: signData	30
5.2.17	Method: signChallenge.....	31
5.2.18	Method: getFingerData	32
5.2.19	Method: authenticateBiometricOnServer.....	33
5.2.20	Method: authenticateCardAndBiometric.....	33
5.2.21	Method: resetPin.....	34
5.2.22	Method: resetPINWithoutAuthenticateBiometric	35
5.2.23	Method: unblockPin	36
5.2.24	Method: readFamilyBook.....	37
5.2.25	Method: updateData.....	38
5.2.26	Method: readData	39
5.2.27	Method: verifySignature	40
5.2.28	Method: validateCardAndCaptureFace.....	41
5.2.29	Method: authenticateFaceOnServer	42
5.3	Class: SignatureValidator	42
5.3.1	Constructor: SignatureValidator	42
5.3.2	Method: ValidateToolkitResponse	43
6	<i>Toolkit Advanced Digital Signature API Reference</i>	43
6.1.1	Method: padesSign	43
6.1.2	Method: padesVerify	44
6.1.3	Method: xadesSign	45
6.1.4	Method: xadesVerify	46
6.1.5	Method: xadesVerify	47
6.1.6	Method: cadesSign	48
6.1.7	Method: cadesVerify.....	49



Abbreviations

AAR	Android Archive
CA	Certificate Authority
CAdES	CMS Advanced Electronic Signatures
CMS	Cryptographic Message Syntax
ICA	Identity and Citizenship Authority
IDE	Integrated Development Environment
JDK	Java Development Kit
JAR	Java Archive
JNI	Java Native Interface
JRE	Java Runtime Environment
PAdES	PDF Advanced Electronic Signatures
PDF	Portable Document Format
SP	Service Providers
VG	Validation Gateway
XAdES	XML Advanced Electronic Signatures
XML	Extensible Markup Language
.NET	Network Enabled Technologies
EF	Elementary File



1 Introduction

The Identity Citizenship Authority (ICA) has developed the ID Card Toolkit to address the requirements of service providers (SP) to integrate, into their business applications: Identification, Authentication, Digital Signature and Non-repudiation services, around the capabilities of the Emirates ID Card. The Toolkit is comprised of a number of Software Development Kits (SDK) supporting different programming languages and platforms. The supported programming languages include: C/C++, Java, Objective C, Swift and C#.

1.1 Document Purpose

This document is a developer's guide to support the development of Java applications that integrate the ICA's ID Card Toolkit SDK.

2 Development Prerequisites& Environment Setup

Following is the Environment setup for Java and Android development;

2.1 Java Environment Prerequisites

- JDK/JRE version 1.6 (Update 43) or higher
- ICA's ID Card Toolkit SDK for Java

2.2 Java Environment Setup

1. Add the **EIDAToolkit.jar** to the classpath.
2. Add the native libraries used by the toolkit to the java library path.

2.3 Android Environment Prerequisites

1. JDK/JRE version 1.8 or higher
2. ID Card Toolkit SDK Android binaries
3. Minimum API Level 15
4. Android Studio

2.4 Android Environment Setup

1. Create a project in Android studio.
2. Set the minimum Android SDK level required by the application. Please note that the minimum supported API level is 15.
3. Add the **EIDAToolkit.aar**, to the project as a module dependency. All required native libraries are included in the archive.
4. Add the device plugin library as module dependency to the project.
5. Create the EIDAToolkit folder in the root of the external storage.
6. Copy the configuration files into the EIDAToolkit folder.
7. Setup the following permissions mentioned in the below sections.

2.4.1 Mandatory Permissions

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
```



2.4.2 Permission Required to Use NFC Interface

```
<uses-permission android:name="android.permission.NFC"/>
```

2.4.3 Permission Required to Use Bluetooth Readers

```
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN"/>  
<uses-permission android:name="android.permission.BLUETOOTH"/>
```

2.4.4 Setting Required to Use USB Connected Readers

```
<uses-feature android:name="android.hardware.usb.host" />
```

3 Developing Your First Program

Following example shows the sequence of API calls required to develop an application using the Toolkit SDK for Java/Android. The example illustrates this by showing how to use the **read public data** service of the Toolkit. To use other services of the Toolkit, developer can follow a similar approach.

Refer to the *Toolkit API Reference* for details of other Toolkit services. A sample C/C++ program is also provided along with the Toolkit for reference.

1. Initialize the toolkit. Toolkit initialization is performed by the constructor of the Toolkit class which takes two parameters :
 - `inProcessMode`: flag indicates how an application wants to run the Toolkit or integrate with the Toolkit. If the flag is `true` the Toolkit will run in-process that is, inside application process. When the flag is `false` application integrates with the Toolkit running as a separate agent process (agent mode). See the “*ID Card Toolkit Technical Overview*” document for further details about the Toolkit integration modes.
 - `configParams`: Additional configuration parameters provided by an application to the Toolkit. Please refer to the “*ID Card Toolkit - Programmer’s Reference*” section #4 for detailed description about the configuration parameters.

Code snippet:

```
// Create a new object of Toolkit  
Toolkit toolkit = new Toolkit(true, configParams);
```

In case of Android, Toolkit initialization is performed by the constructor of the Toolkit class which takes three parameters. First two parameters are same as explained above. The third parameter is context. It is application context obtained from Android environment. Toolkit Initialization will fail if context is not provided.

Code snippet-Android:

```
Context context = getApplicationContext();  
  
// Create a new object of Toolkit  
Toolkit toolkit = new Toolkit(true, configParams, context);
```



2. Application can retrieve a list of connected smartcard readers using the `listReaders` method. This function returns an array of `CardReader` objects connected if no compatible readers found, will return an empty array. When a Toolkit service requiring fingerprint device is invoked by an application, Toolkit automatically detect the fingerprint device connected using available fingerprint device plugins in its configuration. If the Toolkit is unable to find any fingerprint device plugins in its configuration or if the connected fingerprint device is not supported by any of the available plugins, the service will return an error indicating the same.

Code snippet:

```
CardReader[] readers = toolkit.listReaders();
```

3. Connect to a card reader by calling `Connect` method of `CardReader` object retrieved via the `listReaders` method.

Code snippet:

```
// Connect to the first card reader returned by listReaders
CardReader cardReader = readers[0];
cardReader.connect();
```

Note: To directly access a card reader with an Emirates ID card connected to it use the Toolkit `getReaderWithEmiratesId` method as follows. This effectively avoids the need to call the Toolkit `listReaders` method.

```
CardReader cardReader = toolkit.getReaderWithEmiratesId();
cardReader.connect();
```

4. Call `readPublicData` function to retrieve public data from the connected Emirates ID Card. This method takes six parameters. One important point to note here is that the application needs to generate a Request ID and provide as input to the service. The purpose of this Request ID is to bind every Toolkit Service responses to the Request ID which can be verified by the application to protect against replay attacks. Please refer to “*ID Card Toolkit - Programmer’s Reference*” section #5 for the guidelines on generating the Request ID.

Code snippet:

```
CardPublicData cardPublicData = cardReader.readPublicData(
    requestID,
    readNonModifiableData,
    readModifiableData,
    readPhotography,
    readSignatureImage,
    readAddress);
```

5. If successful, the service will return `CardPublicData` object. The response class also points to the digitally signed Toolkit Response in XML format. Please refer to Toolkit Response XML for more details on the Toolkit response format. Applications need to validate the Toolkit Response



digital signature and Request ID to verify the authenticity of the response. Please refer to the Java sample program provided along with the Toolkit for validating Toolkit Response XML.

6. Disconnect from the connected Emirates ID Card by calling the `disconnect` method.

Code snippet:

```
cardReader.disconnect();
```

4 Development Best Practices

4.1 Android Development Recommendations

Following are some of the recommendations when integrating the Toolkit SDK on the Android platform.

4.1.1 Asynchronous Tasks

On the Android platform SDK operations should be implemented on a background thread to prevent impact on application responsiveness. This can be done using RxAndroid thread scheduler, custom threads, AsyncTask, Handler etc.



5 Toolkit API Reference

5.1 Class: Toolkit

This class provides methods required to initialize the `Toolkit` and get the connected smartcard readers.

5.1.1 Constructor: Toolkit

Toolkit constructor initializes the Toolkit context and prepares the SDK for use.

Syntax

```
Java :
public Toolkit(
    boolean inProcessMode,
    String configParams
) throws ToolkitException;

Android :
public Toolkit(
    boolean inProcessMode,
    String configParams,
    Object context
) throws ToolkitException;
```

Parameters

<code>inProcessMode</code>	<p>The flag <code>inProcessMode</code> helps an application to select how it wants the Toolkit shared library to handle the Toolkit API/Service requests. If this flag is <code>true</code>, the Toolkit will fulfill the API/Service requests within the library itself. If the flag is <code>false</code> then the Toolkit shared library in the application process will forward the API/Service requests to Toolkit Agent running in the same system where the application is running.</p> <p>Please refer the “<i>ID Card Toolkit Technical Overview</i>” document for further details about the available integration modes.</p> <p>Please note agent mode is not supported in the android SDK.</p>
<code>configParams</code>	<p>Application specific configuration parameters. Please refer to “<i>ID Card Toolkit - Programmer’s Reference</i>” section #4 for detailed description about the configuration parameters.</p> <p>If configuration parameters are provided as a file, file need to be read and the content of the file need to be provided as a string or else a string containing the configuration parameters need to be provided.</p>
<code>context</code>	Android Application context.



Throws

`ToolkitException` In case of error, the function throws `ToolkitException`.

5.1.2 Method: `getToolkitVersion`

This method is used to get the version of Toolkit being used by the application.

Syntax

```
public String getToolkitVersion() throws ToolkitException;
```

Parameters

No parameters

Return

`String` Returns Toolkit version in the format of `major.minor.patch` version.

Throws

`ToolkitException` In case of error, the function throws `ToolkitException`.

5.1.3 Method: `setNfcMode`

This method is used to set the NFC tag object to the toolkit. This method must be called every time before calling `connect`, `listReaders` or `getReaderWithEmiratesId` method. Please refer to the Android sample for details on the usage of this method.

Please note that this method is used only with Android SDK along with NFC Reader plugin.

Syntax

```
public void setNfcMode(Object tag) throws ToolkitException;
```

Parameters

`Object` NFC tag object. A new tag object is created every time a tag is discovered (or a Card is tapped), even if it is the same card. The tag is invalidated when card is removed or goes out of the range of device or when application calls `disconnect`.

Return

`void`

Throws

`ToolkitException` In case of error, the function throws `ToolkitException`.



5.1.4 Method: cleanup

The method is used to release the resources used by the Toolkit and close the service context. In case of in-process mode, the Toolkit performs cleanup within the Toolkit shared library itself releasing the service context and associated resources. In case of agent mode the Toolkit shared library sends cleanup request to the agent through Web Socket connection. The agent in turn performs cleanup operation and releases resources associated with the service context.

Please note agent mode is not supported in the android SDK.

Syntax

```
public void cleanup() throws ToolkitException;
```

Parameters

No parameters

Return

void

Throws

ToolkitException In case of error, the function throws ToolkitException.

5.1.5 Method: listReaders

This method is used to list all compatible smartcard readers connected to the system. In case of Android this function lists all the smartcard readers of the registered plugins.

Syntax

```
public CardReader[] listReaders() throws ToolkitException;
```

Parameters

No parameters

Return

CardReader[] If the function succeeds, returns array of CardReader object of connected smart card readers. If no readers available, will return an empty array.

Please refer to section #5.2 of this document for the functions related to CardReader class.

Throws

ToolkitException In case of error, the function throws ToolkitException



5.1.6 Method: `getReaderWithEmiratesId`

This method is used to check all the smartcard readers available in the system and return the name of the reader with Emirates ID Card inserted.

Syntax

```
public CardReader getReaderWithEmiratesId() throws ToolkitException;
```

Parameters

No parameters

Return

CardReader	If the function succeeds, returns the CardReader object of card reader with Emirates ID Card inserted.
	Please refer to section #5.2 of this document for the functions related to CardReader

Throws

ToolkitException	In case of error, the function throws ToolkitException.
------------------	---

5.1.7 Method: `prepareRequest`

This method helps to establish secure context for the execution of a Toolkit service. The Request Handle returned by this function in the Toolkit Response XML is used by the applications to encode sensitive information sent via Toolkit Services. This ensures that the sensitive parameters such as User ID, Password & PIN required by some of the Toolkit Services, are protected against sniffing and replay attacks.

Syntax

```
public String prepareRequest(
                        String requestId
                    ) throws ToolkitException
```

Parameters

requestId	Please refer to “ID Card Toolkit - Programmer’s Reference” section #5 for the guidelines on generating the Request ID.
-----------	--

Return

String	String which contains 8-byte request handle in base64 encoded format.
--------	---

Throws

ToolkitException	In case of error, the function throws ToolkitException.
------------------	---



5.1.8 Method: registerDevice

This method is used to register a device with Validation Gateway (VG) against the Service Provider (SP) license. Prior to calling this method application developers must invoke `PrepareRequest` method successfully. The Request Handle returned by `PrepareRequest` method in the Toolkit Response XML is required to encode `encodedUserId` and `encodedPassword` data sent through this method. Only a registered device is allowed to access Toolkit Services except in case of reading public data in offline mode.

Syntax

```
public RegisterDeviceResponse registerDevice(
    String encodedUserId,
    String encodedPassword,
    int deviceReferenceId
) throws ToolkitException;
```

Parameters

<code>encodedUserId</code>	User ID of an authorized Service Provider (SP) encrypted and base64 encoded in accordance with the procedure described in the “ID Card Toolkit - Programmer’s Reference” section #7.
<code>encodedPassword</code>	Password for the <code>encodedUserId</code> encrypted and base64 encoded in accordance with the procedure described in the “ID Card Toolkit - Programmer’s Reference” section #7.
<code>deviceReferenceId</code>	This is a unique ID generated for every registered device by Service Provider (SP) with respect to their internal device management process. ICA stores this device reference ID provided by service provider along with the device information during registration.

Return

`ResgisterDeviceResponse` Register device response data. The attributes of this class are as below.

- `String deviceRegistrationID` - Unique registration ID returned by the Validation Gateway (VG).

This class extends `ToolkitResponse` class. `xmlString` property of `ToolkitResponse` object returns XML response. Applications need to validate the XML response digital signature and Request ID to verify the authenticity of the response.

Please refer to Toolkit Response XML for more details on the Toolkit response format.

Throws

`ToolkitException` In case of error, the function throws `ToolkitException`.



5.1.9 Method: getDeviceId

This method is used to retrieve the unique Identifier of the device. This identifier is required to Deregister a device through ICA Web service REST API.

Syntax

```
public String getDeviceId()throws ToolkitException;
```

Parameters

No parameters

Return

String String which contains a unique identifier of the device.

Throws

ToolkitException In case of error, the function throws ToolkitException.

5.1.10 Method: getDataProtectionKey

This method is used to get the ICA Validation Gateway (VG) public key required for performing data protection. The retrieved data protection key is used to encode sensitive parameters such as User ID, Password & PIN required by some of the Toolkit Services.

Syntax

```
public DataProtectionKey getDataProtectionKey() throws
ToolkitException;
```

Parameters

No parameters

Return

DataProtectionKey If the function succeeds, returns an object of class DataProtectionKey.
The list of functions of this class are as below:

- `getPublicKey()`: Returns the public key in byte array format.
- `getModulus()`: Returns the modulus in string format.
- `getExponent()`: Returns the exponent in string format.

This data protection public key is used to encode PIN, UserID and Password parameters in accordance with the procedure described in to “ID Card Toolkit - Programmer’s Reference” section #6 and section #7.

Throws



ToolkitException
In case of error, the function throws **ToolkitException****5.1.11 Method: parseMRZ**

This method is used to parse the provided MRZ (Machine Readable Zone) string and retrieves the attributes corresponding to the MRZ data provided.

Syntax

```
public MRZData parseMRZ(String mrz) throws ToolkitException;
```

Parameters

mrz	MRZ string to be parsed.
-----	--------------------------

Return

MRZData	If the function succeeds, returns an object of class MRZData .
---------	---

Following are the list of functions corresponding to this class:

- **getCardNumber()**: Returns the parsed card number value from the MRZ data provided.
 - **getIDNumber()**: Returns the parsed Identity number value from the MRZ data provided.
 - **getDocumentType()**: Returns the parsed Document type value from the MRZ data provided.
 - **getIssuedCountry()**: Returns the parsed Issued Country value from the MRZ data provided.
 - **getDateOfBirth()**: Returns the parsed Date of Birth value from the MRZ data provided.
 - **getGender()**: Returns the parsed Gender value from the MRZ data provided.
 - **getNationality()**: Returns the parsed Nationality value from the MRZ data provided.
 - **getFullName()**: Returns the parsed Name value from the MRZ data provided.
 - **getCardExpiryDate()**: Returns the parsed Card Expiry Date value from the MRZ data provided.
-

Throws**ToolkitException**In case of error, the function throws **ToolkitException****5.1.12 Method: getConfigCertificateExpiryDate**

This method is used to get the Expiry date of the certificates corresponding to the Toolkit SDK configuration files issued to the Service Provider.

Syntax

```
public ConfigExpiryDate getConfigCertificateExpiryDate() throws ToolkitException;
```



Parameters

No parameters

Return

`ConfigExpiryDate` Returns `ConfigExpiryDate` class object which contains the Certificate Expiry date of the configuration files.

Methods of this class object returns the expiry date of the certificate corresponding to each configuration file.

Throws

`ToolkitException` In case of error, the function throws `ToolkitException`

5.1.13 Method: `getLicenseExpiryDate`

This method is used to get the Expiry Date of the Toolkit SDK License issued to the Service Provider.

Syntax

```
public String getLicenseExpiryDate() throws ToolkitException;
```

Parameters

No parameters

Return

`String` String which contains Expiry Date of the Toolkit SDK License.

Throws

`ToolkitException` In case of error, the function throws `ToolkitException`

5.1.14 Method: `CaptureFace`

This function performs liveness detection along with face capture.

Please note that this method is currently used only with Android SDK

Syntax

```
public void CaptureFace(Object context,
                        String requestId,
                        int requestCode
                        ) throws ToolkitException;
```

Parameters

`context` Android Application context.

`requestId`

Please refer to “*ID Card Toolkit - Programmer’s Reference*” section #5 for the guidelines on generating the Request ID.

`requestCode`

The `requestCode` helps you to identify from which Intent you came back in **onActivityResult**.

Return

`void`

Callback

Callback will be coming in **onActivityResult**. `onActivityResult()` API is available on the Activity class on all API levels

Process of handling Callback:

1. Overwrite **onActivityResult** method in your activity.
2. Match `requestCode` with the sent `requestCode` in the `validateCardAndCaptureFace` API.
3. Match result code with “`Activity.RESULT_OK`”
4. Get result from **intent** using **getStringExtra** method with “`MyFaceKey.INTENT_RESULT`”
5. Match result with “`MyFaceKey.RESULT_SUCCESS`”.
6. Get captured face data from the intent using **getStringExtra** method with “`MyFaceKey.IMAGE_BASE64`”.
7. Can get the status message from the intent using **getStringExtra** method with “`MyFaceKey.INTENT_MESSAGE`” in case of failure.

Throws

`ToolkitException`

In case of error, the function throws `ToolkitException`.

5.1.15 Method: `verifyFaceOnServerUsingID`

This function performs biometric face authentication with Validation Gateway (VG) service against the captured face image. Prior to calling this function, “`CaptureFace`” needs to be invoked for capturing face image.

Syntax

```
public CardPublicData verifyFaceOnServerUsingID(
    String idn,
    String imageBase64) throws
    ToolkitException;
```

Parameters

`idn`

ID Number of the user whose face needs to be verified

`image`

Base64 face image data, captured using the previous call to “`CaptureFace`” method.

Return

`CardPublicData`

Public data class. The list of attributes of this class are as below:



- idNumber
- cardHolderPhoto
- holderSignatureImage
- ModifiablePublicData
- NonModifiablePublicData

Please refer to “*ID Card Toolkit - Programmer’s Reference*” section #2 for the detailed list of attributes which are part of Emirates ID Card Public data.

xmlString property of CardPublicData object returns XML response. Applications need to validate the XML response digital signature and Request ID to verify the authenticity of the response.

Please refer to Toolkit Response XML for more details on the Toolkit response format.

Throws

ToolkitException In case of error, the function throws ToolkitException.

5.1.16 Method: verifyFaceOnServerUsingPassport

This function performs biometric face authentication with Validation Gateway (VG) service against the captured face image. Prior to calling this function, “CaptureFace” needs to be invoked for capturing face image.

Syntax

```
public CardPublicData verifyFaceOnServerUsingPassport(
    String passportNumber,
    String passportCountry,
    String passportExpiry,
    String dob,
    String imageBase64) throws
    ToolkitException;
```

Parameters

passportNumber	Passport number of the user whose face needs to be verified
passportCountry	Passport issuing country (Ex: UAE)
passportExpiry	Passport expiry date (YYYY-MM-DD)
dateOfBirth	Date of birth on the passport (YYYY-MM-DD)
image	Base64 face image data, captured using the previous call to “CaptureFace” method.

Return



CardPublicData

Public data class. The list of attributes of this class are as below:

- idNumber
- cardHolderPhoto
- holderSignatureImage
- ModifiablePublicData
- NonModifiablePublicData

Please refer to “*ID Card Toolkit - Programmer’s Reference*” section #2 for the detailed list of attributes which are part of Emirates ID Card Public data.

`xmlString` property of `CardPublicData` object returns XML response. Applications need to validate the XML response digital signature and Request ID to verify the authenticity of the response.

Please refer to Toolkit Response XML for more details on the Toolkit response format.

Throws

`ToolkitException`

In case of error, the function throws `ToolkitException`.

5.2 Class: CardReader

This class provides all methods required to perform operations with Emirates ID Card after successful connect operation with the provided `CardReader` class object.

5.2.1 Method: connect

Using `Connect` method an application can establish connection to the ID card in the reader identified by the specified reader name. This method must be called every time an ID card is present to the smartcard reader or if the smartcard reader is unplugged and reconnected. Once connected, application can invoke multiple Toolkit services. It is a good programming practice to reuse the connection handle to perform multiple Toolkit services for a single ID card session. This would improve performance of the application. To get names of all the smartcard readers connected to the system, applications can use `ListReaders` method. On the other hand `GetReaderWithEmiratesID` returns name of the smartcard reader with an Emirates ID Card present.

Syntax

```
public void connect() throws ToolkitException;
```

Parameters

No parameters

Return



void

Throws

ToolkitException In case of error, the function throws ToolkitException.

5.2.2 Method: disconnect

This method is used to disconnect the reader. The ID card can be safely removed from the reader after disconnect.

Syntax

```
public void disconnect() throws ToolkitException;
```

Parameters

No parameter

Throws

ToolkitException In case of error, the function throws ToolkitException.

5.2.3 Method: getReaderName

This method is used to retrieve the connected Emirates ID Card smart card reader name.

Syntax

```
public String getReaderName() throws ToolkitException;
```

Parameters

No parameters

Return

String String which contains connected Emirates ID Card smart card reader name. The reader name is obtained on previous call to getReaderWithEmiratesId API.

Throws

ToolkitException In case of error, the function throws ToolkitException.

5.2.4 Method: getReaderSerialNumber

This method is used to retrieve the connected Emirates ID Card smart card reader name.

Please note that this method is used only with Windows SDK.

Syntax

```
public String getReaderSerialNumber() throws ToolkitException;
```



Parameters

No parameters

Return

String

String which contains connected smart card reader serial number (if available). If the reader serial number is not available this will return null. The reader serial number is obtained on previous call to `getReaderWithEmiratesId` API.

Throws

ToolkitException

In case of error, the function throws ToolkitException.

5.2.5 Method: `getCardVersion`

This method is used to retrieve the version number of the connected Emirates ID Card.

Syntax

```
public int getCardVersion() throws ToolkitException;
```

Parameters

No parameters

Return

int

The version number of the connected Emirates ID Card.

Value	Description
1	Card version V1
2	Card version V2
3	Card version V3
4	Card version V4

Throws

ToolkitException

In case of error, the function throws ToolkitException.

5.2.6 Method: `prepareRequest`

This method helps to establish secure context for the execution of a Toolkit service. The Request Handle returned by this function in the Toolkit Response XML is used by the applications to encode sensitive information sent via Toolkit Services. This ensures that the sensitive parameters such as User ID, Password & PIN required by some of the Toolkit Services, are protected against sniffing and replay attacks.

Syntax

```
public String prepareRequest(
    String requestId
) throws ToolkitException
```



Parameters

requestId	Please refer to “ID Card Toolkit - Programmer’s Reference” section #5 for the guidelines on generating the Request ID.
-----------	--

Return

String	String which contains 8-byte request handle in base64 encoded format.
--------	---

Throws

ToolkitException	In case of error, the function throws ToolkitException.
------------------	---

5.2.7 Method: setNfcAuthenticationParameters

This method is used to set parameters for authentication required for Toolkit to access Emirates ID Card through NFC interface.

This version of the overloaded function takes specific card attributes required to perform NFC authentication. This information is printed on the surface of the ID card.

Syntax

```
public void setNfcAuthenticationParameters(
    String cardNumber,
    String dateOfBirth,
    String expiryDate
) throws ToolkitException
```

Parameters

cardNumber	Card number of the Emirates ID Card holder.
dateOfBirth	Date of birth of Emirates ID Card holder (YYMMDD).

Value	Meaning
YY	Last 2 digits of Year . Ex: For 1956, value is 56
MM	Numeric value for the Month. Ex: 01 – Jan, 02 – Feb, 03-Mar 12 - Dec
DD	Numeric value for day

For 12th April 1956 , the value is 560412

expiryDate	Expiry date of Emirates ID Card (YYMMDD).
------------	---

Value	Meaning
YY	Last 2 digits of Year . Ex: For 2017, value is 17
MM	Numeric value for the Month. Ex: 01 – Jan, 02 – Feb, 03-Mar 12 - Dec
DD	Numeric value for day



For 16th November 2017 , the value is 171116

Return

void

Throws

ToolkitException In case of error, the function throws **ToolkitException**.

5.2.8 Method: setNfcAuthenticationParameters

This method is used to set parameters for authentication required for Toolkit to access Emirates ID Card through NFC interface.

This version of the overloaded function takes the MRZ string from the back side of the ID card. How to acquire the MRZ string from the card surface is outside the scope of the Toolkit. For example applications may use some form of OCR library or MRZ scanner to get this information from the card surface.

Syntax

```
public void setNfcAuthenticationParameters (
                                   String mrzData
                                   ) throws ToolkitException
```

Parameters

mrzData Scanned Machine Readable Zone (MRZ) data that is present on the back side of the ID Card

Return

void

Throws

ToolkitException In case of error, the function throws **ToolkitException**.

5.2.9 Method: checkCardStatus

This method performs the Emirates ID Card validation with ICA Validation Gateway (VG) service and returns the card status.

Syntax

```
public ToolkitResponse checkCardStatus(
                                   String requestId
                                   ) throws ToolkitException;
```

Parameters

`requestId`

Please refer to “*ID Card Toolkit - Programmer’s Reference*” section #5 for the guidelines on generating the Request ID.

Return

`ToolkitResponse`

Returns `ToolkitResponse` object which contains status of the inserted Emirates ID Card.

`xmlString` property of `ToolkitResponse` object returns XML response. Applications need to validate the XML response digital signature and Request ID to verify the authenticity of the response.

Please refer to Toolkit Response XML for more details on the Toolkit response format.

Throws

`ToolkitException`

In case of error, the function throws `ToolkitException`.

5.2.10 Method: `getCardSerialNumber`

This method is used to retrieve the serial number of the Emirates ID Card.

Syntax

```
public String getCardSerialNumber() throws ToolkitException;
```

Parameters

No parameters

Return

`String`

String which contains serial number of the Emirates ID Card.

Throws

`ToolkitException`

In case of error, the function throws `ToolkitException`.

Note: This API can be accessed directly without setting the NFC authentication parameters while using NFC interface

5.2.11 Method: `readPublicData`

This method is used to retrieve data stored on the public areas of the Emirates ID Card. Applications can call this API with flags indicating which data is required to be read. Choosing to read only the data required can improve the performance compared to reading the complete public data every time. The data in the Emirates ID Card is digitally signed to protect the integrity which gets verified within the Toolkit.

Syntax

```
public CardPublicData readPublicData(
    String requestId,
```



```
boolean readNonModifiableData,  
boolean readModifiableData,  
boolean readPhotography,  
boolean readSignatureImage,  
boolean readAddress  
) throws ToolkitException;
```

Parameters

<code>requestId</code>	Please refer to “ <i>ID Card Toolkit - Programmer’s Reference</i> ” section #5 for the guidelines on generating the Request ID.
<code>readNonModifiableData</code>	<p>A value of <code>true</code> indicates that the application wants to read the non-modifiable data from the card, <code>false</code> otherwise.</p> <p>Please refer to “<i>ID Card Toolkit - Programmer’s Reference</i>” section #2.1 for the detailed list of attributes which are part of Emirates ID Card Non-Modifiable data.</p>
<code>readModifiableData</code>	<p>A value of <code>true</code> indicates that the application wants to read the modifiable data from the card, <code>false</code> otherwise.</p> <p>Please refer to “<i>ID Card Toolkit - Programmer’s Reference</i>” section #2.2 for the detailed list of attributes which are part of Emirates ID Card Modifiable data.</p>
<code>readPhotography</code>	A value of <code>true</code> indicates that the application wants to read the photography data from the card, <code>false</code> otherwise.
<code>readSignatureImage</code>	A value of <code>true</code> indicates that the application wants to read the handwritten signature image data from the card, <code>false</code> otherwise.
<code>readAddress</code>	<p>A value of <code>true</code> indicates that the application wants to read the address (both home address and work address) from the card, <code>false</code> otherwise.</p> <p>Please refer to “<i>ID Card Toolkit - Programmer’s Reference</i>” section #2.3 for the detailed list of attributes which are part of Emirates ID Card Address data.</p>

Return

<code>CardPublicData</code>	<p>Public data class. The list of attributes of this class are as below:</p> <ul style="list-style-type: none"> • <code>idNumber</code> • <code>cardNumber;</code> • <code>cardHolderPhoto</code>
-----------------------------	--



- holderSignatureImage
- ModifiablePublicData
- NonModifiablePublicData
- HomeAddress
- WorkAddress

Please refer to “ID Card Toolkit - Programmer’s Reference” section #2 for the detailed list of attributes which are part of Emirates ID Card Public data.

`xmlString` property of `CardPublicData` object returns XML response. Applications need to validate the XML response digital signature and Request ID to verify the authenticity of the response.

Please refer to Toolkit Response XML for more details on the Toolkit response format.

Throws

<code>ToolkitException</code>	In case of error, the function throws <code>ToolkitException</code> .
-------------------------------	---

Note: For V3 Card, it is not required to call **SetNFCAuthenticationParameters**, before reading public data from the Emirates ID Card using NFC

5.2.12 Method: readPublicDataEF

This method is used to retrieve Elementary File data stored on the public areas of the ID card. Applications can call this API with identifier indicating which Elementary File data is required to be read. The Elementary file data in the ID card is digitally signed to protect the integrity which gets verified by the Toolkit when the `validateSignature` flag is set to `true`.

Syntax

```
public byte[] readPublicDataEF(
    int publicDataEFTYPE,
    boolean validateSignature
) throws ToolkitException;
```

Parameters

<code>publicDataEFTYPE</code>	Enumeration value corresponding to the Elementary File to be read.
-------------------------------	--

Value	EF Data
IDN_CN (1)	Identification and Card number
ROOT_CERTIFICATE (2)	Root Certificate
NON_MODIFIABLE_DATA (3)	Non Modifiable Public Data



	MODIFIABLE_DATA (4)	Modifiable Public Data
	PHOTOGRAPHY (5)	Photography
	SIGNATURE_IMAGE (6)	Hand Signature Image
	HOME_ADDRESS (7)	Home Address
	WORK_ADDRESS (8)	Work Address
<hr/>		
<code>validateSignature</code>	A value of <code>true</code> indicates that the application wants to validate the digital signature of the read Elementary file, <code>false</code> otherwise.	
<hr/>		
Return		
<code>byte[]</code>	Elementary File Data as byte array.	
<hr/>		
Throws		
<code>ToolkitException</code>	In case of error, the function throws <code>ToolkitException</code> .	

5.2.13 Method: `parseEFData`

This method is used to parse Elementary File data stored on the public areas of the ID card. Applications can call this API with the Elementary File data which is required to be parsed.

Syntax

public static String parseEFData(byte [] efData) throws ToolkitException;

Parameters

efData

Elementary File data bytes which need to be parsed.

EF data of the following files can be passed.

Value	EF Data
IDN_CN (1)	Identification and Card number
ROOT_CERTIFICATE (2)	Root Certificate
NON_MODIFIABLE_DATA (3)	Non Modifiable Public Data
MODIFIABLE_DATA (4)	Modifiable Public Data
PHOTOGRAPHY (5)	Photography
SIGNATURE_IMAGE (6)	Hand Signature Image
HOME_ADDRESS (7)	Home Address
WORK_ADDRESS (8)	Work Address

Return

String

If the function succeeds, returns the parsed Elementary File Data in a JSON string

Throws



ToolkitException

In case of error, the function throws ToolkitException.

Note: This is a static method, and can be invoked without creating CardReader object

5.2.14 Method: getPkiCertificates

This method reads the authentication and signing certificates from the Emirates ID Card. Prior to calling this service application developers must invoke `PrepareRequest` method successfully. The Request Handle returned by `PrepareRequest` method in the Toolkit Response XML is required to encode `encodedPin` data sent through this method.

Syntax

```
public CardCertificates getPkiCertificates(
    String encodedPin
) throws ToolkitException;
```

Parameters

<code>encodedPin</code>	ID Card PKI Pin which is encrypted and base64 encoded in accordance with the procedure described in the “ID Card Toolkit - Programmer’s Reference” section #6.
-------------------------	--

Return

<code>CardCertificates</code>	If the function succeeds, returns <code>CardCertificates</code> object which contains requested authentication and digital signature certificates retrieved from the inserted Emirates ID Card.
-------------------------------	---

`xmlString` property of `CardCertificates` object returns XML response. Applications need to validate the XML response digital signature and Request ID to verify the authenticity of the response.

Please refer to Toolkit Response XML for more details on the Toolkit response format.

Throws

<code>ToolkitException</code>	In case of error, the function throws <code>ToolkitException</code> .
-------------------------------	---

`getAttemptsLeft` method from `ToolkitException` class returns the number of attempts left to retry this operation before the card gets blocked..

5.2.15 Method: authenticatePki

This method is used to authenticate the ID Card PIN and verify the revocation status of the ID Card authentication certificate. Prior to calling this method application developers must invoke `PrepareRequest` successfully. The Request Handle returned by `PrepareRequest` method in the Toolkit Response XML is required to encode `encodedPin` data sent through this method.



Syntax

```
public ToolkitResponse authenticatePKI(
    String encodedPin
) throws ToolkitException
```

Parameters

encodedPin	ID Card PKI Pin which is encrypted and base64 encoded in accordance with the procedure described in the “ <i>ID Card Toolkit - Programmer’s Reference</i> ” section #6.
------------	---

Return

ToolkitResponse	<p>ToolkitResponse object which contains the PKI authentication status.</p> <p>xmlString property of ToolkitResponse object returns XML response. Applications need to validate the XML response digital signature and Request ID to verify the authenticity of the response.</p> <p>Please refer to Toolkit Response XML for more details on the Toolkit response format.</p>
-----------------	--

Throws

ToolkitException	<p>In case of error, the function throws ToolkitException.</p> <p>getAttemptsLeft method from ToolkitException class returns the number of attempts left to retry this operation before the card gets blocked.</p>
------------------	--

5.2.16 Method: signData

This method is used to digitally sign plain data or hash with signature certificate of the ID card. Prior to calling this method application developers must invoke `PrepareRequest` successfully. The Request Handle returned by `PrepareRequest` method in the Toolkit Response XML is required to encode `encodedPin` data sent through this method.

Syntax

```
public SignatureResponse signData(
    byte[] input,
    boolean isInputHashed,
    String encodedPin
) throws ToolkitException
```

Parameters

input	Data to be signed
isInputHashed	Used to specify if the data is hashed or in plain text. If the data is in



plain text the toolkit will apply a hash before signing.

- true = Input is hashed
- false = Input is plain text

encodedPin

String containing Emirates ID Card Pin, encrypted and base64 encoded in accordance with the procedure described in the “ID Card Toolkit - Programmer’s Reference” section #6.

Return

SignatureResponse

SignatureResponse object which contains Signature and signing status.

getSignature member of SignatureResponse object contains the signature.

xmlString property of SignatureResponse object returns XML response Applications need to validate the XML response digital signature and Request ID to verify the authenticity of the response.

Please refer to Toolkit Response XML for more details on the Toolkit response format.

Throws

ToolkitException

In case of error, the function throws ToolkitException.

getAttemptsLeft method from ToolkitException class returns the number of attempts left to retry this operation before the card gets blocked.

5.2.17 Method: signChallenge

This method is used to digitally sign challenge data with authentication certificate of the ID card. Prior to calling this method application developers must invoke PrepareRequest successfully. The Request Handle returned by PrepareRequest method in the Toolkit Response XML is required to encode encodedPin data sent through this method. This method is used to authenticate users from the application through challenge signing and verification.

Syntax

```
public SignatureResponse signChallenge(
    byte[] input,
    boolean isInputHashed,
    String encodedPin
) throws ToolkitException
```

Parameters

input

Data to be signed

isInputHashed

Used to specify if the data is hashed or in plain text. If the data is in



plain text the toolkit will apply a hash before signing.

- true = Input is hashed
- false = Input is plain text

encodedPin

String containing Emirates ID Card Pin, encrypted and base64 encoded in accordance with the procedure described in the “*ID Card Toolkit - Programmer’s Reference*” section #6.

Return

SignatureResponse

SignatureResponse object which contains Signature and signing status.

getSignature member of SignatureResponse object contains the signature.

xmlString property of SignatureResponse object returns XML response Applications need to validate the XML response digital signature and Request ID to verify the authenticity of the response.

Please refer to Toolkit Response XML for more details on the Toolkit response format.

Throws

ToolkitException

In case of error, the function throws ToolkitException.

getAttemptsLeft method from ToolkitException class returns the number of attempts left to retry this operation before the card gets blocked.

5.2.18 Method: getFingerData

This method is used to get finger index along with finger reference identifiers of the fingerprints stored in the Emirates ID Card.

Syntax

```
public FingerData[] getFingerData() throws ToolkitException;
```

Parameters

No parameter

Return

FingerData

An array of FingerData object which contains biometric authentication reference identifiers and its corresponding finger index values of the inserted Emirates ID Card.

Please refer to “*ID Card Toolkit - Programmer’s Reference*”



section #8 for finger index values.

Throws

`ToolkitException` In case of error, the function throws `ToolkitException`.

5.2.19 Method: `authenticateBiometricOnServer`

This function performs biometric authentication with Validation Gateway (VG) service against the captured finger image.

Syntax

```
public ToolkitResponse authenticateBiometricOnServer(
    String requestId,
    FingerIndex fingerIndex,
    int sensorTimeout
) throws ToolkitException;
```

Parameters

<code>requestId</code>	Please refer to “ <i>ID Card Toolkit - Programmer’s Reference</i> ” section #5 for the guidelines on generating the Request ID.
<code>fingerIndex</code>	Please refer to “ <i>ID Card Toolkit - Programmer’s Reference</i> ” section #8 for finger index values. This should be one of the finger index values returned by a previous call to <code>getFingerData</code> .
<code>sensorTimeout</code>	Timeout in number of seconds for sensor to capture the finger image. The underlying biometric plugin is expected to return error in case the device could not capture a valid fingerprint image within the specified time.

Return

`ToolkitResponse` Returns `ToolkitResponse` object which contains the biometric authentication status.

Throws

`ToolkitException` In case of error, the function throws `ToolkitException`.

Note: For V3 Cards, it is not required to call “`SetNFCAuthenticationParameters`” API, before accessing “`authenticateBiometricOnServer`” API using NFC.

5.2.20 Method: `authenticateCardAndBiometric`

This function validates Emirates ID card and performs biometric authentication with Validation Gateway (VG) service against the captured finger image.

Syntax

```
public ToolkitResponse authenticateCardAndBiometric(
```



```
String requestId,
FingerIndex fingerIndex,
int sensorTimeout
) throws ToolkitException;
```

Parameters

requestId	Please refer to “ <i>ID Card Toolkit - Programmer’s Reference</i> ” section #5 for the guidelines on generating the Request ID.
fingerIndex	Please refer to “ <i>ID Card Toolkit - Programmer’s Reference</i> ” section #8 for finger index values. This should be one of the finger index values returned by a previous call to <code>getFingerData</code> .
sensorTimeout	Timeout in number of seconds for sensor to capture the finger image. The underlying biometric plugin is expected to return <code>error</code> in case the device could not capture a valid fingerprint image within the specified time.

Return

ToolkitResponse	Returns <code>ToolkitResponse</code> object which contains the biometric authentication status.
-----------------	---

Throws

ToolkitException	In case of error, the function throws <code>ToolkitException</code> .
------------------	---

5.2.21 Method: resetPin

This method resets the ID card PIN using the ICA Validation Gateway (VG) service. Prior to calling this method application developers must invoke `PrepareRequest` successfully. The Request Handle returned by `PrepareRequest` method in the Toolkit Response XML is required to encode `encodedPin` data sent through this method. This method involves biometric authentication of the card holder against the captured finger image.

Syntax

```
public ToolkitResponse resetPin(
    String encodedPin,
    FingerData fingerData,
    int sensorTimeout
) throws ToolkitException
```

Parameters

encodedPin	Emirates ID Card PKI Pin which is encrypted and base64 encoded in accordance with the procedure described in the “ <i>ID Card Toolkit - Programmer’s Reference</i> ” section #6.
fingerData	<code>FingerData</code> object referring to finger reference identifier and index values obtained through earlier call to



	<code>getFingerData.</code>
<code>sensorTimeout</code>	Timeout in number of seconds for sensor to capture the finger image. The underlying biometric plugin is expected to return error in case the device could not capture a valid fingerprint image within the specified time.

Return

<code>ToolkitResponse</code>	<p>If the function succeeds, returns <code>ToolkitResponse</code> object.</p> <p><code>xmlString</code> property of <code>ToolkitResponse</code> object returns XML response based on the response from Validation Gateway. Applications need to validate the XML response digital signature and Request ID to verify the authenticity of the response.</p> <p><code>responseStatus</code> property of <code>ToolkitResponse</code> object returns the <code>resetPin</code> status.</p>
------------------------------	--

Throws

<code>ToolkitException</code>	<p>In case of error, the function throws <code>ToolkitException</code>.</p> <p><code>getAttemptsLeft</code> method from <code>ToolkitException</code> class returns the number of attempts left before card gets blocked.</p>
-------------------------------	---

5.2.22 Method: `resetPINWithoutAuthenticateBiometric`

This method resets the ID card PIN using the ICA Validation Gateway (VG) service. Prior to calling this method application developers must invoke `PrepareRequest` successfully. The Request Handle returned by `PrepareRequest` method in the Toolkit Response XML is required to encode `encodedPin` data sent through this method.

Syntax

```
public ToolkitResponse resetPINWithoutAuthenticateBiometric (
    String encodedPin
) throws ToolkitException
```

Parameters

<code>encodedPin</code>	Emirates ID Card PKI Pin which is encrypted and base64 encoded in accordance with the procedure described in the “ID Card Toolkit - Programmer’s Reference” section #6.
-------------------------	---

Return

<code>ToolkitResponse</code>	<p>If the function succeeds, returns <code>ToolkitResponse</code> object.</p> <p><code>xmlString</code> property of <code>ToolkitResponse</code> object returns</p>
------------------------------	---



XML response based on the response from Validation Gateway. Applications need to validate the XML response digital signature and Request ID to verify the authenticity of the response.

`responseStatus` property of `ToolkitResponse` object returns the `resetPin` status.

Throws

<code>ToolkitException</code>	In case of error, the function throws <code>ToolkitException</code> .
-------------------------------	---

`getAttemptsLeft` method from `ToolkitException` class returns the number of attempts left before card gets blocked.

5.2.23 Method: unblockPin

This method is used to unblock the ID Card PIN, if the card is blocked after multiple unsuccessful PIN authentications. Prior to calling this method application developers must invoke `PrepareRequest` successfully. The Request Handle returned by `PrepareRequest` method in the Toolkit Response XML is required to encode `encodedPin` data sent through this method. This method involves biometric authentication of the card holder against the captured finger image.

Syntax

```
public ToolkitResponse unblockPin(
    String encodedPin,
    FingerData fingerData,
    int sensorTimeout
)throws ToolkitException
```

Parameters

<code>encodedPin</code>	Emirates ID Card PKI Pin which is encoded in accordance with the procedure described in the “ <i>ID Card Toolkit - Programmer’s Reference</i> ” section #6.
<code>fingerData</code>	<code>FingerData</code> object referring to finger reference identifier and index values obtained through earlier call to <code>getFingerData</code> .
<code>sensorTimeout</code>	Timeout in number of seconds for sensor to capture the finger image. The underlying biometric plugin is expected to return error in case the device could not capture a valid fingerprint image within the specified time.

Return

<code>ToolkitResponse</code>	If the function succeeds, returns <code>ToolkitResponse</code> object.
------------------------------	--

`xmlString` property of `ToolkitResponse` object returns XML response based on the response from Validation



Gateway. Applications need to validate the XML response digital signature and Request ID to verify the authenticity of the response.

`responseStatus` property of `ToolkitResponse` object returns the `unblockPin` status.

Throws

`ToolkitException` In case of error, the function throws `ToolkitException`.

`getAttemptsLeft` method from `ToolkitException` class returns the number of attempts left before card gets blocked.

5.2.24 Method: readFamilyBook

This method is used to read the family book data from the Emirates ID Card. Family book data is a protected object and access to this requires ID Card Pin Authentication. Prior to calling this method application developers must invoke `PrepareRequest` successfully. The Request Handle returned by `PrepareRequest` method in the Toolkit Response XML is required to encode `encodedPin` data sent through this method.

Syntax

```
public CardFamilyBookData readfamilyBook(
    String encodedPin
) throws ToolkitException;
```

Parameters

`encodedPin` String containing Emirates ID Card Pin, encrypted and base64 encoded in accordance with the procedure described in the “ID Card Toolkit - Programmer’s Reference” section #6.

Return

`CardFamilyBookData` `CardFamilyBookData` object which contains the Family Book data fields retrieved from the inserted Emirates ID Card.

The attributes of `CardFamilyBookData` object are as below.

- `HeadOfFamily`
- `Wife wives[]`
- `Child children[];`

Please refer to “ID Card Toolkit - Programmer’s Reference” section #2.6 for the detailed list of attributes which are part of Emirates ID Card Family Book Data.

`xmlString` property of `CardFamilyBookData` object



returns XML response. Applications need to validate the XML response digital signature and Request ID to verify the authenticity of the response.

Please refer to Toolkit Response XML for more details on the Toolkit response format.

Throws

`ToolkitException`

In case of error, the function throws `ToolkitException`.

`getAttemptsLeft` method from `ToolkitException` class returns the number of attempts left before pin is blocked.

5.2.25 Method: `updateData`

This method is used to update data of the Emirates ID Card. The update of the Emirates ID card can be for existing data (Modifiable data, address and Family Book) or for new containers. The new containers will be personalized and then the data will be updated.

Syntax

```
public ToolkitResponse updateData(
    String requestId,
    int fileType) throws ToolkitException;
```

Parameters

`requestId`

Please refer to “*ID Card Toolkit - Programmer’s Reference*” section #5 for the guidelines on generating the Request ID.

`fileType`

Identifier of the file to be updated.

File Identifier	File / Container value
0	HEALTH_INSURANCE
1	HEALTH_INSURANCE_PROTECTED
2	MOD_DATA
3	HOMEADDRESS
4	WORKADDRESS
5	HEAD_OF_FAMILY
6	WIFE1
7	WIFE2
8	WIFE3
9	WIFE4
10	CHILD1
11	CHILD2
12	CHILD3
13	CHILD4
14	CHILD5
15	CHILD6
16	CHILD7



17	CHILD8
18	CHILD9
19	CHILD10
20	CHILD11
21	CHILD12
22	CHILD13
23	CHILD14
24	CHILD15
25	CHILD16
26	CHILD17
27	CHILD18
28	CHILD19
29	CHILD20

Return

ToolkitResponse

Returns **ToolkitResponse** object which contains status of the inserted Emirates ID Card.

`xmlString` property of **ToolkitResponse** object returns XML response. Applications need to validate the XML response digital signature and Request ID to verify the authenticity of the response.

Please refer to Toolkit Response XML for more details on the Toolkit response format.

Throws

ToolkitException

In case of error, the function throws **ToolkitException**.

5.2.26 Method: readData

This method is used to read the updated container data of the Emirates ID Card. Applications can call this API with file identifier indicating which container data is required to be read. The data in the ID card is digitally signed to protect the integrity which gets verified within the Toolkit.

Syntax

```
public ToolkitResponse readData(
    String requestId,
    int fileType) throws ToolkitException;
```

Parameters

requestId Please refer to “ID Card Toolkit - Programmer’s Reference” section #5 for the guidelines on generating the Request ID.

fileType Identifier of the container file data to be read.

File Identifier	File / Container value
-----------------	------------------------



0	HEALTH_INSURANCE
1	HEALTH_INSURANCE_PROTECTED

Return

`ToolkitResponse`

Returns `ToolkitResponse` object which contains status of the inserted Emirates ID Card.

`xmlString` property of `ToolkitResponse` object returns XML response. Applications need to validate the XML response digital signature and Request ID to verify the authenticity of the response.

Please refer to Toolkit Response XML for more details on the Toolkit response format.

Throws

`ToolkitException`

In case of error, the function throws `ToolkitException`.

5.2.27 Method: `verifySignature`

This method is used to verify digital signature with the corresponding certificate provided by the application.

Syntax

```
public void verifySignature(
    byte[] input,
    Boolean isInputHash,
    byte[] signature,
    byte[] certificate
) throws ToolkitException
```

Parameters

<code>input</code>	Data or hash corresponding to the digital signature to be verified						
<code>isInputHash</code>	<table border="1"> <thead> <tr> <th>Value</th><th>Meaning</th></tr> </thead> <tbody> <tr> <td>true</td><td>Hash data is provided as input parameter</td></tr> <tr> <td>false</td><td>Plain data is provided as input parameter</td></tr> </tbody> </table>	Value	Meaning	true	Hash data is provided as input parameter	false	Plain data is provided as input parameter
Value	Meaning						
true	Hash data is provided as input parameter						
false	Plain data is provided as input parameter						
<code>signature</code>	Buffer containing digital signature.						
<code>certificate</code>	Certificate data corresponding to the signature verification						

Return

`void`

Throws



ToolkitException

In case of error, the function throws ToolkitException.

5.2.28 Method: validateCardAndCaptureFace

This function validates Emirates ID card with Validation Gateway (VG) service and performs liveness detection along with face capture.

Please note that this method is currently used only with Android SDK along with certified NFC Reader / Smart Card Reader plugin.

Syntax

```
public void validateCardAndCaptureFace(
    Object context,
    String requestId,
    int requestCode
) throws ToolkitException;
```

Parameters

context	Android Application context.
requestId	Please refer to “ID Card Toolkit - Programmer’s Reference” section #5 for the guidelines on generating the Request ID.
requestCode	The requestCode helps you to identify from which Intent you came back in onActivityResult .

Return

void

Callback

Callback will be coming in **onActivityResult**. **onActivityResult()** API is available on the Activity class on all API levels

Process of handling Callback:

8. Overwrite **onActivityResult** method in your activity.
9. Match requestCode with the sent requestCode in the validateCardAndCaptureFace API.
10. Match result code with “Activity.RESULT_OK”
11. Get result from **intent** using **getStringExtra** method with “MyFaceKey.INTENT_RESULT”
12. Match result with “MyFaceKey.RESULT_SUCCESS”.
13. Get captured face data from the intent using **getStringExtra** method with “MyFaceKey.IMAGE_BASE64”.
14. Can get the status message from the intent using **getStringExtra** method with “MyFaceKey.INTENT_MESSAGE” in case of failure.

Throws

ToolkitException

In case of error, the function throws ToolkitException.



5.2.29 Method: authenticateFaceOnServer

This function performs biometric face authentication with Validation Gateway (VG) service against the captured face image. Prior to calling this function, “validateCardAndCaptureFace” needs to be invoked for capturing face image.

Syntax

```
public ToolkitResponse authenticateFaceOnServer(
    String imageBase64) throws
    ToolkitException;
```

Parameters

image	Base64 face image data, captured using the previous call to “validateCardAndCaptureFace” method.
-------	--

Return

ToolkitResponse	Returns ToolkitResponse object which contains the biometric face authentication status.
-----------------	---

Throws

ToolkitException	In case of error, the function throws ToolkitException.
------------------	---

5.3 Class: SignatureValidator

This class provides a method to validate Toolkit XML Response received from various Toolkit API operations performed with Emirates ID Card.

5.3.1 Constructor: SignatureValidator

This method is used to set the attributes required to verify Toolkit XML Response with the corresponding certificate and its chain.

Syntax

```
public SignatureValidator (
    Byte [] cert,
    Byte [] certChain
);
```

Parameters

cert	Certificate data corresponding to the Response verification.
certChain	Issuer Certificate chain data corresponding to the certificateData. If null, Issuer Certificate chain validation is ignored.



5.3.2 Method: ValidateToolkitResponse

This method is used to validate Toolkit XML response.

Syntax

```
public void ValidateToolkitResponse (String VGResponse) throws
ToolkitException;
```

Parameters

VGResponse	Toolkit Response XML string
------------	-----------------------------

Return

void

Throws

ToolkitException	In case of error, the function throws ToolkitException.
	If the function succeeds, the attributes of the validated XML response can be retrieved through functions provided within this class.

6 Toolkit Advanced Digital Signature API Reference

6.1.1 Method: padesign

This method is used to digitally sign a PDF document in conformance with the frame work provided by PDF and ISO 32000-1 published by TS 102 778: Electronic Signatures and Infrastructures (ESI); PDF Advanced Electronic Signature Profiles;

Prior to calling this method application developers must invoke `PrepareRequest` successfully. The Request Handle returned by `PrepareRequest` method in the Toolkit Response XML is required to encode encodedPin data sent through `signingContext` of this method.

Syntax

```
public ToolkitResponse padesign(
    PadesSigningContext signingContext,
    String pdfFilePath,
    String signedPdfFilePath
) throws ToolkitException
```

Parameters

signingContext	Properties of the signature. Please refer to “ID Card Toolkit - Programmer’s Reference” section #9 for signing context attributes and its values.
----------------	---



Emirates ID Card PKI Pin which is part of signing context is encrypted and base64 encoded in accordance with the procedure described in the “*ID Card Toolkit - Programmer’s Reference*” section #6.

<code>pdfFilePath</code>	Path of the input document for digital signing
<code>signedPdfFilePath</code>	Path of the signed document

Return

<code>ToolkitResponse</code>	<p>If the function succeeds, returns <code>ToolkitResponse</code> object.</p> <p><code>xmlString</code> property of <code>ToolkitResponse</code> object returns XML response based on the response from Validation Gateway. Applications need to validate the XML response digital signature and Request ID to verify the authenticity of the response.</p> <p><code>responseStatus</code> property of <code>ToolkitResponse</code> object returns the <code>padesSign</code> status.</p>
------------------------------	---

Throws

<code>ToolkitException</code>	In case of error, the function throws <code>ToolkitException</code> .
-------------------------------	---

6.1.2 Method: `padesVerify`

This method is used to digitally verify a PDF in conformance with the frame work provided by PDF and ISO 32000-1 published by TS 102 778: Electronic Signatures and Infrastructures (ESI); PDF Advanced Electronic Signature Profiles;

Syntax

```
public String padesVerify(
    VerificationContext verificationContext,
    String signedPdfFilePath
) throws ToolkitException
```

Parameters

<code>verificationContext</code>	<p>Properties of the signature verification context.</p> <p>Please refer to “<i>ID Card Toolkit - Programmer’s Reference</i>” section #11 for signature verification context attributes and its values.</p>
----------------------------------	---



signedPdfFilePath

Path of the signed PDF document to be verified.

Return

String

Returns the verification report in XML format.

Throws

ToolkitException

In case of error, the function throws ToolkitException.

6.1.3 Method: xadesSign

This method is used to digitally sign a XML document in conformance with [XML-DSig recommendation](#) for advanced electronic signatures maintained and updated by W3C and ETSI (ETSI EN 319 132: Electronic Signatures and Infrastructures; XAdES digital signatures;) together.

Prior to calling this method application developers must invoke `PrepareRequest` successfully. The Request Handle returned by `PrepareRequest` method in the Toolkit Response XML is required to encode encodedPin data sent through `signingContext` of this method.

Syntax

```
public ToolkitResponse xadesSign(
    SigningContext signingContext,
    String xmlFilePath,
    String signedXmlFilePath
) throws ToolkitException
```

Parameters

signingContext

Properties of the signature. Please refer to “ID Card Toolkit - Programmer’s Reference” section #9 for signing context attributes and its values.

Emirates ID Card PKI Pin which is part of signing context is encrypted and base64 encoded in accordance with the procedure described in the “ID Card Toolkit - Programmer’s Reference” section #6.

xmlFilePath

Path of the input XML document for digital signing

`signedXmlFilePath`

Path of the document to store the signed document.

Signing mode	<code>signedXmlFilePath</code>
Detached	<code>null</code>
Enveloped/ Enveloping	Path of the document to store the signed document.

Return

`ToolkitResponse`

If the function succeeds, returns `ToolkitResponse` object.

`xmlString` property of `ToolkitResponse` object returns XML response based on the response from Validation Gateway. Applications need to validate the XML response digital signature and Request ID to verify the authenticity of the response.

`responseStatus` property of `ToolkitResponse` object returns the `xadesSign` status.

`detachedSignature` property of `ToolkitResponse` object returns Signature in case of Detached mode.

Signing mode	<code>detachedSignature</code>
Detached	Signature value
Enveloped/ Enveloping	EMPTY ARRAY

Throws

`ToolkitException`

In case of error, the function throws `ToolkitException`.

6.1.4 Method: `xadesVerify`

This method is used to digitally verify a XML document in conformance with [XML-DSig recommendation](#) for advanced electronic signatures maintained and updated by W3C and ETSI (ETSI EN 319 132: Electronic Signatures and Infrastructures; XAdES digital signatures) together.

Syntax

```
public String xadesVerify(
    VerificationContext verificationContext,
    String signedXmlFilePath
```



) throws ToolkitException

Parameters

verificationContext	Properties of the signature verification context. Please refer to “ID Card Toolkit - Programmer’s Reference” section #11 for signature verification context attributes and its values.						
signedXmlFilePath	Path of the signed XML document containing signature, to be verified. Path of the signed XML document to be verified.						
<table> <tr> <th>Signing Mode</th><th>signedXmlFilePath</th></tr> <tr> <td>Detached</td><td>Path of the XML document to be verified with provided signature</td></tr> <tr> <td>Enveloped/ Enveloping</td><td>Path of the signed XML document</td></tr> </table>		Signing Mode	signedXmlFilePath	Detached	Path of the XML document to be verified with provided signature	Enveloped/ Enveloping	Path of the signed XML document
Signing Mode	signedXmlFilePath						
Detached	Path of the XML document to be verified with provided signature						
Enveloped/ Enveloping	Path of the signed XML document						

Return

String	Returns the verification report in XML format.
--------	--

Throws

ToolkitException	In case of error, the function throws ToolkitException.
------------------	---

6.1.5 Method: xadesVerify

This method is used to digitally verify a detached XML signature in conformance with [XML-DSig recommendation](#) for advanced electronic signatures maintained and updated by W3C and ETSI (ETSI EN 319 132: Electronic Signatures and Infrastructures; XAdES digital signatures) together.

Syntax

```
public String xadesVerify(
    VerificationContext verificationContext,
    String xmlFilePath,
    byte[] signature
) throws ToolkitException
```

Parameters

verificationContext	Properties of the signature verification context. Please refer to “ID Card Toolkit -
---------------------	---



Programmer's Reference" section #11 for signature verification context attributes and its values.

xmlFilePath

Path of the signed XML document containing signature, to be verified.

Path of the signed XML document to be verified.

Signing Mode	signedXmlFilePath
Detached	Path of the XML document to be verified with provided signature
Enveloped/Enveloping	Path of the signed XML document

signature

Signature value.

Signing Mode	signature
Detached	Signature
Enveloped/Enveloping	null

Return

String

Returns the verification report in XML format.

Throws

ToolkitException

In case of error, the function throws ToolkitException.

6.1.6 Method: cadesSign

This method is used to digitally sign various kinds of transactions like purchase requisition, contracts or invoices with extension to [Cryptographic Message Syntax](#) (CMS) signed data making it suitable for [Advanced Electronic Signatures](#) in conformance with ETSI TS 101 733: Electronic Signatures and Infrastructures (ESI); CMS Advanced Electronic Signatures (CAES).

Prior to calling this method application developers must invoke `PrepareRequest` successfully. The Request Handle returned by `PrepareRequest` method in the Toolkit Response XML is required to encode `encodedPin` data sent through `signingContext` of this method.

This method works only with detached signing mode.

Syntax

```
public ToolkitResponse cadesSign(
    SigningContext signingContext,
    String inputFilePath
) throws ToolkitException
```




Parameters

<code>signingContext</code>	<p>Properties of the signature. Please refer to “<i>ID Card Toolkit - Programmer’s Reference</i>” section #9 for signing context attributes and its values.</p> <p>Emirates ID Card PKI Pin which is part of signing context is encrypted and base64 encoded in accordance with the procedure described in the “<i>ID Card Toolkit - Programmer’s Reference</i>” section #6.</p> <p>The Packaging mode in the signing context needs to be Detached.</p>
<code>inputFilePath</code>	Path of the input document for digital signing

Return

<code>ToolkitResponse</code>	<p>If the function succeeds, returns <code>ToolkitResponse</code> object.</p> <p><code>xmlString</code> property of <code>ToolkitResponse</code> object returns XML response based on the response from Validation Gateway. Applications need to validate the XML response digital signature and Request ID to verify the authenticity of the response.</p> <p><code>responseStatus</code> property of <code>ToolkitResponse</code> object returns the <code>adesSign</code> status.</p> <p><code>detachedSignature</code> property of <code>ToolkitResponse</code> object returns Signature in case of Detached mode.</p>
------------------------------	--

Throws

<code>ToolkitException</code>	In case of error, the function throws <code>ToolkitException</code> .
-------------------------------	---

6.1.7 Method: `adesVerify`

This method is used to digitally verify various kinds of transactions like purchase requisition, contracts or invoices with extension to [Cryptographic Message Syntax](#) (CMS) signed data making it suitable for [advanced electronic signatures](#) in conformance with ETSI TS 101 733: Electronic Signatures and Infrastructures (ESI); CMS Advanced Electronic Signatures (CAES).

Syntax



```
public String cadesVerify(  
    VerificationContext verificationContext,  
    String inputFilePath,  
    byte[] signature  
    ) throws ToolkitException
```

Parameters

verificationContext	Properties of the signature verification context. Please refer to “ <i>ID Card Toolkit - Programmer’s Reference</i> ” section #11 for signature verification context attributes and its values.
inputFilePath	Path of the document to be verified.
signature	Signature bytes for verification

Return

String	Returns the verification report in XML format.
--------	--

Throws

ToolkitException	This method throws ToolkitException.
------------------	--------------------------------------