EMV®
Secure Remote Commerce

Specification – JavaScript SDK

Version 1.3

December 2022

© 2019-22 EMVCo, LLC. All rights reserved. Reproduction, distribution and other use of this document is permitted only pursuant to the applicable agreement between the user and EMVCo found at www.emvco.com. EMV® is a registered trademark or trademark of EMVCo, LLC in the United States and other countries.

Legal Notice

The EMV® Specifications are provided “AS IS” without warranties of any kind, and EMVCo neither assumes nor accepts any liability for any errors or omissions contained in these Specifications. EMVCO DISCLAIMS ALL REPRESENTATIONS AND WARRANTIES, EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT, AS TO THESE SPECIFICATIONS.

EMVCo makes no representations or warranties with respect to intellectual property rights of any third parties in or in relation to the Specifications. EMVCo undertakes no responsibility to determine whether any implementation of the EMV® Specifications may violate, infringe, or otherwise exercise the patent, copyright, trademark, trade secret, know-how, or other intellectual property rights of third parties, and thus any person who implements any part of the EMV® Specifications should consult an intellectual property attorney before any such implementation.

Without limiting the foregoing, the Specifications may provide for the use of public key encryption and other technology, which may be the subject matter of patents in several countries. Any party seeking to implement these Specifications is solely responsible for determining whether its activities require a license to any such technology, including for patents on public key encryption technology. EMVCo shall not be liable under any theory for any party’s infringement of any intellectual property rights in connection with the EMV® Specifications.

© 2019-22 EMVCo, LLC. All rights reserved. Reproduction, distribution and other use of this document is permitted only pursuant to the applicable agreement between the user and EMVCo found at www.emvco.com. EMV® is a registered trademark or trademark of EMVCo, LLC in the United States and other countries.

Revision Log – Version 1.3

The following changes have been made to the document since the publication of version 1.2:

- Minor editorial changes throughout the document, with sections and tables renumbered where necessary
- Edits to account for the deprecation of the SRC Data Dictionary
- For each SDK method, the application errors in Section 2.x.3 have been sorted alphabetically
- ACCT_INACCESSIBLE application error has been moved from the individual tables in Section 2.x.3 to Section 2.15 Standard Errors
- The following new SDK methods have been added:
  - Add Billing Address (Section 2.12)
  - Authentication Methods Lookup (Section 2.13)
  - Authenticate (Section 2.14)
- Is Recognized (Section 2.3)
  - Optional `recognitionTokens` of type List<JWT> added to request parameters
  - Optional `recognitionDomainName` of type String added to response parameters
- Identity Lookup (Section 2.5)
  - Optional `lastUsedCardTimestamp` of type String added to response parameters
- Initiate Identity Validation (Section 2.6)
  - Optional `windowRef` of type Window added to request parameters
  - Optional `uriData` of type UriData added to response parameters
  - `maskedValidationChannel` in the response parameters changed from required to conditional
- Complete Identity Validation (Section 2.7)
  - Optional complianceSettings of type ComplianceSettings added to request parameters
  - Optional appInstance of type AppInstance added to request parameters
  - Optional maskedCard of type MaskedCard added to response parameters
  - Optional recognitionToken of type JWT added to response parameters
- Enrol Card16 (Section 2.8)

© 2019-22 EMVCo, LLC. All rights reserved. Reproduction, distribution and other use of this document is permitted only pursuant to the applicable agreement between the user and EMVCo found at www.emvco.com. EMV® is a registered trademark or trademark of EMVCo, LLC in the United States and other countries.

  - Optional encryptedCardholderData of type JWE<CardholderData> added to request parameters
  - Optional `complianceSettings` of type ComplianceSettings added to request parameters
  - Optional `recognitionToken` of type JWT added to response parameters
- Checkout (Section 2.9)
  - Optional `profileOptOut` of type Boolean added to request parameters
  - Optional `AuthenticationMethod` of type AuthenticationMethod added to request parameters
  - Conditional `complianceSettings` of type ComplianceSettings added to request parameters
  - Optional `appInstance` of type AppInstance added to request parameters
  - windowRef in the request parameters changed from optional to conditional
  - Conditional `recognitionToken` of type JWT added to response parameters
  - Optional `Unbindappinstance` of type Boolean deprecated, replaced by optional `bindingStatus` of type BindingStatus in response parameters
  - `idToken` in response parameters: conditionality changed
- Delete Card (Section 2.10)
  - Optional `idToken` of type JWT added to request parameters
- Unbind AppInstance (Section 2.11)
  - Optional `recognitionToken` of type List<JWT> added to request parameters

© 2019-22 EMVCo, LLC. All rights reserved. Reproduction, distribution and other use of this document is permitted only pursuant to the applicable agreement between the user and EMVCo found at www.emvco.com. EMV® is a registered trademark or trademark of EMVCo, LLC in the United States and other countries.

Contents

© 2019-22 EMVCo, LLC. All rights reserved. Reproduction, distribution and other use of this document is permitted only pursuant to the applicable agreement between the user and EMVCo found at www.emvco.com. EMV® is a registered trademark or trademark of EMVCo, LLC in the United States and other countries.

© 2019-22 EMVCo, LLC. All rights reserved. Reproduction, distribution and other use of this document is permitted only pursuant to the applicable agreement between the user and EMVCo found at www.emvco.com. EMV® is a registered trademark or trademark of EMVCo, LLC in the United States and other countries.

Tables

© 2019-22 EMVCo, LLC. All rights reserved. Reproduction, distribution and other use of this document is permitted only pursuant to the applicable agreement between the user and EMVCo found at www.emvco.com. EMV® is a registered trademark or trademark of EMVCo, LLC in the United States and other countries.

© 2019-22 EMVCo, LLC. All rights reserved. Reproduction, distribution and other use of this document is permitted only pursuant to the applicable agreement between the user and EMVCo found at www.emvco.com. EMV® is a registered trademark or trademark of EMVCo, LLC in the United States and other countries.

1 Introduction

Secure Remote Commerce (SRC) is an evolution of remote commerce that provides for secure and interoperable card acceptance established through a standard specification.

This document, the EMV® Secure Remote Commerce Specification – JavaScript SDK, (hereafter the "SRC JavaScript SDK Specification"), contains the definition of the SRC JavaScript SDK for the SRC Initiator. The SDK will be provided by each supported SRC System to be incorporated into the SRC Initiator integration software provided for Digital Payment Applications (e.g. SRC Initiator web application).

It is intended to be used in conjunction with the SRC Specifications (see Section 1.4.2 Published EMVCo Documents).

1.1 Scope

The SRC JavaScript SDK Specification defines a set of JavaScript methods that can be used by a web client to perform SRC operations. The specification defines the method names, request parameters, response attributes and possible errors. It does not define how an implementation of the SDK interacts with the SRC System. This is SRC System proprietary and out of scope.

1.2 Constraints

The SRC JavaScript SDK Specification is designed to work within the constraints described in the SRC Core Specification. In particular, the SRC JavaScript SDK Specification or any implementation of the SRC JavaScript SDK Specification is not intended to replace or interfere with any international, regional, national or local laws and regulations; those governing requirements supersede any industry standards.

1.3 Audience

This document is intended for use by SRC Systems and SRC System Participants.

© 2019-22 EMVCo, LLC. All rights reserved. Reproduction, distribution and other use of this document is permitted only pursuant to the applicable agreement between the user and EMVCo found at www.emvco.com. EMV® is a registered trademark or trademark of EMVCo, LLC in the United States and other countries.

1.4 References

The latest version of any reference, including all published amendments, shall apply unless a publication date is explicitly stated.

1.4.1 Normative References

The standards in Table 1.1 may be associated with SRC JavaScript SDK Specification.

Table 1.1: Normative References

Reference	Publication Name
ISO 3166	Country Codes — ISO 3166
ISO 4217	Currency Codes — ISO 4217
ISO/IEC 7812	Identification cards — Identification of issuers
RFC 7515	JSON Web Signature
RFC 7516	JSON Web Encryption
RFC 7519	JSON Web Token

1.4.2 Published EMVCo Documents

The documents in Table 1.2 are related to or are associated with SRC and are located at www.emvco.com.

© 2019-22 EMVCo, LLC. All rights reserved. Reproduction, distribution and other use of this document is permitted only pursuant to the applicable agreement between the user and EMVCo found at www.emvco.com. EMV® is a registered trademark or trademark of EMVCo, LLC in the United States and other countries.

Table 1.2: EMVCo References

Reference	Publication Name
SRC Core Specification	EMV® Secure Remote Commerce Specification
SRC Reproduction Requirements	EMV® Secure Remote Commerce (SRC): Click to Pay Icon Reproduction Requirements
SRC UI Guidelines and Requirements	EMV® Secure Remote Commerce Specification – User Interface Guidelines and Requirements
SRC API	EMV® Secure Remote Commerce Specification – API
SRC Version Management	EMV® Secure Remote Commerce Version Management for SRC API and SRC JavaScript SDK Specifications
SRC Use Cases	EMV® Secure Remote Commerce Use Cases

Collectively, the term SRC Specifications refers to:

- SRC Core Specification
- SRC Reproduction Requirements
- SRC UI Guidelines and Requirements
- SRC API
- SRC JavaScript SDK (this document)
- SRC Version Management

1.5 Definitions

For the definition of the terms used in the SRC JavaScript SDK Specification, refer to Table 1.3: Definitions in the SRC Core Specification. For definitions of data elements refer to the SRC API.

© 2019-22 EMVCo, LLC. All rights reserved. Reproduction, distribution and other use of this document is permitted only pursuant to the applicable agreement between the user and EMVCo found at www.emvco.com. EMV® is a registered trademark or trademark of EMVCo, LLC in the United States and other countries.

1.6 Notational Conventions

1.6.1 Abbreviations

For the definition of the abbreviations used in the SRC JavaScript SDK Specification, refer to section 1.9.1 Abbreviations in the SRC Core Specification.

1.6.2 Terminology and Conventions

For the definition of the terminology and conventions used in the SRC JavaScript SDK Specification, refer to section 1.9.2 Terminology and Conventions in the SRC Core Specification.

© 2019-22 EMVCo, LLC. All rights reserved. Reproduction, distribution and other use of this document is permitted only pursuant to the applicable agreement between the user and EMVCo found at www.emvco.com. EMV® is a registered trademark or trademark of EMVCo, LLC in the United States and other countries.

2 Web Client SDK

2.1 Data Elements

All primitive and composite data elements are as defined in the SRC API. The request parameters and response attributes (returned if processing is successful) are provided in a single JSON object.

In the request parameters and response attributes tables for each SDK method, the column headed R/C/O in each table refers to whether the parameter/attribute is required, conditional or optional. The following notation is used:

- R = Required – always present
- C = Conditional – present under certain conditions (as specified in the description)
- O = Optional – can be present

2.2 Initialize SRC SDK

This method initialises each SRC System's SDK in a common state. It must be called for each JavaScript SDK incorporated into the SRC Initiator integration software and before any other methods.

Table 2.2.1: Initialize SRC SDK Method

```
init({
      required String srcInitiatorId;
      conditional String srcDpaId;
      optional String srciTransactionId;
      optional DpaTransactionOptions dpaTransactionOptions;
      conditional DpaData dpaData;
      optional String version;
      optional AcceptanceChannelRelatedData
        acceptanceChannelRelatedData;
})
```

```
// Response - empty
```

© 2019-22 EMVCo, LLC. All rights reserved. Reproduction, distribution and other use of this document is permitted only pursuant to the applicable agreement between the user and EMVCo found at www.emvco.com. EMV® is a registered trademark or trademark of EMVCo, LLC in the United States and other countries.

2.2.1 Request Parameters

The request parameters are given in Table 2.2.2.

Table 2.2.2: Request Parameters

Name	R/C/O	Description
srcInitiatorId Type: String	R	The reference identifier generated by the SRC System during Onboarding of the SRC Initiator to the SRC System
srcDpaId Type: String	C	The reference identifier that may have been generated either by the SRC Initiator or by the SRC System. During DPA Registration with the SRC System (if it occurs) one, or both of these, value(s) will identify the DPA for subsequent initialisations **Conditionality**: Required when `dpaData` is not supplied. When both the `srcDpaId` and `dpaData` are supplied, then an SRC System can (as an implementation choice) choose how each should be used
srciTransactionId Type: String	O	Transaction-unique identifier assigned by the SRC Initiator
dpaTransactionOptions Type:DpaTransactionOptions	O	These options can be used to override transaction options for the DPA that were configured during the DPA Registration
dpaData Type: DpaData	C	Present when a DPA has not been registered with the SRC System or can be present in order to dynamically update previously registered `DPA Data` in the SRC System (e.g. presentation name) **Conditionality**: Required when `srcDpaId` is not supplied. When both `srcDpaId` and `dpaData` are supplied, then an SRC System can (as an implementation choice) choose how each should be used

© 2019-22 EMVCo, LLC. All rights reserved. Reproduction, distribution and other use of this document is permitted only pursuant to the applicable agreement between the user and EMVCo found at www.emvco.com. EMV® is a registered trademark or trademark of EMVCo, LLC in the United States and other countries.

Name	R/C/O	Description
version Type: String	O	SDK versioning
acceptanceChannelRelatedData Type: AcceptanceChannelRelatedData	O	Passes specific acceptance channel data. Refer to the SRC API Specification for details

2.2.2 Response Attributes

This method does not have any response attributes.

2.2.3 Application Errors

The application errors are given in Table 2.2.3.

Table 2.2.3: Application Errors

Reason Code	Description
DPA_ID_OR_DATA_MISSING	The `srcDpaId` and `dpaData` parameters are missing: at least one must be supplied
SRCI_ID_MISSING	The `srcInitiatorId` parameter is missing

2.3 Is Recognized

This method uses a Device Identity (derived from a First Party Token) to determine whether it is bound to an SRC Profile and, if so, returns a Federated ID Token. or client application.

If a Federated ID Token is returned, the SRC Initiator integration software may then provide this Federated ID Token in the getSrcProfile() method of the SDKs of the other SRC Systems.

© 2019-22 EMVCo, LLC. All rights reserved. Reproduction, distribution and other use of this document is permitted only pursuant to the applicable agreement between the user and EMVCo found at www.emvco.com. EMV® is a registered trademark or trademark of EMVCo, LLC in the United States and other countries.

Table 2.3.1: Is Recognized Method

```
isRecognized({
      optional List<JWT> recognitionTokens;
})
```

```
// Response
{
      required Boolean recognized;
      optional String recognitionDomainName;
      conditional List<JWT> idTokens;
}
```

2.3.1 Request Parameters

The request attributes are given in Table 2.3.2.

Table 2.3.2: Request Attributes

Name	R/C/O	Description
recognitionTokens Type: List<JWT>	O	List of explicit recognition tokens issued by this SRC System

2.3.2 Response Attributes

The response attributes are given in Table 2.3.3.

Table 2.3.3: Response Attributes

Name	R/C/O	Description
recognized Type: Boolean	R	Flag indicating whether the Consumer Device (e.g. browser or client application) is recognised by the SRC System
recognitionDomainName Type: String	O	A redirection to a domain that may facilitate recognition of the Consumer Device (e.g. browser or client application) by the SRC System

© 2019-22 EMVCo, LLC. All rights reserved. Reproduction, distribution and other use of this document is permitted only pursuant to the applicable agreement between the user and EMVCo found at www.emvco.com. EMV® is a registered trademark or trademark of EMVCo, LLC in the United States and other countries.

Name	R/C/O	Description
idTokens Type: List<JWT>	C	List of Federated ID Tokens identifying the primary Consumer Identity bound to the recognised SRC Profiles **Conditionality**: Required when the value of `recognized` is set to `true` (i.e. one or more SRC Profiles are recognised)

2.3.3 Application Errors

There are no additional application errors other than those given in the standard errors (see Table 2.15.1, Section 2.15 Standard Errors).

2.4 Get SRC Profile

This method takes a list of Federated ID Tokens and returns SRC Profile data to enable card selection. The SRC Initiator aggregates Federated ID Tokens received from multiple SRC Systems and provides these to individual SRC Systems in order to fetch a complete card list.

Note: The `authorization` data element will not be present in the `profiles` returned to the SRC Initiator.

Table 2.4.1: Get SRC Profile Method

```
getSrcProfile({
      optional List<JWT> idTokens;
})
```

```
// Response
{
      required List<SrcProfile> profiles;
      optional String srcCorrelationId;
}
```

2.4.1 Request Parameters

The request parameters are given in Table 2.4.2.

© 2019-22 EMVCo, LLC. All rights reserved. Reproduction, distribution and other use of this document is permitted only pursuant to the applicable agreement between the user and EMVCo found at www.emvco.com. EMV® is a registered trademark or trademark of EMVCo, LLC in the United States and other countries.

Table 2.4.2: Request Parameters

Name	R/C/O	Description
idTokens Type: List<JWT>	O	List of the Federated ID Tokens received from one or more SRC Systems

2.4.2 Response Attributes

The response attributes are given in Table 2.4.3.

Table 2.4.3: Response Attributes

Name	R/C/O	Description
profiles Type: List<SrcProfile>	R	List of SRC Profile(s) associated with each recognised Consumer Identity. If no SRC Profiles are recognised, then an empty list is returned
srcCorrelationId Type: String	O	Reference identifier returned by the SRC System.

2.4.3 Application Errors

There are no additional application errors other than those given in the standard errors (see Table 2.15.1, Section 2.15 Standard Errors).

2.5 Identity Lookup

This method checks whether a specified Consumer Identity (email address or mobile phone number) is known to the SRC System.

Table 2.5.1: Identity Lookup Method

```
identityLookup({

      required ConsumerIdentity consumerIdentity;

})
```

© 2019-22 EMVCo, LLC. All rights reserved. Reproduction, distribution and other use of this document is permitted only pursuant to the applicable agreement between the user and EMVCo found at www.emvco.com. EMV® is a registered trademark or trademark of EMVCo, LLC in the United States and other countries.

```
// Response
{
      required Boolean consumerPresent;
      optional List<IdentityValidationChannel>
        supportedValidationChannels;
      optional String lastUsedCardTimestamp;
}
```

2.5.1 Request Parameters

The request parameters are given in Table 2.5.2.

Table 2.5.2: Request Parameters

Name	R/C/O	Description
consumerIdentity Type: ConsumerIdentity	R	An email or phone number that will be used to determine whether an SRC Profile with this primary Consumer Identity is present

2.5.2 Response Attributes

The response attributes are given in Table 2.5.3.

Table 2.5.3: Response Attributes

Name	R/C/O	Description
consumerPresent Type: Boolean	R	Flag indicating whether an SRC Profile with the provided Consumer Identity is present
lastUsedCardTimestamp Type: String (Numeric)	O	Timestamp of the last used card in the SRC Profile associated with the provided Consumer Identity (UTC time in Unix epoch format)
supportedValidationChannels Type: List<IdentityValidationChannel>	O	List of channels that can be used to perform identity validation. If returned by the SRC System, these choices could be presented to the Consumer

© 2019-22 EMVCo, LLC. All rights reserved. Reproduction, distribution and other use of this document is permitted only pursuant to the applicable agreement between the user and EMVCo found at www.emvco.com. EMV® is a registered trademark or trademark of EMVCo, LLC in the United States and other countries.

2.5.3 Application Errors

The application errors are given in Table 2.5.4.

Table 2.5.4: Application Errors

Reason Code	Description
CONSUMER_ID_MISSING	The `consumerIdentity` parameter is missing
ID_FORMAT_UNSUPPORTED	Unsupported Consumer Identity type

2.6 Initiate Identity Validation

This method initiates a process to validate that the Consumer is in possession of, or has access to, the Consumer Identity claimed.

Table 2.6.1: Initiate Identity Validation Method

```
initiateIdentityValidation({

      optional String requestedValidationChannelId;
      optional Window windowRef;
})
```

```
// Response
{

      conditional String maskedValidationChannel;
      optional String validationMessage;
      optional List<IdentityValidationChannel>
        supportedValidationChannels;
      optional UriData uriData;
}
```

2.6.1 Request Parameters

The request parameters are given in Table 2.6.2.

© 2019-22 EMVCo, LLC. All rights reserved. Reproduction, distribution and other use of this document is permitted only pursuant to the applicable agreement between the user and EMVCo found at www.emvco.com. EMV® is a registered trademark or trademark of EMVCo, LLC in the United States and other countries.

Table 2.6.2: Request Parameters

Name	R/C/O	Description
requestedValidationChannelId Type: String	O	Identifier of the channel over which the identity validation should be initiated
windowRef Type: Window	O	A handle to a browser window which is used to open the SRC System facilitated authentication in embedded iframe

2.6.2 Response Attributes

The response attributes are given in Table 2.6.3.

Table 2.6.3: Response Attributes

Name	R/C/O	Description
maskedValidationChannel Type: String	C	Masked value of the channel (e.g. email/phone) that the SRC System used to deliver the validation data (e.g. OTP) Conditionality: Required when `windowRef` is not provided in the input
validationMessage Type: String	O	Message returned by the SRC System to provide a locale-specific advisory to the Consumer about the identity validation process
supportedValidationChannels Type: List<IdentityValidationChannel>	O	List of additional channels that are supported and can be used to perform identity validation. If returned by the SRC System, these choices may be presented to the Consumer
uriData Type: UriData	O	URI of the authentication mobile application used to facilitate out of band identity validation

2.6.3 Application Errors

The application errors are given in Table 2.6.4.

© 2019-22 EMVCo, LLC. All rights reserved. Reproduction, distribution and other use of this document is permitted only pursuant to the applicable agreement between the user and EMVCo found at www.emvco.com. EMV® is a registered trademark or trademark of EMVCo, LLC in the United States and other countries.

Table 2.6.4: Application Errors

Reason Code	Description
OTP_SEND_FAILED	The OTP could not be sent to the recipient
RETRIES_EXCEEDED	The limit for the number of retries exceeded

2.7 Complete Identity Validation

This method determines whether data, provided by the Consumer as part of a second step of an identity validation process, is valid. It can also be used to check whether an out-of-band service was successful.

Table 2.7.1: Complete Identity Validation Method

```
completeIdentityValidation({

      conditional String validationData;
      optional ComplianceSettings complianceSettings;
      optional AppInstance appInstance;
})
```

```
// Response
{

      required JWT idToken;
      optional MaskedCard maskedCard;
      optional JWT recognitionToken;
}
```

2.7.1 Request Parameters

The request parameters are given in Table 2.7.2.

© 2019-22 EMVCo, LLC. All rights reserved. Reproduction, distribution and other use of this document is permitted only pursuant to the applicable agreement between the user and EMVCo found at www.emvco.com. EMV® is a registered trademark or trademark of EMVCo, LLC in the United States and other countries.

Table 2.7.2: Request Parameters

Name	R/C/O	Description
validationData Type: String	C	The validation data (e.g. the OTP value) entered by the user **Conditionality**: Required when the content of the requested `identityValidationChannelType` is set to a value other than OUT_OF_BAND
complianceSettings Type: ComplianceSettings	O	The compliance settings specifying the consents provided by the Consumer for binding in the Remember Me scenario
appInstance Type: AppInstance;	O	Information for binding in the Remember Me use case

2.7.2 Response Attributes

The response attributes are given in Table 2.7.3.

Table 2.7.3: Response Attributes

Name	R/C/O	Description
idToken Type: JWT	R	The Federated ID Token returned following successful validation of the `validationData`
maskedCard Type: MaskedCard	O	The masked card representing the last used card
recognitionToken Type: JWT	O	The explicit recognition token issued by the SRC System

2.7.3 Application Errors

The application errors are given in Table 2.7.4.

© 2019-22 EMVCo, LLC. All rights reserved. Reproduction, distribution and other use of this document is permitted only pursuant to the applicable agreement between the user and EMVCo found at www.emvco.com. EMV® is a registered trademark or trademark of EMVCo, LLC in the United States and other countries.

Table 2.7.4: Application Errors

Reason Code	Description
CODE_EXPIRED	The `validationData` is expired
CODE_INVALID	The supplied `validationData` is invalid
RETRIES_EXCEEDED	The limit for the number of retries exceeded
VALDATA_MISSING	The `validationData` parameter is missing
VALIDATION_IN_PROGRESS	The requested `identityValidationChannelType` is set to OUT_OF_BAND and no result is available

2.8 Enrol Card

This method enrols a new PAN to the SRC System during checkout. The PAN may be enrolled to an existing / identified SRC Profile, or to a newly-created SRC Profile, or (in the case of guest checkout) may not be added to an SRC Profile at all.

Table 2.8.1: Enrol Method

```
enrollCard({

      optional String srciTransactionId;
      required JWE encryptedCard;
      optional JSONObject threeDsInputData;
      conditional JWT idToken;
      optional JWE<Consumer> encryptedConsumer;
      optional JWE<CardholderData> encryptedCardholderData;
      optional JSONObject srcTokenRequestData;
      optional AssuranceData assuranceData;
      optional ComplianceSettings complianceSettings;
})
```

```
// Response
{
```

© 2019-22 EMVCo, LLC. All rights reserved. Reproduction, distribution and other use of this document is permitted only pursuant to the applicable agreement between the user and EMVCo found at www.emvco.com. EMV® is a registered trademark or trademark of EMVCo, LLC in the United States and other countries.

```
      required MaskedCard maskedCard;
      optional String srcCorrelationId;
      optional JWT recognitionToken;
}
```

2.8.1 Request Parameters

The request parameters are given in Table 2.8.2.

Table 2.8.2: Request Parameters

Name	R/C/O	Description
srciTransactionId Type: String	O	Transaction-unique identifier assigned by the SRC Initiator
encryptedCard Type: JWE<Card>	R	The card being enrolled with the SRC System. Encrypted using a public key of the SRC System to which the card is being enrolled
threeDsInputData Type: JSONObject	O	Merchant provided 3DS data
idToken Type: JWT	C	Federated ID Token used to check whether an SRC Profile exists and, if not, used by the DCF to provide the Consumer with hints to use a Consumer Identity that is consistent across SRC Systems **Conditionality:** Required when it is determined that the Consumer has previously enrolled with, and been recognised by, another SRC System
encryptedConsumer Type: JWE<Consumer>	O	Consumer information related to the card being enrolled. Encrypted using the public key of the SRC System to which the card is being enrolled

© 2019-22 EMVCo, LLC. All rights reserved. Reproduction, distribution and other use of this document is permitted only pursuant to the applicable agreement between the user and EMVCo found at www.emvco.com. EMV® is a registered trademark or trademark of EMVCo, LLC in the United States and other countries.

Name	R/C/O	Description
encryptedCardholderData Type: JWE<CardholderData>	O	Cardholder information related to the card being enrolled. Encrypted using the public key of the SRC System to which the card is being enrolled
srcTokenRequestData Type: JSONObject	O	SRC System-specific data (provided by the merchant) to support a Token Request
complianceSettings Type: ComplianceSettings	O	Indication that compliance consent has been collected from the Consumer when the enrolled card is added to an SRC Profile
assuranceData Type: AssuranceData	O	Assurance data related to the enrolment

2.8.2 Response Attributes

The response attributes are given in Table 2.8.3.

Table 2.8.3: Response Attributes

Name	R/C/O	Description
maskedCard Type: MaskedCard	R	Masked data related to the enrolled Digital Card
srcCorrelationId Type: String	O	Reference identifier returned by the SRC System
recognitionToken Type: JWT	O	The explicit recognition token issued by the SRC System

2.8.3 Application Errors

The application errors are given in Table 2.8.4.

Table 2.8.4: Application Errors

Reason Code	Description
AUTH_INVALID	Invalid `idToken`

© 2019-22 EMVCo, LLC. All rights reserved. Reproduction, distribution and other use of this document is permitted only pursuant to the applicable agreement between the user and EMVCo found at www.emvco.com. EMV® is a registered trademark or trademark of EMVCo, LLC in the United States and other countries.

Reason Code	Description
CARD_ADD_FAILED	Unable to add the provided card
CARD_EXP_INVALID	Invalid card expiry date
CARD_INVALID	Invalid `primaryAccountNumber`
CARD_MISSING	The `encryptedCard` parameter is missing
CARD_SECURITY_CODE_MISSING	Card security code must be supplied in the `encryptedCard` parameter

2.9 Checkout

This method performs checkout using the specified Digital Card or PAN. If successful, the response contains summary checkout information and, conditionally, an encrypted payload signed by the SRC System containing PCI and/or PII data.

Table 2.9.1: Checkout Method

```
checkout({
      optional String srciTransactionId;
      conditional String srcCorrelationId;
      conditional String srcDigitalCardId;
      conditional JWE<Card> encryptedCard;
      optional JWE<Consumer> encryptedConsumer;
      conditional JWT idToken;
      conditional DpaTransactionOptions dpaTransactionOptions;
      optional PayloadTypeIndicator payloadTypeIndicatorCheckout;
      optional String recipientIdCheckout;
      optional PayloadTypeIndicator payloadTypeIndicatorPayload;
      optional String recipientIdPayload;
      optional AssuranceData assuranceData;
      optional SrciActionCode srciActionCode;
      ~~optional~~conditional Window windowRef; Changed from Optional
         to Conditional
```

© 2019-22 EMVCo, LLC. All rights reserved. Reproduction, distribution and other use of this document is permitted only pursuant to the applicable agreement between the user and EMVCo found at www.emvco.com. EMV® is a registered trademark or trademark of EMVCo, LLC in the United States and other countries.

```
      optional AcceptanceChannelRelatedData
        acceptanceChannelRelatedData;
      optional Boolean profileOptOut;
      optional AuthenticationMethod authenticationMethod;
      conditional ComplianceSettings complianceSettings;
      optional AppInstance appInstance;
})
```

```
// Response
{
      required DcfActionCode dcfActionCode;
      conditional JWT idToken;
      conditional JWT recognitionToken;
      conditional JWS<CheckoutPayloadResponse> checkoutResponse;
      optional Boolean unbindAppInstance;DEPRECATED
      Replaced by
      optional BindingStatus bindingStatus;
}
```

2.9.1 Request Parameters

The request parameters are given in Table 2.9.2.

Table 2.9.2: Request Parameters

Name	R/C/O	Description
srciTransactionId Type: String	O	A transaction-unique identifier assigned by the SRC Initiator
srcCorrelationId Type: String	C	Reference identifier returned by the SRC System **Conditionality:** Required when the SRC System returned it in prior calls within the same transaction
srcDigitalCardId Type: String	C	A reference identifier of the card to be used for checkout

© 2019-22 EMVCo, LLC. All rights reserved. Reproduction, distribution and other use of this document is permitted only pursuant to the applicable agreement between the user and EMVCo found at www.emvco.com. EMV® is a registered trademark or trademark of EMVCo, LLC in the United States and other countries.

Name	R/C/O	Description
		Conditionality: Required for checkout when a Digital Card is selected from a Candidate List
encryptedCard Type: JWE<Card>	C	The card being enrolled with the SRC System. Encrypted using a public key of SRC System to which the card is being enrolled **Conditionality**: Required for a combined flow where this card is being enrolled during checkout
encryptedConsumer Type: JWE<Consumer>	O	Consumer information related to the card being enrolled. Encrypted using the public key of the SRC System to which the card is being enrolled
idToken Type: JWT	C	Federated ID Token used to check whether an SRC Profile exists and, if not, used by the DCF to provide the Consumer with hints to use a Consumer Identity that is consistent across SRC Systems **Conditionality:** Required when it is determined that the Consumer has previously enrolled with, and been recognised by, another SRC System
dpaTransactionOptions Type: DpaTransactionOptions	C	These options can be used to override transaction options for the DPA that were configured during the DPA Registration **Conditionality**: Required when not provided earlier in the init() method call and the DPA has not been Registered with the SRC System
payloadTypeIndicatorCheckout Type: PayloadTypeIndicator	O	Indicates the scope of the encrypted payload, if any, to be provided in the `checkoutResponse` attribute in the response to this method

© 2019-22 EMVCo, LLC. All rights reserved. Reproduction, distribution and other use of this document is permitted only pursuant to the applicable agreement between the user and EMVCo found at www.emvco.com. EMV® is a registered trademark or trademark of EMVCo, LLC in the United States and other countries.

Name	R/C/O	Description
recipientIdCheckout Type: String	O	Identifier of the ultimate recipient of the encrypted payload returned in the `checkoutResponse` attribute in the response to this method. Used by the SRC System to determine which key is used for encryption of the payload
payloadTypeIndicatorPayload Type: PayloadTypeIndicator	O	Indicates the type of payload to be provided when the payload is returned in a subsequent interaction (e.g. when the Get Payload operation is called)
recipientIdPayload Type: String	O	Identifier of the recipient that will subsequently request the encrypted payload. Used by the SRC System to identify which key to use for encryption of the payload
assuranceData Type: AssuranceData	O	Assurance data supplied to support risk management
srciActionCode Type: SrciActionCode	C	A code indicating a non-typical behaviour on the SRC Initiator that should be addressed by the DCF **Conditionality**: Required when non-typical behaviour has occurred
windowRef Type: Window	~~O~~C	A reference to a browsing context, e.g. iframe or new pop-up window ***Changed from Optional to Conditional*** **Conditionality**: Required when the DCF needs to open a custom URI or the SRC System facilitates authentication
acceptanceChannelRelatedData Type: AcceptanceChannelRelatedData	O	This field is used to pass along specific acceptance channel related data Refer to the SRC API Specification for details
profileOptOut Type: Boolean	O	If set to TRUE, the SRC Profile should not be created. Default value is FALSE

© 2019-22 EMVCo, LLC. All rights reserved. Reproduction, distribution and other use of this document is permitted only pursuant to the applicable agreement between the user and EMVCo found at www.emvco.com. EMV® is a registered trademark or trademark of EMVCo, LLC in the United States and other countries.

Name	R/C/O	Description
authenticationMethod Type: AuthenticationMethod	O	The preferred authentication method as selected by the SRC Initiator
complianceSettings Type: ComplianceSettings	C	The compliance settings specifying the consents provided by the Consumer **Conditionality**: Required when `encryptedConsumer` is supplied and `profileOptOut` is FALSE
appInstance Type: AppInstance;	O	Information for binding in the Remember Me use case

2.9.2 Response Attributes

The response attributes are given in Table 2.9.3.

Table 2.9.3: Response Attributes

Name	R/C/O	Description
dcfActionCode Type: DcfActionCode	R	A code indicating the behaviour to be handled by the SRC Initiator
idToken Type: JWT	C	A Federated ID Token related to the current SRC Profile ~~**Conditionality**: Required when:~~ ~~• Requested by the DCF during its processing of the checkout session (e.g. DCF requesting identity validation be performed by the SRC System); *or*~~ ~~• The `unbindAppInstance` attribute returned in this response is set to `true`; *or*~~ ~~• A new / updated `idToken` is generated~~ ***Conditionality changed to:*** **Conditionality**: Required when:

© 2019-22 EMVCo, LLC. All rights reserved. Reproduction, distribution and other use of this document is permitted only pursuant to the applicable agreement between the user and EMVCo found at www.emvco.com. EMV® is a registered trademark or trademark of EMVCo, LLC in the United States and other countries.

Name	R/C/O	Description
		• Requested by the DCF during its processing of the checkout session (e.g. DCF requesting identity validation be performed by the SRC System); *or* • `bindingStatus` is set to BIND or UNBIND; *or* • A new / updated `idToken` is generated
recognitionToken Type: JWT	C	Long-lived First Party Token representing a device or app bound to the SRC Profile **Conditionality:** Required when the Consumer consented to be remembered (`complianceResource` object of ComplianceType REMEMBERME present in `complianceSettings`)
checkoutResponse Type: JWS <CheckoutPayloadResponse>	C	Signed structure **Conditionality**: Required when the `dcfActionCode` is set to COMPLETE or PENDING_AUTHENTICATION `encryptedPayload` or `encryptedSignedPayload` will be not present within the `checkoutResponse` (see SRC API) when: • `payloadTypeIndicatorCheckout` is set to SUMMARY *or* • `dcfActionCode` is set to PENDING_AUTHENTICATION
~~**unbindAppInstance**~~ ~~Type: Boolean~~ **DEPRECATED**	~~O~~	~~Flag indicating whether the Consumer has chosen to be 'un-remembered' from the Consumer Device. The default value is assumed to be `false` if this field is not provided.~~ ~~If this attribute is set to `true`, then the SRC Initiator shall proceed to call the unbindAppInstance() method for all available SDKs with the `idToken` attribute returned in this response~~

© 2019-22 EMVCo, LLC. All rights reserved. Reproduction, distribution and other use of this document is permitted only pursuant to the applicable agreement between the user and EMVCo found at www.emvco.com. EMV® is a registered trademark or trademark of EMVCo, LLC in the United States and other countries.

Name	R/C/O	Description
Replaced by **bindingStatus** Type: BindingStatus	O	Status of the binding/unbinding request. The value BIND indicates that the Consumer has chosen to be “remembered” on the Consumer Device

© 2019-22 EMVCo, LLC. All rights reserved. Reproduction, distribution and other use of this document is permitted only pursuant to the applicable agreement between the user and EMVCo found at www.emvco.com. EMV® is a registered trademark or trademark of EMVCo, LLC in the United States and other countries.

2.9.3 Application Errors

The application errors are given in Table 2.9.4.

Table 2.9.4: Application Errors

Reason Code	Description
AUTH_INVALID	Invalid `idToken`
BILLING_ADDRESS_REQUIRED	The billing address must be provided
CARD_ADD_FAILED	Unable to add the card when combined flow (enrol and checkout) is occurring
CARD_EXP_INVALID	Invalid card expiry date
CARD_INVALID	Invalid `primaryAccountNumber` when combined flow (enrol and checkout) is occurring
CARD_MISSING	The `srcDigitalCardId` or `encryptedCard` parameter is required but is missing
CARD_NOT_RECOGNIZED	The provided `srcDigitalCardId` is not recognised
CARD_SECURITY_CODE_MISSING	Card Security Code (CSC) must be supplied in the `encryptedCard` parameter when a combined flow (enrol and checkout) is occurring
MERCHANT_CATEGORY_CODE_MISSING	The merchant category code must be provided for 3DS_ON_BEHALF
MERCHANT_COUNTRY_CODE_MISSING	The merchant country code must be provided for 3DS_ON_BEHALF
MERCHANT_DATA_INVALID	Merchant data is invalid
TERMS_AND_CONDITIONS_NOT_ACCEPTED	Terms and Conditions not accepted
UNABLE_TO_CONNECT	Unable to connect to / Launch DCF

© 2019-22 EMVCo, LLC. All rights reserved. Reproduction, distribution and other use of this document is permitted only pursuant to the applicable agreement between the user and EMVCo found at www.emvco.com. EMV® is a registered trademark or trademark of EMVCo, LLC in the United States and other countries.

2.10 Delete Card

This method deletes a Digital Card from an SRC Profile.

Table 2.10.1: Delete Card Method

```
deleteCard({
      required String srcDigitalCardId;
      optional JWT idToken;
})
```

```
// Response
{
      optional String srcCorrelationId;
}
```

2.10.1 Request Parameters

The request parameters are given in Table 2.10.2.

Table 2.10.2: Request Parameters

Name	R/C/O	Description
srcDigitalCardId Type: String	R	A reference identifier of the card to be deleted
idToken Type: JWT	O	May be provided for additional security when the Consumer has previously enrolled with, and been recognised by, another SRC System

2.10.2 Response Attributes

The response parameters are given in Table 2.10.3.

© 2019-22 EMVCo, LLC. All rights reserved. Reproduction, distribution and other use of this document is permitted only pursuant to the applicable agreement between the user and EMVCo found at www.emvco.com. EMV® is a registered trademark or trademark of EMVCo, LLC in the United States and other countries.

Table 2.10.3: Response Attributes

Name	R/C/O	Description
srcCorrelationId Type: String	O	Reference identifier returned by the SRC System

2.10.3 Application Errors

The application errors are given in Table 2.10.4.

Table 2.10.4: Application Errors

Reason Code	Description
AUTH_INVALID	The client does not have authorisation to perform the operation
CARDID_MISSING	The `srcDigitalCardId` parameter is missing
CARD_NOT_RECOGNIZED	The provided `srcDigitalCardId` is not recognised

2.11 Unbind AppInstance

This method unbinds a Device Identity (an application instance) from an SRC Profile.

Table 2.11.1: Unbind AppInstance Method

```
unbindAppInstance({

      required List<JWT> idTokens;

      optional List<JWT> recognitionToken;

})
```

© 2019-22 EMVCo, LLC. All rights reserved. Reproduction, distribution and other use of this document is permitted only pursuant to the applicable agreement between the user and EMVCo found at www.emvco.com. EMV® is a registered trademark or trademark of EMVCo, LLC in the United States and other countries.

```
// Response
{
      optional String srcCorrelationId;
}
```

2.11.1 Request Parameters

The request parameters are given in Table 2.11.2.

Table 2.11.2: Request Parameters

Name	R/C/O	Description
idTokens Type: List<JWT>	R	Each Federated ID Token indicates an SRC Profile(s) from which the Device Identity should be unbound
recognitionToken Type: List<JWT>	O	The explicit recognition token issued by the SRC System

2.11.2 Response Attributes

The response parameters are given in Table 2.11.3.

Table 2.11.3: Response Attributes

Name	R/C/O	Description
srcCorrelationId Type: String	O	Reference identifier returned by the SRC System

2.11.3 Application Errors

The application errors are given in Table 2.11.4.

Table 2.11.4: Application Errors

Reason Code	Description
AUTH_INVALID	Invalid `idToken`

© 2019-22 EMVCo, LLC. All rights reserved. Reproduction, distribution and other use of this document is permitted only pursuant to the applicable agreement between the user and EMVCo found at www.emvco.com. EMV® is a registered trademark or trademark of EMVCo, LLC in the United States and other countries.

Reason Code	Description
AUTH_MISSING	Missing `idTokens` list, or empty `idTokens` list provided

2.12 Add Billing Address

This method adds a billing address for the given Digital Card or updates an existing billing address.

Table 2.12.1: Add Billing Address Method

```
addBillingAddress({

      required String srcDigitalCardId;
      required Address billingAddress;
      optional JWT idToken;
})
```

```
// Response
{

      required MaskedCard maskedCard;
}
```

2.12.1 Request Parameters

The request parameters are given in Table 2.12.2.

Table 2.12.2: Request Parameters

Name	R/C/O	Description
srcDigitalCardId Type: String	R	An identifier of the card that the billing address should be set for.
billingAddress Type: Address	R	A component containing the billing address details to be added to the card.

© 2019-22 EMVCo, LLC. All rights reserved. Reproduction, distribution and other use of this document is permitted only pursuant to the applicable agreement between the user and EMVCo found at www.emvco.com. EMV® is a registered trademark or trademark of EMVCo, LLC in the United States and other countries.

Name	R/C/O	Description
idToken Type: JWT	O	May be provided when it is determined that the Consumer has previously enrolled with, and been recognised by, another SRC System

2.12.2 Response Attributes

The response parameters are given in Table 2.12.3.

Table 2.12.3: Response Attributes

Name	R/C/O	Description
maskedCard Type: MaskedCard	R	Masked card containing the new billing address.

2.12.3 Application Errors

The application errors are given in Table 2.12.4.

Table 2.12.4: Application Errors

Reason Code	Description
AUTH_INVALID	Invalid `idToken`
AUTH_MISSING	Missing `idTokens` list, or empty `idTokens` list provided
BILLING_ADDRESS_MISSING	The `billingAdress` is missing
CARD_NOT_RECOGNIZED	The provided `srcDigitalCardId` is not recognised
CARDID_MISSING	The `srcDigitalCardId` parameter is missing

2.13 Authentication Methods Lookup

This method obtains a proposed list of authentication methods relevant to the criteria specified by the client.

© 2019-22 EMVCo, LLC. All rights reserved. Reproduction, distribution and other use of this document is permitted only pursuant to the applicable agreement between the user and EMVCo found at www.emvco.com. EMV® is a registered trademark or trademark of EMVCo, LLC in the United States and other countries.

Table 2.13.1: Authentication Methods Lookup Method

```
authenticationMethodsLookup({

      required AccountReference accountReference;
      required AuthenticationContext authenticationContext;
      optional String srcCorrelationId;
      optional String srciTransactionId;
})
```

```
// Response
{

      required List<AuthenticationMethod> authenticationMethods;
      optional AssuranceData assuranceData;
      optional String srcCorrelationId;
      optional String srciTransactionId;
}
```

2.13.1 Request Parameters

The request parameters are given in Table 2.13.2.

Table 2.13.2: Request Parameters

Name	R/C/O	Description
accountReference Type: AccountReference	R	Must be one of either: • `srcDigitalCardId`; *or* • `consumerIdentity` (e.g. an email address or a mobile phone number)
authenticationContext Type: AuthenticationContext	R	Data provided as per context of the authentication (e.g. `transactionAmount`, `dpaData`, `srcDpaId`)
srcCorrelationId Type: String	O	Reference identifier returned by the SRC System

© 2019-22 EMVCo, LLC. All rights reserved. Reproduction, distribution and other use of this document is permitted only pursuant to the applicable agreement between the user and EMVCo found at www.emvco.com. EMV® is a registered trademark or trademark of EMVCo, LLC in the United States and other countries.

Name	R/C/O	Description
srciTransactionId Type: String	O	Transaction-unique identifier assigned by the SRC Initiator

2.13.2 Response Attributes

The response parameters are given in Table 2.13.3.

Table 2.13.3: Response Attributes

Name	R/C/O	Description
authenticationMethods Type: List<AuthenticationMethod>	R	List of supported authentication methods
assuranceData Type: AssuranceData	O	Assurance data supplied to support risk management
srcCorrelationId Type: String	O	Reference identifier returned by the SRC System
srciTransactionId Type: String	O	Transaction-unique identifier assigned by the SRC Initiator

2.13.3 Application Errors

The application errors are given in Table 2.13.4.

Table 2.13.4: Application Errors

Reason Code	Description
ACCT_REF_FORMAT_UNSUPPORTED	Unsupported `accountReference`
ACCT_REF_MISSING	The `accountReference` parameter is missing

2.14 Authenticate

This method can be used to:

© 2019-22 EMVCo, LLC. All rights reserved. Reproduction, distribution and other use of this document is permitted only pursuant to the applicable agreement between the user and EMVCo found at www.emvco.com. EMV® is a registered trademark or trademark of EMVCo, LLC in the United States and other countries.

- Initiate an authentication based on specified input criteria
- Complete an in-band validation, passing validation data for assessment, e.g. OTP value
- Check progress / status on an on-going out-of-band validation, where validation occurs on another channel

Table 2.14.1: Authenticate Method

```
authenticate({

      conditional AccountReference accountReference;
      conditional AuthenticationContext authenticationContext;
      required AuthenticationMethod authenticationMethod;
      optional Window windowRef;
      optional String srcCorrelationId;
      optional String srciTransactionId;
})
```

```
// Response
{

      conditional AuthenticationResult authenticationResult;
      required AuthenticationStatus authenticationStatus;
      conditional AssuranceData assuranceData;
      conditional JSONObject methodAttributes;
      optional String srcCorrelationId;
      optional String srciTransactionId;
}
```

2.14.1 Request Parameters

The request parameters are given in Table 2.14.2.

© 2019-22 EMVCo, LLC. All rights reserved. Reproduction, distribution and other use of this document is permitted only pursuant to the applicable agreement between the user and EMVCo found at www.emvco.com. EMV® is a registered trademark or trademark of EMVCo, LLC in the United States and other countries.

Table 2.14.2: Request Parameters

Name	R/C/O	Description
accountReference Type: AccountReference	C	Must be one of either: • `srcDigitalCardId`; *or* • `consumerIdentity` (e.g. an email address or a mobile phone number) **Conditionality**: Required when not available from a previous API call
authenticationContext Type: AuthenticationContext	C	Reasons for authentication such as for an enrolment or a transaction **Conditionality**: Required when not available from a previous API call
authenticationMethod Type: AuthenticationMethod	R	Contains `authenticationMethodType` selected and relevant `methodAttributes` (refer to the SRC API Specification for details)
windowRef Type: Window	O	A reference to a browsing context, e.g. iframe or new pop-up window used to facilitate the DCF opening a custom URI or SRC System facilitating the Cardholder authentication
srcCorrelationId Type: String	O	Reference identifier returned by the SRC System
srciTransactionId Type: String	O	Transaction-unique identifier assigned by the SRC Initiator

2.14.2 Response Attributes

The response parameters are given in Table 2.14.3.

Table 2.14.3: Response Attributes

Name	R/C/O	Description
authenticationResult Type: AuthenticationResult	C	**Conditionality**: Required when `authenticationStatus` is COMPLETE

© 2019-22 EMVCo, LLC. All rights reserved. Reproduction, distribution and other use of this document is permitted only pursuant to the applicable agreement between the user and EMVCo found at www.emvco.com. EMV® is a registered trademark or trademark of EMVCo, LLC in the United States and other countries.

Name	R/C/O	Description
authenticationStatus Type: AuthenticationStatus	R	Current status of the initiated authentication event
assuranceData Type: AssuranceData	C	**Conditionality**: Required when `authenticationStatus` is COMPLETE
methodAttributes Type: JSONObject	C	Any relevant attributes supplied by the SRC System **Conditionality**: Required as specified in the SRC API
srcCorrelationId Type: String	O	Reference identifier returned by the SRC System
srciTransactionId Type: String	O	Transaction-unique identifier assigned by the SRC Initiator

2.14.3 Application Errors

The application errors are given in Table 2.14.4.

Table 2.14.4: Application Errors

Reason Code	Description
3DS_INPUT_DATA_MISSING	The `threeDsInputData` is missing or insufficient
ACCT_REF_FORMAT_UNSUPPORTED	Unsupported `accountReference`
ACCT_REF_MISSING	The `accountReference` parameter is missing
AUTHENTICATION_METHOD_NOT_SUPPORTED	The supplied authentication method doesn't match the authentication context
CVC_DATA_MISSING	The CSC data is missing
OTP_SEND_FAILED	The OTP could not be sent to the recipient
RETRIES_EXCEEDED	The limit for the number of retries exceeded
VAL_DATA_MISSING	The `validationData` parameter is missing

© 2019-22 EMVCo, LLC. All rights reserved. Reproduction, distribution and other use of this document is permitted only pursuant to the applicable agreement between the user and EMVCo found at www.emvco.com. EMV® is a registered trademark or trademark of EMVCo, LLC in the United States and other countries.

Reason Code	Description
VAL_DATA_EXPIRED	The `validationData` is expired
VAL_DATA_INVALID	The supplied `validationData` is invalid

2.15 Standard Errors

SDK method errors can be application errors or standard errors.

- Standard errors can be returned by any SDK method and should be handled in a common way. They are described in Table 2.15.1
- Application errors that are only returned for specific SDK methods are described in the application errors section for the specific SDK method

Table 2.15.1: Standard Errors

Reason Code	Description
ACCT_INACCESSIBLE	The SRC Profile exists but is not currently accessible (e.g. is locked)
AUTH_ERROR	The server cannot perform the authentication necessary to process the request
BLOCKED	The method is blocked due to security reasons. Invocation of additional SDK methods may be blocked for the session and card
INVALID_PARAMETER	The value provided for one or more request parameters is considered invalid. This error is also generated in case of a missing, required, request parameter. Notes: • Whenever possible client-side validation of request parameters should be performed to avoid a round trip to the server. Simple validation constraints are documented as part of the SRC API • If the content of the request parameter is dependent on Consumer input, prompt the user to enter a value or enter an appropriately formatted value

© 2019-22 EMVCo, LLC. All rights reserved. Reproduction, distribution and other use of this document is permitted only pursuant to the applicable agreement between the user and EMVCo found at www.emvco.com. EMV® is a registered trademark or trademark of EMVCo, LLC in the United States and other countries.

Reason Code	Description
INVALID_REQUEST	The server is not able to adequately parse the request. Usually occurs when some request parameter is expected to be in a particular format but is not. Examples: • base64 decoding failed • The field is not in a particular format. The message field may provide additional clarification of what part/parameter of the request is considered incorrect
NOT_FOUND	The requested resource/business entity does not exist. The resource might also be hidden for security reasons
REQUEST_TIMEOUT	Request timeout
SERVICE_ERROR	Unexpected behaviour on the server caused the error. Either show a generic message or retry the same request again (it might succeed)
UNKNOWN_ERROR	Unknown error

© 2019-22 EMVCo, LLC. All rights reserved. Reproduction, distribution and other use of this document is permitted only pursuant to the applicable agreement between the user and EMVCo found at www.emvco.com. EMV® is a registered trademark or trademark of EMVCo, LLC in the United States and other countries.

***** END OF DOCUMENT *****

© 2019-22 EMVCo, LLC. All rights reserved. Reproduction, distribution and other use of this document is permitted only pursuant to the applicable agreement between the user and EMVCo found at www.emvco.com. EMV® is a registered trademark or trademark of EMVCo, LLC in the United States and other countries.