American Express

Click to Pay

Technical Specification Guide -Version 1.1.1

SDK and Server-Side API Specifications

Legal Notices

This document contains sensitive, confidential and trade secret information and may not be disclosed to third parties without the prior written consent of American Express Travel Related Services Company, Inc.

The policies, procedures, and rules in this specification are subject to change from time to time by American Express Global Network Services.

© 2021 American Express Travel Related Services Co., Inc.

To the maximum extent permitted by law, American Express does not make and hereby disclaims any and all representations, warranties, and liabilities, whether express or implied, or arising by law or from a course of dealing or usage of trade, including implied warranties of merchantability or fitness for a particular purpose or any warranty of title or non-infringement. Each Participant must comply with laws and regulations applicable to the subject matter of this document. These laws and regulations can differ from country to country, and each Participant is solely responsible for being aware and adhering to them in all countries where applicable.

All Rights Reserved.
All trademarks used herein are the property of their respective owners.

Table of Contents

©2021 AMERICAN EXPRESS. ALL RIGHTS RESERVED.

©2021 AMERICAN EXPRESS. ALL RIGHTS RESERVED.

Document History

Release	Date (DD/MM/YY)	Author	Modifications
V1.1.1	06/07/2021	Chenna Kesava Maduru	Added ThreeDsOutputData to Payload
V1.1.1	07/15/2021	Manasa Sreevani Channagiri	Updating the dynamicDataType as “NONE” in case of Safe key payment payload (skPA payload)
V1.1.1	07/30/2021	Manasa Sreevani Channagiri	Added NONE also as the valid dynamicDataType. Updated that, the dynamicDataType DYNAMIC_CARD_SECURITY_CODE will be sent for both TOKEN and PAN transactions. Updated the dynamicDataExpiration description. Updated the dynamicDataValue description. Removed CARD_APPLICATION_CRYPTOGRAM_LONG_FORM from the valid dynamicDataTypes.

©2021 AMERICAN EXPRESS. ALL RIGHTS RESERVED.

1. Technical Specification Guide

1.1. Introduction

This document specifies the manual onboarding process for an SRCi to start working with Amex SRC System.

There are certain steps involved to onboard SRCi. SRC System will first provide SRCi initiator id, followed by sharing the backend payload URL and SRC system software's CDN URLs and then exchanging security keys through well-known URL or secure email. This document explains each onboarding step in detail and provides the payload API specification along with the security details on Payload API.

1.2. SRCi initiator Id

The Initiator id is a unique identity for each SRCi to get identified at SRC system. Amex as an SRC system will generate a UUID for each SRCi. After generating the UUID Amex will share the ID through a secure email.

1.3. Security Keys

Below is the requirement for security keys between Amex as an SRC System and any SRCi system

©2021 AMERICAN EXPRESS. ALL RIGHTS RESERVED.

1.4. IDToken

- Amex will generate an asymmetric key pair and provide the public certificate to other SRC systems to validate the IDToken generated by Amex. The public certificate and its intermediate and root certs will be provided on well-known url.
- Other SRC systems will provide their public certificate, intermediate and root certs on a well-known URL to Amex so that Amex can validate their IDToken signature

1.5. Payload API (server-side) and Checkout response (client-side)

- Amex will use asymmetric key for JWS message signing. Amex will provide the public certificate and its intermediate and root certs on well-known url.
- SRCi will share the public certificate and its intermediate and root certs on well-known url for Amex to encrypt the payload.

1.6. PAN Encryption

Amex will generate asymmetric key pair for Card Object encryption on browser. The public key will be shared with SRCi so that they can encrypt the Card object on the browser.

1.6.1. Asymmetric key sizes and Algorithms

Key length is 3072 bits for all asymmetric keys.

Algorithms used -

Payload signing - {"alg":"RS256"}

IDToken signing - {"alg":"RS256"}

Payload encryption – {"alg":"RSA-OAEP-256","enc":"A256GCM"}

©2021 AMERICAN EXPRESS. ALL RIGHTS RESERVED.

1.7. Mutual TLS

Amex supports mutual transport layer security (mTLS) on payload API. Amex will provide the mTLS public certificate and its intermediate and root certs on well-known url. Key length is 3072 bits.

1.8. Well known URL

Below is the well-known url for QA and Production environment.

- QA: https://openid-qa.americanexpress.com/keys
- Production: https://openid.americanexpress.com/keys

1.9. System Software

Amex will provide the SRC system software in the form of a JavaScript library on content delivery network (CDN). Below is the CDN url for System Software and the steps to integrate with - SRCi needs to add the following script in merchant's web page to load the Amex's SDK

Below are the SDK CDN urls based on the environment

a. QA: https://qwww.aexpstatic.com/akamai/remotecommerce/scripts/amexSDK-1.0.0.js
b. Prod: https://www.aexp-static.com/cdaas/remotecommerce/scripts/amexSDK-1.0.0.js

Example: <script src= https://qwww.aexp-static.com/akamai/remotecommerce/scripts/amexSDK-1.0.0.js ></script>

1. Once the script is loaded it will create a global object on the window. The name of the global object is AmexSS.

2. After the above script has loaded, Amex SDK then creates an iframe to communicate with Amex's server-side APIs. Use the below code to create Amex SDK instance

©2021 AMERICAN EXPRESS. ALL RIGHTS RESERVED.

```
let amexSDK = window.AmexSS.getInstance();
```

3. Methods exposed by Amex SDK will return a promise. If the promise has resolved successfully it will return the response otherwise return an error. The structure for response error codes are based on the Amex Spec

4. One simple example of accessing the Amex' SDK

```
let amexSDK = window.AmexSS.getInstance();
amexSDK.getCustomerProfile(idTokens: string[])
  .then((response) => {
    console.log('PROFILES', response.customerProfile')
  })
  .catch((errors) => {
   console.log('ERRORS', errors')
})
```

2. Data Types

The following sections defines various data types used by this SRC API document.

2.1. Primitive Data Types

Table 1.1: Primitive Types

Type	Base Type	Constraints	Description
String	String	[A-Z][a-z][0-9]	Wherever String type is used it is assumed that the format will be [A-Z][a-z][0-9], no wild characters and any variation to this rule will be mentioned explicitly.
Enum	String		Only a list of specific values are allowed. The list is defined as part of the API specification. All enum values by default are assumed to have maximum size of 255 ASCII characters.
Boolean	Enum		Enumeration field with values "true" or "false".
AlphaNumeric	String	[A-Z][a-z][0-9]	Guid
Identifier	String	[A-Z][a-z][0-9,-]	Generic identifier.
Identifier2	String	[0-9,A-Z,a-z,-,_]	Generic identifier, accepting '_' character.

©2021 AMERICAN EXPRESS. ALL RIGHTS RESERVED.

AddressLine	String	[a-zA-Z0-9 °º.',:_#/()ÁáÀàÂâÄäÃãÇçÉéÈèÊêËëÍíÎîÏï ÑñÓóÒôÖõŒœÚú ÙùÛûÜüŸÿÆæĄąĆćĘęŁłŃńŚśŹźŻż/-]+ size: 140	Used for Address Line as part of address specification.
CountryCode2	String	size: 2	Country in ISO 3166-1 alpha-2 format, eg "US"
Locale	String	size: 5	Based on ISO format for language (ISO 639-1) and alpha-2 country code (ISO 3166-1 alpha-2). The language and country should be separated using a "_". For example- "locale": "en_US".
EpochDateTime	Numeric		UTC Format in epoch seconds E.g. "date": 6763725673
Base64	String	[A-Z][a-z][0-9]=	Base64 encoded string, per RFC4648.
Base64URL	String		Base64URL encoded string. The encoding is the same as Base64, but uses '-' character instead of '+' and '_' character instead of '/' character.
CurrencyCode	String	size: 3	ISO 4217 format
Amount	String	Float	This type is used to represent an amount. Both Digits before and after the '.' are required. Maximum 9 digits before an optional decimal point and 4 decimal digits E.g. "9.00"
Digits	Numeric	[0-9]	A sequence of digits represented as a string. E.g. "122345898"
AccountNumber	String	[1-6,8]{1}[0-9]+ min size: 13 max size: 19	
TwoDigits	Numeric	[0-9]{2}	
FourDigits	Numeric	[0-9]{4}	
SixDigits	Numeric	[0-9]{6}	
SystemId	String	[0-9,A-Z, a-z,+,/,=]	
UUID	String	[A-Fa-f0-9|A-Fa-f0-9]{8}\-([A-Fa-f0-9|A-Fa-f0-9]{4}\-){3}[AFa-f0-9|A-Fa-f09]{12}	UUID
CardType	Enum		Valid Values: • CREDIT • DEBIT • CHARGE • PREPAID • DEFERRED DEBIT • OTHER

©2021 AMERICAN EXPRESS. ALL RIGHTS RESERVED.

CardBrand	Enum		Brand of payment instrument. It is one of the following values: • VISA • MASTERCARD • AMEX • DISCOVER
			• JCB • DINERSCLUB • UNIONPAY • ELECTRON (Brazil only) • ELO (Brazil only)
SecurityCode	Numeric	([0-9]{3,4})?	3-4 digit CVV2 code
URL	String	Max size: 256	HTTPS URL

2.2. Custom Data Structure

2.2.1. DPA Data

This structure represents the configuration parameters that are common across all transactions and it is originates from the DPA.

Note

- Amex do not support dpaAddress in DpaData object

```
dictionary DpaData {
        optional String dpaPresentationName;
        optional String dpaUri;
        optional String dpaLogoUri;
        required String dpaName;
        optional String dpaEmailAddress;
        optional PhoneNumber dpaPhoneNumeber;
        optional String dpaSupportEmailAddress;
        optional PhoneNumber
dpaSupportPhoneNumber;
        optional String dpaSupportUri;
        optional Enum applicationType;
}
```

©2021 AMERICAN EXPRESS. ALL RIGHTS RESERVED.

Description of data elements:

Name	Type	Description
dpaPresentationName	String Optional	Display name of the DPA. Ex. 'Disney Online'.
dpaUri	String Optional	Website URL, Ex. 'http://www.disneyonline.com'
dpaLogoUri	String Optional	URI of the logo of the DPA.
dpaName	String Required	Legal name of registered DPA.
dpaEmailAddress	String Optional	DPA's contact email address.
dpaPhoneNumber	String Optional	DPA's contact phone number.
dpaSupportEmailAddress	String Optional	DPA's support contact email address.
dpaSupportPhoneNumber	String Optional	DPA's support contact phone number.
dpaSupportUri	String Optional	DPA's support URI.
dpaApplicationType	Enum Optional	Type of DPA • WEB_BROWSER • MOBILE_APP • IOT_DEVICE • OTHER

©2021 AMERICAN EXPRESS. ALL RIGHTS RESERVED.

2.2.2. DPA Transaction Options

This structure represents the DPA transaction configuration parameters that are specific to the transaction.

Note

- reviewAction, checkoutDescription are present in Amex spec which are not present in EMV1.1v. Please refer the field for description.
- merchantCategoryCode is optional in EMV1.1v, but Amex spec made it conditional
- merchantCountryCode is optional in EMV1.1v, but Amex spec made it conditional

```
dictionary DpaTransactionOptions {
 optional String dpaLocale;
 optional List<String> dpaAcceptedBillingCountries;
 optional List<String> dpaAcceptedShippingCountries;
 optional Enum dpaBillingPreference;
 optional Enum dpaShippingPreference;
 optional Boolean consumerNameRequested;
 optional Boolean consumerEmailAddressRequested;
 optional Boolean consumerPhoneNumberRequested;
 optional List<PaymentOptions> paymentOptions;
 optional Enum reviewAction;
 optional String checkoutDescription;
 optional Enum transactionType;
 optional String orderType;
 conditional TransactionAmount transactionAmount;
 required Enum threeDsPreference;
 conditional JSONObject threeDsInputData;
 optional JSONObject srcTokenRequestData;
 optional JSONObject customInputData;
 optional String merchantOrderId;
 Conditional String merchantCategoryCode;
 Conditional String merchantCountryCode;
 }

TransactionAmount {
 required Number transactionAmount;
 required String transactionCurrencyCode;
}
```

©2021 AMERICAN EXPRESS. ALL RIGHTS RESERVED.

Description of data elements:

Name	Type	Description
dpaAcceptedBillingCountries	List<String> Optional	Billing Restrictions. Array of country codes in ISO 3166-1 alpha-2 format - Payments from all the listed billing countries are accepted. E.g., ["US","CA","AU"]. If this list is empty, then it means all countries are accepted.
dpaAcceptedShippingCountries	List<String> Optional	Shipping restrictions. Array of country codes in ISO 3166-1 alpha-2 format - Shipping region country codes that limits the selection of eligible shipping addresses. If this list is empty, then it means all countries are accepted
dpaBillingPreference	Enum Optional	Verbosity of address required by the DPA. Valid values for enum: • ALL • POSTAL_COUNTRY • NONE If not provided, the default is 'ALL'
dpaShippingPreference	Enum Optional	Whether DSA wants to have shipping address collected. Valid values for enum: • ALL

©2021 AMERICAN EXPRESS. ALL RIGHTS RESERVED.

		• POSTAL_COUNTRY • NONE If not provided the default is 'ALL'
consumerNameRequested	Boolean Optional	Default is true
consumerEmailAddressRequested	Boolean Optional	Default is true
consumerPhoneNumberRequested	Boolean Optional	Default is true
paymentOptions	List<PaymentOptions> Optional	Payment options requested by the DPA.
reviewAction	Enum Optional	'pay' or 'continue' to indicate to the user if payment will be processed after selection or if only data returned to SRCi/merchant to confirm first. The default value is 'continue'
checkoutDescription	String Optional	Review message to go with action
transactionType	Enum Optional	Type of the transaction. The Possible values are: • PURCHASE • BILL_PAYMENT • MONEY_TRANSFER If not provided, the default value is 'PURCHASE'

©2021 AMERICAN EXPRESS. ALL RIGHTS RESERVED.

orderType	String Optional	Type of order. The Possible values are • REAUTHORIZATION • RECURRING • INSTALLMENT
TransactionAmount.transactioinAmount	Number Conditional	The amount of the transaction. E.g. 99.95 Conditionality: Required if 3DS is requested
TransactionAmount.transactionCurrencyCode	String Conditional	Currency in which the transaction amount is expressed. ISO 4217 three- digit currency code Conditionality: Required if amount is present and 3DS is requested
ThreeDsPreference	Enum Required	Merchant's 3DS preferences Possible values are • NONE • SELF • ONBEHALF
ThreeDsInputData	JSON Conditional	Merchant's 3DS input data. Conditionality: Must be supplied if 3DS is to be performed by SRC System. Possible values are acquirerId, acquirerMid and requestorId. acquirerMid is mandatory if ThreeDsPreference = ONBEHALF
merchantOrderId	String Optional	The order identifier generated by the DPA.
merchantCategoryCode	String Conditional	MCC of the merchant Conditionality: Required if 3DS is requested

©2021 AMERICAN EXPRESS. ALL RIGHTS RESERVED.

merchantCountryCode	String Conditional	Country code of the merchant Conditionality: Required if 3DS is requested
srcTokenTokenRequestData	JSON Optional	Data to support a Token Request.
customInputData	JSON Optional	Extensible container that allows DPA to pass network-specific set of data to SRC System.

2.2.3. PaymentOptions

This structure details of this transaction.

```
dictionary PaymentOptions {
      optional Enum dynamicDataType;
      optional String dpaDynamicDataTtlMinutes;
}
```

Description of data elements:

Name	Type	Description
dynamicDataType	Enum Optional	If not provided, in the response PAN payload will be provided. If provided but issuer doesn't support tokenization that case PAN payload will be provided in the response Can be the following values: • DYNAMIC_CARD_SECURITY_CODE (This is 4-digit dynamic CID) • CARD_APPLICATION_CRYPTOGRAM_SHORT_FORM • CARD_APPLICATION_CRYPTOGRAM_LONG_FORM • CARDHOLDER_AUTHENTICATION_CRYPTOGRAM • NONE Only DYNAMIC_CARD_SECURITY_CODE will be returned as other forms are not supported as of now.
dpaDynamicDataTtlMinutes	String Optional	Requested "Time to Live" (expiry period) of the dynamic data, specified in minutes.

©2021 AMERICAN EXPRESS. ALL RIGHTS RESERVED.

2.2.4. SrcProfile

This is the detailed structure of the Customer Profile object.

Note
• Amex do not support maskedShippingAddresses

```
dictionary SrcProfile {
    optional List<MaskedCard> maskedCards;
    conditional MaskedConsumer maskedConsumer;
    required JWT authorization;
}
```

Description of data elements:

Name	Type	Description
maskedConsumer	MaskedConsumer Conditional	Masked information of a recognized consumer.
maskedCards	List<MaskedCard> Optional	Card list of recognized/authenticated user.
authorization	JWT Required	See the description of SRC IDToken in the section above in this document. The idTokens are returned only for the consumers recognized by this SRC system.

2.2.5. Address

This is the structure of the Address object.

Note
• Amex does not support createTime, lastUsedTime. • addressId is optional at EMVCo V1.1 but Amex made it required

©2021 AMERICAN EXPRESS. ALL RIGHTS RESERVED.

```
dictionary Address {
    required String addressId; //UUID
    optional String name;
    optional String line1;
    optional String line2;
    optional String line3;
    optional String city;
    optional String state;
    optional String zip;
    optional String countryCode;
}
```

Description of data elements:

Name	Type	Description
AddressId	String Required	Identifier of the address in the SRC system. This is UUID.
name	String Optional	This will be the recipient name for shipping address and cardholder name for billing address.
line1	String Optional	Line 1 of the address.
line2	String Optional	Line 2 of the address.
line3	String Optional	Line 3 of the address.
city	String Optional	Name of the city.
state	String Optional	Name of the state.
zip	String Optional	Postal code for this address.
countryCode	String Optional	Country code for this address

©2021 AMERICAN EXPRESS. ALL RIGHTS RESERVED.

2.2.6. Masked Card

This is the structure of the Masked card object.

Note

- Amex does not support these fields under Masked Card
  - (a) paymentCardDescriptor
  - (b) srcPaymentCardId
  - (c) digitalCardFeatures
  - (d) countryCode
  - (e) dcf
  - (f) serviced
  - (g) paymentAccountReference
- srcDigitalCardId is conditional at EMVCo V1.1 but Amex made it required
- paymentCardType is optional at EMVCo V1.1 but Amex made it required

dictionary MaskedCard {

```
  required String srcDigitalCardId;
  required String panBin;
  required String panLastFour;
  conditional String tokenBinRange;
  conditional String tokenLastFour;
  conditional String panExpirationMonth;
  conditional String panExpirationYear;
  required DigitalCardData digitalCardData;
  required EpochDateTime dateofCardCreated;
  optional EpochDateTime dateofCardLastUsed;
  optional MaskedAddress maskedBillingAddress;
  required String paymentCardType
}
```

©2021 AMERICAN EXPRESS. ALL RIGHT S RESERVED.

Description of data elements:

Name	Type	Description
srcDigitalCardId	UUID Required	Unique Identifier of the card (UUID)
panBin	String Required	BIN of the card
panLastFour	String Required	Last 4 digits of the card.
tokenBinRange	String Conditional	BIN of the token. Provided if the transaction is a Token based.
tokenLastFour	String Conditional	Last 4 digits of the token. Provided if the transaction is a Token based.
panExpirationMonth	String Conditional	2-digit PAN expiration month. This is required if card supports expiry
panExpirationYear	String Conditional	4-digit PAN expiration year. This is required if card supports expiry
digitalCardData	DigitalCardData Required	This is metadata about the card
dateofCardCreated	EpochDateTime Required	Timestamp of when this card was enrolled. Format: Unix timestamp, milliseconds since epoch. Eg. 1536926400
dateofCardLastUsed	EpochDateTime Optional	Timestamp of when this card was last used. Format: Unix timestamp, milliseconds since epoch. Eg. 1536926400

©2021 AMERICAN EXPRESS. ALL RIGHTS RESERVED.

maskedBillingAddress	MaskedAddress Optional	Masked billing address for display purposes.
paymentCardType	String Required	Always “CREDIT” for SRC System.

2.2.7. Digital Card Data

This is the structure of the Digital Card object.

Note
• pendingEvents: Amex does not support pendingEvents

```
dictionary DigitalCardData {
        Required Enum status;
        optional String presentationName;
        Required String descriptorName;
        Required String artUri;
        optional Number artHeight;
        optional Number artWidth;
}
```

Description of data elements:

Name	Type	Description
status	Enum Required	Status of the Card in the SRC system, Amex only supports Active
presentationName	String Optional	Name that can be used for display

©2021 AMERICAN EXPRESS. ALL RIGHTS RESERVED.

descriptorName	String required	Presentation text defined by the SRC Programme that describes the PAN presented as a Digital Card. This descriptor is the same across all DCFs.
artUri	String required	HTTPS (full) URL for the cardArt. Can be card or issuer specific value. E.g. 'https://network.com/card/1234.png'
artHeight	Number Optional	Height of the card art image in pixels
artWidth	Number Optional	Width of the card art image in pixels

2.2.8. Card

This is the structure of the Card object.

Note

- Amex does not support paymentAccountReference
- panExpirationMonth is conditional at EMVCo V1.1 but Amex made it required
- panExpirationYear is conditional at EMVCo V1.1 but Amex made it required

```
dictionary Card {
  required String primaryAccountNumber;
  required String panExpirationMonth;
  required String panExpirationYear;
  optional String cardSecurityCode;
  optional String cardholderFullName;
  optional String cardholderFirstName;
  optional String cardholderLastName;
  optional Address billingAddress;
}
```

©2021 AMERICAN EXPRESS. ALL RIGHTS RESERVED.

Description of data elements:

Name	Type	Description
primaryAccountNumber	String Required	This is the PAN
panExpirationMonth	String Required	Two-digit expiry month
panExpirationYear	String Required	Four-digit expiry year
cardSecurityCode	String Optional	Some cards may not have a security code value.
cardHolderFullName	String Optional	Card holder name
cardHolderFirstName	String Optional	Card holder Last name
cardHolderLastName	String Optional	Billing address of the card member
billingAddress	Address Optional	

2.2.9. PaymentToken

This is the structure of the Token object.

©2021 AMERICAN EXPRESS. ALL RIGHTS RESERVED.

Note

- Amex does not support these fields under PaymentToken
  (a) cardholderFullName
  (b) cardholderFirstName
  (c) cardholderLastName
  (d) paymentAccountReference.
- tokenExpirationMonth is conditional at EMVCo V1.1 but Amex made it required
  tokenExpirationYear is conditional at EMVCo V1.1 but Amex made it required

```
dictionary PaymentToken {
   required String paymentToken;
   required String tokenExpirationMonth;
   required String tokenExpirationYear;
}
```

Description of data elements:

Name	Type	Description
paymentToken	String Required	This is the Token
tokenExpirationMonth	String Required	Two-digit expiry month
tokenExpirationYear	String Required	Four-digit expiry year

2.2.10. Masked Address

This details the structure of the Address object.

```
dictionary MaskedAddress {
    required String addressId;
    optional String name;
    optional String line1;
```

©2021 AMERICAN EXPRESS. ALL RIGHTS RESERVED.

```
    optional String line2;
    optional String line3;
    optional String city;
    optional String state;
    optional String zip;
    optional String countryCode;
    optional String createTime;
    optional String lastUsedTime;
}
```

Description of data elements:

Name	Type	Description
addressId	String required	Identifier of the address in the SRC system
line1	String Optional	Masked Line 1 of the address. E.g. '1** M*** St'
line2	String Optional	Masked Line 2 of the address.
line3	String Optional	Masked Line 3 of the address.
city	String Optional	Masked name of the city.
state	String Optional	Masked name of the state.
zip	String Optional	Postal code for this address.
countryCode	String Optional	Country code for this address

©2021 AMERICAN EXPRESS. ALL RIGHTS RESERVED.

name	String Optional	Name of the individual receiving the delivered goods or services. Only applicable for the shipping address
createdTime	Epoch format Optional	Data and time the address was created
lastUsedTime	Epoch format Optional	Data and time the address was last used

2.2.11. Dynamic Data

This is the structure of the DynamicData object.

Note
• dynamicDataExpiration is optional at EMVCo V1.1 but Amex made it conditional

```
dictionary DynamicData {
    conditional String dynamicDataValue;
    required Enum dynamicDataType;
    conditional String dynamicDataExpiration;
}
```

Description of data elements:

Name	Type	Description
dynamicDataValue	String Conditional	The value of the cryptogram or CVV2 code as determined by the dynamic data type field. **Condition:** If DYNAMIC_CARD_SECURITY_CODE is enabled for initiator, market, card issuer.

©2021 AMERICAN EXPRESS. ALL RIGHTS RESERVED.

dynamicDataType	Enum Required	Type of dynamic data provided by the SRC system. Valid values: • DYNAMIC_CARD_SECURITY_CODE • NONE Note: CARD_APPLICATION_CRYPTOGRAM_LONG_FORM is in the near-term roadmap. Launch timeline TBD. NONE will be returned when Amex SRC System has performed 3DS (SafeKey) authentication. DYNAMIC_CARD_SECURITY_CODE will be sent for both TOKEN and PAN transactions.
dynamicDataExpiration	String Conditional	Indicates validity period returned by the SRC system. **Condition:** If DYNAMIC_CARD_SECURITY_CODE is enabled for initiator, market, card issuer.

2.2.12. Checkout Payload Response

Note

- Amex does not support customOutputData
- srcCorrelationId is conditional at EMVCo V1.1 but Amex made it required
- maskedConsumer is conditional at EMVCo V1.1 but Amex made it required
- Field name is payload in EMVCo V1.1 but Amex made it encryptedPayload
- Field of type in JWE<JWS<Payload>> in EMV1.1v, but Amex made Payload<JWE>

```
dictionary CheckoutPayloadResponse {
  required UUID srcCorrelationId;
  conditional String srciTransactionId;
  required MaskedConsumer maskedConsumer;
```

©2021 AMERICAN EXPRESS. ALL RIGHTS RESERVED.

```
  required MaskedCard maskedCard;
  conditional String shippingAddressZip;
  conditional String shippingAddressCountryCode;
  conditional Payload<JWE> encryptedPayload;
  optional AssuranceData assuranceData;
  optional EventHistory eventHistory;
}
```

Description of data elements:

Name	Type	Description
srcCorrelationId	UUID Required	This is the unique identifier generated by SRC system to track and link SRC messages. This will be used as the transaction identifier assigned by the SRC system for this particular transaction.
srciTransactionId	String Conditional	A unique id used to track the user journey. This is used for analytics to be able to correlate a single user "session" from button impression to the end of the transaction. This field may be created on the merchant page by the SRC Initiator and needs to be passed through to all the networks (SRC Systems). It is passed all the way to the DCFs as well. Conditionality: Will be provided in the response if SRCi sent when they requested for an SRC transaction.
maskedCard	MaskedCard	Masked card information.
	Required	

©2021 AMERICAN EXPRESS. ALL RIGHTS RESERVED.

shippingAddressZIP	String Conditional	ZIP Code of the address being shipped to. Provided if it was asked in the DPATransctionOption in Init or checkout.
shippingAddressCountryCode	String Conditional	Country Code of the address being shipped to. Provided if it was asked in the DPATransctionOption in Init or checkout.
maskedConsumer	MaskedConsumer Required	Masked information about the consumer for display purposes.
EncryptedPayload	Payload<JWE> Conditional	This is the encrypted Payload. For the structure of this element, see Payload. This is required only when PCI/PII data is requested by the DPA during an SRC transaction. This is a conditional field and will not be sent back when only masked information is requested by the DPA. See payloadTypeIndicator field for more info. We will send this in two places 2.2.12.1.1.1. Browser Checkout In case payloadTypeIndicatorCheckout is one of the following ✦ SUMMARY ✦ PAYMENT ✦ FULL ✦ NON_PAYMENT

©2021 AMERICAN EXPRESS. ALL RIGHTS RESERVED.

		2.2.12.1.1.2. Backend Payload In case payloadTypeIndicatorPayload is one of the following ✦ PAYMENT ✦ FULL
eventHistory	EventHistory Optional	Event history related to checkout flow
assuranceData	AssuranceData Optional	Assurance data related to checkout flow

2.2.13. Payload

This is the structure of the encrypted Payload object that are a part of the JWE claims. See the JWE section in the appendix for encryption details. All these individual data elements are directly added to the claim set (custom claims) of the JWE

Note

- Amex does not support srcTokenResultsData
- dynamicData is required at EMVCo V1.1 but Amex made it conditional.

```
dictionary Payload {
conditional Card card;
conditional PaymentToken token;
conditional Address shippingAddress;
conditional List<DynamicData> dynamicData;
conditional Address billingAddress;
conditional String consumerEmailAddress;
conditional String consumerFirstName;
conditional String consumerLastName;
conditional String consumerFullName;
conditional MobileNumber consumerMobileNumber;
conditional JSONObject threeDsOutpoutData
}
```

©2021 AMERICAN EXPRESS. ALL RIGHTS RESERVED.

Description of data elements:

Name	Type	Description
card	Card Conditional	If merchant needs PAN payload, this would contain the PAN number.
token	PaymentToken Conditional	If merchant needs Token payload, this would contain the token number.
dynamicData	List<DynamicData > Conditional	Dynamic data (cryptograms) applicable for the transaction.
billingAddress	Address Conditional	Billing address of the card selected. This must be present if the billing address was requested by the DPA.
shippingAddress	Address Conditional	Shipping address of the consumer. This must be present if the shipping address was requested by the DPA.
consumerEmailAddress	String Conditional	Consumer-provided email address. Conditionality: Must be provided if: • Consumer Email Address is available in the SRC Profile AND • Email address is requested (consumerEmailAddress Requested is true) AND • PayloadTypeIndicatorCheckout is “FULL” or “NON_PAYMENT”
consumerFirstName	String Conditional	Consumer-provided first name. Conditionality: Must be provided if: • Consumer First Name is available in the SRC Profile AND

©2021 AMERICAN EXPRESS. ALL RIGHTS RESERVED.

		• Consumer Name is requested (consumerNameRequested is true) AND • PayloadTypeIndicatorCheckout is “FULL” or “NON_PAYMENT”
consumerLastName	String Conditional	Consumer-provided last name. Conditionality: Must be provided if: • Consumer Last Name is available in the SRC Profile AND • Consumer Name is requested (consumerNameRequested is true) AND • PayloadTypeIndicatorCheckout is “FULL” or “NON_PAYMENT”
consumerFullName	String Conditional	Consumer-provided full name. Conditionality: Must be provided if: • Consumer Full Name is available in the SRC Profile AND • Consumer Name is requested (consumerNameRequested is true) AND • PayloadTypeIndicatorCheckout is “FULL” or “NON_PAYMENT”
consumerMobileNumber	MobileNumber Conditional	Consumer-provided mobile number. Conditionality: Must be provided if: • Consumer Mobile Name is available in the SRC profile AND • Consumer Mobile number is requested (consumerPhoneNumber Requested is true) AND

©2021 AMERICAN EXPRESS. ALL RIGHTS RESERVED.

		• PayloadTypeIndicatorCheckout is “FULL” or “NON_PAYMENT”
threeDsOutpoutData	JSON (String) Conditional	If threeDs transaction is not NONE, then Amex 3DS specific fields will be populated in JSON format. If request is Payment authenticated 3DS request, then this JSON will populate with below fields. authenticationValue eci transStatus transStatusReason dsTransID acsTransID

2.2.14. Masked Consumer

This is the structure of the Masked Consumer object.

Note
• Amex does not support fields under Masked Consumer (a) maskedEmailAddress (b) dateConsumerLastUsed (c) maskedNationalIdentifier (d) complianceSettings.

```
dictionary MaskedConsumer {
 optional String srcConsumerId;
 required MaskedConsumerIdentity maskedConsumerIdentity;
 optional String maskedFirstName;
 optional String maskedLastName;
 optional String maskedFullName;
 optional PhoneNumber maskedMobileNumber;
 optional String countryCode;
 optional Locale languageCode;
 required enum status;
 required EpochDateTime dateConsumerAdded;
}
```

©2021 AMERICAN EXPRESS. ALL RIGHTS RESERVED.

Description of data elements:

Name	Type	Description
srcConsumerId	String Optional	Returned by the SRC system if available.
maskedFirstName	String Optional	Masked first name of the consumer.
maskedLastName	String Optional	Masked last name of the consumer.
maskedFullName	String Optional	Masked full name of the consumer. Required when first name and last name is not present.
maskedMobileNumber	PhoneNumber Optional	Masked mobile number of the consumer. Required if emailAddress is not present.
countryCode	CountryCode Optional	Consumer provided country code
languageCode	String Optional	Consumer provided language choice
status	enum Required	Can be ACTIVE, SUSPENDED and LOCKED
dateConsumerAdded	EpochDateTime Required	Date consumer was added to SRC system
maskedConsumerIdentity	MaskedConsumerIdentity Required	Masked value of the primary verifiable Consumer Identifier within an SRC Profile. For example, an email address or a mobile phone number.

©2021 AMERICAN EXPRESS. ALL RIGHTS RESERVED.

2.2.15. AssuranceData

This is the structure of the AssuranceData object.

Note
• Amex only supports cardVerificationMethod under AssuranceData.

Description of data elements:

Name	Type	Description
cardVerificationMethod	String Optional	Card verification check to validate that the PAN is active and valid at the Card Issuer. Valid values are: • 01 $0 authorisation, or single unit of currency authorisation • 02 Card Verification Number validation • 03 Postal code and address verification, where supported • 04 - 20 EMVCo future use • 21-99SRCSystem specific

2.2.16. EventHistory

This is the structure of EventHistory object

Note
• Amex only supports ageOfSrcPanEnrolmentSinceCreated under EventHistory.

```
dictionary EventHistory {
    optional String ageOfSrcPanEnrolmentSinceCreated;  }
```

Description of data elements:

Name	Type	Description

©2021 AMERICAN EXPRESS. ALL RIGHTS RESERVED.

ageOfSrcPanEnrolmentSinceCreated	String Optional	Age, in days, of the SRC Profile as it exists in the SRC System since the time it was created.

2.2.17. PhoneNumber

This is the structure of the PhoneNumber object.

Note
• countryCode is required at EMVCo V1.1 but Amex made it optional

```
dictionary PhoneNumber {
      optional String countryCode;
      required String phoneNumber;
}
```

Description of data elements:

Name	Type	Description
countryCode	String Optional	Phone number Country code.
phoneNumber	String Required	Phone number

2.2.18. Consumer Identity

This is the structure of the ConsumerIdentity object.

```
dictionary ConsumerIdentity {
  optional Enum identityProvider;
  required String identityValue;
  required Enum identityType;
}
```

©2021 AMERICAN EXPRESS. ALL RIGHTS RESERVED.

Description of data elements:

Name	Type	Description
identityProvider	Enum Optional	The Identity Provider – The default value is 'SRC'.
identityValue	String Required	Value of the consumer identity – eg., 'user@example.com'
identityType	Enum Required	The type of the consumer identity. Values: • EMAIL_ADDRESS • MOBILE_PHONE_NUMBER • CUSTOM_IDENTIFIER

2.2.19. MaskedConsumerIdentity

This is the structure of the MaskedConsumerIdentity object.

```
dictionary MaskedConsumerIdentity {
  optional Enum identityProvider;
  required Enum identityType;
  required String maskedIdentityValue;
}
```

Description of data elements:

Name	Type	Description
identityProvider	Enum Optional	The Identity Provider – The default value is 'SRC'.
maskedIdentityValue	String Required	Value of the consumer identity – eg., 'user@example.com'

©2021 AMERICAN EXPRESS. ALL RIGHTS RESERVED.

identityType	Enum Required	The type of the consumer identity. Values: • EMAIL_ADDRESS • MOBILE_NUMBER • CUSTOM_IDENTIFIER

2.2.20. IdentityValidationChannel

This is the structure of the IdentityValidationChannel object.

```
dictionary IdentityValidationChannel {
        required String ValidationChannelId;
        required Enum identityType;
        optional String maskedValidationChannel;
}
```

Description of data elements:

Name	Type	Description
ValidationChannelId	Enum Required	Id of the validation channel.
identityType	String Required	The type of the consumer identity. e.g. • EMAIL • SMS
maskedValidationChannel	Enum Optional	Masked ID validation channel (e.g. masked email, masked mobile number).

©2021 AMERICAN EXPRESS. ALL RIGHTS RESERVED.

2.2.21. Confirmation Data

This is the structure of the ConfirmationData object.

Note

- Amex does not support assuranceData
- checkoutEventType is optional at EMVCo V1.1 but Amex made it required
- checkoutEventStatus is optional at EMVCo V1.1 but Amex made it required
- transactionAmount is optional at EMVCo V1.1 but Amex made it required

```
dictionary ConfirmationData {
    required String checkoutEventType;
    required String checkoutEventStatus;
    optional String confirmationStatus;
    optional String confirmationReason;
    optional String confirmationTimestamp;
    optional String networkAuthorizationCode;
    optional String networkTransactionIdentifier;
    optional Enum paymentNetworkReference;
    required TransactionAmount transactionAmount;
}
```

Description of data elements:

Name	Type	Description
checkoutEventType	String Required	Event type associated with the update. Valid values are: • 01 Authorise • 02 Capture • 03 Refund • 04 Cancel • 05 Fraud • 06 Chargeback • 07 Other

©2021 AMERICAN EXPRESS. ALL RIGHTS RESERVED.

checkoutEventStatus	String Required	Event type associated with the order. Valid values are: • 01 Created • 02 Confirmed • 03 Cancelled • 04 Fraud Cancelled • 05 Others • 06 - 50 EMVCo future use • 51 - 99 SRC System specific
confirmationStatus	String Optional	Status of the event as provided by the SRC Initiator in the Confirmation message. Valid values are: • 01 Success • 02 Failure • 03 Other
confirmationReason	String Optional	Description of the reason for the event associated with the order.
confirmationTimestamp	String Optional	Date and time of the event completion corresponding to the Confirmation event by the SRC Initiator.
networkAuthorizationCode	String Optional	Authorisation code associated with an approved transaction.
networkTransactionIdentifier	String Optional	Unique authorisation related tracing value assigned by a Payment Network and provided in an authorisation response.
paymentNetworkReference	String Optional	Transaction identifier as provided by a Payment Network after authorisation has been complete.
transactionAmount	TransactionAmount Required	Amount of the transaction.

©2021 AMERICAN EXPRESS. ALL RIGHTS RESERVED.

2.3. SRC ID Token

This allows for other SRC Systems to generate a session using the trust with SRC System that did the user verification.

This is a token that allows the SRC Systems to communicate to each other the SRC user identity verification. The SRC systems can exchange this token for an access token.

This SRC IDToken should be discarded at the end of an SRC checkout transaction.

Table 1.2: SRC ID Token

Name	Cardinality	Notes
JWS Header		
alg	1	Algorithm used to sign this id token. Supported values/algorithms: • "RS256" to mean RSASSA-PKCS1-v1_5 using SHA-256
kid	1	Client's (Issuer's) API key to look up the appropriate key When the SRC System generates this IDToken, this API key is used to lookup the "Outbound Message Authentication" key. When the SRC System verifies the signature of this IDToken, this API key is used to lookup the "Inbound Message Authentication" key.
typ	1	The media type of the token Must be "JWT+ext.id_token"
JWS Claim Set		
iss	1	Identifies the IDToken's Issuer on the domain of the receiver of this token. E.g. 'https://www.americanexpress.com'

©2021 AMERICAN EXPRESS. ALL RIGHTS RESERVED.

sub	1	Subject Identifier. A locally unique and never reassignedidentifier within the Issuer for the End-User, which is intended to be consumed by the Client, e.g., 24400320 or AItOawmwtWwcT0k51BayewNvutrJUq svl6qs7A4. It MUST NOT exceed 255 ASCII characters in length. The sub value is a case sensitive string. SRC specific primary identifier of the consumer that MAY BE used to locate consumer's SRC Profile.
aud	1..n	Who this token is intended for. The domains of the audience. After the user is successfully authenticated, this is a white-space delimited list of URLs (protocol, port, hostname, uri path, without query params) of all SRCs. It is possible that there may be more than one URL per SRC. Like https:/mastercard.com, https://americanexpress.com, https://discover.com, etc.
exp	1	Expiration time in UTC and unix/epoch format The current date/time MUST be before the expiration date/time listed in the value. Implementers MAY provide for some small leeway, usually no more than a few minutes, to account for clock skew.

©2021 AMERICAN EXPRESS. ALL RIGHTS RESERVED.

iat	1	Issuance time in UTC and unix/epoch format Time at which the JWT was issued. This should not be before the current date/time. Implementers MAY provide for some small leeway, usually no more than a few minutes, to account for clock skew.
auth_time	0..1	Time when the End-User authentication occurred. If this is not provided, then it is the same as iat. This specifies when the actual user authentication occurred. The time when the user actually provided the credentials for authentication on this device.
jti	0..1	The "jti" (JWT ID) claim provides a unique identifier for the JWT. The value is a case-sensitive string.
amr	1..n	Authentication Methods References Partner that authenticated the user informs the relying party a list of credential types provided by the end-user during their authentication. It is an JSON array of strings. For the specific details of each of the values, see: https://tools.ietf.org/html/draft-ietf-oauth-amr-values-04#page-3 Here are the valid values: • email_otp • cookie

©2021 AMERICAN EXPRESS. ALL RIGHTS RESERVED.

phone_number	0..1	Hash of End-User's preferred mobile phone number in E.164 [E.164] format for example, +1 (425) 555- 1212) or+56 (2) 687 2400. Extensions are not permitted. Used by Relying Party to help to identify a matching customer profile. The hash value must be base64 encoded with '=' padding.
phone_number_verifi ed	0..1	True if the End-User's phone number has been verified; otherwise false. When this Claim Value is true, this means that the OP took affirmative steps to ensure that this phone number was controlled by the End-User at the time the verification was performed. The means by which a phone number is verified is context-specific, and dependent upon the trust framework or contractual agreements within which the parties are operating. For SRC, this value MUST be true ONLY IF the OP can deterministically confirm that the phone number was verified by the user authenticated on this app instance.
email	0..1	Hash of End-User's preferred e-mail address. Its value MUST conform to the RFC 5322 [RFC5322] addr-spec syntax. The RP MUST NOT rely upon this value being unique. The hash value must be base64 encoded with '=' padding.

©2021 AMERICAN EXPRESS. ALL RIGHT S RESERVED.

email_verified	0..1	True if the End-User's e-mail address has been verified; otherwise false. When this Claim Value is true, this means that the OP took affirmative steps to ensure that this e-mail address was controlled by the End-User at the time the verification was performed. The means by which an e-mail address is verified is context-specific, and dependent upon the trust framework or contractual agreements within which the parties are operating. For SRC: This value MUST be true ONLY IF the OP can deterministically confirm that the email address was verified by the user authenticated on this app instance.
src_phone_number_ mask	0..1	Masked consumer mobile phone number. This MUST use E.164 format with SRC-specific masking rules. Used by Relying Party to properly render UI and allow frictionless onboarding User Experience. Masked consumer e-mail address in RFC 5322 format with SRC specific masking rules.Used by Replying Party to properly render UI and allow frictionless onboarding User Experience. Masking rules are defined in the Appendix section. Masking rules are defined in the Appendix section.
src_email_mask	0..1	Masked consumer e-mail address in RFC 5322 format with SRC specific masking rules. Used by Replying Party to properly render UI and allow frictionless onboarding User Experience. Masking rules are defined in the Appendix section.

©2021 AMERICAN EXPRESS. ALL RIGHTS RESERVED.

3. JavaScript APIs for SRC Initiators

This section presents the SRC JavaScript SDK. The SDK will be provided as source code to be consumed by the SRCI web application, and this SDK will implement the actual API calls to the SRC system.

Notes for JS APIs:

- All the SRC JS SDK APIs other than the init methods use JS promises instead of events/callbacks. The SRC SDK always responds back with a promise.
- If the promises are fulfilled, then the success responses are documented in the below "Response Attributes" sections for each of the APIs. If the promises are rejected (for errors), the error detail structure is documented in the errors section above.
- The WindowRef is used to host the UI. When presenting SRCI UI, the SRCI JS implementation will control the window (Popup or iframe). And, when DCF is presenting the UI, SRC SDK implementation will pass the WindowRef to the DCF appropriately so that the DCF UI can be rendered on that window. This should work seamlessly for Popups or iframes.
- All the SRC SDKs are initialized within the SRCI Communicator iframe, the postMessage calls are all prefixed by the network name. This means that the network SRC SDKs should only listen to their own postMessage calls and ignore others. So, when a network card is selected by the user, the subsequent postMessage calls are all made with that network prefix only until the user selects/switches-to a card of different network.

©2021 AMERICAN EXPRESS. ALL RIGHTS RESERVED.

3.1. Initialize SRC SDK

Initialize the app with common state. Must be called before any other methods. Synchronous, no response.

Note

- Amex does not support srcDpaId.
- Version is an optional element for SRCi to pass version number of specification (e.g. 1.1) they want to execute. Amex internally setup base version and supported versions for every SRCi after mutual agreement. After this setup SRCi can switch between supported versions, otherwise it will be defaulted to base version.
- DpaData is conditional at EMVCo V1.1 but Amex made it required

```
init ({
 optional String srciTransactionId;
 required String srcInitiatorId;
 required DpaData dpaData;
 optional DpaTransactionOptions dpaTransactionOptions;
 optional String version;
}
```

```
// Response - empty
```

Request Parameters:

Name	Type	Description
srciTransactionId	String Optional	A unique id used to track the user journey. This is used for analytics to be able to correlate a single user "session" from button impression to the end of the transaction. This field may be created on the merchant page by the SRC Initiator and need to be
		passed-through to all the networks (SRC Systems). It is passed all the way to the DCFs as well.

©2021 AMERICAN EXPRESS. ALL RIGHTS RESERVED.

srcInitiatorId	UUID Required	SRCI identifier. This is generated by the SRC System during onboarding.
dpaData	DpaData Required	DPA Registration data should be provided to SRC system.
dpaTransactionOptions	DpaTransactionOptions Optional	DPA configuration data that can override any configuration sent last time.

Business Errors:

Reason Code	Description
SRCI_ID_MISSING	The identifier for the SRC Initiator is missing.
DPA_ID_MISSING	Both srciDpaId & srcDpaId is missing. There would be no way for an SRC System to identify the DPA.
SRCI_TXN_ID_MISSING	srciTransactionId is missing.

3.2. Recognize Consumer

Determine if the user is recognized (e.g. by detecting the presence of a local cookie in the browser environment) and, if so, obtain JWT to optionally pass to getCustomerProfile call to other SRC Systems. In case the user is recognized, the method may then (as an optimization) initiate the getCustomerProfile request.

isRecognized()

// Response dictionary
{
required Boolean recognized;
conditional List<JWT>
idTokens;
}

©2021 AMERICAN EXPRESS. ALL RIGHTS RESERVED.

Response Attributes:

Name	Type	Description
recognized	Boolean Required	This indicates if the user is recognized/identified by the SRC System.
idTokens	List<JWT> Conditional	See the description of SRC IDToken in the section above in this document. The idToken is returned only for the recognized user.

Business Errors:

Reason Code	Description
ACCT_INACCESSIBLE	The account exists but is not currently accessible (e.g. is locked).

3.3. Get SRC Profile

Takes JWT (or browser provided HTTP-only, secure cookie), and returns masked card and other account profile data for enabling card selection.

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

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

©2021 AMERICAN EXPRESS. ALL RIGHTS RESERVED.

Request Parameters:

Name	Type	Description
idTokens	List<JWT> Optional	See the description of SRC IDToken in the section above in this document. SRCI can provide idTokens received from SRC systems based on consumer recognition performed via OTP (for unrecognized flow) or cookie (for recognized flow). SRCI aggregate idTokens from SRC systems and provide to individual SRC systems to fetch card list. If SRCI does not provide idTokens in the request, SRC systems might still fetch cards based on cookie recognition or decide to return empty card list if the consumer is not recognized.

Response Attributes:

Name	Type	Description
profiles	SrcProfile Required	Customer Profiles associated to each recognized user using JWT (idToken) Need to send empty list if no profiles found
srcCorrelationId	UUID Optional	This is the unique identifier generated by SRC system to track and link SRC messages. This will be used as the transaction identifier assigned by the SRC System for this particular transaction. This field is required when the cards are returned successfully.

©2021 AMERICAN EXPRESS. ALL RIGHTS RESERVED.

Business Errors:

Reason Code	Description
AUTH_INVALID	Invalid Authorization
ACCT_INACCESSIBLE	The account exists but is not currently accessible (e.g. is locked).
EMPTY_PROFILES	No Profiles found for IdTokens

3.4. Identity Lookup

This method is to lookup users (user accounts) for a given user identity as email

```
 identityLookup(
 required ConsumerIdentity consumerIdentity;
)
// Response dictionary
{
 required Boolean consumerPresent;
 optional List<IdentityValidationChannel>
supportedValidationChannels
}
```

Request Parameters:

Name	Type	Description
consumerIdentity	ConsumerIdentity Required	Identifier to be used to lookup user across SRC systems. It is email id for Amex SRC System.

Response Attributes:

Name	Type	Description
consumerPresent	Boolean Required	This indicates if the customer is present in the SRC System.
supportedValidationalChannels	List<IdentityValidationChannel> Optional	List of supported validation channels. This may be returned by the SRC System to indicate the channels supported.

©2021 AMERICAN EXPRESS. ALL RIGHTS RESERVED.

Business Errors:

Reason Code	Description
FRAUD	The user account was locked or disabled.
ID_FORMAT_UNSUPPORTED	Unsupported Id format
CONSUMER_ID_MISSING	Consumer identity is missing in the request
ACCT_INACCESSIBLE	The account exists but is not currently accessible (e.g. is locked)

3.5. Initiate Identity Validation

This method is used to send validation code (like Email/SMS OTP) to a given user id after a successful response is received for the Id Lookup request.

```
initiateIdentityValidation (
 optional String requestedValidationChannelId;
)
```

```
// Response dictionary {
 required String maskedValidationChannel;
 optional String validationMessage;
 optional List<IdentityValidationChannel>
supportedValidationChannels;
}
```

©2021 AMERICAN EXPRESS. ALL RIGHTS RESERVED.

Request Parameters:

Name	Type	Description
requestedValidationChannelId	String Optional	This validation channel id was selected from the UI by user if SRCi has a facility for user to choose the validation channel between PHONE and EMAIL. If not provided, Amex SRC system will decide where to send the OTP. Amex can send the validation OTP to both SMS and EMAIL.

Response Attributes:

Name	Type	Description
maskedValidationChannel	String Required	Masked value of the channel used to deliver the Validation code (like OTP). Channel can be either of masked Email/Phone in case user selected the validation channel from sendOTP screen OR Amex SRC system decided to send it to particular channel. For some use cases, Amex SRC system decided to send the OTP to both the
		channels, and we may send the comma separated masked channels in this field.

©2021 AMERICAN EXPRESS. ALL RIGHTS RESERVED.

validationMessage	String Optional	Optional display message which may contain a locale-specific advisory to the Consumer about the identity validation process.
supportedValidationChannels	List<RequestedValidationChannelId> Optional	The list of supported validation channel.

Business 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 for OTP generation was exceeded.
ID_INVALID	Invalid ID.
UNRECOGNIZED_CONSUMER_ID	Consumer ID could not be recognized.
ACCT_INACCESSIBLE	The account exists but is not currently accessible (e.g. is locked).

3.6. Complete Identity Validation Process

This method is to complete the validation of the Validation code (like OTP value) submitted by the

consumer has received via maskedValidationChannel.

```
completeIdentityValidation (
required String validationData;
)
// Response dictionary {
required JWT idToken;
}
```

©2021 AMERICAN EXPRESS. ALL RIGHTS RESERVED.

Request Parameters:

Name	Type	Description
validationData	String Required	OTP value entered by the user.

Response Attributes:

Name	Type	Description
idToken	JWT Required	SRC IDToken created by the SRC System after successful consumer authentication

Business Errors:

Reason Code	Description
CODE_INVALID	The supplied OTP value was invalid. Try again.
CODE_EXPIRED	The OTP is expired. Regenerate the OTP and try again.
RETRIES_EXCEEDED	The limit for the number of retries for OTP generation was exceeded.
VALIDATION_DATA_MISSING	Validation data missing.
ACCT_INACCESSIBLE	The account exists but is not currently accessible (e.g. is locked).

3.7. Checkout

This method performs checkout using the specified card. If successful, the response contains summary checkout information and, conditionally, an encrypted payload containing PCI and/or PII data, depending on the configuration of the DpaTransactionOptions.

This method is called after the Consumer has chosen a MaskedCard for checkout (resulting via selection of a card from the SRC Candidate List OR as a response to enrollCard() method).

Typically, the SRCi would call back to the DPA to retrieve any additional data that the DPA may

©2021 AMERICAN EXPRESS. ALL RIGHTS RESERVED.

have, e.g. updated DpaTransactionOptions, based on the selected card. If the DPA does return some data via this callback, then the SRCI MUST insert that data without modification into the Checkout Request.

In addition to primary use-case Checkout method also supports card being provided at checkout. When the combined flow is executed the client should provide the encrypted Card object as input parameter instead of ID of digital card identifier. The card will be enrolled to SRC system and used for checkout.

Note

- Amex does not support fields in the checkout request

  i) recipientIdCheckout ii) recipientIdPayload

- SrciActionCode is optional at EMVCo V1.1, but Amex made it conditional
- DpaData is not in EMVco 1.1v, but Amex will use this information to identify merchant name.
- windowRef is optional at EMVCo V1.1, but Amex made it required
- unbindAppInstance is not present in EMV 1.1v, but Amex made it required

```
checkout ({
 optional String srciTransactionId;
 conditional String srcDigitalCardId;
 conditional JWE<Card> encryptedCard;
 conditional JWT idToken;
 conditional DpaTransactionOptions dpaTransactionOptions;
 optional PayloadTypeIndicator payloadTypeIndicatorPayload;
 optional PayloadTypeIndicator payloadTypeIndicatorCheckout;
 optional AssuranceData assuranceData;
 Conditional SrciActionCode srciActionCode
 optional DpaData dpaData; required windowRef;
})
```

©2021 AMERICAN EXPRESS. ALL RIGHTS RESERVED.

```
// Response dictionary {
 required Enum dcfActionCode;
 conditional JWT idToken;
 conditional JWS<CheckoutPayloadResponse>
checkoutResponse;
 required String unbindAppInstance
}
```

Request Parameters:

Name	Type	Description
srciTransactionId	String Optional	Session reference identifier generated by the SRC Initiator. This is a SRCi-specific field that is passed-in by SRCi in the checkout request so that the SRC system sends this back in the checkout response. This field can be used to tie the SRCi/DPA-specific session or transaction information to the srcCorrelationId because both of the fields, srciTransactionId and srcCorrelationId will be passed back in the signed checkoutResponse. It is recommended that this is passed-in by the SRCi in the checkout request.
		When this is passed-in, the SRC system must send the same value back in the checkout response.
srcDigitalCardId	String Conditional	ID of the selected card. This should be provided if any card is selected from the src profile returned by the SRC system

©2021 AMERICAN EXPRESS. ALL RIGHTS RESERVED.

encryptedCard	JWE<Card> Conditional	An encrypted Card object describing the card to be enrolled with the SRC System. Encrypted using the public key of the target SRC system. This is a JWE with a single composite JSON Card object at the root with "card" as the claimset element. This should be provided if a new card is added to the src profile.
idToken	JWT Conditional	3rd party federated identity token is supplied; this may be used to provide the consumer hints in the DCF UX to enter their identity credentials consistently across SRC systems. This is required when consumer enters a new card particularly in the scenarios Multiple IDTokens are returned by an SRC System SRC system never returned SRC profile
srciActionCode	SrciActionCode Conditional	Indicates the action code to be supplied to the DCF. SrciActionCode enum: NEW_USER: new-user flow AUTH_FAILED: consumer identity authentication was attempted but failed
		AUTH_SKIPPED: consumer identity authentication was initiated but was "skipped Conditionality: Must be present if one of the above outcomes occurred or is determined to have occurred by the SRCI.
dpaData	DpaData Optional	DPA Registration data should be provided to SRC system.
windowRef	Window Required	Window handle required to open a custom URI in the popup/iframe window.
dpaTransactionOptions	DpaTransactionOptions Optional	May be supplied, if necessary, to replace the existing DPA transaction options.

©2021 AMERICAN EXPRESS. ALL RIGHTS RESERVED.

assuranceData	String Optional	May be supplied to support risk management
payloadTypeIndicatorCheckout	Enum Optional	Type of encrypted payload to be returned by the checkout invocation. • SUMMARY • FULL • PAYMENT • NON_PAYMENT
payloadTypeIndicatorPayload	Enum Optional	Type of encrypted payload to be created for the retrieval by the Payload API • FULL • PAYMENT

Response Attributes:

Name	Type	Description
CheckoutPayloadResponse	JWS(CheckoutPayloadResponse) Conditional	This is the body of the CheckoutPayloadResponse. See CheckoutPayloadResponse for more info. Conditionality: Only supplied in case the dcfActionCode is "COMPLETE".
dcfActionCode	Enum Required	Indicates the action code received from the DCF. DcfActionCode enum: "COMPLETE": DCF processing completed normally "CHANGE_CARD": consumer wishes to select an alternative card "CANCEL": consumer wishes to cancel the flow processing cannot continue

©2021 AMERICAN EXPRESS. ALL RIGHTS RESERVED.

unbindAppInstance	String Required	This is required field and True if user wants to unbind from all the devices on all the network.
idToken	JWT Conditional	This is a conditional field and will be populated with the idToken if unbindAppInstance is set to true.

Business Errors:

Reason Code	Description
CARD_ADD_FAILED	Unable to add card.
CARD_SECURITY_CODE_MISSING	Card security must be supplied.
CARD_INVALID	Invalid card number.
CARD_EXP_INVALID	Invalid card expiry date.
CARDID_MISSING	The card ID was required but is missing.
CARD_NOT_RECOGNIZED	The specified card was not recognized.
ACCT_INACCESSIBLE	The account exists but is not currently accessible (e.g. is locked).
MERCHANT_DATA_INVALID	Invalid merchant data.
UNABLE_TO_CONNECT	Unable to connect to DCF.
TERMS_AND_CONDITIONS_NOT_ACCEPTED	Terms and Conditions not accepted
GUEST_CHECKOUT_NOT_SUPPORTED	Amex has no support for a guest checkout if SRC is not able to validate the user.

©2021 AMERICAN EXPRESS. ALL RIGHTS RESERVED.

3.8. Unbind app instance

This method disassociates the Consumer Device from the Consumer's SRC Profile.

When a consumer unbinds himself from Amex DCF, checkout API will provide IdTokens and set unbindAppInstance to true. This will be true for any dcfActionCode, which means whether that consumer completes the transaction or not, he/she will be unremembered from SRC.

Note

- In EMV 1.1v, idTokens is of type JWT, but Amex made it of type List<JWT>
- idTokens is optional in EMV1.1v, but Amex made it required
- srcCorrelationId is optional in EMV1.1v, but Amex made it required

```
unbindAppInstance (
    required List<JWT> idTokens
)
```

```
// Response dictionary
{
  required String srcCorrelationId
}
```

Request Parameters:

Name	Type	Description
idTokens	List<JWT> required	SRCI can provide idTokens received from SRC systems based on consumer recognition performed via OTP (for unrecognized flow) or cookie (for recognized flow). SRCI aggregate idTokens from multiple consumers on a given device from SRC systems and provide to individual SRC systems to fetch card list.

©2021 AMERICAN EXPRESS. ALL RIGHTS RESERVED.

Response Attributes:

Name	Type	Description
srcCorrelationId	UUID required	This is the unique identifier generated by SRC system to track and link SRC messages. This will be used as the transaction identifier assigned by the SRC System for transaction. this particular

Business Errors:

Reason Code	Description
AUTH_INVALID	Invalid Authorization

3.9. Payload API

The Get Payload operation allows the SRC Initiator to retrieve the payload data from the SRC System for authorization purposes.

The Get Payload operation is server-side API intended for server-based communication.

In case “payloadTypeIndicator” in this request or “payloadTypeIndicatorPayload” in the Checkout API request is set to SUMMARY this request should return error code 400 with error description indicating illegal request.

Note
• Accept: This is an optional element for SRCi to pass version number of specification (e.g. 1.1) they want to execute. Amex internally setup base version and supported versions for every SRCi after mutual agreement. After this setup SRCi can switch between supported versions, otherwise it will be defaulted to base version. • PayloadTypeIndicator is optional in EMV 1.1v, but Amex made it required

©2021 AMERICAN EXPRESS. ALL RIGHTS RESERVED.

```
payload (
 required String srcClientId;
 required String srcCorrelationId;
 required String payloadTypeIndicator;
 optional String recipientId;
 optional String srciTransactionId;
 optional String accept;
 conditional String srcDpaId;
 optional String serviceId;
)

// Response
{
 required JWS<CheckoutPayloadResponse> payloadResponse;
}
```

Request Parameters:

Name	Datatype	Description
srcClientId	String Required	The unique identifier assigned to the SRC entity by the SRC System during manual onboarding. Identifies the connecting client, e.g. SRCI, DCF, SRCPI. For SRCi it is always Srci initiator id (generated during manual onboarding)
SrcCorrelationId	String Required	SRC Correlation Id corresponding to this SRC checkout transaction
PayloadTypeIndicator	String Required	The verbosity of payload requested. The possible values are • FULL • PAYMENT Any other value other than the above will result HTTP 400 error.

©2021 AMERICAN EXPRESS. ALL RIGHTS RESERVED.

RecipientId	String Optional	The expected value is the key identifier (kid) which will be used to pull the public certificate for payload encryption. Certificate can reside locally with SRC system if requesting party does not have Well-Known url facility else SRC
		system would go to well-known url of SRCi to fetch the payload encryption key. The SRCi can share their well-known url at time of manual onboarding.
SrciTransactionId	String Optional	SrciTransactionId generated by the SRC Common Software or SRC initiator. If not provided SRC system still can process the transaction
Accept	String Optional	Version information for server API ie. version=1.1

Response Attributes:

Name	Type	Description
payloadResponse	JWS<CheckoutPayloadResponse> Required	This is the encrypted CheckoutPayloadResponse. For the structure of this element, see CheckoutPayloadResponse.

3.10. Confirmation API

This API is provided by the SRC System to the SRC Initiator to confirm the result of checkout (user action) as well as results from payment authorization. This call is on server side

©2021 AMERICAN EXPRESS. ALL RIGHTS RESERVED.

Note
Amex does not support following elements in Confirmation API • srcDpaId • serviceId

```
confirmation (
    required String srcClientId;
     required String SrcCorrelationId;
    required                    ConfirmationData
confirmationData;
     optional String SrciTransactionId;
)
// Response  - empty
```

Request Parameters:

Name	Type	Description
srcCorrelationId	String, required	A unique SRC Correlation ID generated by the SRC System.
srciTransactionId	String, Optional	A Unique id to track API request end to end.
confirmationData	ConfirmationData, required	Confirmation Data from SRCI to SRC System. This is to confirm the results of transaction or the outcome of payment authorization. See Data Structure section for details.
srcClientId	String required	The unique identifier assigned to the SRC entity by the SRC System during manual onboarding. Identifies the connecting client, e.g. SRCI, DCF, SRCPI. For SRCi it is always Srci initiator id (generated during manual onboarding)

©2021 AMERICAN EXPRESS. ALL RIGHTS RESERVED.

3.11. Authenticated Data Token for Checkout Response

The purpose of this authentication data token is to provide authentication and integrity protection to the SRC summary payload.

- The authenticated data is valid only for one particular request
- It is valid maximum for only 1 minute
- This is recommended to be used as an authorization header with PoP (Proof-of-Possession) authentication scheme to connect to the SRC Initiator servers to fetch the full merchant payload.

Table 3.1: Authenticated Data Token

Header

Parameter Name	M/O	Description
alg	M	Algorithm used to digitally sign the JWS payload according to JWS specification. Signing algorithm needs to be RS256/HS256, specific to an SRC System.
kid	M	Id of the cryptographic public key of SRC System signing the checkout request. Relaying party SHOULD use the ID to select appropriate key to verify the signature. Type of the public key identified by the ID MUST match type of the signing algorithm.

©2021 AMERICAN EXPRESS. ALL RIGHTS RESERVED.

Checkout Claimset

Claim Name	Cardinality	Notes
iss	1	The value has to be URI or other ID of the SRC System that generated this JWS. The format of the ID is specific to SRC Programme. Sample value: https://srcsystem1.com
aud	1..n	This is the URI or other ID of the SRCI that this payload is sent to. The format of the ID is specific to SRC Programme. Sample value: https://srci.com
exp	1	Expiration time in UTC and unix/epoch format
iat	1	Issuance time in UTC and unix/epoch format Time at which the JWS was issued. This should not be before the current date/time.
jti	0..1	The "jti" (JWS ID) claim provides a unique identifier for the JWS. The value is a casesensitive string.
srcCorrelationId	1	SRC Correlation Id corresponding to this SRC checkout transaction
srciTransactionId	0..1	Populated if there is one provided in the input
maskedCard	0..1	This is a complex JSON object, MaskedCard
shippingAddressZIP	0..1	Shipping address zip code value
shippingCountryCode	0..1	Shipping country code value
maskedConsumer	0..1	This is a complex JSON object, MaskedConsumer
encryptedPayload	0..1	JWE/Encrypted payload that contains payment and/or consumer data.

©2021 AMERICAN EXPRESS. ALL RIGHTS RESERVED.

3.11.1. Successful responses

A successful response means that the promise is fulfilled. The action relating to the promise has succeeded.

3.12. Errors

An error response means that the promise is rejected. The action relating to the promise has failed.

API errors can be Business Errors or Standard Errors. An error object is passed for every error.

The structure of the error is described in the next section.

Standard errors can be returned by any API and should be handled in a common way. Business errors are returned by some APIs only and are described in the Errors section for each API.

Section Error Handling provides a guideline how to handle different errors.

3.12.1. Error Structure

This is the Standard Response Structure, which will be returned in case of an error.

Use “error.reason” field to drive your error handling logic.

Errors are also provided with a human readable error description in “error.message” field. This field should be used only to understand the problem.

```
dictionary error {
   "message": "Input parameters validation failed.",
   "reason": "invalidParameter",
   "details":
   [ // Optional structure, used with input data validation error
     { // Types to specify the fields with errors
       "location": "creditCard",
       "message": "Should be a numeric value"
     }
   ]
 }
```

©2021 AMERICAN EXPRESS. ALL RIGHTS RESERVED.

Name	Type	Description
message	String Optional	Every error should have a human readable message describing the error. The message can very even for the same reason, thus providing additional insides about the problem. This approach should be used with client bugs to avoid introducing new reason and still provide enough information to API consumer and helping him/her to fix the bug. API providers are free to change the message at any point of time. API consumers should not use this field to drive their business logic or to expose it directly to the clients via UI or other means
reason	String Required	API consumers should use this field to drive their error handling business logic
details	Object Optional	Array of fields - Data validation errors should use this field to specify which fields failed the validation using location field.
details.location	String Required	The value of this field is using XPATH expression to point to the field, which failed validation.
details.message	String Optional	The specific error for this field.

3.12.2. Standard Errors

These errors can be returned by any API and should be handled consistently in a unified way. Standard Errors are not going to be documented as part of each individual API description.

Reason Code	**Description & What to do**
UNKNOWN_ERROR	Unknown error

©2021 AMERICAN EXPRESS. ALL RIGHTS RESERVED.

REQUEST_TIMEOUT	Request timeout
SERVER_ERROR	General server error
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 field. Notes: Whenever possible should provide client-side validation to avoid round trip to the server. Simple validation constraints are documented as part of the API specification Handle this error by prompting the user to change the value. If the value was not provided by a user
INVALID_REQUEST	The server could not even understand the request. Usually, these are the cases, when some data field must 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/field of the request is considered incorrect. Please, refer to the API specification for the structure, format, and constraints on the API request.
AUTH_ERROR	The server does understand the request but cannot authenticate.
NOT_FOUND	The requested resource/business entity does not exist. The resource might also be hidden for security reasons.

©2021 AMERICAN EXPRESS. ALL RIGHTS RESERVED.

RATE_LIMIT_EXCEEDED	Too many requests have been sent in a given amount of time. Intended for use with rate limiting schemes. Wait before sending the next request and decrease the rate of sending API requests. Consider implementing Exponential Backoff. In this algorithm, the delay before you retry is defined as: retryDelay = (2 ^ N) * 1000 + randomDelayMs retryDelay Retry delay in milliseconds. N your retry count (0, 1, 2, 3, …). randomDelayMs random delay in milliseconds (0 .. 1000).
SERVICE_ERROR	Some error happened on the server which caused the error. Either show a generic message or retry the same request again (it might succeed).

3.12.3. Error Handling

This section provides a set of guidelines how to handle API errors for the purpose of simplifying API consumer error handling logic and achieving backward compatibility with the newer versions of the APIs.

It is assumed the API consumer follow these guidelines.

- Drive your error handling logic based on the value of errorResponse.reason field
- Field errorResponse.message is for informational purpose only ○ Do not parse this field to drive your error handling logic ○ Do not expose this field to the end user either
- It is assumed the there is a default "catch all" type of error handling logic

©2021 AMERICAN EXPRESS. ALL RIGHTS RESERVED.

- Any errors with codes, “invalidParameter” or “invalidRequest” status codes are considered as integration errors. You can choose to show a message, or just fail processing this operation

4. JWE Details and Composition

JSON Web Encryption (JWE) is used to encode sensitive information. The following naming convention should be followed for fields requiring encryption in this document.

“enc<FIELD>” : “JWE (…) “

The JSON Web Encryption (JWE) content should be signed or encrypted using the shared secret that was provided to client at the time of onboarding.

Notes

- Compact serialization style (elements separated by “.”)
- All fields are base 64 – Url Safe encoded.
- Use a CEK (Content Encryption Key) of 256bits size.
- Use AES-GCM-256KW algorithm for encryption of content encryption key and Use an Initialization Vector (IV) for encryption. Size of IV is to be 96 bit.
- Use AES-GCM-256 algorithm for encryption of payment instrument and Use an Initialization Vector (IV) for encryption. Size of IV is to be 96 bit.
- Authentication Tag will be generated as an additional output of the AES-GCM-256 encryption. Size of this field is 128 bits.
- All String to byte and vice-versa conversions are done with UTF-8 charset.

©2021 AMERICAN EXPRESS. ALL RIGHTS RESERVED.

4.1. Sample JWE using API Key/Shared Secret

JWE Header

```
 "header": {
   "alg": "AGCM256KW", // Encryption algorithm to be used for encryption of CEK
   "iv":  "<SizeofIVshouldbe96bit.>", // IV to be used for encryption of CEK
   "tag": "<128bitvalue>", // HMAC generated from applying AES-256-GCM-KW to the CEK
   "kid": "50charAPIKey", // API key
   "enc": "AGCM256",
   "iat": "1429837145"
 }
```

JWE Body

//base64 encoded form. CEK encrypted using AGCM256KW (alg) algorithm and the CEK IV

“encrypted_key”: “UghlOgu ... MR4gp_A=”,

//base64 encoded form. IV for the text encryption. Size of IV is to be 96 bit Base64 encoded form

“iv”: “AxY8DctDa….GlsbGljb3RoZQ=”,

//Encrypted blob generated using the AES-GCM encryption (enc) of the text to encrypt

“ciphertext”: “KDITthhZTGufMY…….xPSUrfmqCHXaI9wOGY=”,

// base64 encoded form . HMAC generated using the AES-GCM encryption of the text to encrypt. The size of the tag should be 128 bits.

“tag”: “Mz-VPPyU4…RlcuYv1IwIvzw=”

Note: The JWE Protected Header is input as the AAD (Additional Authenticated Data) parameter of the authenticated encryption (AES-GCM) of the “text to encrypt”.

JWE composition

BASE64URL (UTF8 (JWE Header)) || ‘.’ ||

BASE64URL (JWE Encrypted Key) || ‘.’ ||

©2021 AMERICAN EXPRESS. ALL RIGHTS RESERVED.

BASE64URL (JWE IV) || '.' ||
BASE64URL (JWE Ciphertext) || '.' ||

BASE64URL (JWE Authentication Tag)

JWE/JWS specification requires BASE64URL encoding with NO padding.

5. Data Masking Rules

The following masking rules are followed wherever applicable in the APIs:

- Plaintext exposure cannot exceed 3 characters or 30% of the total PI field length, whichever is shorter
- Masking of the remaining 70% can either be the first 70% of the PI field, the last 70% of the PI field, or a random 70% of the field
- Use the special character, * for masking the characters when being presented to enforce consistency in masking logic
- If the PI field is email address, the domain name should be left in the clear

©2021 AMERICAN EXPRESS. ALL RIGHTS RESERVED.