EMV® Specification Bulletin No. 241
First Edition October 2019

Secure Remote Commerce Specification – JavaScript SDK

This Specification Bulletin updates EMV® Secure Remote Commerce Specification – JavaScript SDK, version 1.0 to correct errors in the original specification.

Applicability

This Specification Bulletin applies to:

- *EMV® Secure Remote Commerce Specification – JavaScript SDK, version 1.0*

Related Documents

- *EMV® Secure Remote Commerce Specification*
- *EMV® Secure Remote Commerce Specification – API*

Effective Date

- *The date of publication*

Description

With the publication of the *EMV® Secure Remote Commerce Specification, version 1.0*, EMVCo published two further documents:

- *EMV® Secure Remote Commerce Specification - API*
- *EMV® Secure Remote Commerce Specification – JavaScript SDK*

These describe:

- APIs to be used for the transmission of data between SRC Systems to SRC System Participants.
- SRC JavaScript SDK for the SRC Initiator, provided as source code to be consumed by the SRC Common Software

This bulletin updates the JavaScript SDK specification to correct minor errors in version 1.0 and to improve consistency between the JavaScript SDK and API specifications.

In the changes that follow, new text is highlighted in red. For clarify, deleted text is not shown.

Specification Changes

In Section 2.2.1, Table 2.2.2, update the entry for *srciTransactionId* to read as follows:

srciTransactionId Type: String	O	Transaction-unique identifier assigned by the SRCI.

In Section 2.4, Table 2.4.1 change the Response parameter *srcCorrelationId* from required to optional. The Response definition should now read:

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

In Section 2.4.2, Table 2.4.3, update the entry for *srcCorrelationId* to read as follows:

srcCorrelationId Type: String	O	SRC Correlation ID may be returned by SRC System. If it is returned, it must be quoted in all subsequent calls to the SRC System within the same transaction.

In Section 2.5, Table 2.5.1, add a new parameter, *supportedValidationChannels*, to the response dictionary. The response dictionary should now read:

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

In Section 2.5.2, Table 2.5.3, add the following row at the bottom of the table:

supportedValidationChannels Type: List<IdentityValidationChannel>	O	List of supported validation channels. This may be returned by the SRC System to indicate the channels supported.

In Section 2.8, Table 2.8.1, add two new parameters, *encryptedCard* and *threeDsInputData* to the method definition and change the parameter *srcCorrelationId* in the response dictionary from conditional to optional. The table should now read:

```
enrollCard ({
      optional String srciTransactionId;
      required JWE encryptedCard;
      optional JSONObject threeDsInputData;
      conditional JWT idToken;
})
```

```
// Response
dictionary {
      required MaskedCard maskedCard;
      optional String srcCorrelationId;
}
```

In Section 2.8.1, Table 2.8.2, update the entry for *srcTransactionId* to read as follows, add a new parameter (threeDSInputData) as the third row in the table and add four new entries as the last four rows of the table.

srciTransactionId Type: String	O	Transaction-unique identifier assigned by the SRCI.

threeDsInputData Type: JSONObject	O	Merchant's 3DS input data.

In Section 2.8.2, Table 2.8.3, update the entry for *srcCorrelationId* to read as follows:

srcCorrelationId Type: String	O	SRC Correlation ID may be returned by SRC System. If it is returned, it must be quoted in all subsequent calls to the SRC System within the same transaction.

In Section 2.9, Table 2.9.1:

- In the Method definition, change the parameter *srcCorrelationId* from optional to conditional
- In the Response dictionary, replace the parameters *checkoutResponseJWS* and *checkoutResponse*, with a single parameter, checkoutResponse

- Delete the CheckoutResponse dictionary and replace with the following text: “For details of the CheckoutPayloadResponse object, refer to SRC API Section 2.4.2 Checkout Payload Response.”

The table should read as follows:

```
checkout ({
      optional String srciTransactionId;
      conditional String srcCorrelationId;
      conditional String srcDigitalCardId;
      conditional JWE<Card> encryptedCard;
      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 Window windowRef;
})
```

```
// Response
dictionary {
      required DcfActionCode dcfActionCode;
      conditional JWT idToken;
      conditional JWS<CheckoutPayloadResponse> checkoutResponse;
}
For details of the CheckoutPayloadResponse object, refer to SRC API
Section 2.4.2 Checkout Payload Response.
```

In Section 2.9.1, Table 2.9.2, update the entries for *srciTransactionId, srcCorrelationId* and *srciActionCode* to read as follows:

srciTransactionId Type: String	O	Transaction-unique identifier assigned by the SRCI.
srcCorrelationId Type: String	C	Correlation identifier. An identifier previously provided by the SRC System to which the card is being enrolled and/or with which checkout is occurring.

		Conditionality: must be provided if SRC System returned it in prior calls within the same transaction, otherwise, it is not required
srciActionCode Type: SrciActionCode	C	Action code. A code to be supplied to the DCF indicating a specific action based on the following: 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.

In Section 2.9.2, Table 2.9.3, delete the entry for *checkoutResponseJWS* and update the entry for *checkoutResponse* to read as follows:

checkoutResponse Type: JWS <CheckoutPayloadResponse>	C	checkoutResponse signed structured as defined in SRC API Section 2.4.2 Checkout Payload Response. **Conditionality**: Only supplied in case the dcfActionCode is "COMPLETE"

Delete Section 2.9.3 CheckoutResponse Type Definitions, including Table 2.9.4

Renumber:

- Section 2.9.4 Application Errors to Section 2.9.3 Application Errors
- Table 2.9.5 Application Errors to Table 2.9.4 Application Errors

Change the reference from "*The application errors are given in Table 2.9.5"* to "*The application errors are given in Table 2.9.4"*

Section 2.9.3 now reads:

2.9.3 Application Errors

The application errors are given in Table 2.9.4.

Table 2.9.4: Application Errors

The contents of the table remain as is.

In Section 2.10, Table 2.10.1, in the response dictionary, change the parameter *srcCorrelationId* from required to optional. The response dictionary should now read:

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

In Section 2.10.2, Table 2.10.3, update the entry for *srcCorrelationId* to read as follows:

srcCorrelationId Type: String	O	SRC Correlation ID may be returned by SRC System if it is within a particular checkout transaction context.

In Section 2.11, Table 2.11.1, in the response dictionary, change the parameter *srcCorrelationId* from required to optional. The response dictionary should now read:

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

In Section 2.11.2, Table 2.11.3, update the entry for *srcCorrelationId* to read as follows:

srcCorrelationId Type: String	O	SRC Correlation ID may be returned by SRC System if it is within a particular checkout transaction context.

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