

SDK response update 3.1.0.0

As part of the SDK update, starting now we will generate standardized response objects, including the full detail of fields and their possible values.

Additionally, we improved the tagging logic to correctly detect icons and related events. With these improvements, when a spoofing-related event is detected, a transaction will be generated and the event will be recorded with its corresponding details.

If spoofing is detected, an evidence image will also be sent to the customer-provided **webhook**. Moreover, from now on the transaction details (including the transaction number) will be delivered through the webhook to ensure an auditable record of the event.

Expected response: Transaction details sent to the webhook.

SDK response level (PSA): This information will be returned by the SDK as part of the standardized response object.

```
{
  "processType": "liveness",
  "statusCode": 1,
  "detail": {
    "status": "approved",
    "reason": {
      "name": "NOT_ALERT",
      "description": "The transaction was approved."
    },
  },
  "uidDevice": "1da62007d8974a9fa71e18cba76f6fc1",
  "keyProcessLiveness": "69c265aeca9d4863b61b1fc6a8099f52",
  "processId": "3f2b9c1e-7a4d-4f2a-9c0c-6c4c2e9b8d12",
  "image": {
    "encoding": "base64",
    "mimeType": "image/jpeg",
    "data": "<BASE64>"
  }
}
```

Field description

- **processType**: Type of process executed.
 - Possible values: "liveness" | "cardcapture"
 - Notes:
 - If `processType = "liveness"`, the field `keyProcessLiveness` **may be present**.
 - If `processType = "cardcapture"`, `keyProcessLiveness` **must NOT be returned**.
- **Side**: Indicates which side of the document was captured.
 - Possible values: "front" | "back"
 - Notes:
 - If `processType = "liveness"`, the field **Side must NOT be returned**.
 - If `processType = "cardcapture"`, **Side may be present**.
- **statusCode**: Numeric result code of the process.
 - 1 = approved
 - 2 = rejected
 - 3 = failed
- **detail**: Object containing the detailed outcome.
 - **status**: Text status of the result.
 - Possible values: "approved" | "rejected" | "failed"
 - **reason**: Human-readable reason for the result.
 - **name**: Short label describing the reason (e.g., "No alerts").
 - **description**: Longer explanation of the reason (e.g., "The transaction was approved.").
 - **uidDevice**: Unique device identifier (string).
 - **keyProcessLiveness** (*only for liveness*): Unique identifier for the liveness process (string).
 - Returned **only** when `processType = "liveness"`.
 - **processId**: Unique process identifier (GUID/UUID) generated per execution to correlate SDK responses and webhook events.
 - **image**: Image object associated with the process.
 - **encoding**: Encoding type used for the image payload.
 - Typical value: "base64"
 - **mimeType**: MIME type of the image data.
 - Examples: "image/jpeg", "image/png"
 - **data**: Base64-encoded image content (string).
 - Usually provided **without** the `data:image/...;base64,` prefix.

The component's possible responses are split by functionality: CardCapture (document capture) or **Liveness** (proof of life). Below is a practical, complete list of outcomes you should handle (success, pending, failures, cancellations, and technical errors).

Liveness

Status (code)	Nombre (name)	Descripción (description)	Objeto (reason)
NONE	NOT_ALERT	The transaction was approved with no alerts detected.	<code>{"code":"NONE","name":"NOT_ALERT","description":"The transaction was approved with no alerts detected."}</code>
NOT_ALIVE	Not alive detected	The liveness check indicates the subject is not alive.	<code>{"code":"NOT_ALIVE","name":"Not alive detected","description":"The liveness check indicates the subject is not alive."}</code>
CLOSED_EYES	Eyes closed detected	The subject's eyes were detected as closed during the capture.	<code>{"code":"CLOSED_EYES","name":"Eyes closed detected","description":"The subject's eyes were detected as closed during the capture."}</code>
SPOOFING	Spoofing detected	A potential spoofing attempt was detected during the liveness process.	<code>{"code":"SPOOFING","name":"Spoofing detected","description":"A potential spoofing attempt was detected during the liveness process."}</code>
PAD	Presentation attack detected (PAD)	A potential presentation attack was detected (PAD confidence triggered).	<code>{"code":"PAD","name":"Presentation attack detected (PAD)","description":"A potential presentation attack was detected (PAD confidence triggered)."} </code>
PERMISSION_DENIED	Camera permission denied	Camera access was denied by the user or the operating system.	<code>{"code":"PERMISSION_DENIED","name":"Camera permission denied","description":"Camera access was denied by the user or the operating system."}</code>
CAMERA_NOT_FOUND	Camera not found	No camera was detected on the device, or it is unavailable.	<code>{"code":"CAMERA_NOT_FOUND","name":"Camera not found","description":"No camera was detected on the device, or it is unavailable."}</code>
FACE_NOT_DETECTED	Face not detected	No face was detected in the camera frame.	<code>{"code":"FACE_NOT_DETECTED","name":"Face not detected","description":"No face was detected in the camera frame."}</code>
NETWORK_ERROR	Network error	A network issue prevented the process from completing.	<code>{"code":"NETWORK_ERROR","name":"Network error","description":"A network issue prevented the process from completing."}</code>
SDK_ERROR	SDK error	An internal SDK error occurred during the process.	<code>{"code":"SDK_ERROR","name":"SDK error","description":"An internal SDK error occurred during the process."}</code>
USER_CANCELED	User canceled	The user canceled the process before completion.	<code>{"code":"USER_CANCELED","name":"User canceled","description":"The user canceled the process before completion."}</code>

Example:

As an example response, the JSON below represents a **rejected transaction (Status 2)**. The **reason** field may come as an **object**, including the **rejection reason details** and its **description**.

It also includes identifiers such as **uidDevice** and **keyProcessLiveness**, plus **processId** (the unique process UID).

Evidence (e.g., an image) can be returned as **Base64**, including the **encoding type**.

```
{
  "processType": "liveness",
  "statusCode": 2,
  "detail": {
    "status": "rejected",
    "reason": {
      "name": "SPOOFING",
      "description": "The liveness check indicates the subject is not alive."
    },
    "uidDevice": "1da62007d8974a9fa71e18cba76f6fc1",
    "keyProcessLiveness": "69c265aeca9d4863b61b1fc6a8099f52",
    "processId": "3f2b9c1e-7a4d-4f2a-9c0c-6c4c2e9b8d12",
    "image": {
      "encoding": "base64",
      "mimeType": "image/jpeg",
      "data": "<BASE64>"
    }
  }
}
```

CardCapture

Status (code)	Nombre (name)	Descripción (description)	Objeto (reason)
NONE	NOT_ALERT	The transaction was approved with no alerts detected.	{ "code": "NONE", "name": "NOT_ALERT", "description": "The transaction was approved with no alerts detected." }
PERMISSION_DENIED	Camera permission denied	Camera access was denied by the user or the operating system.	{ "code": "PERMISSION_DENIED", "name": "Camera permission denied", "description": "Camera access was denied by the user or the operating system." }

Status (code)	Nombre (name)	Descripción (description)	Objeto (reason)
CAMERA_NOT_FOUND	Camera not found	No camera was detected on the device, or it is unavailable.	{ "code": "CAMERA_NOT_FOUND", "name": "Camera not found", "description": "No camera was detected on the device, or it is unavailable." }
CAMERA_IN_USE	Camera in use	The camera is currently being used by another application.	{ "code": "CAMERA_IN_USE", "name": "Camera in use", "description": "The camera is currently being used by another application." }
NOT_SUPPORTED	Device not supported	The device does not meet the requirements for this capture process.	{ "code": "NOT_SUPPORTED", "name": "Device not supported", "description": "The device does not meet the requirements for this capture process." }
DOCUMENT_NOT_DETECTED	Document not detected	No document was detected in the camera frame.	{ "code": "DOCUMENT_NOT_DETECTED", "name": "Document not detected", "description": "No document was detected in the camera frame." }
NETWORK_ERROR	Network error	A network issue prevented the process from completing.	{ "code": "NETWORK_ERROR", "name": "Network error", "description": "A network issue prevented the process from completing." }
SDK_ERROR	SDK error	An internal SDK error occurred during the process.	{ "code": "SDK_ERROR", "name": "SDK error", "description": "An internal SDK error occurred during the process." }
USER_CANCELED	User canceled	The user canceled the process before completion.	{ "code": "USER_CANCELED", "name": "User canceled", "description": "The user canceled the process before completion." }
FOCUS_LOST	Focus lost / component interrupted	The process was interrupted because the app lost focus or the component was closed, minimized, or moved to the background.	{ "code": "FOCUS_LOST", "name": "Focus lost / component interrupted", "description": "The process was interrupted because the app lost focus or the component was closed, minimized, or moved to the background." }

Example:

For the **document capture** process, it will be mapped under **processType: "cardCapture"** and will use **statusCode: 3**, which groups any **cancellation scenario** that affects the execution of the flow, for example:

- **Process cancelled by the user.**
- **Component focus loss during the session** (e.g., the app goes to the background or the user navigates to another screen).

The response will include the **reason** (cancellation reason) and **status** (state), as well as the **processId** (unique process identifier) and its **linked ID** (if applicable). Additionally, the **imageDetail** object will be returned, where the result of the processed image will be provided.

```
{
  "processType": "cardcapture",
```

```

"side": "front",
"statusCode": 3,
"detail": {
  "status": "failed",
  "reason": {
    "name": "USER_CANCELED",
    "description": "The user canceled the process before completion."
  },
  "uidDevice": "1da62007d8974a9fa71e18cba76f6fc1",
  "processId": "3f2b9c1e-7a4d-4f2a-9c0c-6c4c2e9b8d12",
  "image": {
    "encoding": "base64",
    "mimeType": "image/jpeg",
    "data": "<BASE64>"
  }
}
}
}

```

This process will trigger an **event to the webhook**, where a record will be **registered** and linked by a **UID** (for example, `processId` as the unique process identifier). The relationship between the **transaction** and the **webhook event** will be kept for traceability.

The payload sent will follow the structure below. This response **only applies to the webhook** when `reasonCode` is one of the following values: **SPOOFING**, **PAD**, **CLOSED_EYES**, **NOT_ALIVE**.

```

{
  "transactionId": "10743",
  "processId": "3f2b9c1e-7a4d-4f2a-9c0c-6c4c2e9b8d12",
  "transactionStatus": "rejected",
  "stages": [
    {
      "processType": "liveness",
      "status": "Spoofing",
      "reason": {
        "name": "SPOOFING",
        "description": "A potential spoofing attempt was detected during the liveness process."
      }
    },
  ]
}

```

```
"uidDevice": "1da62007d8974a9fa71e18cba76f6fc1",  
"keyProcessLiveness": "69c265aeca9d4863b61b1fc6a8099f52",  
"processId": "3f2b9c1e-7a4d-4f2a-9c0c-6c4c2e9b8d12"  
  }  
]  
}
```

[webhook-risk-validation.yaml](#)

Revision #12

Created 26 February 2026 20:11:00 by roger de avila

Updated 6 March 2026 15:49:52 by roger de avila