Click to Pay

General overview

Click to Pay is a universal digital wallet for payment cards, designed to work seamlessly across a wide range of online retailers. It offers a faster, safer, and more convenient checkout experience by allowing cardholders to pay with their credit or debit cards without re-entering details for every purchase. The Click to Pay solution is based on the EMVCo Secure Remote Commerce Specifications and supported by major card payment networks such as Visa and Mastercard.

How it works

1. **One-time setup:** The cardholder registers its card(s) with Click to Pay through its card issuer.
2. **At checkout:** When the cardholder is shopping online and sees the Click to Pay icon, it can be chosen as the payment method.
3. **Quick access:** After verifying the cardholder's identity, the cardholder picks the saved card in Click to Pay and completes the payment—no need to re-enter numbers or billing addresses.

Key Benefits

- **Speed:** Faster checkout with fewer steps.
- **Security:** Uses tokenization and built-in fraud protection.
- **Convenience:** Works across multiple merchants and supports various cards under one login.

DECTA solution for Click to Pay AVALIABLE ONLY FOR VISA TOKENIZED CARDS

DECTA provides Visa card issuers with multiple endpoints to enroll and manage a cardholder's tokenized card in the Visa Click to Pay service in accordance with Visa requirements. DECTA has integration with the Visa Click to Pay service and acts as an intermediary between the card issuer and the Visa Click to Pay service

Pre-conditions to start using DECTA API Click to Pay endpoints

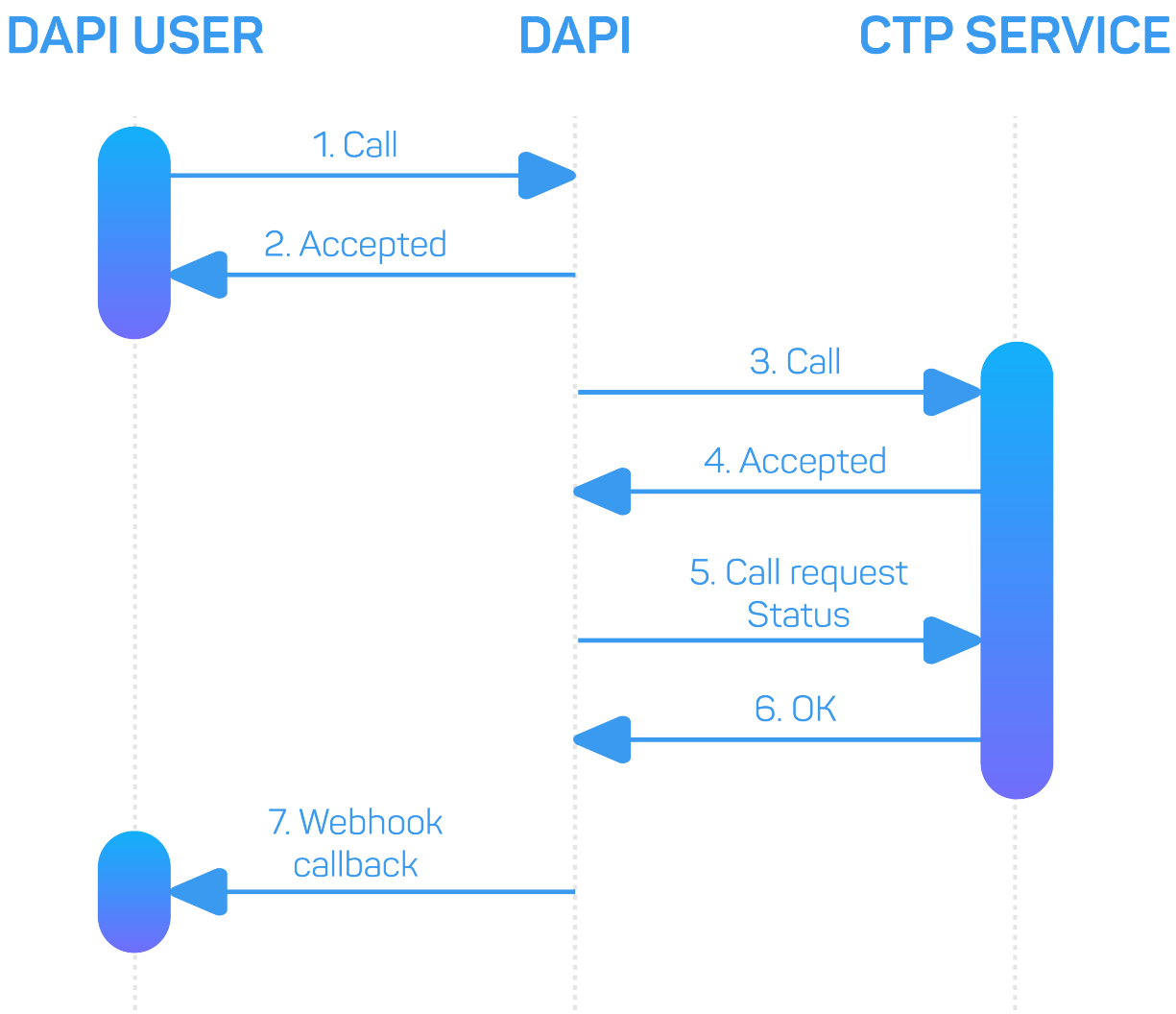
- Completed Visa card tokenization project at least for one digital wallet;
- Opened Click to Pay implementation project with Visa;
- Configured test environment in Visa Developer Center (VDP) for Click to Pay project;

- Assigned access rights by DECTA

Important:

- In the enrollment process, you can use your own cardholder's profile data or existing data that have been saved in the Card Management System on the DECTA side. **The preferred source of profile data should be agreed upon with your DECTA Manager before access rights are granted by DECTA.**
- If you use existing data saved in the Card Management System to enroll the card, then you should NOT send a payload.
- The postcode (zipCode) field must be no longer than 9 symbols. If you use saved data in the Card Management System, the exceeded postcode length will be truncated to 9 symbols.
- Mobile phone format must follow the ITU-T E.164 (2010) format structure. Do not include the "+" symbol.
- Email value must not contain diacritic symbols, otherwise the request will be rejected by Visa API.
- Address must be a **real**, otherwise the request will be rejected by the Visa Address Verification Service.
- Card object fields must not contain the following symbols: `[]{}:&$^!=;*#`<>”`. Otherwise the request will be rejected by Visa API.

General scheme for DECTA API requests to enroll, manage and delete data in Click to Pay



Get Click to Pay Erolled Data

When you want to receive the cardholder's profile data saved in Click to Pay along with the cards assigned to this profile, you should use the request `GET /v2/api/click-to-pay/{clientId}`. The request is processed synchronously with the Visa Click to Pay service, and all requested data is returned in the response.

Enroll data to Click to Pay

Use `POST /v2/api/click-to-pay/{ppan}` to enroll data to Click to Pay.

This endpoint creates a profile in the Click to Pay system and assigns a card to it. During the saving process Visa tokenizes the card, making it ready for use in Click to Pay functionality.

In response you will receive an "Accepted" status, indicating that the request will be processed asynchronously. Once the request is processed in the Click to Pay service, you will receive a webhook notification containing the result status of the request.

Manage Click to Pay data

Use `PATCH /v2/api/click-to-pay/{ppan}` to manage Click to Pay data.

This endpoint allows you to update data saved in your Click to Pay profile, such as email, mobile phone, billing address, and more. For details on JSON models, please refer to the DECTA API API SPACE documentation.

In response you will receive an "Accepted" status, indicating that the request will be processed asynchronously. Once the request is processed in the Click to Pay service, you will receive a webhook notification containing the result status of the request.

Delete Click to Pay data

Use `DELETE /v2/api/click-to-pay/{ppan}` to delete Click to Pay data.

This endpoint allows you to remove cards from your Click to Pay profile.

Use this endpoint immediately after a card is blocked or when the cardholder wants to remove a card from their Click to Pay profile. Please note that if you delete the last card in a profile, then entire profile will be deleted. To create a new cardholder profile with a new assigned card, use the Enroll data to Click to Pay endpoint.

If the cardholder does not use Click to Pay payment service, then the cardholder should delete their saved data in Click to Pay. Please ensure that unused data is not retained!

In response, you will receive an "Accepted" status, indicating that the request will be processed asynchronously. Once the request is processed in the Click to Pay service, you will receive a webhook notification containing the result status of the request.

Webhook callback overview

This functionality is used to process asynchronous requests from the Click to Pay service. Within the webhook data, the Customer can understand the request's process status (SUCCESS or FAILED) along with additional details. The Customer should create an endpoint on their side that can process incoming requests and send a 200 HTTP code in response. The web service should be ready to receive HTTP POST calls with a payload in JSON format. If the web service uses basic authentication, the credentials should be

provided to DECTA for configuration. Each webhook call represents a separate event message. If the Customer's designated URL is not available or DECTA receives an HTTP error, then DECTA will continue to resend the webhook every 20 seconds for 24 hours.

How to set up?

1. Create webservice to get webhook data and send the endpoint data to your DECTA manager.
2. DECTA will configurate your profile.

How to use?

1. Send request to `POST /v2/api/click-to-pay/{ppan}` or `PATCH /v2/api/click-to-pay/{ppan}` or `DELETE /v2/api/click-to-pay/{ppan}`.
2. Get response 202 Accepted.
3. Wait for webhook with request result status.
4. Map incoming response with initial request using `requestId` from request header and `requestId` from response payload.

Webhook payload example:

```
{
   "endpoint": "UPDATE_CLICK_TO_PAY_DATA",
   "requestId": "11a11b1c-1d11-11ef-1g11-h1fi1jk11l11",
   "data": [{
           "status": "FAILED",
           "clientId": "CR0000000000000001",
           "errors": [{
                   "code": "DAPI-ISS-ERR:0104",
                   "message": "Incoming data processing error",
                   "details": "Click To Pay Profile card data update failed",
                   "link": "https://dapidocs.decta.com/Additional-info/Errors#DAPI-ISS-
ERR:0104"
               }
           ]
       },
       {
           "status": "SUCCESS",
           "clientId": "CR0000000000000001",
           "details": "Client data successfully updated in Click to Pay profile"
       }
   ]
}
```

Data models

Main webhook body

```
{
    "endpoint": "UPDATE_CLICK_TO_PAY_DATA",
    "requestId": "11a11b1c-1d11-11ef-1g11-h1fi1jk11l11",
    "data": []
}
```

Field	Data type	Description
endpoint	`string`	Endpoint short name.
requestId	`string`	Request ID from initial request. For request/response mapping.
data	`array`	Webhook payload data array can include error or success objects.

Success result

```
{
    "status": "SUCCESS",
    "clientId": "CR0000000000000001",
    "details": "Client data successfully updated in Click to Pay profile"
}
```

Field	Data type	Description
status	`string`	Request status. Example: `["SUCCESS", "FAILED"]`
clientId	`string`	Client identification number equal to profile id.
details	`string`	Additional details about request resolution.

Failed result

```
{
    "status": "FAILED",
    "clientId": "CR0000000000000001",
    "errors": [{
            "code": "DAPI-ISS-ERR:0104",
            "message": "Incoming data processing error",
            "details": "Click To Pay Profile card data update failed",
            "link": "https://dapidocs.decta.com/Additional-info/Errors#DAPI-ISS-ERR:0104"
        }
    ]
}
```

Field	Data type	Description
status	`string`	Request status. Example: `["SUCCESS", "FAILED"]`
clientId	`string`	Client identification number equal to profile id.
errors	`array`	Errors data array. v2 endpoints standard error model.