

Cards

Learn how to create/manage cards, receive and/or update card's information

A set of functionalities allows to manage important aspects of the card lifecycle, to retrieve, update and oversee key card details.

Create Card

Use `POST /v2/api/cards` Create card V2 to create a physical or virtual card for a new private or business client, as well as to create a supplementary card to existing card account. To create a card, you should submit all mandatory fields as indicated in API Space documentation. If the Create card request was submitted successfully then you will receive PPAN, card account and client ID in response. This endpoint requires additional access rights and is proposed for TPP clients only. To start using it, please contact your DECTA manager.

All Create Card scenarios:

Private Base Cards:

Private Base Cards

1. Base Private Card with New Client and New Agreement

CreateCard Base Private [2.6]

REQUEST BODY SCHEMA (application/json)

privateClient required	object	mandatory block for New Base Card
↳ id optional	string	Can be absent (in order to create new Client) / Can be filled with own value (in order to create new Client with own id)
↳ name, surname, mobilePhone required	string	mandatory parameters
card		

required	object	mandatory block
↳ productCode required	string	mandatory parameter
↳ cardAccount optional	string	Can be absent (in order to create new "card.cardAccount") / Can be filled with own value (in order to create new "card.cardAccount" with own id)
↳ currencies required	array	mandatory parameter
↳ holder ignored	object	parameters will be ignored (holder is a Private Client)
↳ deliveryAddress required	object	mandatory (will be used in Client Agreement)
↳ supplementary required	string	FALSE is mandatory for BASE card

REQUEST SAMPLE

```
{
  "privateClient": {
    "name": "{{name}}",
    "surname": "{{surname}}",
    "mobilePhone": "{{mobilePhone}}"
  },
  "card": {
    "productCode": "{{productCode}}",
    "currencies": [
      "{{currencies}}"
    ],
    "deliveryAddress": {
      "country": "{{country}}",
      "city": "{{city}}",
      "street": "{{street}}"
    },
    "supplementary": "false"
  }
}
```

VALIDATIONS



CreateCard will be rejected if provided "privateClient.id" already exists;
CreateCard will be rejected if "privateClient.name" OR "privateClient.surname" is not provided;

2. Attached Private Card with New Agreement

CreateCard Base Private [2.6]

REQUEST BODY SCHEMA (application/json)

privateClient required	object	mandatory block for New Base Card
↳ id required	string	mandatory parameter, existing ID (in order to use existing Client data)
[other parameters]	Other parameters will be ignored	
card required	object	mandatory block
↳ productCode required	string	mandatory parameter
↳ cardAccount required	string	mandatory parameter, existing Account (in order to use existing "card.cardAccount")
↳ currencies required	array	mandatory parameter
↳ holder absent	object	block should be absent (holder is a Private Client)
↳ deliveryAddress required	object	mandatory (will be used in Client Agreement)
↳ supplementary required	string	FALSE is mandatory for BASE card

REQUEST SAMPLE

```

{
  "privateClient": {
    "id": "{{id}}"
  },
  "card": {
    "productCode": "{{productCode}}",
    "cardAccount": "{{cardAccount}}",
    "currencies": [
      "{{currencies}}"
    ],
    "supplementary": "false"
  }
}

```

VALIDATIONS

-  CreateCard will be rejected if provided "card.productCode" not registered;
- CreateCard will be rejected if provided "card.currencies" doesn't match with registered "card.productCode";
- CreateCard will be rejected if provided "privateClient.id" not exists;

3. Attached Private Card with Existing Agreement

CreateCard Base Private [2.6]

REQUEST BODY SCHEMA (application/json)

privateClient required	object	mandatory block for New Base Card
↳ id required	string	mandatory parameter, existing ID (in order to use existing Client data)
[other parameters]	Other parameters will be ignored	
card required	object	mandatory block
↳ productCode required	string	mandatory parameter
↳ cardAccount required	string	mandatory parameter, existing Account (in order to use existing "card.cardAccount")

↳ currencies required	array	mandatory parameter
↳ holder absent	object	block should be absent (holder is a Private Client)
↳ deliveryAddress ignored	object	parameters will be ignored
↳ supplementary required	string	FALSE is mandatory for BASE card

REQUEST SAMPLE

```
{
  "privateClient": {
    "id": "{{id}}"
  },
  "card": {
    "productCode": "{{productCode}}",
    "cardAccount": "{{cardAccount}}",
    "currencies": [
      "{{currencies}}"
    ],
    "supplementary": "false"
  }
}
```

VALIDATIONS

- ❗ CreateCard will be rejected if provided "privateClient.id" not exists;
- CreateCard will be rejected if provided "card.productCode" does not match the details of the existing agreement;
- CreateCard will be rejected if provided "card.currencies" does not match the details of the existing "card.cardAccount";

Private Supplementary Cards

1. Supplementary Private Card with new Client

CreateCard Supplementary Private [2.6]

REQUEST BODY SCHEMA (application/json)

privateClient**required**

object

mandatory block for New Base Card

↳ id

required

string

mandatory parameter, existing ID (in order to use existing Client data)

[other parameters]

*Other parameters will be ignored***card****required**

object

mandatory block

↳ productCode

required

string

mandatory parameter

↳ cardAccount

required

string

mandatory parameter, existing Account (in order to use existing "card.cardAccount")

↳ currencies

required

array

mandatory parameter

↳ holder

required

object

mandatory block should contain Client data (holder as Supplementary Card owner)

↳ id

optional

string

Can be absent (in order to create new Client) / Can be filled with own value (in order to create new Client with own id)

↳ name,
surname,
mobilePhone**required**

string

mandatory parameters

↳ deliveryAddress

ignored

object

parameters will be ignored

↳ supplementary

required

string

True is mandatory for Supplementary card

REQUEST SAMPLE

```
{
  "privateClient": {
    "id": "{{id}}"
  },
  "card": {
    "productCode": "{{productCode}}",
    "cardAccount": "{{cardAccount}}",
    "currencies": [
      "{{currencies}}"
    ],
    "holder": {
      "name": "{{name}}",
      "surname": "{{surname}}",
      "mobilePhone": "{{mobilePhone}}"
    },
    "supplementary": "true"
  }
}
```

VALIDATIONS

-  CreateCard will be rejected If provided "card.productCode" does not match the product of the found agreement;
- CreateCard will be rejected if provided "card.currencies" does not match the details of the existing card account;
- CreateCard will be rejected if provided "privateClient.id" or "card.productCode" could not be found;
- CreateCard will be rejected if provided "card.cardAccount" do not match requested "privateClient.id";

2. Supplementary Private Card with existing Client

CreateCard Supplementary Private [2.6]

REQUEST BODY SCHEMA (application/json)

privateClient

required

object mandatory block for New Base Card

↳ id

required

string mandatory parameter, existing ID (in order to use existing Client data) ("privateClient.id" should be used from initial client, base Card owner)

[other parameters]

Other parameters will be ignored

card

object mandatory block

required

↳ productCode

string

mandatory parameter

required

↳ cardAccount

string

mandatory parameter, existing Account (in order to use existing "card.cardAccount") ("card.cardAccount" should be used from initial client, base Card owner)

required

↳ currencies

array

mandatory parameter

required

↳ holder

object

mandatory block should contain Client data

required

↳ id

string

mandatory parameter, existing 2nd Client. (in order to use existing Client data) ("card.holder.id" should be used from another client, planned to be used as Supplementary Card owner)

required

↳ deliveryAddress

object

parameters will be ignored

ignored

↳ supplementary

string

True is mandatory for Supplementary card

required

REQUEST SAMPLE

```
{
  "privateClient": {
    "id": "{{id}}"
  },
  "card": {
    "productCode": "{{productCode}}",
    "cardAccount": "{{cardAccount}}",
    "currencies": [
      "{{currencies}}"
    ],
    "holder": {
      "id": "{{id}}"
    },
    "supplementary": "true"
  }
}
```

VALIDATIONS

- ❗ CreateCard will be rejected If provided "card.productCode" does not match the product of the found agreement;
CreateCard will be rejected if provided "card.currencies" does not match the details of the existing "card.cardAccount";
CreateCard will be rejected if provided "privateClient.id" or "card.cardAccount" could not be found;
CreateCard will be rejected if provided "card.cardAccount" do not match requested "privateClient.id";

Business Cards:

Business Base Cards

1. Base Business Card with New Client and New Agreement

CreateCard Base Business [2.6]

REQUEST BODY SCHEMA (application/json)

businessClient required	object	mandatory block for Business Card (Company data)
↳ registrationNumber required	string	mandatory parameter
card required	object	mandatory block
↳ productCode required	string	mandatory parameter
↳ cardAccount optional	string	Can be absent (in order to create new "card.cardAccount") / Can be filled with own value (in order to create new "card.cardAccount" with own id)
↳ currencies required	array	mandatory parameter
↳ holder required	object	mandatory block should contain Client data (Business Cardholder data)

↳ **id** string Can be absent (in order to create new Client) / Can be filled with own value (in order to create new Client with own id)
optional

↳ **name,** string mandatory parameters
surname,
mobilePhone
required

↳ **deliveryAddress** object mandatory (will be used in Client Agreement)
required

↳ **supplementary** string FALSE is mandatory for BASE card
required

REQUEST SAMPLE

```
{
  "businessClient": {
    "registrationNumber": "{{registrationNumber}}"
  },
  "card": {
    "productCode": "{{productCode}}",
    "currencies": [
      "{{currencies}}"
    ],
  },
  "holder": {
    "name": "{{name}}",
    "surname": "{{surname}}",
    "mobilePhone": "{{mobilePhone}}"
  },
  "deliveryAddress": {
    "country": "{{country}}",
    "city": "{{city}}",
    "street": "{{street}}"
  },
  "supplementary": "false"
}
```

VALIDATIONS



CreateCard will be rejected if provided "businessClient" details do not match existing business client;

2. Additional Business Card with New Client and New Agreement

CreateCard Base Business [2.6]

REQUEST BODY SCHEMA (application/json)

businessClient required	object	mandatory block for Business Card
↳ registrationNumber required	string	mandatory parameter, existing RegNum (in order to create Business Card within the same Company)
[other parameters]	<i>other parameters should match if used</i>	
card required	object	mandatory block
↳ productCode required	string	mandatory parameter
↳ cardAccount optional	string	Can be absent (in order to create new "card.cardAccount") / Can be filled with own value (in order to create new "card.cardAccount" with own id)
↳ currencies required	array	mandatory parameter
↳ holder required	object	mandatory block should contain Client data (Business Cardholder data)
↳ id optional	string	Can be absent (in order to create new Client) / Can be filled with own value (in order to create new Client with own id)
↳ name, surname, mobilePhone required	string	mandatory parameters
↳ deliveryAddress required	object	mandatory (will be used in Client Agreement)
↳ supplementary required	string	FALSE is mandatory for BASE card

REQUEST SAMPLE

```
{
  "businessClient": {
    "registrationNumber": "{{registrationNumber}}"
  },
  "card": {
    "productCode": "{{productCode}}",
    "currencies": [
      "{{currencies}}"
    ],
    "holder": {
      "name": "{{name}}",
      "surname": "{{surname}}",
      "mobilePhone": "{{mobilePhone}}"
    },
    "deliveryAddress": {
      "country": "{{country}}",
      "city": "{{city}}",
      "street": "{{street}}"
    },
    "supplementary": "false"
  }
}
```

VALIDATIONS

 CreateCard will be rejected if provided "businessClient" details do not match existing business client;

3. Attached Business Card with New Agreement (existing business client and existing cardholder)

CreateCard Base Business [2.6]

REQUEST BODY SCHEMA (application/json)

businessClient required	object	mandatory block for Business Card
↳ registrationNumber required	string	mandatory parameter, existing RegNum (in order to create Business Card within the same Company)
[other parameters]	Other parameters will be ignored	

card

required

object

mandatory block

↳ productCode

required

string

mandatory parameter

↳ cardAccount

optional

string

Can be absent (in order to create new "card.cardAccount") / Can be filled with own value (in order to create new "card.cardAccount" with own id)

↳ currencies

required

array

mandatory parameter

↳ holder

required

object

mandatory block should contain Client data

↳ id

required

string

mandatory parameter, existing Client ID (in order to use existing Business Cardholder data)

[other parameters]

Other parameters will be ignored

↳ deliveryAddress

required

object

mandatory (will be used in Client Agreement)

↳ supplementary

required

string

FALSE is mandatory for BASE card

REQUEST SAMPLE

```
{
  "businessClient": {
    "registrationNumber": "{{registrationNumber}}"
  },
  "card": {
    "productCode": "{{productCode}}",
    "cardAccount": "{{cardAccount}}",
    "currencies": [
      "{{currencies}}"
    ],
    "holder": {
      "id": "{{id}}"
    },
    "deliveryAddress": {
      "country": "{{country}}",
      "city": "{{city}}",
      "street": "{{street}}"
    },
    "supplementary": "false"
  }
}
```

VALIDATIONS

- ❗ CreateCard will be rejected if provided "businessClient.registrationNumber" could not be found;
- CreateCard will be rejected if provided "card.holder.id" could not be found;
- CreateCard will be rejected if provided "businessclient.registrationnumber" does not match the registration number of original Business Client;

4. Attached Business Card with existing Agreement (existing business client and existing cardholder)

CreateCard Base Business [2.6]

REQUEST BODY SCHEMA (application/json)

businessClient	object	mandatory block for Business Card
required		
↳ registrationNumber	string	mandatory parameter, existing RegNum (in order to create Business Card within the same Company)
required		
[other parameters]		Other parameters will be ignored

card

object mandatory block

required

↳ productCode string mandatory parameter

required

↳ cardAccount string mandatory parameter, existing Account (in order to use existing "card.cardAccount")

required

↳ currencies array mandatory parameter

required

↳ holder object mandatory block should contain Client data

required

↳ id string mandatory parameter, existing Client ID (in order to use existing Business Cardholder data)

required

[other parameters] Other parameters will be ignored

↳ deliveryAddress object parameters will be ignored

ignored

↳ supplementary string FALSE is mandatory for BASE card

required

REQUEST SAMPLE

```
{
  "businessClient": {
    "registrationNumber": "{{registrationNumber}}"
  },
  "card": {
    "productCode": "{{productCode}}",
    "cardAccount": "{{cardAccount}}",
    "currencies": [
      "{{currencies}}"
    ],
    "holder": {
      "id": "{{id}}"
    },
    "supplementary": "false"
  }
}
```

VALIDATIONS

- i** CreateCard will be rejected if provided "businessclient.registrationnumber", "card.holder.id" or CardAccount could not be found;
CreateCard will be rejected if provided "card.holder.id" does not own the same "businessclient.registrationnumber" as provided in businessClient part;
CreateCard will be rejected if provided "card.productCode" or "card.currencies" does not match the "card.productCode" or "card.currencies" of the found agreement;

Business Supplementary Cards

1. Supplementary Business Card with new Holder

CreateCard Supplementary Business[2.6]

REQUEST BODY SCHEMA (application/json)

businessClient required	object	mandatory block for Business Card
↳ registrationNumber required	string	mandatory parameter, existing ID (in order to create Business Card within the same Company)
[other parameters]	// other parameters should match if used	
card required	object	mandatory block
↳ productCode required	string	mandatory parameter
↳ cardAccount required	string	mandatory parameter, existing Account (in order to use existing "card.cardAccount")
↳ currencies required	array	mandatory parameter
↳ holder required	object	mandatory block should contain Client data (Business Cardholder data)
↳ id optional	string	Can be absent (in order to create new Client) / Can be filled with own value (in order to create new Client with own id)

↳ name, string mandatory parameters
surname,
mobilePhone
required

↳ deliveryAddress object parameters will be ignored
ignored

↳ supplementary string True is mandatory for Supplementary card
required

REQUEST SAMPLE

```
{
  "businessClient": {
    "registrationNumber": "{{registrationNumber}}"
  },
  "card": {
    "productCode": "{{productCode}}",
    "cardAccount": "{{cardAccount}}",
    "currencies": [
      "{{currencies}}"
    ],
    "holder": {
      "name": "{{name}}",
      "surname": "{{surname}}",
      "mobilePhone": "{{mobilePhone}}"
    },
    "supplementary": "true"
  }
}
```

VALIDATIONS

- ❗ CreateCard will be rejected If provided "card.productCode" does not match the product of the found agreement;
CreateCard will be rejected if provided "card.currencies" does not match the details of the existing card account;
CreateCard will be rejected if provided RegistrationNumber or CardAccount could not be found;
CreateCard will be rejected if provided RegistrationNumber or "card.cardAccount" do not match requested ones;

2. Supplementary Business Card with existing Holder

CreateCard Supplementary Business[2.6]

REQUEST BODY SCHEMA (application/json)

businessClient required	object	mandatory block for Business Card
↳ registrationNumber required	string	mandatory parameter, existing RegNum (in order to create Business Card within the same Company) (Reg.num should match initial client, base Business Card owner and Supplementary Client owner)
[other parameters]	Other parameters will be ignored	
card required	object	mandatory block
↳ productCode required	string	mandatory parameter
↳ cardAccount required	string	mandatory parameter, existing Account (in order to use existing "card.cardAccount") (Account should be used from another client, planned to be used as Supplementary card owner)
↳ currencies required	array	mandatory parameter
↳ holder required	object	mandatory block should contain Client data
↳ id required	string	mandatory parameter, existing Client ID (in order to use existing Business Cardholder data) ("card.holder.id" should be used from initial client, base Card owner)
[other parameters]	Other parameters will be ignored	
↳ deliveryAddress ignored	object	parameters will be ignored
↳ supplementary required	string	True is mandatory for Supplementary card

REQUEST SAMPLE

```
{
  "businessClient": {
    "registrationNumber": "{{registrationNumber}}"
  },
  "card": {
    "productCode": "{{productCode}}",
    "cardAccount": "{{cardAccount}}",
    "currencies": [
      "{{currencies}}"
    ],
    "holder": {
      "id": "{{id}}"
    },
    "supplementary": "true"
  }
}
```

VALIDATIONS

-  CreateCard will be rejected if provided "businessclient.registrationnumber" does not match the "businessclient.registrationnumber" of the Holder;
- CreateCard will be rejected if provided "card.currencies" does not match the details of the existing card account;
- CreateCard will be rejected if provided "card.holder.id" or CardAccount could not be found;
- CreateCard will be rejected if provided "card.cardAccount" do not match requested "card.holder.id";

Batch create card

When you want to create cards in batch request, you should use the request `POST /v2/api/cards/batch` to create more than one card per one request with the same input data. The endpoint requires additional access rights and is proposed for TPP clients only. Please note that batch card creation works asynchronously and requires additional integration with webhook handling.

How to set up?

1. Contact your DECTA manager about access to this endpoint.

2. Create a web service to get webhook data and send the endpoint data to your DECTA manager.
3. Start using the batch card creation endpoint once DECTA has notified you about it.

How to use?

1. Send request to `POST /v2/api/cards/batch` with generated id for `request-id` header and `userDefinedField12` field in payload.
2. Get response `202 Accepted`.
3. Save `request-id` header for webhook mapping and `userDefinedField12` for card data search request.
4. Wait for webhook with end result.
5. Use `GET /v1/api/cards` with `userDefinedField12` as a query parameter to get card data.

Important:

- Include the `request-id` header and `userDefinedField12` in all batch requests.
- Per batch card request can be created up to 6000 cards per hour → up to 4 mlj cards a month starting from 1 date of month until the 30 date of month.
- To create cards with the same expiry date across one or multiple batch requests, it is recommended to submit the requests at the beginning of the month to ensure consistent expiry alignment.

Webhook callback overview

The webhook notifies the Customer about completion of the batch card creation processing to refrain the Customer from using looped order status checks via API. 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 call represents a separate event message received. 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.

Payload example

```
{
  "endpoint": "BATCH_CARD_CREATE",
  "requestId": "11a11b1c-1d11-11ef-1g11-h1filjk11111",
  "data": [
    {
      "count": "100",
      "created": "55",
      "details": "Error during card creation",
      "errors": [
        {
          "code": "DAPI-ISS-ERR:0001",
          "message": "General internal error",
          "details": "Some services got unexpected error. Please contact DECTA support.",
          "link": "https://dapidocs.decta.com/Additional-info/Errors#DAPI-ISS-ERR:0001"
        }
      ]
    }
  ]
}
```

Field	Data type	Description
endpoint	string	Endpoint short name.
requestId	string	Request ID from initial request.
data	array	Array which contains action result.
data.count	string	The requested cards count.
data.created	string	The created cards count.
data.details	string	The result short description.
data.errors	array	The v2 default error model.

Get Cards

When you want to retrieve information about all your issued cards, you should use the request `GET /v1/api/cards` to get a list of cards with detailed information.

Retrieve Card information

To receive detailed information about a specific card and client, use the request `GET /v2/api/cards/{ppan}` (instead of old version - `GET /v1/api/cards/{ppan}`) and select necessary PPAN.

Retrieve Card data 5

Use the request `GET /v1/api/cards/{ppan}/card-data5` to get sensitive card information, like full PAN and CVV2 or CVC2 for a specific PPAN. DECTA takes data security seriously, so please contact your DECTA Manager to learn more about the availability of this additional feature.

Retrieve Card data 3

If you are cooperating with a token service provider, you should use request `GET /v1/api/cards/{ppan}/card-data3` to get selected card data including a clear pan, cardholder name, and expiry date for the specific card. As an additional layer of security, it is necessary to use the OTP Secret in the request. OTP secret allows card issuers to delegate permissions for a token service provider to fetch sensitive data for a specific PPAN.

Retrieve Card data 4

If you are cooperating with a third-party data service provider, it should use request `GET /v1/api/cards/{ppan}/card-data4` to get selected card data including a clear pan, cardholder name, CVV2/CVC2 and expiry date for the the specific card. As an additional layer of security, it is necessary to use the OTP secret in the request. OTP secret allows card issuers to delegate permissions for a token service provider to fetch sensitive data for a specific PPAN.

Retrieve Card data 6

If you are cooperating with a third-party data service provider, it should use request `GET /v2/api/cards/{ppan}/card-data6` to get PIN in encrypted format for a selected card. As an additional layer of security, it is necessary to use the OTP secret in the request. OTP secret allows card issuers to delegate permissions for a third-party data service provider to fetch sensitive data for a specific PPAN.

Retrieve Card data 7

Use request `GET /v2/api/cards/{ppan}/card-data7` to get sensitive card information, like PIN code for a specific PPAN. DECTA takes data security seriously, so please contact your DECTA Manager to learn more about the availability of this additional feature.

Get a One-Time Password (OTP)

Use request `GET /v1/api/cards/{ppan}/otp-secret` if you are cooperating with a token service provider or a third-party data service provider to allow them to retrieve a sensitive data in card-data 3 and card-data 4, card-data 6 requests for a specific PPAN. OTP secret is a single use code which can be used up to 2 times in 15 sec period.

Assign PIN code

You have the option to offer your clients the possibility to change their PIN code by using the request `POST /v1/api/cards/{ppan}/assign-pin`. The PIN code length is 4 digits. After PIN code update via API request the cardholder must confirm it in ATM. To find out more about this option and how you can start using it, contact your Account Manager.

Replace a card

When your client needs to replace a damaged or stolen/lost card, use the request `POST /v1/api/cards/{ppan}/replace`. The clients previous card should be permanently blocked.

Renew a card

When your client needs to renew the plastic card that expires within 2 months use the request `POST /v1/api/cards/{ppan}/renew`. This request allows to mark the card by selected PPAN which should be renewed and included in embossing file during the next file generation session. This request is intended only for plastic cards which are in status `UNBLOCKED` or `BLOCKED_BY_HOLDER`. If the card is in status `BLOCKED_BY_HOLDER`, then it will be automatically changed to status `UNBLOCKED` and marked for renewal. Use `PATCH /v1/api/cards/{ppan}/state` to set again `BLOCKED_BY_HOLDER` for the old card.

Renewed cards are made with the same account and card conditions configured in card product setting using the same PAN or a new PAN.

For renewed cards with the same PAN:

- The old card should be used until it expires and then the renewed card can be activated;
- Card token is updated.
- CVV/CVC is updated.

For renewed cards with a new PAN:

- The old card can be closed any time upon activating the renewed card;
- Card token is updated.
- CVV/CVC is updated.

Duplicate a card

DECTA supports Digital First card issuance, enabling issuers to create virtual cards that can later be duplicated into physical cards. If a specialized card product with duplication capability is configured on the DECTA side, authorized users can call `/v2/api/cards/{ppan}/duplicate`.

Digital First Overview

The Digital First model allows cardholders to immediately receive and use a virtual card upon issuance, with seamless tokenization into mobile wallets and the option to request a corresponding physical card.

Benefits

Cardholder Benefits

- **Immediate usability** Card usage and transactional capabilities.
- **Tokenization support** Enables instant provisioning to mobile wallets and other payment channels.
- **Unified card data** Maintains consistent card data and security credentials between the virtual and corresponding physical card.

Issuer Benefits

- **Improved card products** More flexible card offerings with faster access to payment functionality.
- **Scheme program advantages** Access to benefits provided by card schemes when enrolled in supported programs (e.g. Mastercard Digital First™).

Preconditions

A specialized card product with enabled duplication must be configured on the DECTA side. For details, contact your DECTA Account Manager.

User flow

1. Create a virtual card using: `POST /v2/api/cards`
2. Duplicate the card using: `POST /v2/api/cards/{ppan}/duplicate`
3. Wait for card personalization and physical card dispatch.

Important:

- A virtual card can be converted to a physical card at any time during its lifecycle as long as card is not closed.
- Virtual cards and converted physical cards cannot be renewed or replaced.
- When converted, the physical card retains the same PAN, CVV, and expiration date as the virtual card.
- While the virtual card is active, the physical card is also active during delivery. Issuers are advised to apply card spending limits to secure online-only use or other restrictions until the cardholder receives the physical card.
- Converted cards are identifiable in Cards report according to the Issuing Report Files Specification.
- Virtual card provisioning to mobile wallets and other payment channels is possible if the issuer has implemented cards tokenization.

Card Status Change

To change the card status, you should request `PATCH /v1/api/cards/{ppan}/state` that allows to activate, block or unblock a card by selected pseudo-pan.

Activate Card

To activate a physical card that has been received by the cardholder and has the status 1-Suspended, make a `PATCH /v1/api/cards/{ppan}/state` request and set the state to `ACTIVE`. In this request the cardholder should indicate the last 6 digits of the card.

Suspend Card

To suspend a card make a `PATCH /v1/api/cards/{ppan}/state` request and set the state to `BLOCKED_BY_HOLDER` or `BLOCKED_BY_PARTNER`. A suspended card can still be unblocked or blocked permanently.

Card statuses with converted card blocking reason codes to Visa/Mastercard systems:

Card status via DAPI function	ISO reason code (Authorisation response code)	Converted reason code to Visa	Converted reason code to Mastercard	Description
BLOCKED_BY_HOLDER	104 (decline, do not honor)	78	62	Selected card is temporarily blocked, possible to unblock
BLOCKED_BY_PARTNER	107 (decline, refer to card issuer)	5C	01	Selected card is temporarily blocked, possible to unblock

Reactivate Suspended Card

To activate a card that was suspended, make a `PATCH /v1/api/cards/{ppan}/state` request and set the state to `UNBLOCKED`.

Card Closure

Closing a card is equal to blocked permanently card status and can't be reversed. Make `PATCH /v1/api/cards/{ppan}/state` request and set the state to `BLOCKED_PERMANENTLY` or `BLOCKED_TERMINATED` or `BLOCKED_NOT_RECEIVED`.

Card statuses with converted card blocking reason codes to Visa/Mastercard systems:

Card status via DAPI function	ISO reason code (Authorisation response code)	Converted reason code to Visa	Converted reason code to Mastercard	Description
BLOCKED_PERMANENTLY	203 (pick-up, card acceptor contact card acquirer)	04	04	Selected card is blocked forever for any reason, not possible to unblock
BLOCKED_TERMINATED	204 (pick-up, restricted card)	46	04	Selected card is blocked forever due to agreement termination with the client, not possible to unblock
BLOCKED_NOT_RECEIVED	208 (pick-up, lost)	41	41	Selected card is blocked forever because the plastic card is not received/lost, not possible to unblock

Change card user defined fields

If you want to change userDefinedField7, then use this request `PATCH /v1/api/cards/{ppan}/user-defined-fields`. UserDefinedField7 is a free custom use field which allows to store specific information/conditions associated with card in the database.