

RED interchange API - Card Operations

The TI smartcards contain user certificates and secrets for authorization and electronic signature operations. To provide a two-factor-authorization smartcards also require their user to enter personal PIN codes for certain operations.

- new cards come with transport-PINs that must be replaced by the user before the first use of the card
- card pins can be changed on user request

All card operations can be executed via the telematik API.

GetCards

GetCards is a GET request returning all cards currently available in the connected card readers.

Example for a GetCards request

```
{{baseURL}}/{{tenant}}/TI?action=$getcards
```

Example: request for all cards in all connected card readers

See [RED Interchange API - Postman Collection 0400 - TI - GetCards](#)

In order to return cards from a specific card terminal only the request can be extended by the IP address of the card terminal. The card terminal IP address can be set in the card terminal settings by the system administrator.

Example for a GetCards request from a specific terminal

```
{{baseURL}}/{{tenant}}/TI?action=$getcards&cardreader=172.20.129.41
```

Example: request for all cards in all connected card reader with IP address 172.20.129.41

See [RED Interchange API - Postman Collection 0401 - TI - GetCards by IP](#)

GetCards returns an XML data set with all cards found in all card readers connected. For each card the card handle is returned which is to be used for identifying the card in further requests.

Example of a GetCards request

```
<ConnectorResponse>
  <Card>
    <CardType>SMC-B</CardType>
    <CardHandle>6974e884-465d-4f23-abad-5ae61a538d26</CardHandle>
    <Iccsn>80276883110000117848</Iccsn>
    <CtId>00:0D:F8:08:AD:79</CtId>
    <SlotId>4</SlotId>
    <CardHolderName>Praxis Dr. Steffi SänderTEST-ONLY</CardHolderName>
  </Card>
  <Card>
    <CardType>HBA</CardType>
    <CardHandle>aabff9f7-001f-4f43-bb4d-a2d1bf7906d7</CardHandle>
    <Iccsn>80276883110000121207</Iccsn>
    <CtId>00:0D:F8:08:AD:79</CtId>
    <SlotId>2</SlotId>
    <CardHolderName>Helmut Stauffenberg-KleinschmidtTEST-ONLY</CardHolderName>
  </Card>
</ConnectorResponse>
```

TelematikID

For some purpose the telematik-ID of a SMC-B or HBA is required. The telematik-ID is a unique ID of the signature card.

Telematikids is a GET request returning all cards currently available in the connected card readers with their telematik IDs.

Example for a GetCards request

```
{{baseUrl}}/{{tenant}}/TI?action=$telematikids
```

Example: request for all cards in all connected card readers with telematik IDs

See [RED Interchange API - Postman Collection 0406 - TI - TelematikIDs](#)

TelematikIDs returns an XML data set with all cards found in all card readers connected. For each card the card handle is returned which is to be used for identifying the card in further requests.

Result of a TelematikIDs request

```
<ConnectorResponse>
  <Card>
    <CardHandle>d4e5c3cf-e01b-4b50-a56a-e43f695d4339</CardHandle>
    <CardType>SMC-B</CardType>
    <Iccsn>80276883110000117848</Iccsn>
    <telematikid>1-SMC-B-Testkarte-883110000117848</telematikid>
  </Card>
  <Card>
    <CardHandle>78d69a2c-c5a2-4d9e-9a02-a33e6e30a7d6</CardHandle>
    <CardType>HBA</CardType>
    <Iccsn>80276883110000136667</Iccsn>
    <telematikid>1-HBA-Testkarte-883110000136667</telematikid>
  </Card>
</ConnectorResponse>
```

GetPINStatus

GetPINStatus is a GET request returning the current status of all pins of all cards in all card terminals. This request fetches all cards and then requires each card to return its PIN status. Processing time of this request depends on the number of terminals connected.

Example for GetPINStatus request

```
{{baseUrl}}/{{tenant}}/TI?action=$getpinstatus
```

Example: GetPINStatus request

See [RED Interchange API - Postman Collection 0402 - TI - PinStatus](#)

In order to get the PIN status of all cards from a specific card terminal only the request can be extended by the IP address of the card terminal.

```
{{baseUrl}}/{{tenant}}/TI?action=$getpinstatus&cardreader=172.20.129.41
```

Example of a GetPINStatus by IP request for card terminal with IP address 172.20.129.41

See [RED Interchange API - Postman Collection 0403 - TI - PinStatus by IP](#)

GetPINStatus returns an XML data set including the PIN status for each card (PIN.SMC or PIN.HBA).

- VERIFIED - this card has been confirmed by a user PIN entry and may be used for operations
- VERIFIABLE, NOT VERIFIED - confirmation is pending, user must enter the PIN in order to use the card for operations

- TRANSPORT_PIN, EMPTY_PIN - card has not been initialized yet, a new PIN must be set by user
- REJECTED - a wrong PIN was entered
- BLOCKED, NOWBLOCKED, WASBLOCKED - wrong PIN has been entered too many times, card has been blocked and must be verified using the PUK

For further information see [gematik Implementierungsleitfaden für Primärsysteme](#)

Example for GetPINStatus request

```
<ConnectorResponse>
  <Card>
    <CardType>SMC-B</CardType>
    <CtId>00:0D:F8:08:AD:79</CtId>
    <CardHandle>6974e884-465d-4f23-abad-5ae61a538d26</CardHandle>
    <PinStatus>VERIFIED</PinStatus>
    <Status>OK</Status>
  </Card>
  <Card>
    <CardType>HBA</CardType>
    <CtId>00:0D:F8:08:AD:79</CtId>
    <CardHandle>aabff9f7-001f-4f43-bb4d-a2d1bf7906d7</CardHandle>
    <PinStatus>NOT_VERIFIED</PinStatus>
    <Status>4072</Status>
  </Card>
</ConnectorResponse>
```

VerifyPIN

In order to activate a card the user must enter the card PIN at the card terminal keyboard. This must be triggered by a verifyPIN request sent to the card terminal which then will prompt the user for PIN entry. The request may be sent without the card terminal IP address given which would prompt all connected card terminals for PIN entry, but it is recommended to include the IP address of the card terminal in order to minimize the user effort.

Example of a VerifyPIN request to a specific card terminal

```
{{baseURL}}/{{tenant}}/TI?cardreader=172.20.129.41&action=$verifypin
```

After successful execution the request returns the XML data object with all cards and their PIN status as described in GetPINStatus.

See [RED Interchange API - Postman Collection 0404 - TI - Verifypin by IP](#)

SMC-B Remote-PIN Activation

The card terminal Cherry ST1506 offers automation of PIN entry for SMC-B cards (remote-PIN function). If configured in RED a websocket connection is established between RED and card terminal. Any time a PIN entry is required (e.g. after sending a verifyPIN request) the card terminal will send a request to RED for the PIN. RED will return the stored PIN to the card terminal. Automatisations of HBA PIN entries are not possible.

[Erkönig - Cherry ST1506 Remote-PIN](#)

If a remote card terminal is set up its PIN status can be activated with the request "SMBPinUpdate"

Example of UpdateSMCPIN request to activate all connected remote card terminals

```
{{baseURL}}/{{tenant}}/TI?action=$updatesmcpin
```

After successful execution the request returns Status 201.

Activate Komfortsignatur

If activated for a signature card (HBA) the Komfortsignatur feature allows to create up to 250 electronic signatures without additional PIN entry. In order to use this comfort feature this option must be activated for a named signature card.

Activation of Komfortsignatur is (depending on Konnektor settings) limited to 250 signatures, a given time frame of up to 12 hours max and the signature card session. The session starts when the card is inserted in the card terminal and ends when it is removed.

Example for activation of Komfortsignatur for card with name

```
{{baseURL}}/{{tenant}}/TI?action=$activatecomfort&cardname=HBAArzt
```

After successful activation RED returns an XML object with status of signature, number of available signatures (signaturemax) and remaining time.

Example of result after Komfortsignatur activation

```
<status>
  <signaturestatus>OK</signaturestatus>
  <signaturemax>250</signaturemax>
  <signatureremaining>PT11H59M58S</signatureremaining>
</status>
```

IsAlive

This call allows to verify the current operational status of the interface. The call returns status for Card operations (such as GetCards) and the FHIR interface.

Example for an isalive-call

```
{{baseURL}}/{{tenant}}/TI?action=$isalive
```

Example of isalive-call

```
<result>
  <tioperations status="true" miliseconds="2339"/>
  <fhir status="true" miliseconds="27"/>
</result>
```