



General notes

Register application

Applications that want to access the figo Connect must be registered beforehand. If you'd like to create a partner application, please [email us](#). We will generate a client identifier and client secret for your application.

Calling the figo Connect API

All requests to the figo Connect API must contain an access token in the `Authorization: Bearer` HTTP header. Before any of the REST endpoints can be called, an access token has to be requested from the figo Connect authorization server.

Command line demo

Try out our figo Connect API:

```
curl -H "Authorization: Bearer ASHWLIkouP2O6_bgA2wWReRhletgWKHYjLqDaqb0LFfaP_cjLiRiSyQXyt2kM1eXU2XLFZQ0Hro15HikJQT_eNeT_9XQ" https://api.figo.me/rest/
```

figo SDK

Integrate figo Connect easier and faster with one of our figo SDKs.

Language	Download
Python	https://github.com/figo-connect/python-figo
Ruby	https://github.com/figo-connect/ruby-figo
Node.js	https://github.com/figo-connect/node-figo
PHP	https://github.com/figo-connect/php-figo
Java	https://github.com/figo-connect/java-figo

Error handling

As long as not specified otherwise, standard HTTP error codes are used for error handling.

HTTP code	Description
200	Success
301	Redirect
400	Malformed request
401	Missing, invalid or expired access token. A new access token must be requested or the user must be re-authenticated.
403	Insufficient permission
404	Requested object does not exist
405	Unexpected request method
503	Exceeded rate limit

Version information

/version

DESCRIPTION Get figo Connect server and client application versions.

REQUEST URL `GET https://api.figo.me/version`

PARAMETERS **product_name** Product name of client application.
product_version Product version of client application.
product_environment Environment of client application; Possible values are `production`, `sandbox` or `ad-hoc`.
platform Platform name and version.
device Device type and revision.

Sample request

```
product_name=figo+iOS&product_version=3&product_environment=p
platform=iOS+6.1.3&device=iPhone4,1
```

RESPONSE **product_name** Product name of figo Connect server.
product_version Product version of figo Connect server.
product_environment Environment of figo Connect server; Possible values are

production, sandbox or staging.

ssl_fingerprints List of SHA-1 fingerprints of all valid figo Connect server SSL/TLS certificates.

client Depending on the parameters in the request, this object contains information about the client application. The information are in particular the minimum version of the client application compatible with the figo Connect server, the most recent version of the client application available and an update URL.

notice Depending on the parameters in the request, this object contains a message to be displayed by the client application. Next to the message there is an ID number to avoid displaying the same message twice, as well as an URL pointing to further information.

Sample JSON response

```
{
  "product_name": "figo Connect",
  "product_version": "929f8b8",
  "product_environment": "production",
  "ssl_fingerprints": [
    "A6:FE:08:F4:A8:86:F9:C1:BF:4E:70:0A:BD:72:AE:B8:8E:B7:78",
    "AD:A0:E3:2B:1F:CE:E8:44:F2:83:BA:AE:E4:7D:F2:AD:44:48:7F",
    "B0:8B:03:C2:07:93:B3:B8:5B:A0:15:03:FB:A4:71:7D:86:9C:8F"
  ],
  "client": {
    "current_version": "3",
    "minimum_version": "2",
    "url": "https://itunes.apple.com/app/id567739104?mt=8"
  },
  "notice": {
    "id": 42,
    "message": "Scheduled maintenance on midnight.",
    "url": "http://figo.me/notices/maintenance.html"
  }
}
```

Client management

/client/purchase_code

DESCRIPTION Purchase a voucher code which can be redeemed later for any user.

The `Authorization: Basic` HTTP header must contain the client identifier as username and the client secret as password.

REQUEST URL POST `https://api.figo.me/client/purchase_code`

PARAMETERS **premium_days** *mandatory* The number of days the figo Account plan will become premium when redeemed. The recommended value is 30 days.

Sample JSON request

```
{
  "premium_days": 30
}
```

RESPONSE **code** Voucher code.

Sample JSON response

```
{
  "code": "qdc-lhb-eaj-hld"
}
```

Authentication

We use **OAuth 2.0** for authentication and authorization.

/auth/code

DESCRIPTION Before your application can request an access token, it must first obtain an authorization code.

Your application initiates the authorization process by directing the user's web browser to a popup window of figo Connect.

The popup window will ask the user for his or her figo Account credentials. The user will see the permissions as specified in `scope` and a list of accounts to choose from. If the user grants access to your application, the figo Connect server sends an authorization code to the callback URL by redirecting the browser to `redirect_uri`.

POPUP WINDOW URL `https://api.figo.me/auth/code`

PARAMETERS **response_type** *mandatory* This parameter must be set to `code`.

client_id *mandatory* The client identifier obtained during application registration.

redirect_uri The authorization code will be sent to this callback URL. It must match one of the URLs registered during application registration.

scope A space delimited set of requested **permissions**. The requested permissions can be narrower but not broader than the permissions agreed during application registration. If this parameter is omitted, the permissions agreed during application registration are used in place.

state *mandatory* Any kind of string that will be forwarded in the response message. It serves two purposes: The value is used to maintain state between this request and the callback, e.g. it might contain a session ID from your application. The value should also contain a random component, which your application checks to mitigate cross-site request forgery.

Sample popup window URL

```
https://api.figo.me/auth/code?response_type=code&\
client_id=CaESKmC8MAhNpDe5rvmWnSkRE_7pkkVIIgMwclgzGcQY&\
scope=accounts%3Dro+balance%3Dro+transactions%3Dro&\
state=xqD6gjWygsBlF0uB
```

REDIRECT ON SUCCESS **code** The authorization code for obtaining an access token. It can only be used once and will expire after a few minutes.

state The `state` parameter from the request.

Sample redirect

```
GET /callback?code=OkA7jUIro6JNmOF7i5f7QmepCLLoIFug6621ADHLMjrl0wVZNkheCv5nMkH3hZ0v24immWGaI5Gc0P0F9ue0sgnV7DqwmM&state=xq
```

REDIRECT ON ERROR **error** One of the error codes as defined in section 4.1.2.1 of the OAuth 2.0 specification.

error_description A human-readable error message intended for the application developer.

state The `state` parameter from the request.

/auth/token (authorization code)

DESCRIPTION After your application obtained an authorization code, it may exchange the authorization code for an access token.

The `Authorization: Basic` HTTP header must contain the client identifier as username and the client secret as password.

REQUEST URL POST `https://api.figo.me/auth/token`

PARAMETERS **grant_type** *mandatory* This parameter must be set to `authorization_code`.

code *mandatory* The authorization code returned from the initial request.

redirect_uri If the callback URL was specified in the initial request, then it must also be included in this request.

Sample request

```
grant_type=authorization_code&code=OkA7jUIro6JNmOF7i5f7QmepCI  
MjK6oimFYK1x5xy5rl0wVZNkheCv5nMkH3hZ0v24immWGaI5aGc0P0F9ue0sg
```

RESPONSE ON SUCCESS **access_token** The access token

token_type This response parameter is set to `Bearer`.

expires_in The remaining live time of the access token in seconds.

refresh_token A refresh token that may be used to request new access tokens. Refresh tokens remain valid until the user revokes access to your application. This response parameter is only present if the permission `offline` has been requested in the authorization code request.

scope A space delimited set of requested [permissions](#).

Sample JSON response

```
{  
  "access_token": "ASHWLikouP206_bgA2wWReRhletgWKHYjLqDaqb0LF  
                  22ujRIP_cjLiRiSyQXyt2kM1eXU2XLFZQ0Hro15Hik  
  "token_type": "Bearer",  
  "expires_in": 3600,  
  "refresh_token": "RTfI2WNyK78NozupDH9ai8GPRbjjdVsXPpTmobD2p  
                  oT1TkFspMQOzN1CZpIrZREHrSdYeObea3Qda7hk2Q
```

```
"scope": "accounts=ro balance=ro transactions=ro offline"
}
```

RESPONSE ON ERROR **error** One of the error codes as defined in section 5.2 of the OAuth 2.0 specification.

error_description A human-readable error message intended for the application developer.

/auth/token (refresh token)

DESCRIPTION Request new access token with refresh token. New access tokens can be obtained as long as the user has not revoked the access granted to your application.

The `Authorization: Basic` HTTP header must contain the client identifier as username and the client secret as password.

REQUEST URL POST `https://api.figo.me/auth/token`

PARAMETERS **grant_type** *mandatory* This parameter must be set to `refresh_token`.
refresh_token *mandatory* The refresh token returned from the authorization code exchange.
scope A space delimited set of requested **permissions**. The requested permissions can be narrower but not broader than the permissions supplied in the authorization code request. If this parameter is omitted, the permissions supplied in the authorization code request are used in place.

Sample request

```
grant_type=refresh_token&refresh_token=RTfI2WNYK78NozupDH9ai8  
obD2p_1epZUYmidZA0oT1TkFspMQOzNlCZpIrZREHrSdYeObea3Qda7hk2Q6P
```

RESPONSE The response is similar to the response documented in the **previous** section.

/auth/token (password)

DESCRIPTION Native clients may directly ask the user for his figo Account credentials and use them to obtain an access token.

The `Authorization: Basic` HTTP header must contain the client identifier as username and the client secret as password.

Two-factor authentication is not yet available for figo Connect. Client applications must be prepared to display an appropriate error message whenever they receive the error code `two_factor_authentication`. The error message should contain the hint, that the client application must be updated to a newer version.

REQUEST URL POST `https://api.figo.me/auth/token`

PARAMETERS

grant_type *mandatory* This parameter must be set to `password`.

username *mandatory* The figo Account email address.

password *mandatory* The figo Account password.

device_name *mandatory* Human-readable name for the device where the client application runs.

device_type *mandatory* One of the device types `iPhone`, `iPad`, `WWW`, `Windows`, `Mac` or `Linux`.

device_udid *mandatory* Device UDID for associating notification subscriptions with individual devices.

scope A space delimited set of requested **permissions**. The requested permissions can be narrower but not broader than the permissions agreed during application registration. If this parameter is omitted, the permissions agreed during application registration are used in place.

Sample request

```
grant_type=password&username=demo@figo.me&password=demo1234&\ndevice_name=My+Phone&device_type=iPhone&device_udid=b4ec8a40e
```

RESPONSE

The response is similar to the response documented in the **previous** section. In addition to the error codes defined in the previous section, the error code `locked_user` might be returned.

/auth/revoke

DESCRIPTION

When a user unsubscribes from your application, your application must also revoke its access to figo Connect.

No `Authorization` HTTP header is required for this task.

The supplied token can either be an access token or a refresh token. If the token is an access token and it has a corresponding refresh token, the refresh token will also be revoked.

REQUEST URL

```
GET https://api.figo.me/auth/revoke
```

PARAMETERS

token A refresh token or access token.

Sample request

```
token=RTfI2WNyK78NozupDH9ai8GPRbjjdVsXPpTmobD2p_1epZUYmidZAOcZpIrZREHrSdYeObea3Qda7hk2Q6PO5BEVF3GBh0
```

RESPONSE

A successful response contains no content. On error the following response parameters will be returned:

error One of the error codes `invalid_request` or `invalid_grant`.

error_description A human-readable error message intended for the application developer.

/auth/user

DESCRIPTION Register a new figo Account. Only native clients may create new figo Accounts.

REQUEST URL POST https://api.figo.me/auth/user

PARAMETERS

name *mandatory* First and last name.

email *mandatory* Email address; It must obey the figo **username & password** policy.

send_newsletter *mandatory* This flag indicates whether the user has agreed to be contacted by email.

language *mandatory* Two-letter code of preferred language.

password *mandatory* New figo Account password; It must obey the figo **username & password** policy.

affiliate_user Base64 encoded email address of the user promoting the new user.

affiliate_client_id Client ID of the figo Connect partner from which the user was redirected to the registration form.

Sample JSON request

```
{
  "name": "John Doe",
  "email": "demo@figo.me",
  "send_newsletter": true,
  "language": "en",
  "password": "demo1234"
}
```

RESPONSE ON SUCCESS **recovery_password** Auto-generated recovery password.

Sample JSON response

```
{
  "recovery_password": "tpiz-khyt-plii-dkpd-ezkw"
}
```

RESPONSE ON ERROR **error** One of the error codes `invalid request`, `user_exists`, `username_policy_error` **Or** `password_policy_error`.

error_description A human-readable error message intended for the application developer.

/auth/user/resend_unlock_code

DESCRIPTION Re-send unlock code email.

PERMISSIONS user=rw

REQUEST URL POST https://api.figo.me/auth/user/resend_unlock_code

PARAMETERS **username** *mandatory* The figo Account email address

Sampe request

```
{"username": "demo@figo.me"}
```

RESPONSE A sucessfull response contains no content.

/auth/unlock

DESCRIPTION Native clients may unlock figo Accounts. This request must be issued with the same client identifier which was used to request the unlock code.

REQUEST URL POST <https://api.figo.me/auth/unlock>

PARAMETERS **username** *mandatory* The figo Account email address.

unlock_code *mandatory* The unlock code sent to the user by email.

recovery_password *mandatory* The figo Account recovery password.

new_password *mandatory* New figo account password; It must obey the figo **username & password** policy.

Sample request

```
username=demo@figo.me&password=demo1234&unlock_code=57105548&recovery_password=tpiz-khyt-plii-dkpd-ezkw&new_password=demo5
```

RESPONSE A sucessfull response contains no content. On error the following response parameters will be returned:

error One of the error codes `invalid_request`, `invalid_unlock_code`, `invalid_recovery_password`, `invalid_credentials`, `recovery_locked` or `password_policy_error`.

error_description A human-readable error message intended for the application developer.

User management

/rest/user (GET)

DESCRIPTION Get figo Account settings.

PERMISSIONS `user=ro`

REQUEST URL GET <https://api.figo.me/rest/user>

RESPONSE **user_id** Internal figo Connect user ID.

name First and last name.

- email** Email address.
- address** Postal address for bills, etc.
- verified_email** This flag indicates whether the email address has been verified.
- send_newsletter** This flag indicates whether the user has agreed to be contacted by email.
- language** Two-letter code of preferred language.
- premium** This flag indicates whether the figo Account plan is free or premium.
- premium_expires_on** Timestamp of premium figo Account expiry.
- premium_subscription** Provider for premium subscription or Null of no subscription is active.
- join_date** Timestamp of figo Account registration.
- force_reset** If this flag is set then all local data must be cleared from the device and re-fetched from the figo Connect server.
- recovery_password** Auto-generated recovery password. This response parameter will only be set once and only for the figo iOS app and only for legacy figo Accounts. The figo iOS app must display this recovery password to the user.

Sample JSON request

```
{
  "user_id": "U12345",
  "name": "John Doe",
  "email": "demo@figo.me",
  "address": {
    "company": "figo",
    "street": "Ritterstr. 2-3",
    "postal_code": "10969",
    "city": "Berlin"
  },
  "verified_email": true,
  "send_newsletter": true,
  "language": "en",
  "premium": true,
  "premium_expires_on": "2014-04-19T17:25:54.000Z",
  "premium_subscription": "paymill",
  "join_date": "2012-04-19T17:25:54.000Z"
}
```

/rest/user (PUT)

DESCRIPTION **Modify figo Account settings.**

PERMISSIONS `user=rw`

REQUEST URL `PUT https://api.figo.me/rest/user`

- REQUEST
- name** First and last name.
 - email** Email address. It must obey the figo **username & password** policy. If this parameter is set, then the parameter `password` must be set, too.
 - address** Postal address. It must always be set in full (i.e. company, street, postal_code & city)

send_newsletter This flag indicates whether the user has agreed to be contacted by email.

language Two-letter code of preferred language.

password Current figo Account password.

new_password New figo Account password; It must obey the figo **username & password** policy. If this parameter is set, then the parameter `password` must be set, too.

Sample JSON request

```
{
  "email": "new_mail@figo.me",
  "password": "demo1234"
}
```

RESPONSE A sucessfull response contains no content. On error the following response parameters will be returned:

error One of the error codes `invalid_request`, `missing_password`, `locked_user`, `user_exists`, `username_policy_error` or `password_policy_error`.

error_description A human-readable error message intended for the application developer.

/rest/user (DELETE)

DESCRIPTION Delete figo Account.

PERMISSIONS `user=rw`

REQUEST URL `DELETE https://api.figo.me/rest/user`

RESPONSE A sucessfull response contains no content.

/rest/user/resend_verification

DESCRIPTION Re-send verification email.

PERMISSIONS `user=rw`

REQUEST URL `POST https://api.figo.me/rest/user/resend_verification`

RESPONSE A sucessfull response contains no content.

/rest/user/purchase_subscription

DESCRIPTION Purchase a premium membership with a debit or credit card.

PERMISSIONS `user=rw`

REQUEST URL `POST https://api.figo.me/rest/user/purchase_subscription`

PARAMETERS

token *mandatory* Paymill card token from the Paymill JavaScript bridge.

plan *mandatory* Which plan the customer would like to purchase: `monthly` or `yearly`.

sandbox *mandatory* This flag indicates whether the token has been issued by the Paymill test account.

Sample JSON request

```
{
  "token": "(Paymill token here)",
  "plan": "monthly",
  "sandbox": false
}
```

RESPONSE

premium This flag indicates whether the figo Account plan is free or premium.

premium_expires_on Timestamp of premium figo Account expiry.

Sample JSON response

```
{
  "premium": true,
  "premium_expires_on": "2014-04-19T17:25:54.000Z"
}
```

/rest/user/cancel_subscription

DESCRIPTION Cancel the premium membership

PERMISSIONS `user=rw`

REQUEST URL `POST https://api.figo.me/rest/user/cancel_subscription`

RESPONSE A sucessfull response contains no content.

/rest/user/verify_receipt

DESCRIPTION Verify iOS App Store receipt from In-App Purchase.

PERMISSIONS `user=rw`

REQUEST URL `POST https://api.figo.me/rest/user/verify_receipt`

PARAMETERS

receipt *mandatory* iOS App Store receipt data.

sandbox *mandatory* This flag indicates whether the receipt has been issued by the Apple sandbox server.

Sample JSON request

```
{
  "receipt": "(receipt bytes here)",
  "sandbox": false
}
```

```
}
```

RESPONSE: **premium** This flag indicates whether the figo Account plan is free or premium.

premium_expires_on Timestamp of premium figo Account expiry.

Sample JSON response

```
{
  "premium": true,
  "premium_expires_on": "2014-04-19T17:25:54.000Z"
}
```

/rest/user/redeem_code

DESCRIPTION Redeem a voucher code.

PERMISSIONS `user=ro`

REQUEST URL `POST https://api.figo.me/rest/user/redeem_code`

PARAMETERS **code** *mandatory* Voucher code.

Sample JSON response

```
{
  "code": "qdc-lhb-eaj-hld"
}
```

RESPONSE ON SUCCESS: **premium** This flag indicates whether the figo Account plan is free or premium.

premium_expires_on Timestamp of premium figo Account expiry.

Sample JSON response

```
{
  "premium": true,
  "premium_expires_on": "2014-04-19T17:25:54.000Z"
}
```

RESPONSE ON ERROR: **error** One of the error codes `invalid_request`, `invalid_code` or `already_redeemed`.

error_description A human-readable error message intended for the application developer.

Client applications

/rest/clients

DESCRIPTION Get a list of all current and former authorized figo Connect clients.

PERMISSIONS `clients=ro`

REQUEST URL

GET https://api.figo.me/rest/clients

RESPONSE

clients List of **client** objects.

Sample JSON response

```
{
  "clients": [
    {
      "client_id": "901cd43bd27e3875f59b0e51f4965e68\
034e3f47122f0a21fcd18fc5ce847349",
      "name": "DEMO",
      "homepage": "http://figo.me",
      "description": "figo demo client",
      "icon": "https://api.figo.me/assets/clients/figo.png",
      "scope": "accounts=ro balance=ro transactions=ro offlin
      "valid": true,
      "last_access": "2013-04-10T12:00:00.000Z",
      "accounts": [
        "A1.1",
        "A1.2"
      ]
    }
  ]
}
```

/rest/clients/{client_id} (PUT)

DESCRIPTION

Modify client details.

PERMISSIONS

clients=rw

REQUEST URL

GET https://api.figo.me/rest/clients/{client_id}

client_id Internal figo Connect client ID.

PARAMETERS

accounts List of account IDs. The client will have access to these accounts.

Sample JSON request

```
{
  "accounts": [
    "A1.1",
    "A1.2",
    "A1.3"
  ]
}
```

RESPONSE

A sucessfull response contains no content.

/rest/clients/{client_id} (DELETE)

DESCRIPTION

Unlink client.

PERMISSIONS

clients=rw

REQUEST URL

DELETE https://api.figo.me/rest/clients/{client_id}

client_id Internal figo Connect client ID.

RESPONSE

A sucessfull response contains no content.

/rest/devices

DESCRIPTION

Get a list of all authorized devices.

PERMISSIONS

clients=ro

REQUEST URL

GET https://api.figo.me/rest/devices

RESPONSE

devices List of **device** objects.

Sample JSON response

```
{
  "devices": [
    {
      "device_id": "901cd43bd27e3875f59b0e51f4965e68\
034e3f47122f0a21fcd18fc5ce847349",
      "name": "figo's iPhone",
      "icon": "https://api.figo.me/assets/devices/iphone.png",
      "last_access": "2013-04-10T12:00:00.000Z"
    }
  ]
}
```

/rest/devices/{device_id} (PUT)

DESCRIPTION

Modify device details.

PERMISSIONS

clients=rw

REQUEST URL

GET https://api.figo.me/rest/devices/{device_id}

device_id Internal figo Connect device ID.

PARAMETERS

name Device name.

Sample JSON request

```
{
  "name": "My Phone"
}
```

RESPONSE

A sucessfull response contains no content.

/rest/devices/{device_id} (DELETE)

DESCRIPTION	Unlink device.
PERMISSIONS	clients=rw
REQUEST URL	DELETE https://api.figo.me/rest/devices/{device_id}
	device_id Internal figo Connect device ID.
RESPONSE	A sucessfull response contains no content.

REST API

Bank accounts and transactions can easily accessed via a **RESTful** web service with **JSON** objects.

/rest/accounts (GET)

DESCRIPTION	List all accounts that the user has chosen to share with your application.
PERMISSIONS	accounts=ro
REQUEST URL	GET https://api.figo.me/rest/accounts
RESPONSE	accounts List of account objects.
	Sample JSON response

```
{
  "accounts": [
    {
      "account_id": "A1.1",
      "bank_id": "B1.1",
      "name": "Girokonto",
      "owner": "figo",
      "account_number": "4711951500",
      "bank_code": "90090042",
      "bank_name": "Demobank",
      "currency": "EUR",
      "iban": "DE67900900424711951500",
      "bic": "DEMODE01",
      "type": "Giro account",
      "icon": "https://api.figo.me/assets/accounts/demokonto.",
      "status": {
        "code": 1,
        "sync_timestamp": "2013-04-11T19:16:00.000Z",
        "success_timestamp": "2013-04-11T19:16:00.000Z"
      }
    },
    {
      "account_id": "A1.2",
      "bank_id": "B1.1",
      "name": "Sparkonto",
      "owner": "figo",
      "account_number": "4711951501",
      "bank_code": "90090042",
      "bank_name": "Demobank",
      "currency": "EUR",
      "iban": "DE05900900424711951501",
      "bic": "DEMODE01",
```



```
    "type": "Savings account",
    "icon": "https://api.figo.me/assets/accounts/demokonto.",
    "status": {
      "code": -1,
      "message": "Could not get transactions.",
      "sync_timestamp": "2013-04-11T19:16:00.000Z",
      "success_timestamp": "2013-04-11T15:16:00.000Z"
    }
  }
]
```

/rest/accounts (POST)

DESCRIPTION Add new bank account. The figo Connect server will transparently create or modify a bank contact to add additional bank accounts.

This call will return immediately. Your application should monitor the progress by using the task token with `/task/progress`. After `/task/progress` has reported `is_ended`, your application can then use the REST API to retrieve any added accounts. NB: In this context, `/task/progress` will report `is_erroneous` instead of `is_waiting_for_pin`.

PERMISSIONS `accounts=rw`

REQUEST URL `POST https://api.figo.me/rest/accounts`

PARAMETERS **bank_code** *mandatory* Bank code.

country *mandatory* Two-letter country code; must be set to `de`.

credentials *mandatory* List of login credential strings. They must be in the same order as in the `credentials` list from the login settings.

save_pin This flag indicates whether the user has chosen to save the PIN on the figo Connect server. It is mandatory if the authentication type in the login settings of the bank code is `pin` and will be ignored otherwise.

disable_first_sync * This flag indicates whether the initial sync of the transactions and balances of the newly created accounts should be omitted. If this is the case certain listed accounts might actually be invalid (e.g. old creditcards) and will be automatically removed at the first sync.

Sample JSON request

```
{
  "bank_code": "90090042",
  "country": "de",
  "credentials": [
    "figo",
    "figo"
  ],
  "save_pin": true
}
```

RESPONSE **task_token** Task token.

Sample JSON response

```
{
  "task_token": "YmB-BtvbWufLnbwgAVfP7XfLatwhrtu0sATfnZNR7LGP
                LdZI--EqEPnwh_h6mCxToLEBhtA7LVd4uM4gTcZG8F6U
}
```

/rest/accounts (PUT)

DESCRIPTION Set account sort order.

PERMISSIONS accounts=rw

REQUEST URL PUT https://api.figo.me/rest/accounts

PARAMETERS **accounts** *mandatory* List of JSON objects with the field `account_id` set to the internal figo Connect account ID. The accounts will be sorted in the list order.

Sample JSON request

```
{
  "accounts": [
    {
      "account_id": "A1.1"
    },
    {
      "account_id": "A1.2"
    },
    {
      "account_id": "A1.3"
    }
  ]
}
```

RESPONSE A successful response contains no content.

/rest/accounts/{account_id} (GET)

DESCRIPTION Get information about a particular bank account.

PERMISSIONS accounts=ro

REQUEST URL GET https://api.figo.me/rest/accounts/{account_id}

account_id Internal figo Connect account ID.

RESPONSE A single **account** object.

Sample JSON response

```
{
  "account_id": "A1.1",
  "bank_id": "B1.1",
  "name": "Girokonto",
  "owner": "figo",
  "account_number": "4711951500",
  "task_token": "YmB-BtvbWufLnbwgAVfP7XfLatwhrtu0sATfnZNR7LGP
                LdZI--EqEPnwh_h6mCxToLEBhtA7LVd4uM4gTcZG8F6U
}
```

```
"bank_code": "90090042",
"bank_name": "Demobank",
"currency": "EUR",
"iban": "DE67900900424711951500",
"bic": "DEMODE01",
"type": "Giro account",
"icon": "https://api.figo.me/assets/accounts/demokonto.png"
"status": {
  "code": 1,
  "sync_timestamp": "2013-04-11T19:16:00.000Z",
  "success_timestamp": "2013-04-11T19:16:00.000Z"
}
}
```

/rest/accounts/{account_id} (PUT)

DESCRIPTION **Modify bank account settings.**

PERMISSIONS `accounts=rw`

REQUEST URL `PUT https://api.figo.me/rest/accounts/{account_id}`

account_id Internal figo Connect account ID.

PARAMETERS **name** Account name

owner Account owner

auto_sync This flag indicates whether the account will be automatically synchronized. It will be ignored for offline accounts and cash accounts.

account_number Account number; this parameter can only be set for offline accounts.

bank_code Bank code; this parameter can only be set for offline accounts.

account_number Account number; this parameter can only be set for offline accounts.

currency Three-character currency code; this parameter can only be set for offline accounts.

iban IBAN; this parameter can only be set for offline accounts.

bic BIC; this parameter can only be set for offline accounts.

type [Account type](#)

in_total_balance This flag indicates whether the balance of this account is added to the total balance of accounts.

preview This flag indicates whether this account is only shown as preview for an unpaid premium plan.

Sample JSON request

```
{
  "name": "Demokonto",
  "owner": Jon Doe,
  "in_total_balance": false
}
```

RESPONSE A successful response contains no content.

/rest/accounts/{account_id} (DELETE)

DESCRIPTION Remove bank account. Once the last remaining account of a bank contact has been removed, the bank contact will be automatically removed as well.

PERMISSIONS `accounts=rw`

REQUEST URL `DELETE https://api.figo.me/rest/accounts/{account_id}`

account_id Internal figo Connect account ID.

RESPONSE A successful response contains no content.

/rest/accounts/{account_id}/balance (GET)

DESCRIPTION Get balance and account limits.

PERMISSIONS `balance=ro`

REQUEST URL `GET https://api.figo.me/rest/accounts/{account_id}/balance`

account_id Internal figo Connect account ID.

RESPONSE **balance** Account balance; This response parameter will be omitted if the balance is not yet known.

balance_date Bank server timestamp of balance; This response parameter will be omitted if the balance is not yet known.

credit_line Credit line.

monthly_spending_limit User-defined spending limit.

status *Synchronization status* object.

Sample JSON response

```
{
  "balance": 3250.31,
  "balance_date": "2013-04-11T12:00:00.000Z",
  "credit_line": 0.00,
  "monthly_spending_limit": 1000.00,
  "status": {
    "code": 1,
    "sync_timestamp": "2013-04-11T19:16:00.000Z",
    "success_timestamp": "2013-04-11T19:16:00.000Z"
  }
}
```

/rest/accounts/{account_id}/balance (PUT)

DESCRIPTION Modify balance or account limits.

PERMISSIONS `balance=rw`

REQUEST URL

PUT https://api.figo.me/rest/accounts/{account_id}/balance

account_id Internal figo Connect account ID.

PARAMETERS

balance Account balance.

credit_line Credit line.

monthly_spending_limit User-defined spending limit.

Sample JSON request

```
{
  "credit_line": 500.00,
  "monthly_spending_limit": 250.00
}
```

RESPONSE

A successful response contains no content.

/rest/transactions

DESCRIPTION

Get transactions of all accounts that the user has chosen to share with your application.

PERMISSIONS

transactions=ro

REQUEST URL

GET <https://api.figo.me/rest/transactions>

PARAMETERS

since This parameter can either be a transaction ID or a date.

If this parameter is a transaction ID, then the transactions which are newer than the referenced transaction will be returned in the response. The exact meaning of newer depends on the value of `since_type`.

If this parameter is a date, then the transactions which were booked on or after this date will be returned in the response (and `since_type` will hence be ignored).

since_type This parameter defines how the parameter `since` will be interpreted.

If this parameter is set to `booked` (the default), then the value of the parameter `since` will be compared to the booking date of the transaction.

If this parameter is set to `created`, then the value of the parameter `since` will be compared to the creation time of the transaction on the figo Connect server.

If this parameter is set to `modified`, then the value of the parameter `since` will be compared to the last modification time of the transaction on the figo Connect server. The response parameter `transactions` will contain not only newly created transactions but also modified transactions. The response parameter `deleted` will contain a list of removed transactions.

filter Filter expression to narrow down the returned transactions.

start_id Do only return transactions which were booked after the start transaction ID. In combination with the `count` parameter this can be used to paginate the result list.

count Limit the number of returned transactions. The default number is 1000. In combination with the `start_id` parameter this can be used to paginate the result list.

offset Offset into the implicit list of transactions used as starting point for the returned transactions. In combination with the `count` parameter this can be used to paginate the result list.

include_pending This flag indicates whether pending transactions should be included in the response. Pending transactions are always included as a complete set, regardless of the `since` parameter. Before caching a copy of the pending transactions, all existing pending transactions for the same account must be removed from the local cache.

Example request

```
since=2013-04-10&include_pending=1
```

RESPONSE **transactions** List of **transaction** objects.

deleted List of deleted transaction IDs.

status **Synchronization status** object. The error messages of all accounts will be concatenated to a single error message. For oldest timestamp of all accounts will be used.

Sample JSON response

```
{
  "transactions": [
    {
      "transaction_id": "T1.1.24",
      "account_id": "A1.1",
      "name": "Dr. House Solutions GmbH",
      "account_number": "4711951501",
      "bank_code": "90090042",
      "bank_name": "Demobank",
      "amount": -300.00,
      "currency": "EUR",
      "booking_date": "2013-04-10T12:00:00.000Z",
      "value_date": "2013-04-10T12:00:00.000Z",
      "purpose": "Miete Vertragsnr. 12993",
      "type": "Direct debit",
      "booking_text": "Lastschrift",
      "booked": true,
      "creation_timestamp": "2013-04-10T08:21:36.000Z",
      "modification_timestamp": "2013-04-11T13:54:02.000Z",
      "visited": true
    },
    {
      "transaction_id": "T1.2.15",
      "account_id": "A1.2",
      "name": "Girokonto",
      "account_number": "4711951500",
      "bank_code": "90090042",
      "bank_name": "Demobank",
      "amount": 200.00,
      "currency": "EUR",
      "booking_date": "2013-04-11T12:00:00.000Z",
      "value_date": "2013-04-11T12:00:00.000Z",
      "purpose": "Sparen",
      "type": "Standing order",
      "booking_text": "Dauerauftrag",
      "booked": true,
      "creation_timestamp": "2013-04-11T13:54:42.000Z",
      "modification_timestamp": "2013-04-11T13:54:42.000Z",

```

```
        "visited": false
      }
    ],
    "status": {
      "code": -1,
      "message": "Could not get transactions.",
      "sync_timestamp": "2013-04-11T19:16:00.000Z",
      "success_timestamp": "2013-04-11T15:16:00.000Z"
    }
  }
}
```

/rest/accounts/{account_id}/transactions

DESCRIPTION Get transactions of a particular account.

PERMISSIONS transactions=ro

REQUEST URL GET https://api.figo.me/rest/accounts/{account_id}/transactions

account_id Internal figo Connect account ID.

PARAMETERS The same parameters as documented in the [previous](#) section.

RESPONSE **transactions** List of [transaction](#) objects.

deleted List of deleted transaction IDs.

status [Synchronization status](#) object.

Sample JSON response

```
{
  "transactions": [
    {
      "transaction_id": "T1.1.25",
      "account_id": "A1.1",
      "name": "Rogers Shipping, Inc.",
      "account_number": "4711951501",
      "bank_code": "90090042",
      "bank_name": "Demobank",
      "amount": -17.90,
      "currency": "EUR",
      "booking_date": "2013-04-11T12:00:00.000Z",
      "value_date": "2013-04-11T12:00:00.000Z",
      "purpose": "Ihre Sendung 0815 vom 01.03.2012, Vielen Da",
      "type": "Transfer",
      "booking_text": "Überweisung",
      "booked": false,
      "creation_timestamp": "2013-04-11T13:54:02.000Z",
      "modification_timestamp": "2013-04-11T13:54:02.000Z",
      "visited": true
    },
    {
      "transaction_id": "T1.1.24",
      "account_id": "A1.1",
      "name": "Dr. House Solutions GmbH",
      "account_number": "4711951501",
      "bank_code": "90090042",
      "bank_name": "Demobank",
      "amount": -300.00,
      "currency": "EUR",
```

```
        "booking_date": "2013-04-10T12:00:00.000Z",
        "value_date": "2013-04-10T12:00:00.000Z",
        "purpose": "Miete Vertragsnr. 12993",
        "type": "Direct debit",
        "booking_text": "Lastschrift",
        "booked": true,
        "creation_timestamp": "2013-04-10T08:21:36.000Z",
        "modification_timestamp": "2013-04-11T13:54:02.000Z",
        "visited": true
    }
],
"status": {
    "code": 1,
    "sync_timestamp": "2013-04-11T19:16:00.000Z",
    "success_timestamp": "2013-04-11T19:16:00.000Z"
}
```

/rest/accounts/{account_id}/transactions (POST)

DESCRIPTION Insert transaction into transaction list.

PERMISSIONS transactions=rw

REQUEST URL POST https://api.figo.me/rest/accounts/{account_id}/transactions

account_id Internal figo Connect account ID.

PARAMETERS **name** Name of originator or recipient.

account_number Account number of originator or recipient.

bank_code Bank code of originator or recipient.

amount *mandatory* Transaction amount.

currency Three-character currency code.

booking_date *mandatory* Booking date.

value_date Value date.

purpose Purpose text.

type Transaction type.

booking_text Booking text.

booked This flag indicates whether the transaction is booked or pending.

visited This flag indicates whether the transaction has already been marked as visited by the user.

Sample JSON request

```
{
  "amount": 1.00,
  "booking_date": "2013-07-01",
  "purpose": "Donation",
}
```


RESPONSE A single **transaction** object.

Sample JSON response

```
{
  "transaction_id": "T1.1.26",
  "account_id": "A1.1",
  "name": "",
  "account_number": "",
  "bank_code": "",
  "bank_name": "",
  "amount": 1.00,
  "currency": "EUR",
  "booking_date": "2013-07-01T12:00:00.000Z",
  "value_date": "2013-07-01T12:00:00.000Z",
  "purpose": "Donation",
  "type": "Unknown",
  "booking_text": "",
  "booked": true,
  "creation_timestamp": "2013-07-01T11:46:40.000Z",
  "modification_timestamp": "2013-01-07T11:46:40.000Z",
  "visited": true
}
```

/rest/accounts/{account_id}/transactions (PUT)

DESCRIPTION Update all transactions in the respective account

PERMISSIONS transactions=rw

REQUEST URL PUT https://api.figo.me/rest/accounts/{account_id}/transactions

account_id Internal figo Connect account ID.

PARAMETERS **visited** This flag indicates whether the transaction has already been marked as visited by the user.

Sample JSON request

```
{
  "visited": true
}
```

RESPONSE A successful response contains no content.

/rest/accounts/{account_id}/transactions/{transaction_id} (GET)

DESCRIPTION Get information about a particular transaction.

PERMISSIONS transactions=ro

REQUEST URL GET https://api.figo.me/rest/accounts/{account_id}/transactions/{transaction_id}

account_id Internal figo Connect account ID.

transaction_id Internal figo Connect transaction ID.

RESPONSE A single **transaction** object.

Sample JSON response

```
{
  "transaction_id": "T1.1.25",
  "account_id": "A1.1",
  "name": "Rogers Shipping, Inc.",
  "account_number": "4711951501",
  "bank_code": "90090042",
  "bank_name": "Demobank",
  "amount": -17.90,
  "currency": "EUR",
  "booking_date": "2013-04-11T12:00:00.000Z",
  "value_date": "2013-04-11T12:00:00.000Z",
  "purpose": "Ihre Sendung 0815 vom 01.03.2012, Vielen Dank",
  "type": "Transfer",
  "booking_text": "Überweisung",
  "booked": false,
  "creation_timestamp": "2013-04-11T13:54:02.000Z",
  "modification_timestamp": "2013-04-11T13:54:02.000Z",
  "visited": true
}
```

/rest/accounts/{account_id}/transactions/{transaction_id} (PUT)

DESCRIPTION **Modify transaction.**

PERMISSIONS transactions=rw

REQUEST URL PUT https://api.figo.me/rest/accounts/{account_id}/
transactions/{transaction_id}

account_id Internal figo Connect account ID.

transaction_id Internal figo Connect transaction ID.

PARAMETERS **name** Name of originator or recipient.

account_number Account number of originator or recipient.

bank_code Bank code of originator or recipient.

amount Transaction amount.

currency Three-character currency code.

booking_date Booking date.

value_date Value date.

purpose Purpose text.

type **Transaction type.**

booking_text Booking text.

booked This flag indicates whether the transaction is booked or pending.

visited This flag indicates whether the transaction has already been marked as visited by the user.

Sample JSON request

```
{
  "visited": true
}
```

RESPONSE A successful response contains no content.

/rest/accounts/{account_id}/transactions/{transaction_id} (DELETE)

DESCRIPTION Delete transaction.

PERMISSIONS transactions=rw

REQUEST URL

```
DELETE https://api.figo.me/rest/accounts/{account_id}/
transactions/{transaction_id}
```

account_id Internal figo Connect account ID.

transaction_id Internal figo Connect transaction ID.

RESPONSE A successful response contains no content.

/rest/banks/{bank_id} (GET)

DESCRIPTION Get bank contact settings.

PERMISSIONS accounts=ro

REQUEST URL

```
GET https://api.figo.me/rest/banks/{bank_id}
```

bank_id Internal figo Connect bank ID.

RESPONSE **sepa_creditor_id** SEPA direct debit creditor ID.

save_pin This flag indicates whether the user has chosen to save the PIN on the figo Connect server.

Sample JSON response

```
{
  "sepa_creditor_id": "DE02ZZZ0123456789",
  "save_pin": true
}
```

/rest/banks/{bank_id} (PUT)

DESCRIPTION Modify bank contact settings.

PERMISSIONS accounts=rw

REQUEST URL	PUT <code>https://api.figo.me/rest/banks/{bank_id}</code>
	bank_id Internal figo Connect bank ID.
PARAMETERS	sepa_creditor_id SEPA direct debit creditor ID.
	Sample JSON request
	<pre>{ "sepa_creditor_id": "DE02ZZZ0123456789" }</pre>
RESPONSE	A successful response contains no content.

/rest/banks/{bank_id}/remove_pin

DESCRIPTION	Remove the stored PIN of a bank contact from the figo Connect server.
PERMISSIONS	accounts=rw
REQUEST URL	POST <code>https://api.figo.me/rest/banks/{bank_id}/remove_pin</code>
	bank_id Internal figo Connect bank ID.
RESPONSE	A successful response contains no content.

Payments

/rest/accounts/{account_id}/payments (GET)

DESCRIPTION	List all payments known to the figo Connect server.
PERMISSIONS	payments=ro
REQUEST URL	GET <code>https://api.figo.me/rest/accounts/{account_id}/payments</code>
	account_id Internal figo Connect account ID.
PARAMETERS	since This parameter can either be a payment ID or a date. If this parameter is a payment ID, then the payments which were created after the referenced payment will be returned in the response. If this parameter is a date, then the payments which were created on or after this date will be returned in the response. count Limit the number of returned payments. The default number is 1000.
	Example request
	<code>since=2013-06-15</code>

RESPONSE **payments** List of **payment** objects. If one of the payment objects is a container for multiple payments, then the field `container` will be set to the empty list. The complete payment object can be fetched with `/rest/accounts/{account_id}/payments/{payment_id}`.

Sample JSON response

```
{
  "payments": [
    {
      "payment_id": "P1.1.234",
      "account_id": "A1.1",
      "type": "Transfer",
      "name": "figo",
      "account_number": "4711951501",
      "bank_code": "90090042",
      "amount": 0.89,
      "currency": "EUR",
      "purpose": "Thanks for all the fish.",
      "text_key": 51,
      "text_key_extension": 0,
      "notification_recipient": "",
      "creation_timestamp": "2013-07-16T13:53:56.000Z",
      "modification_timestamp": "2013-07-16T13:53:56.000Z"
    }
  ]
}
```

/rest/accounts/{account_id}/payments (POST)

DESCRIPTION Insert payment into payment list.

PERMISSIONS payments=rw

REQUEST URL POST https://api.figo.me/rest/accounts/{account_id}/payments

account_id Internal figo Connect account ID.

PARAMETERS **type** *mandatory* **Payment type**

name Name of creditor or debtor; This field is mandatory unless `container` is set, in which case this field will be ignored.

account_number Account number or IBAN of creditor or debtor; This field is mandatory unless `container` is set, in which case this field will be ignored.

bank_code Bank code or BIC of creditor or debtor; This field is mandatory unless `container` is set, in which case this field will be ignored.

amount Order amount; This field is mandatory unless `container` is set, in which case this field will be ignored.

currency Three-character currency code; This field defaults to the account's currency and will be ignored if `container` is set.

purpose Purpose text; This field will be ignored if `container` is set.

text_key DTA text key; This field will be ignored if `container` is set.

text_key_extension DTA text key extension; This field will be ignored if `container` is set.

notification_recipient Recipient of the payment notification, should be an email address.

scheduled_date Scheduled date specified according to ISO 8601. Recurring time intervals for standing orders are specified according ISO 8601, too.

container List of **payment** objects. This payment can only be a container for multiple payments if the field `supported_file_formats` of the **payment parameters** is not empty.

Sample JSON request

```
{
  "type": "Transfer",
  "name": "figo",
  "account_number": "4711951501",
  "bank_code": "90090042",
  "amount": 0.89,
  "purpose": "Thanks for all the fish."
}
```

RESPONSE A single **payment** object.

Sample JSON response

```
{
  "payment_id": "P1.1.234",
  "account_id": "A1.1",
  "type": "Transfer",
  "name": "figo",
  "account_number": "4711951501",
  "bank_code": "90090042",
  "bank_name": "Demobank",
  "amount": 0.89,
  "currency": "EUR",
  "purpose": "Thanks for all the fish.",
  "text_key": 51,
  "text_key_extension": 0,
  "notification_recipient": "",
  "creation_timestamp": "2013-07-16T13:53:56.000Z",
  "modification_timestamp": "2013-07-16T13:53:56.000Z"
}
```

/rest/accounts/{account_id}/payments (file upload)

DESCRIPTION Insert container with multiple payments into payment list.

PERMISSIONS `payments=rw`

REQUEST URL `POST https://api.figo.me/rest/accounts/{account_id}/payments`

account_id Internal figo Connect account ID.

PARAMETERS **file_content** *mandatory* Base64 encoded file with payments.

file_format *mandatory* Payment **file format**.

Sample JSON request

```
{
  "file_content": "(file content here)",
  "file_format": "DTAUS0"
}
```

RESPONSE A single **payment** object. The amount will be set to the total amount of all payments in the container.

Sample JSON response

```
{
  "payment_id": "P1.1.235",
  "account_id": "A1.1",
  "type": "Transfer",
  "amount": 700.00,
  "currency": "EUR",
  "container": [
    {
      "payment_id": "P1.1.235.1",
      "account_id": "A1.1",
      "type": "Transfer",
      "name": "figo",
      "account_number": "4711951501",
      "bank_code": "90090042",
      "bank_name": "Demobank",
      "amount": 400.00,
      "currency": "EUR",
      "purpose": "Order nb. 888",
      "text_key": 51,
      "text_key_extension": 0,
      "notification_recipient": "",
      "creation_timestamp": "2013-07-17T11:23:47.000Z",
      "modification_timestamp": "2013-07-17T11:23:47.000Z"
    },
    {
      "payment_id": "P1.1.235.2",
      "account_id": "A1.1",
      "type": "Transfer",
      "name": "figo",
      "account_number": "4711951502",
      "bank_code": "90090042",
      "bank_name": "Demobank",
      "amount": 300.00,
      "currency": "EUR",
      "purpose": "Order nb. 999",
      "text_key": 51,
      "text_key_extension": 0,
      "notification_recipient": "",
      "creation_timestamp": "2013-07-17T11:23:47.000Z",
      "modification_timestamp": "2013-07-17T11:23:47.000Z"
    }
  ]
  "creation_timestamp": "2013-07-17T11:23:47.000Z",
  "modification_timestamp": "2013-07-17T11:23:47.000Z"
}
```

/rest/accounts/{account_id}/payments/{payment_id} (GET)

DESCRIPTION Get information about a particular payment.

PERMISSIONS payments=ro

REQUEST URL GET https://api.figo.me/rest/accounts/{account_id}/payments/{payment_id}

account_id Internal figo Connect account ID.

payment_id Internal figo Connect payment ID.

RESPONSE A single **payment** object.

Sample JSON response

```
{
  "payment_id": "P1.1.234",
  "account_id": "A1.1",
  "type": "Transfer",
  "name": "figo",
  "account_number": "4711951501",
  "bank_code": "90090042",
  "bank_name": "Demobank",
  "amount": 0.89,
  "currency": "EUR",
  "purpose": "Thanks for all the fish.",
  "text_key": 51,
  "text_key_extension": 0,
  "notification_recipient": "",
  "creation_timestamp": "2013-07-16T13:53:56.000Z",
  "modification_timestamp": "2013-07-16T13:53:56.000Z"
}
```

/rest/accounts/{account_id}/payments/{payment_id} (PUT)

DESCRIPTION **Modify payment in list.**

If the payments has already been submitted to the bank server, i.e. the field `submission_timestamp` is set, then the only field that can be modified with this method is `transaction_id`.

PERMISSIONS `payments=rw`

REQUEST URL `PUT https://api.figo.me/rest/accounts/{account_id}/payments/{`

account_id Internal figo Connect account ID.

payment_id Internal figo Connect payment ID.

PARAMETERS **name** Name of creditor or debtor

account_number Account number of creditor or debtor

bank_code Bank code of creditor or debtor

amount Order amount

currency Three-character currency code

purpose Purpose text

text_key DTA text key

text_key_extension DTA text key extension

notification_recipient Recipient of the payment notification, should be an email address.

scheduled_date Scheduled date specified according to ISO 8601. Recurring time intervals for standing orders are specified according ISO 8601, too.

container List of **payment** objects. This payment can only be a container for multiple payments if the field `supported_file_formats` of the **payment parameters** is not empty.

transaction_id This field is set to the transaction ID that has been matched to the payment.

Sample JSON request

```
{
  "amount": 8.99,
  "purpose": "Thanks for all the fish."
}
```

RESPONSE A sucessfull response contains no content.

/rest/accounts/{account_id}/payments/{payment_id} (DELETE)

DESCRIPTION Delete payment from list.

PERMISSIONS `payments=rw`

REQUEST URL `DELETE https://api.figo.me/rest/accounts/{account_id}/payment`

account_id Internal figo Connect account ID.

payment_id Internal figo Connect payment ID.

RESPONSE A sucessfull response contains no content.

/rest/accounts/{account_id}/payments/{payment_id}/submit

DESCRIPTION Submit payment to bank server.

This call will return immediately. Your application must either direct the user’s web browser to a pop window of figo Connect with the URL `/task/start`, or moniter the progress itself by using the task token with `/task/progress`.

PERMISSIONS `submit_payments`

REQUEST URL `POST https://api.figo.me/rest/accounts/{account_id}/payments/`

account_id Internal figo Connect account ID.

payment_id Internal figo Connect payment ID.

PARAMETERS **redirect_uri** *mandatory* At the end of the submission process a response will be sent to this callback URL.

state *mandatory* Any kind of string that will be forwarded in the callback response message. It

serves two purposes: The value is used to maintain state between this request and the callback, e.g. it might contain a session ID from your application. The value should also contain a random component, which your application checks to mitigate cross-site request forgery.

tan_scheme_id *mandatory* TAN scheme ID of user-selected TAN scheme.

Sample JSON request

```
{
  "redirect_uri": "https://api.figo.me/callback",
  "state": "4HgwtQP0dnX4UjPf",
  "tan_scheme_id": "M1.1.1"
}
```

RESPONSE **task_token** Task token.

Sample JSON response

```
{
  "task_token": "YmB-BtvbWufLnbgwAVfP7XfLatwhrtu0sATfnZNR7LGF
                LdZI--EqEPnwh_h6mCxToLEBhtA7LVd4uM4gTcZG8F6U"
}
```

/rest/accounts/{account_id}/scheduled (GET)

DESCRIPTION List all scheduled payments known to the bank server.

PERMISSIONS payments=ro

REQUEST URL GET https://api.figo.me/rest/accounts/{account_id}/scheduled

account_id Internal figo Connect account ID.

RESPONSE **payments** List of **payment** objects. If one of the payment objects is a container for multiple payments, then the field `container` will be set to the empty list. The complete payment object can be fetched with `/rest/accounts/{account_id}/scheduled/{payment_id}`.

In addition to the usual fields in the **payment** objects two more fields are used: `can_be_modified` and `can_be_deleted`. The first one is a list of fields which are modifiable on the bank server. The second one is a flag which is set to `true` if the payment can be deleted from the bank server.

status **Synchronization status** object.

Sample JSON response

```
{
  "payments": [
    {
      "payment_id": "P1.1.236",
      "account_id": "A1.1",
      "type": "Transfer",
      "name": "figo",
      "account_number": "4711951501",

```

```

        "bank_code": "90090042",
        "amount": 0.89,
        "currency": "EUR",
        "purpose": "Thanks for all the fish.",
        "text_key": 51,
        "text_key_extension": 0,
        "scheduled_date": "2013-07-22",
        "creation_timestamp": "2013-07-18T11:57:59.000Z",
        "modification_timestamp": "2013-07-18T11:57:59.000Z",
        "can_be_modified": [
            "name",
            "amount",
            "purpose"
        ],
        "can_be_deleted": true
    }
],
"status": {
    "code": 1,
    "sync_timestamp": "2013-04-11T19:16:00.000Z",
    "success_timestamp": "2013-04-11T19:16:00.000Z"
}
}

```

/rest/accounts/{account_id}/scheduled/{payment_id} (GET)

DESCRIPTION Get information about a particular scheduled payment.

PERMISSIONS payments=ro

REQUEST URL GET https://api.figo.me/rest/accounts/{account_id}/scheduled/

account_id Internal figo Connect account ID.

payment_id Internal figo Connect payment ID.

RESPONSE A single **payment** object.

In addition to the usual fields in the **payment** objects two more fields are used:

`can_be_modified` and `can_be_deleted`. The first one is a list of fields which are modifiable on the bank server. The second one is a flag which is set to `true` if the payment can be deleted from the bank server.

Sample JSON response

```

{
  "payment_id": "P1.1.236",
  "account_id": "A1.1",
  "type": "Transfer",
  "name": "figo",
  "account_number": "4711951501",
  "bank_code": "90090042",
  "amount": 0.89,
  "currency": "EUR",
  "purpose": "Thanks for all the fish.",
  "text_key": 51,
  "text_key_extension": 0,
  "scheduled_date": "2013-07-22",
  "creation_timestamp": "2013-07-18T11:57:59.000Z",
  "modification_timestamp": "2013-07-18T11:57:59.000Z",
  "can_be_modified": [

```

```
        "name",
        "amount",
        "purpose"
    ],
    "can_be_deleted": true
}
```

/rest/accounts/{account_id}/scheduled/{payment_id} (PUT)

DESCRIPTION **Modify scheduled payment on bank server.**

This call will return immediately. Your application must either direct the user's web browser to a pop window of figo Connect with the URL `/task/start`, or monitor the progress itself by using the task token with `/task/progress`.

PERMISSIONS `submit_payments`

REQUEST URL PUT `https://api.figo.me/rest/accounts/{account_id}/scheduled/`

account_id Internal figo Connect account ID.

payment_id Internal figo Connect payment ID.

PARAMETERS **redirect_uri** *mandatory* At the end of the submission process a response will be sent to this callback URL.

state *mandatory* Any kind of string that will be forwarded in the callback response message. It serves two purposes: The value is used to maintain state between this request and the callback, e.g. it might contain a session ID from your application. The value should also contain a random component, which your application checks to mitigate cross-site request forgery.

tan_scheme_id *mandatory* TAN scheme ID of user-selected TAN scheme.

name Name of creditor or debtor

account_number Account number of creditor or debtor

bank_code Bank code of creditor or debtor

amount Order amount

currency Three-character currency code

purpose Purpose text

text_key DTA text key

text_key_extension DTA text key extension

scheduled_date Scheduled date. Recurring time intervals for standing orders are specified according ISO 8601.

Sample JSON request

```
{
  "redirect_uri": "https://api.figo.me/callback",
  "state": "4HgwtQP0dnX4UjPf",
  "tan_scheme_id": "M1.1.1",
  "scheduled_date": "2017-01-01T00:00:00",
  "amount": 100,
  "currency": "EUR",
  "purpose": "Monthly payment",
  "text_key": "DTA",
  "text_key_extension": "EXT",
  "can_be_deleted": true
}
```

```
"amount": 8.99,  
"purpose": "Thanks for all the fish."  
}
```

RESPONSE **task_token** Task token.

Sample JSON response

```
{  
  "task_token": "YmB-BtvbWufLnbwgAVfP7XfLatwhrtu0sATfnZNR7LGP  
                LdZI--EqEPnwh_h6mCxToLEBhtA7LVd4uM4gTcZG8F6U  
}
```

/rest/accounts/{account_id}/scheduled/{payment_id} (DELETE)

DESCRIPTION **Delete scheduled payment from bank server.**

This call will return immediately. Your application must either direct the user's web browser to a pop window of figo Connect with the URL `/task/start`, or monitor the progress itself by using the task token with `/task/progress`.

PERMISSIONS `submit_payments`

REQUEST URL `DELETE https://api.figo.me/rest/accounts/{account_id}/scheduled/{payment_id}`

account_id Internal figo Connect account ID.

payment_id Internal figo Connect payment ID.

PARAMETERS **redirect_uri** *mandatory* At the end of the submission process a response will be sent to this callback URL.

state *mandatory* Any kind of string that will be forwarded in the callback response message. It serves two purposes: The value is used to maintain state between this request and the callback, e.g. it might contain a session ID from your application. The value should also contain a random component, which your application checks to mitigate cross-site request forgery.

tan_scheme_id *mandatory* TAN scheme ID of user-selected TAN scheme.

RESPONSE **task_token** Task token.

Sample JSON response

```
{  
  "task_token": "YmB-BtvbWufLnbwgAVfP7XfLatwhrtu0sATfnZNR7LGP  
                LdZI--EqEPnwh_h6mCxToLEBhtA7LVd4uM4gTcZG8F6U  
}
```

Catalog

/rest/catalog/services/de

DESCRIPTION	Get list of supported credit card and payment services.
PERMISSIONS	accounts=rw
REQUEST URL	GET https://api.figo.me/rest/catalog/services/de

RESPONSE **services** List of **service** objects.

Sample JSON response

```
{
  "services": [
    {
      "name": "Demokonto",
      "bank_code": "90090042",
      "icon": "https://api.figo.me/assets/accounts/demokonto."
    }
  ]
}
```

/rest/catalog/banks/de/{bank_code}

DESCRIPTION	Get login settings for bank code.
PERMISSIONS	accounts=rw
REQUEST URL	GET https://api.figo.me/rest/catalog/banks/de/{bank_code}

bank_code Bank code

RESPONSE **bank_name** Bank name.

supported This flag indicates whether this bank is supported by figo.

credentials List of **credential** objects.

auth_type If the authentication type is `pin`, then the user must have the option to save or not to save his or her PIN. If the authentication type is `none` or `token`, then there is no such option.

advice Help text.

icon Icon URL.

Sample JSON response

```
{
  "bank_name": "Demobank",
  "supported": true,
  "credentials": [
    {
      "label": "Benutzername"
    },
    {
      "label": "PIN",
      "masked": true
    }
  ]
}
```

```
},
"auth_type": "pin",
"advice": "Benutzername: figo, PIN: figo",
"icon": "https://api.figo.me/assets/accounts/demokonto.png"
}
```

Synchronization

During synchronization the figo Connect server fetches the current balance, recent transactions and scheduled payments from the banks, payment providers or financial service providers.

Usually the figo Connect server synchronizes all bank accounts on a daily basis. However, the synchronization has to be triggered manually either if the user disabled automatic synchronization or there was a PIN error or the PIN has not been saved on the figo Connect server.

/rest/sync

DESCRIPTION	Your application initiates the synchronization process by requesting a task token.
PERMISSIONS	balance=ro or transactions=ro or payments=ro
REQUEST URL	POST https://api.figo.me/rest/sync
PARAMETERS	<p>redirect_uri <i>mandatory</i> At the end of the synchronization process a response will be sent to this callback URL.</p> <p>state <i>mandatory</i> Any kind of string that will be forwarded in the callback response message. It serves two purposes: The value is used to maintain state between this request and the callback, e.g. it might contain a session ID from your application. The value should also contain a random component, which your application checks to mitigate cross-site request forgery.</p> <p>disable_notifications This flag indicates whether notifications should be sent to your application. Since your application will be notified by the callback URL anyway, you might want to disable any additional notifications.</p> <p>if_not_synced_since If this parameter is set, only those accounts will be synchronized, which have not been synchronized within the specified number of minutes.</p> <p>account_ids Only sync the accounts with these IDs</p> <p>account_filter Python re expression to filter the accounts to be synced. The re will be applied to the bank codes and bank names</p>

Sample JSON request

```
{
  "redirect_uri": "https://api.figo.me/callback",
  "state": "4HgwtQP0dnX4UjPf",
  "disable_notifications": true,
  "if_not_synced_since": 15
}
```

RESPONSE

task_token Task token.

Sample JSON response

```
{
  "task_token": "YmB-BtvbWufLnbwgAVfP7XfLatwhrtu0sATfnZNR7LGP
                LdZI--EqEPnwh_h6mCxToLEBhtA7LVd4uM4gTcZG8F6U
}
```

/task/start

DESCRIPTION Start communication with bank server. After your application obtained a task token, it may direct the user’s web browser to a popup window of figo Connect.

The popup window will ask the user for his or her account PIN, perform two-factor authentication or ask for a TAN. It also displays status messages as the task proceeds.

After the task has been finished, the figo Connect server sends a response message to the callback URL by redirecting the browser to `redirect_uri`. Your application can then use the REST API to retrieve the current balance, recent transactions and scheduled payments.

POPUP WINDOW URL `https://api.figo.me/task/start?id={id}`

id Task token from the initial request.

REDIRECT **state** The `state` parameter from the request.

Sample redirect

```
GET /callback?state=4HgwtQP0dnX4UjPf
```

/task/progress

DESCRIPTION While the figo Connect server communicates with a bank server, your application can monitor its progress by periodically polling this method.

REQUEST URL `POST https://api.figo.me/task/progress?id={id}`

id Task token.

PARAMETERS **continue** This flag signals to continue after an error condition or to skip a PIN or challenge-response entry.

pin Submit PIN. If this parameter is set, then the parameter `save_pin` must be set, too.

response Submit response to challenge.

save_pin This flag indicates whether the user has chosen to save the PIN on the figo Connect server.

Example request

```
pin=figo&save_pin=0
```


RESPONSE **account_id** Account ID of currently processed account.

message Status message or error message for currently processed account.

is_waiting_for_pin If this flag is set, then the figo Connect server waits for a PIN.

is_waiting_for_response If this flag is set, then the figo Connect server waits for a response to the parameter `challenge`.

is_erroneous If this flag is set, then an error occurred and the figo Connect server waits for a continuation.

is_ended If this flag is set, then the communication with the bank server has been completed.

challenge **Challenge** object.

Example JSON response

```
{
  "account_id" = "A1.2",
  "message" = "Getting balance...",
  "is_waiting_for_pin": false,
  "is_waiting_for_response": false,
  "is_erroneous": false,
  "is_ended": false
}
```

/task/cancel

DESCRIPTION **Cancel bank server communication.**

REQUEST URL `POST https://api.figo.me/task/cancel?id={id}`

id Task token.

RESPONSE **A sucessfull response contains no content.**

Notifications

Whenever the bank account of a user received new transactions, the figo Connect server can asynchronously notify your application with **webooks**.

/rest/notifications (GET)

DESCRIPTION **List all notifications your application did register for the user on the current device.**

REQUEST URL `GET https://api.figo.me/rest/notifications`

RESPONSE **notifications** List of **registered notification** objects.

Sample JSON response

```
{
  "notifications": [
    {
      "notification_id": "N1.7",
      "observe_key": "/rest/transactions?include_pending=0",
      "notify_uri": "https://api.figo.me/callback",
      "state": "cjLaN3lONdeLJQH3"
    }
  ]
}
```

/rest/notifications (POST)

DESCRIPTION Register your application to receive notifications.

REQUEST URL POST `https://api.figo.me/rest/notifications`

PARAMETERS **observe_key** *mandatory* Notification key

notify_uri *mandatory* Notification messages will be sent to this URL. The URL schemes `https://` and `http://` are used for webhooks. URLs with the scheme `apns://` denote the Apple Push Notification Service and have the form

`apns://{bundle_id}/{apns_token}?sandbox={sandbox}`, where `bundle_id` is the iOS App bundle identifier, `apns_token` is the device APNS token issued by Apple and `sandbox` is a flag which indicates whether the token has been issued by the Apple sandbox server.

state Mandatory for URL schemes `https://` and `http://` and ignored for URL scheme `apns://`. The value can be any kind of string that will be forwarded in the notification message. It serves two purposes: The value is used to maintain state between this request and the notification message, e.g. it might contain an user ID from your application. The value should also contain a random component, which your application checks to mitigate cross-site request forgery.

Sample JSON request

```
{
  "observe_key": "/rest/transactions?include_pending=0",
  "notify_uri": "https://api.figo.me/callback",
  "state": "cjLaN3lONdeLJQH3"
}
```

RESPONSE **notification_id** Internal figo Connect notification ID.

Remainder Properties of the newly created notification

Sample JSON response

```
{
  "notification_id": "N1.7"
  "observe_key": "/rest/transactions?include_pending=0",
  "notify_uri": "https://api.figo.me/callback",
  "state": "cjLaN3lONdeLJQH3"
}
```

/rest/notifications/{notification_id} (GET)

DESCRIPTION Get information about a particular notification.

REQUEST URL `GET https://api.figo.me/rest/notifications/{notification_id}`

notification_id Internal figo Connect notification ID.

RESPONSE A single **registered notification** object.

Sample JSON response

```
{
  "notification_id": "N1.7",
  "observe_key": "/rest/transactions?include_pending=0",
  "notify_uri": "https://api.figo.me/callback",
  "state": "cjLaN3lONdeLJQH3"
}
```

/rest/notifications/{notification_id} (PUT)

DESCRIPTION Modify notification.

REQUEST URL `PUT https://api.figo.me/rest/notifications/{notification_id}`

notification_id Internal figo Connect notification ID.

REQUEST: **observe_key** Notification key

notify_uri Notification messages will be sent to this URL.

state Any kind of string that will be forwarded in the notification message.

Sample JSON request

```
{
  "notify_uri": "https://api.figo.me/callback",
}
```

RESPONSE A sucessfull response contains no content.

/rest/notifications/{notification_id} (DELETE)

DESCRIPTION Unregister notification.

REQUEST URL `DELETE https://api.figo.me/rest/notifications/{notification_id}`

notification_id Internal figo Connect notification ID.

RESPONSE A sucessfull response contains no content.

Webhooks

DESCRIPTION	Whenever the figo Connect server sends a notification message to your webhook URL, your application can update its accounts, balance or transactions with the REST API.
WEBHOOK PARAMETERS	<p>notification_id Internal figo Connect notification ID from the notification registration response.</p> <p>observe_key Notification key</p> <p>state The <code>state</code> parameter from the notification registration request.</p> <p>Sample webhook message</p> <pre>{ "notification_id": "N1.7", "observe_key": "/rest/transactions?include_pending=0", "state": "cjLaN3lONdeLJQH3" }</pre>

Notification keys

Account balance

DESCRIPTION	Triggered when the respective account balance changes.
PERMISSIONS	<code>balance=ro</code>
NOTIFICATION KEY	<code>/rest/accounts/{account_id}/balance</code>
	account_id Internal figo Connect account ID.
PARAMETERS	inferior_limit Trigger if the balance of the account is under the provided value.

Account transactions

DESCRIPTION	Triggered when an account has received new transactions.
PERMISSIONS	<code>transactions=ro</code>
NOTIFICATION KEY	<code>/rest/accounts/{account_id}/transactions</code>
	account_id Internal figo Connect account ID.
PARAMETERS	<p>include_pending Trigger not only for booked but also for pending transactions.</p> <p>more_expenses_then_deposits Trigger only if the sum of expenses in the current month exceeds the sum of deposits in the same time. Only combinable with <code>include_pending</code>.</p> <p>current_month_expense_goal Trigger only if the sum of expenses in the current month exceeds the provided value. Only combinable with <code>include_pending</code>.</p> <p>single_expense_goal Trigger only for expense transactions exceeding the provided value. Only</p>

combinable with `include_pending`.

single_deposit_goal Trigger only for expense transactions exceeding the provided value. Only combinable with `include_pending`.

purpose Trigger only on transactions whose purpose contains the provided value. Only combinable with `include_pending`.

name Trigger only on transactions whose sender/receiver name contains the provided value. Only combinable with `include_pending`.

All transactions

DESCRIPTION	Triggered when any of the accounts that the user has chosen to share with your application has received new transactions.
PERMISSIONS	<code>transactions=ro</code>
NOTIFICATION KEY	<code>/rest/transactions</code>
PARAMETERS	include_pending Trigger not only for booked but also for pending transactions.

Test notification

DESCRIPTION	Triggered immediately. This special notification key can be used to test the delivery of notifications. The notification message will be sent immediately and no registration occurs.
NOTIFICATION KEY	<code>/rest/notifications/test</code>

Common JSON objects

Permissions

Permission	Description
<code>accounts=ro</code>	Read-only access to the list of accounts and account details.
<code>accounts=rw</code>	Read-write access to the list of accounts and account details.
<code>balance=ro</code>	Read-only access to the account balance and limits.
<code>balance=rw</code>	Read-write access to the account balance and limits.
<code>transactions=ro</code>	Read-only access to the account transactions.
<code>transactions=rw</code>	Read-write access to the account transactions.

payments=ro	Read-only access to the account payments.
payments=rw	Read-write access to the account payments.
user=ro	Read-only access to the figo Account settings.
user=rw	Read-write access to the figo Account settings.
clients=ro	Read-only access to the figo Connect client management.
clients=rw	Read-write access to the figo Connect client management.
submit_payments	Allow submission of payments to bank server.
offline	Access figo Connect when the user is not present.

Client object

Field	Description
client_id	Internal figo Connect client ID
name	Client name
homepage	Homepage URL
description	Client description
icon	Icon URL
scope	A space delimited set of permissions for the client.
valid	This flag indicates whether the client is still authorized.
last_access	Timestamp of the last request this client made.
accounts	List of account IDs. The client has access to these accounts.

Device object

Field	Description
device_id	Internal figo Connect device ID
name	Device name
icon	Icon URL
last_access	Timestamp of the last request this device made.

Service object

Field	Description
name	Credit card or payment service name
bank_code	Bank code
icon	Icon URL

Account object

Field	Description
account_id	Internal figo Connect account ID
bank_id	Internal figo Connect bank ID
name	Account name
owner	Account owner
auto_sync	This flag indicates whether the account will be automatically synchronized.
account_number	Account number
bank_code	Bank code
bank_name	Bank name
currency	Three-character currency code
iban	IBAN
bic	BIC
type	Account type
icon	Account icon URL
supported_payments	List of payment types with payment parameters objects
supported_tan_schemes	List of TAN scheme objects
preferred_tan_scheme	ID of the TAN scheme preferred by the user
in_total_balance	This flag indicates whether the balance of this account is added to the total balance of accounts.
save_pin	This flag indicates whether the user has chosen to save the PIN on the figo Connect server.
preview	This flag indicates whether this account is only

shown as preview for an unpaid premium plan.	
status	Synchronization status object

Account types

Account type constants
Giro account
Savings account
Credit card
Loan account
PayPal
Cash book
Unknown

Transaction object

Field	Description
transaction_id	Internal figo Connect transaction ID
account_id	Internal figo Connect account ID
name	Name of originator or recipient
account_number	Account number of originator or recipient
bank_code	Bank code of originator or recipient
bank_name	Bank name of originator or recipient
amount	Transaction amount
currency	Three-character currency code
booking_date	Booking date
value_date	Value date
purpose	Purpose text
type	Transaction type
booking_text	Booking text
booked	This flag indicates whether the transaction is booked or pending.
creation_timestamp	Internal creation timestamp on the figo

	Connect server.
modification_timestamp	Internal modification timestamp on the figo Connect server.
visited	This flag indicates whether the transaction has already been marked as visited by the user.

Transaction types

Transaction type constants
Transfer
Standing order
Direct debit
Salary or rent
Electronic cash
GeldKarte
ATM
Charges or interest
Unknown

Payment object

Field	Description
payment_id	Internal figo Connect payment ID
account_id	Internal figo Connect account ID
type	Payment type
name	Name of creditor or debtor
account_number	Account number of creditor or debtor
bank_code	Bank code of creditor or debtor
bank_name	Bank name of creditor or debtor
bank_icon	Icon of creditor or debtor bank
amount	Order amount
currency	Three-character currency code
purpose	Purpose text

<code>text_key</code>	DTA text key
<code>text_key_extension</code>	DTA text key extension
<code>scheduled_date</code>	Scheduled date. Recurring time intervals for standing orders are specified according ISO 8601.
<code>container</code>	If this payment object is a container for multiple payments, then this field is set and contains a list of payment objects.
<code>submission_timestamp</code>	Timestamp of submission to the bank server.
<code>creation_timestamp</code>	Internal creation timestamp on the figo Connect server.
<code>modification_timestamp</code>	Internal modification timestamp on the figo Connect server.
<code>transaction_id</code>	Transaction ID. This field is only set if the payment has been matched to a transaction.

Payment types

Payment type constants
<code>Tansfer</code>
<code>Scheduled transfer</code>
<code>Standing order</code>
<code>Direct debit</code>
<code>SEPA transfer</code>
<code>Scheduled SEPA transfer</code>
<code>SEPA standing order</code>
<code>SEPA direct debit</code>

Payment parameters object

Field	Description
<code>allowed_recipients</code>	List of account IDs. The payment recipient must be one of these accounts. No restriction applies if this field is omitted.
<code>max_purpose_length</code>	Maximum string length of purpose text.
<code>supported_text_keys</code>	List of supported DTA text keys.

min_scheduled_date	Earliest scheduled date.
max_scheduled_date	Latest scheduled date.
supported_file_formats	List of supported payment file formats for file upload.

Payment file formats

Payment file format constants
DTAUS0

TAN scheme object

Field	Description
tan_scheme_id	Internal figo Connect TAN scheme ID.
name	TAN scheme name.
medium_name	TAN medium name.

Challenge object

Field	Description
title	Challenge title
label	Response label
format	Challenge data format. Possible values are <code>Text</code> , <code>HTML</code> or <code>HHD</code> .
data	Challenge data

Synchronization status

Field	Description
code	Internal figo Connect status code
message	Human-readable error message. The figo Connect server tries to localize the message as requested by the <code>Accept-Language</code> HTTP header.
sync_timestamp	Timestamp of last synchronization.
success_timestamp	Timestamp of last successful synchronization.

Status codes

Code	Description
1	Success
-1	General error
-2	PIN error

Credential object

Field	Description
label	Label for text input field
masked	This indicates whether the this text input field is used for password entry and therefore should be masked.
optional	This flag indicates whether this text input field is allowed to contain the empty string.

Registered notification object

Field	Description
notification_id	Internal figo Connect notification ID from the notification registration response.
observe_key	Notification key
notify_uri	Notification messages will be sent to this URL.
state	The <code>state</code> parameter from the notification registration request.

Username & password policy

The figo username & password policy is enforced by the figo Connect server. Usernames and passwords should also be checked by the client to show appropriate error messages.

Usernames must be at least 3 characters long, must contain exactly one @-character and must neither start nor end with an @-character.

There are no requirements for passwords.

MEHR ÜBER UNS

Unternehmen
Jobs


SPRICH DARÜBER!



 



 


 

[Presse](#)

[Kontakt](#)

[Für Entwickler](#)

[figo Dienstleister](#)

[Impressum](#)

[Allgemeine Geschäftsbedingungen](#)

[Datenschutzerklärung](#)



F Empfehlen



Tweet



+1



F Empfehlen



Tweet



+1



FOLLOW US!

