1. General API Information

- **API Version:** v3
- **Production Base URL:**

https://api.infakt.pl/api/v3

- **Sandbox (Testing) Base URL:**

https://api.sandbox-infakt.pl/api/v3

- **Protocols & Encryption:**

All connections are made over HTTPS using TLS 1.2 or TLS 1.3. Older TLS versions are not supported, and the list of supported cipher suites may change over time.

---

2. Authentication and Headers

- **API Key:**

You need to generate your API key from your inFakt account settings.

The key must be sent via the HTTP header:

X-inFakt-ApiKey: YOUR_API_KEY

- **Required Headers:**
- Accept: application/json
- Content-Type: application/json (required for POST, PUT, PATCH, and DELETE methods)
- **Scopes:**

When creating the API key, you define its permissions (scopes) which determine which resources and operations the key can access. For example:

- api:invoices:read – read invoice, client, and product data (e.g. listing invoices, viewing details, generating PDFs).
- api:invoices:write – manage invoices, clients, and products (e.g. adding, editing, deleting invoices, and sending invoice emails).
- Other available scopes include api:accounting:read, api:accounting:write, api:sensitive:bank_accounts:write, api:costs:read, api:costs:write, and api:ksef:integration:write.

3. Main Endpoints and Operations

Invoices

- **List Invoices:**
- **Endpoint:** GET /invoices.json
- **Usage:**

Retrieves a list of invoices from your inFakt account.

You can filter the results (for example, q[status_eq]=sent or q[status_eq]=printed), paginate, and sort the results (e.g. order=invoice_date desc).

You may also limit the returned fields using the fields parameter (e.g. fields=number,invoice_date,status).

- **Invoice Details:**
- **Endpoint:** GET /invoices/{invoice_uuid}.json
- **Usage:**

Retrieves full details for a specific invoice.

The URL requires the unique invoice identifier (UUID).

Clients

- **List Clients:**
- **Endpoint:** GET /clients.json
- **Usage:**

Retrieves a list of clients with similar options for filtering, pagination, and field selection.

- **Client Details:**
- **Endpoint:** GET /clients/{client_id}.json
- **Usage:**

Retrieves detailed information for a specific client.

Additional Operations

- **Filtering:**

The q parameter, along with appropriate modifiers, allows you to narrow down the results:

- **_eq:** Exact match (e.g. q[number_eq]=1/09/2013).
- **_cont:** Checks if a field contains the specified value (for text fields).
- For dates, modifiers such as _lt, _gt, _lteq, and _gteq are available.
- **Pagination and Sorting:**
- **Pagination:** Use offset (starting point) and limit (maximum number of records, up to 100).
- **Sorting:** Use the order parameter to specify the field and sort direction (e.g. order=invoice_date desc).

---

4. Request Structure and Data Format

- **Request Format:**

Requests—especially those that modify data (POST, PUT, PATCH, DELETE)—should include a request body in JSON format.

- **Encoding:**

All string data must be UTF-8 encoded.

- **Example Request (cURL):**

```
curl -H "X-inFakt-ApiKey: YOUR_API_KEY" \
    -H "Content-Type: application/json" \
    -X PUT \
    -d '{"client":{"company_name":"Infakt Biuro Rachunkowe Sp. z o.o."}}' \
    https://api.infakt.pl/api/v3/clients/1.json
```

---

5. Response Structure

- **Main Components of the Response:**

The API response is a JSON object containing:

- **metainfo:** Metadata such as the number of records (count, total_count) and links to the next or previous pages (next, previous).
- **entities:** An array of resource objects (e.g. invoices, clients).
- **Example Response for Invoices:**

```
{
  "metainfo": {
    "count": 3,
    "total_count": 15,
    "next": "https://api.infakt.pl/api/v3/invoices.json?offset=10&limit=5",
    "previous": "https://api.infakt.pl/api/v3/invoices.json?offset=0&limit=5"
  },
  "entities": [
    {
      "id": 5,
      "number": "4/05/2022",
      "invoice_date": "2022-05-04",
      "status": "sent",
      "gross_price": 123450
      // other fields
    }
  ]
}
```

---

6. Key Fields of the Invoice Object

When integrating with the API, pay close attention to the following invoice fields:

- **Identification and Dates:**
- id: Unique invoice identifier.
- number: Invoice number.
- invoice_date: Invoice issue date (YYYY-MM-DD).
- payment_date: Payment due date.
- paid_date: Date when the invoice was paid.
- **Amounts (provided in grosze):**
- gross_price: Total gross amount.
- net_price: Net amount.
- tax_price: VAT amount.
- paid_price: Amount already paid.
- left_to_pay: Remaining amount to be paid.
- **Status and Payment Method:**

- status: Invoice status (e.g. draft, sent, printed, paid).
- payment_method: Payment method (e.g. transfer, cash, card).
- kind: Type of invoice (e.g. vat, proforma).
- **Client Information:**
- client_id: Client identifier.
- client_company_name: Client company name (if no client_id is provided).
- client_nip: Client’s tax identification number.
- Optionally: personal data (first name, last name) and client address.
- **Line Items (services):**
- The services array contains details for each line item on the invoice, such as:
- name: Name of the product or service.
- tax_symbol: VAT rate.
- quantity, unit_net_price, net_price, gross_price, tax_price, and other related details.

---

7. API Limits and Constraints

Request Rate Limits

inFakt enforces rate limits to protect its infrastructure and prevent abuse. Depending on your plan (currently for non-accountant plans), typical limits are as follows:

- **GET Requests (Data Reading):**
- **Per Minute:** Up to 50 requests per client account.
- **Per Hour:** Up to 200 requests per client account.
- **Per Day:** Up to 1000 requests per client account.
- **Other Methods (POST, PUT, PATCH, DELETE):**
- **Per Minute:** Up to 10 requests per IP address.
- **Per Hour:** Up to 100 requests per IP address.
- **Per Day:** Up to 500 requests per IP address.

Additional Limits

- **Email Sending:**
- **Free Plans:** Up to 20 emails per day.
- **Paid Plans:** Up to 3000 emails per day.
- **Number of Records in a Response:**

The maximum number of records returned when listing resources is 100 (controlled by the limit parameter).

Error Handling for Rate Limits

- **HTTP 429 (Too Many Requests):**

If you exceed the rate limits, the API will respond with a 429 status code. Implement a retry mechanism with an appropriate back-off interval in such cases.

- **HTTP 5xx Errors:**

Server errors may also indicate temporary issues. A retry mechanism is advisable for these errors as well.

Notes on Rate Limits

- Limits are enforced at both the client account level and the individual IP address level.
- The sandbox (testing) environment also has limits, which may be even more restrictive—this is important to keep in mind during testing.
- It is recommended to monitor API responses and log any errors related to exceeding rate limits so that your request strategy can be dynamically adjusted.

---

8. Additional Information for Developers

- **Filtering and Modifiers:**

Filtering is performed using parameters like q[PARAMETER_eq] where the _eq modifier enforces an exact match. For partial matches, use _cont. For dates, modifiers like _lt, _gt, _lteq, and _gteq are available.

- **Field Selection:**

To improve performance, you can specify which fields should be returned using the fields parameter.

- **Sorting and Pagination:**

Ensure your integration handles pagination for large data sets. Sorting can help obtain ordered results (e.g. by invoice date).

- **Security:**

Make sure your HTTP client supports TLS 1.2/1.3 and current cipher suites. All requests are transmitted over HTTPS, which is critical for protecting sensitive data.

- **Back-End Integration:**

The existing back‑end already includes a dedicated API client (in api_client.py) that implements the logic for retrieving invoice lists, filtering by status (e.g. sent, printed), pagination, and HTTP error handling.