fluorochem.

Fluorochem customer API guide

Product information stock pricing and molecule search

Use this guide to connect your application to the Fluorochem API. Your API key identifies your account and selects your agreed currency and price level. Fluorochem supplies the key separately. Public IP restrictions apply only when configured for your account.

```
Base URL: https://api.fluorochem.co.uk
API header: x-api-key: YOUR_API_KEY
```

Make your first request

Send an HTTPS GET request for a product code. Replace YOUR_API_KEY with your supplied key. The example below uses curl with shell line continuations.

```
curl "https://api.fluorochem.co.uk/product/F003409" \
  -H "x-api-key: YOUR_API_KEY"
```

For Postman or another HTTP client, choose GET, enter the URL and add x-api-key under Headers. Do not put the key, customer ID or price level in the URL.

Available routes

Method	Path	Returns
GET	/products	Ten complete products
GET	/product/{id}	One product including variants
GET	/product/{id}/variants	Pack sizes stock and prices
POST	/molecules/search	Products matching an InChIKey
GET	/products/changes	Changed codes, SMILES and stock locations

Route availability varies by account. Replace {id} with the exact Fluorochem product code, for example F003409. Optional variant filters are on page 3. Trailing slashes are accepted. These routes return JSON. Bulk file downloads and examples are covered later in this guide.

Access requirements

Requests need your API key and permission to use the route. If IP restrictions are configured, requests must also come from an allowed address. Contact Fluorochem to enable a route or amend your addresses. Keep the key in server-side configuration; avoid exposing it in website JavaScript, shared URLs or logs.

Product information and pricing

Retrieve a product

```
GET /product/F003409
x-api-key: YOUR_API_KEY
```

A successful response is a product object with id, type, weblink, meta, properties, categories, hazards, transport, regulatory, sds_links and variants. Product metadata includes the description, CAS number, molecular formula and structural identifiers where available. Missing fields may be null.

The abbreviated example below illustrates the response structure. Values are illustrative and do not represent a current quotation. Some product fields and pack sizes have been omitted.

```
{
  "id": "F003409",
  "type": "product",
  "weblink": "https://fluorochem.co.uk/product/F003409",
  "meta": {
    "fc_code": "F003409",
    "description": "1,1,1,3,3,3-Hexafluoro-2-propanol",
    "cas": "920-66-1"
  },
  "variants": [{
    "sku": "F003409-100G",
    "size": "100",
    "size_unit": "G",
    "stock": {
      "uk": "true", "germany": "false",
      "china": "true", "any": "true"
    },
    "price": 112
  }]
}
```

Interpret prices and stock

price is a JSON number or null. A null value means no price is available for your assigned level on that pack size; it is not a zero price. Currency and price level are assigned to your account and are not repeated in each variant. Query parameters cannot change them.

Stock values are the strings "true" and "false", not Boolean values or quantities. stock.any is "true" when at least one listed warehouse has stock. size and size_unit are strings.

Retrieve a product sample

```
GET /products
```

Returns an array of ten complete product objects. This route is a limited listing and provides no pagination or full-catalogue export. Use a known product code for individual retrieval.

Pack sizes stock and price filters

`GET /product/F003409/variants`

Returns a JSON array of variants with sku, size, size_unit, stock and price, using the same structure as the variants in a product response. Add any combination of the following query parameters.

Parameter	Example	Effect
instock	instock=1	Keep variants where at least one warehouse has stock.
priceonly	priceonly=1	Keep variants with a numeric price. A zero price is retained.
country	country=uk	Add leadtime text based on the preferred warehouse with stock. Does not change prices or remove stock locations.

Combined example

```
curl -G "https://api.fluorochem.co.uk/product/F003409/variants" \
  -d "instock=1" -d "priceonly=1" -d "country=uk" \
  -H "x-api-key: YOUR_API_KEY"
```

Append ?instock=1&priceonly=1&country=uk to the variants URL to combine all three parameters. instock and priceonly accept 1, true, yes or on, ignoring case. If omitted, the corresponding filter is not applied. An empty result after filtering is returned as an empty array [].

Country values

Supported values are uk or unitedkingdom; germany; france; spain; italy; netherlands; belgium; usa or unitedstates; and china. Values are case-insensitive. Use the names without spaces as listed here. Unrecognised values add no leadtime field.

For UK requests the order of preference is UK, Germany, then China. For supported European countries it is Germany, UK, then China. For USA it is UK, Germany, then China. For China it is China, UK, then Germany.

When a preferred warehouse has stock, leadtime is added as text: Next Day, 7-10 Days or 10-14 Days from that warehouse, according to its position in the preference order. If no warehouse has stock, leadtime is omitted.

Example filtered variant

```
[{
  "sku": "F003409-100G", "size": "100", "size_unit": "G",
  "stock": {"uk":"true", "germany":"false",
            "china":"true", "any":"true"},
  "price": 112,
  "leadtime": "Next Day from the United Kingdom"
}]
```

Example values are illustrative.

Molecule search

```
POST /molecules/search
x-api-key: YOUR_API_KEY
Content-Type: application/json
```

Search for an exact InChIKey. Supply the full key in the JSON property inchi_key, using the standard uppercase format. This endpoint does not accept a molecule image, SMILES, a CAS number, a partial identifier or a substructure query.

Request example

```
curl "https://api.fluorochem.co.uk/molecules/search" \
  -X POST \
  -H "x-api-key: YOUR_API_KEY" \
  -H "Content-Type: application/json" \
  -d '{"inchi_key":"BYEAHWXPCBROCE-UHFFFAOYSA-N"}'
```

Response

A successful response is an array of matching products. Each contains id, weblink, meta and properties. This response omits stock, variants and prices. Use each returned id with the product or variants route to retrieve those details.

Abbreviated example with additional metadata and properties omitted:

```
[{
  "id": "F003409",
  "weblink": "https://fluorochem.co.uk/product/F003409",
  "meta": {
    "fc_code": "F003409",
    "description": "1,1,1,3,3,3-Hexafluoro-2-propanol",
    "inchi_key": "BYEAHWXPCBROCE-UHFFFAOYSA-N"
  },
  "properties": {"molecular_weight": "168.038"}
}]
```

Retrieve pricing for a match

```
GET /product/F003409/variants
```

When nothing matches

The service responds with HTTP 404 and the following JSON. A missing or malformed inchi_key instead produces HTTP 400.

```
{"message":"No matches found"}
```

Changed and added products

```
GET /products/changes?since=2026-09-01&limit=500
x-api-key: YOUR_API_KEY
```

Available to accounts with Product changes enabled. This route returns product codes, canonical SMILES and UK, Germany and China stock availability for products whose catalogue creation or modification date is after your chosen date. No quantities or prices are included.

Choose a starting date

Parameter	Use
since	Required on the first request. Use YYYY-MM-DD for midnight UTC, or an ISO 8601 timestamp with an explicit Z or numeric offset. The lower bound is exclusive: only dates strictly after since qualify.
limit	Optional maximum product codes per page. Default 500; minimum 1; maximum 1000. Set it on the first request only.
cursor	Use the token from a previous response to continue that request. Supply cursor alone, with no since, limit or other query parameters.

For example, 2026-09-01 means 2026-09-01T00:00:00.000Z. A timestamp without a time-zone suffix is invalid. URL-encode query values, particularly a positive offset such as +01:00. Using Z avoids that ambiguity.

What counts as a change

A current published product qualifies when its creation date or modification date in the catalogue is after since and at or before until. Source dates are interpreted as UTC, including values stored without an offset. These are the dates recorded in the catalogue; they are not an independently verified record of when every underlying change happened.

One consistent catalogue snapshot

The route reads SMILES and product-level stock flags from Elasticsearch in one bulk lookup per page. These flags summarize availability across variants as last indexed. It does not make an Azure request for each product. The date window, catalogue details and stock flags remain in the same fixed snapshot across all pages.

Changes response

The existing productCodes list is retained. The products array contains matching records in the same order, so you can read the requested structural and availability data directly.

```
{
  "productCodes": ["F003409"],
  "products": [{
    "fc_code": "F003409",
    "smiles": "OC(C(F)(F)F)C(F)(F)F",
    "stock": {"uk": true, "germany": true, "china": false}
  }],
  "since": "2026-09-01T00:00:00.000Z",
  "until": "2026-09-15T08:00:00.000Z",
  "hasMore": true,
  "nextCursor": "OPAQUE_CURSOR"
}
```

Values are illustrative. since is normalised to UTC. until is fixed at the first request time and is inclusive. Product codes are sorted and deduplicated.

Read the stock location vector

Value	Meaning for that location
true	At least one variant is available in that location.
false	No variant is available in that location.
null	The catalogue stock flag is missing; availability is unknown. Do not treat this as confirmed out of stock.

These are JSON booleans, not the strings "true" and "false" used by the existing variants route. The stock object always includes uk, germany and china. It reports availability as last indexed, not live warehouse quantities.

Missing structural or stock data

A missing canonical SMILES returns smiles: null. Newly added products may have no stock flags yet; in that case all three locations are null. The product remains in the change feed so that you can track the addition and retry missing details later.

Continue a changes request

```
GET /products/changes?cursor=OPAQUE_CURSOR
x-api-key: YOUR_API_KEY
```

Replace OPAQUE_CURSOR with the exact nextCursor value returned by the previous page. Keep the same API key and URL-encode the token. Do not add since or limit to continuation requests.

Process every page before saving your progress

Step	Action
1	Request /products/changes with since and an optional limit.
2	Store the returned products locally and continue promptly. Collect every page before making any additional product or variant requests.
3	While hasMore is true, request the next page using cursor alone. A page can be empty, including the final page; rely on hasMore and nextCursor rather than the number of codes.
4	When hasMore is false and nextCursor is null, update or insert the collected products by fc_code. Fetch any further details through enabled routes as needed, then save until as the completed checkpoint.

Use an overlap for routine polling

For your next poll, use the previous completed until minus 24 hours as since, then deduplicate or update existing products by code. This overlap helps with delayed indexing. If a poll is interrupted, resume promptly with its cursor, or restart from the original since and process duplicate codes safely.

An overlap cannot guarantee detection of late or backdated imports. Keep a periodic full-catalogue reconciliation if you need to recover those records or detect removals. The /products sample route is not a full export.

Cursor lifetime and request limits

A changes snapshot expires after 20 minutes without use or 24 hours in total. A service restart may also invalidate it. HTTP 410 means you must restart from the original since. Finish one polling run before starting another: a customer can have at most two unfinished changes requests. Further starts return HTTP 429.

Each page and follow-up product request uses the same request allowance. Collect the code pages promptly; pace detail requests afterwards. Respect Retry-After on HTTP 429. For larger result sets, use limit=1000 to reduce page requests. HTTP 503 may mean direct Elasticsearch access is not configured; contact Fluorochem if it persists.

Scope of this route

This is a catalogue date query, not a full database change log. It does not report deletions, products that have been unpublished, or independent stock and price changes in Azure. Refresh stock and pricing separately if your integration needs those updates even when the catalogue record has not changed.

Errors limits and data freshness

Check the HTTP status before using response data. Most errors use {"error":"description"}; a molecule search without matches uses {"message":"No matches found"}.

Status	Meaning	Recommended action
400	Invalid input	Check the product code, JSON, InChIKey or date/cursor parameters.
403	Access denied	Check the key, account, allowed IP and route permission. Contact Fluorochem if a route is unavailable.
404	Not found	Check the route and product code, or handle no molecule matches.
405 / 415	Wrong method or media type	Use the documented method and application/json for POST.
410	Changes cursor expired	Restart with the original since; deduplicate or update by code.
429	Request or snapshot limit	Follow Retry-After. Finish existing changes runs before starting another.
500	Internal error	Retry with backoff; report persistent failures.
502 / 503 / 504	Source failure unavailable data or busy service	Retry with backoff. Persistent missing data/configuration needs review.

Request limits

Live routes share 100 requests per rolling 15 minutes and 10 per 10 seconds, with two concurrent requests per customer. Bulk routes have separate limits. Respect Retry-After on 429/503; back off rather than retrying rapidly. Correct invalid requests before retrying.

Data freshness

Underlying stock and pricing are updated overnight. Product and pricing documents are cached for up to 24 hours, with daily expiry at 06:15 UK time. The first request after expiry fetches fresh source data. Availability is not a live stock reservation. The changes route uses its own fixed catalogue snapshot, as described on pages 5-6.

Single-product and variant responses include X-Product-Fetched-At and X-Pricing-Fetched-At. X-Pricing-Last-Modified is present when the pricing source supplies it. X-Product-Cache and X-Pricing-Cache indicate HIT or MISS. Use these headers when investigating freshness.

Request support

If a request fails repeatedly, send Fluorochem the route, UTC request time, HTTP status and X-Request-Id response header, plus the public IP your system used. Remove API keys and other credentials from any example you share.

This guide covers the customer product, molecule and bulk API. Administrative and ChemScene integration routes use separate access arrangements and are outside this guide.

Bulk catalogue or export?

A catalogue is the prepared product snapshot. An export is the background job that prepares that snapshot. The two names describe different steps, not different product datasets.

What you need	Request
Download the latest prepared file	GET /bulk/catalogue/download
Preview products as normal JSON	GET /bulk/catalogue?limit=1
Check the file date and record count	GET /bulk/catalogue/manifest
Request fresh preparation	POST /bulk/exports

Start with a small JSON preview

```
GET /bulk/catalogue?limit=1
x-api-key: YOUR_API_KEY
```

This returns a normal JSON object containing a products array. It reads the same prepared snapshot used by the download. It does not fetch live product data or start a new export.

What Postman is showing

In Postman, check the status and Content-Type first. For a 200 application/gzip response, use Send and Download (where available) or save the response to a file, keep the .ndjson.gz extension, then extract it. Binary-looking text in the response viewer is expected for a compressed file. To see ordinary JSON without downloading the file, use GET /bulk/catalogue?limit=1. An error response is JSON, not a gzip file.

Postman controls vary by version. For large files, prefer Send and Download where available. Downloadable request and saved-response examples are available from the API dashboard, under Routes and Bulk exports. The collection contains no API key; enter yours locally. Do not run the whole collection automatically because POST /bulk/exports starts work.

All examples are illustrative and abbreviated, not current stock or quotations. Bulk permissions are granted separately. Preparation follows the schedule configured for your account, which may be weekly or monthly. Downloading again does not refresh the file.

The downloaded file

```
GET /bulk/catalogue/download
x-api-key: YOUR_API_KEY
```

A successful download is a gzip file, not a JSON array, CSV or ZIP archive. The HTTP headers identify the file and the prepared version. These example IDs and values are illustrative.

```
HTTP 200 OK
Content-Type: application/gzip
Content-Disposition: attachment; filename="fluorochem-catalogue-11111111-1111-4111-
8111-111111111111.ndjson.gz"
X-Catalogue-Version: 11111111-1111-4111-8111-111111111111
```

After extracting the gzip file

The result is an .ndjson text file. NDJSON means one JSON object per physical line, with no enclosing square brackets and no commas between records. Each line is a product in the same format as /product/{id}, including its variants and your assigned pricing. The abbreviated lines below wrap on the printed page; actual records each occupy one line.

```
{"id":"F003409","variants":[{"sku":"F003409-100G","size":"100","size_unit":"g",
"stock":{"uk":"true","germany":"false","china":"true","any":"true"},"price":112}]}
{"id":"F000002","variants":[]}
```

These examples omit product metadata and other fields. An empty variants array means no variant data was available, not proof that there is no stock. File prices are already selected for your account; check currency and priceLevel in the manifest. Check Content-Type before extracting: a failed request returns a JSON error instead of a gzip file.

Which download URL?

GET /bulk/catalogue/download chooses the latest compatible completed file. GET /bulk/exports/{jobId}/download retrieves the file from one specific ready job. Both use the same format and require your API key. Downloading either file does not create a new export.

JSON pages and file information

A page of products

```
GET /bulk/catalogue?limit=1
```

```
{
  "version": "11111111-1111-4111-8111-111111111111",
  "scope": {
    "kind": "full"
  },
  "catalogueComplete": true,
  "total": 533000,
  "products": [
    {
      "id": "F003409",
      "variants": []
    }
  ],
  "hasMore": true,
  "next": "/bulk/catalogue?version=11111111-1111-4111-8111-
111111111111&after=F003409&limit=1"
}
```

Abbreviated example. products contains this page; total counts the whole snapshot. Follow next exactly, with the same API key, until hasMore is false and next is null. Default page size is 100; maximum 500. Prefer the compressed file for a complete catalogue.

The manifest describes the file

```
GET /bulk/catalogue/manifest
```

```
{
  "version": "11111111-1111-4111-8111-111111111111",
  "recordCount": 533000,
  "catalogueComplete": true,
  "currency": "GBP",
  "generatedAt": "2026-10-04T10:00:00.000Z",
  "file": {
    "bytes": 12345678,
    "sha256": "ILLUSTRATIVE_CHECKSUM"
  }
}
```

This abbreviated JSON contains file information, not products. The full manifest also identifies the scope, price level and generation period. file.sha256 is a 64-character SHA-256 checksum of the compressed file. A weekly or monthly schedule means a download may be older than today; downloading again does not refresh it.

Requesting a fresh export

1. Request preparation

```
POST /bulk/exports
x-api-key: YOUR_API_KEY
Content-Type: application/json

{}
```

This starts background work, or returns the already-pending job. HTTP 202 means accepted, not finished. The response is job information rather than product data. One new generation is allowed per 24 hours.


```
{
  "id": "11111111-1111-4111-8111-111111111111",
  "status": "queued",
  "created": "2026-10-04T08:00:00.000Z",
  "nextGenerationAt": "2026-10-05T08:00:00.000Z",
  "progress": 0,
  "totalProducts": null,
  "statusUrl": "/bulk/exports/11111111-1111-4111-8111-111111111111",
  "downloadUrl": null
}
```


2. Check the returned status URL

Poll statusUrl about every 30 seconds, respecting Retry-After. The job moves from queued to running, then ready or failed. A 200 status response does not itself mean the job succeeded.


```
{
  "id": "11111111-1111-4111-8111-111111111111",
  "status": "ready",
  "progress": 533000,
  "totalProducts": 533000,
  "phase": "complete",
  "finished": "2026-10-04T10:00:00.000Z",
  "statusUrl": "/bulk/exports/11111111-1111-4111-8111-111111111111",
  "downloadUrl": "/bulk/exports/11111111-1111-4111-8111-111111111111/download"
}
```


Abbreviated ready response. progress is a product count, not a percentage. totalProducts may be null while counting. If status is failed, inspect error; do not repeatedly start new jobs.

3. Download when ready

Send an authenticated GET request to downloadUrl. It returns the same compressed NDJSON format as the prepared catalogue download. Keep the imported version for future reconciliation.

Changes between prepared files

```
GET /bulk/changes?from=22222222-2222-4222-8222-222222222222
x-api-key: YOUR_API_KEY
```

Replace from with the version you last imported successfully. This route compares prepared files and includes changed product/stock/pricing values and removals from those snapshots. It is different from /products/changes, which uses catalogue timestamps and excludes independent pricing changes and deletions.

When a newer version exists

HTTP 200 with Content-Type application/gzip returns a compressed NDJSON delta. The extracted lines below are abbreviated. Upsert means insert or replace the entire product by id; remove means delete that id from your local catalogue.

```
{"type":"upsert","product":{"id":"F003409","variants":[{"sku":"F003409-100G","size":
"100","size_unit":"g","stock":{"uk":"true","germany":"false","china":"true","any":
"true"},"price":112}]}}
{"type":"remove","id":"F000002"}
```

Apply every line successfully, then store the target version from the X-Catalogue-Version response header. Do not advance your saved version after an incomplete download or import.

When you already have the current version

The response is normal JSON, not a gzip file:

```
{
  "from": "11111111-1111-4111-8111-111111111111",
  "to": "11111111-1111-4111-8111-111111111111",
  "changes": [],
  "complete": true,
  "scope": {
    "kind": "full"
  }
}
```

When your version is too old

Changes cover skipped generations within a rolling 90-day history window, measured from the version you last successfully imported. Updates are idempotent final-state records. A checkpoint outside the window or with unavailable history returns HTTP 409 with code BULK_DOWNLOAD_REQUIRED; import the latest full file before saving its version. Only one completed full download is retained per customer.

Postman reference: learning.postman.com/docs/use/send-requests/response-data/responses (saving responses); support.postman.com/hc/en-us/articles/360033875073 (large response downloads).