The UI and API examples on this page are representative mockups — hostnames, keys, names and figures shown are illustrative, not live screenshots.

IQX AI Gateway

The **IQX AI Gateway** is the single front door for every AI feature in IQX. Instead of each part of the product talking to an AI provider directly, everything — the **IQXAssist** chat your consultants use, the wiki help assistant, CV reformatting, document and photo reading, audio transcription — calls *one* internal service. That service picks a model, routes the request, retries elsewhere if something is down, meters the spend, logs the usage, and hands the answer back.

In short, it is an AI *router* and *control plane*: the clever bits (which model, failover, cost caps, encryption) live in one place, so every IQX feature behaves the same way and we can add or swap models without touching the apps.

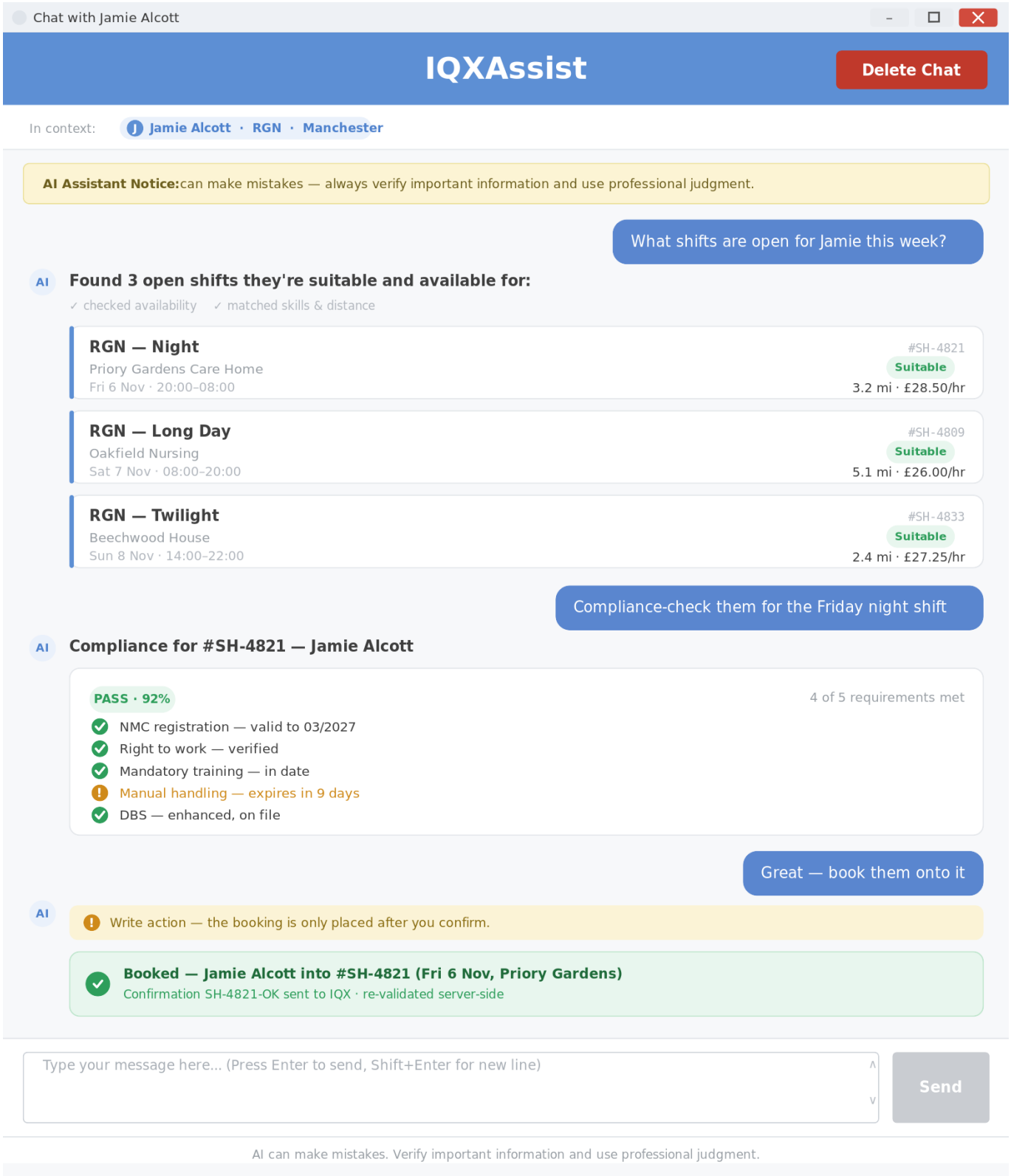
At a glance

- **EU-hosted, open-weight models** — nothing is sent to the big US AI vendors.
- **One API key per tenant/client**, with per-key capabilities and a monthly **spend cap**.
- **Automatic failover** — if a model is unavailable, the gateway transparently tries another.
- **OpenAI-compatible** — point any OpenAI SDK at `/v1` and it just works.
- Chat history stored in **MongoDB in the UK**, **encrypted per message**, auto-deleted **30 days** after last use.

—-

IQXAssist — the in-app chat

IQXAssist is the assistant your consultants talk to inside IQX. Unlike a standard “question and answer” bot, it is **context-aware**: it knows which candidate or contact record you are viewing and can pull live data specific to them.



Two modes

The chat works in two modes depending on what you ask:

- **Conversational** — answers general questions, drafts text, or summarises information already in the chat.

- **Agent (live data)** — when you ask for specific records or actions (e.g. “show me open shifts”), it uses secure tools to fetch real-time data from IQX before answering.

Every session is tied to the record you are viewing. Navigate to a different candidate and a new chat begins, so data can't cross over between records.

What it can pull in (read)

IQXAssist has strictly defined **tools** that let it read specific parts of a record. These are read-only and never change anything.

Data type	What it can see	Best way to ask
Timesheets / records	Pay status, hours, rates, notes, placements	“Show timesheets from 1 Jan to 1 Feb”
Shifts	Open shifts the candidate is suitable and available for	“What shifts are open for them next week?”
Availability	Holidays and unavailability blocks	“Are they available in March?”
Compliance	Licence details, training, medicals, right to work	“Compliance-check them for the Friday night shift”



The “date range” rule
To avoid overloading the system, the AI looks for a specific timeframe. If you ask for records without a date it may default to the last 30 days or ask you to clarify. Try to specify: “last month”, “next week”, or “since Jan 2024”.

Taking actions (write)

IQXAssist can also do things on your say-so — propose a **shift booking** or record an **availability change**.



Write actions are **only ever carried out after you explicitly confirm them**, and every action is re-validated server-side (availability, compliance, booking rules, still-open) before anything is saved. If a rule fails, the action is refused and the reason is shown.

Drafting & formatting

One of the handiest features is drafting communications:

- **Emails** — ask it to “write an email…” and it produces a professional, contact-event-ready email

(subject and body).
- **Text / SMS** — ask for a “text message” and it keeps the wording short and casual, suitable for mobile.

Smart touches

- **Streaming responses** — answers appear in real time as the AI works, rather than waiting for the whole reply.
- **Auto-titling** — each session is given a short title from the first few messages, so it's easy to find later.

—-

The API — endpoint by endpoint

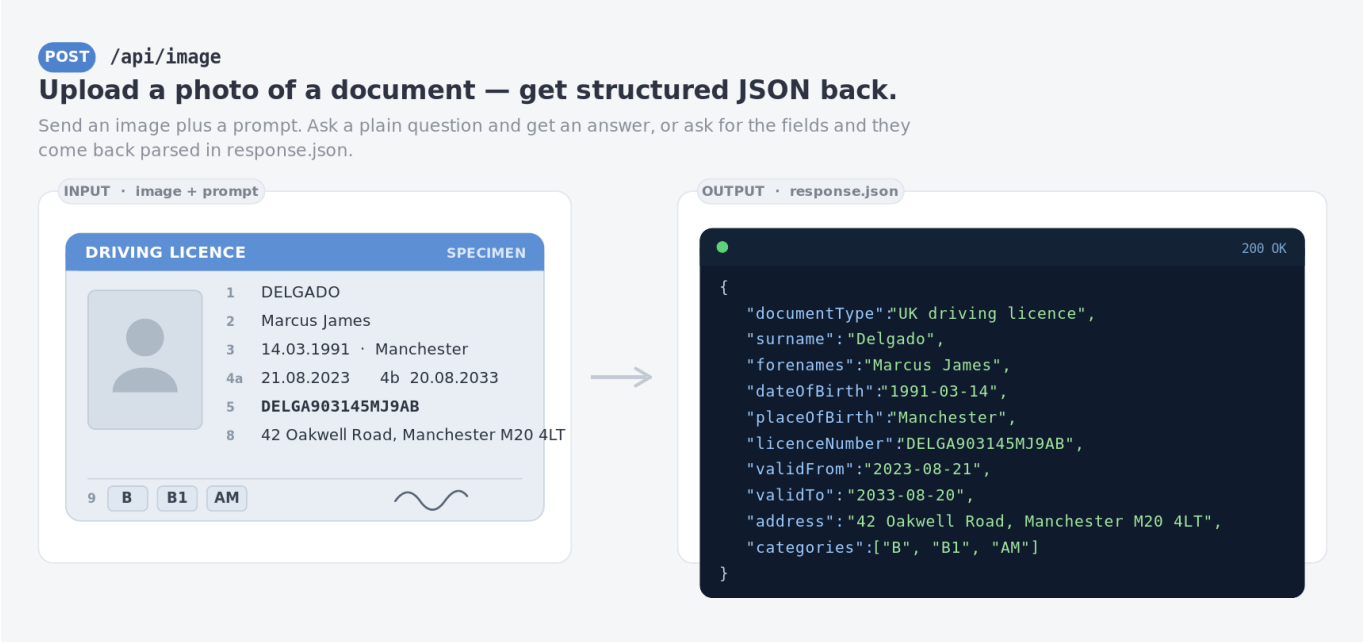
Every capability is also a single HTTP call, so any IQX service or script can use it. Authenticate with a Bearer key — the optional user id after the dot attributes usage to a person:

```
Authorization: Bearer YOUR_KEY.USER_ID
```

Here is what each endpoint takes in and gives back.

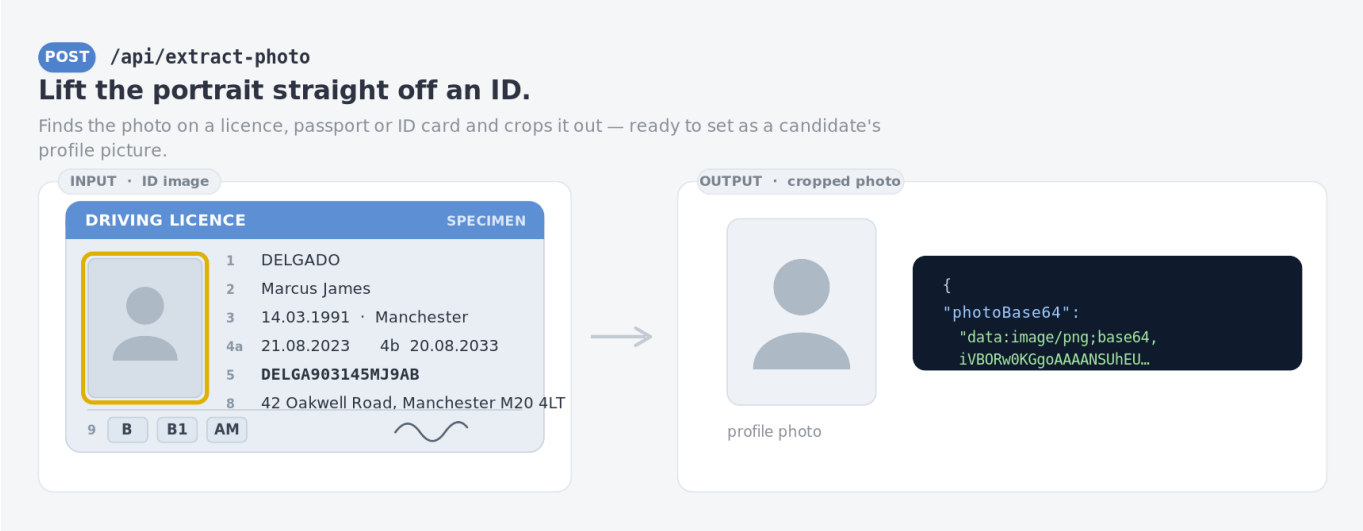
Read a document — get JSON out

Send an image plus a prompt. Ask a plain question and get an answer, or ask for the fields and they come back parsed in `response.json`. Driving licences, passports, DBS certificates, timesheets, whiteboards — all fair game.



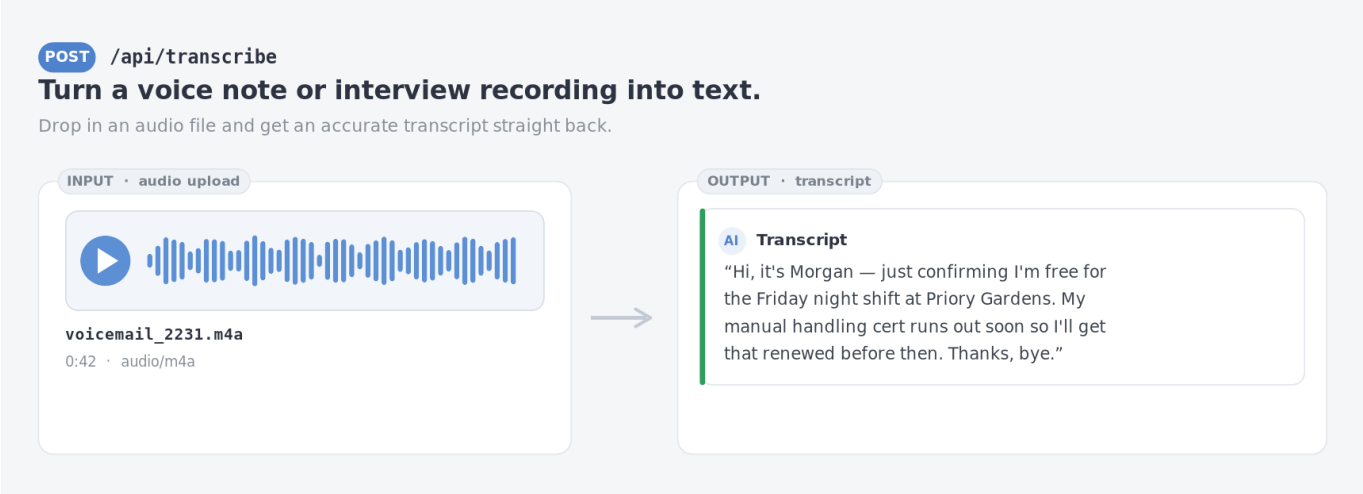
Lift the photo off an ID

Finds the portrait on a licence, passport or ID card and crops it out — ready to save straight to the candidate's record as their profile picture.



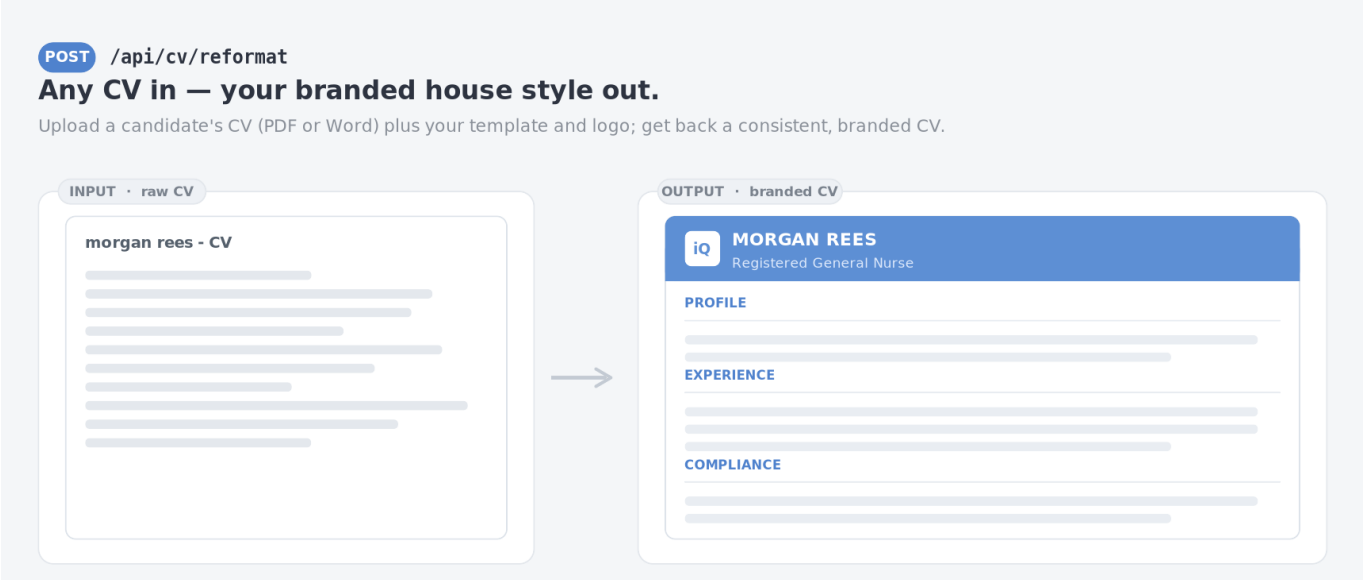
Transcribe audio

Drop in a voice note or interview recording and get an accurate transcript straight back.



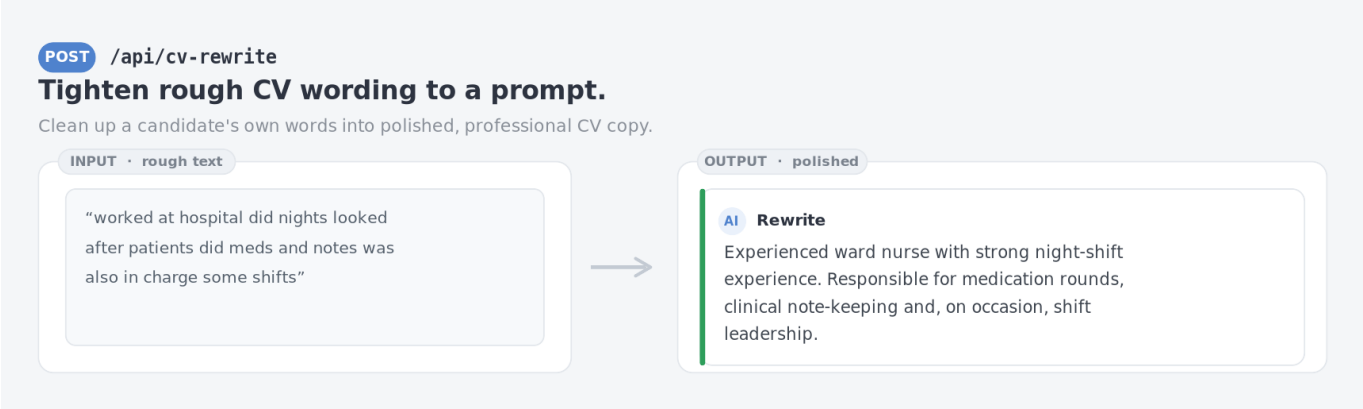
Reformat a CV to house style

Upload a candidate's CV (PDF or Word) plus your template and logo; get back a consistent, branded CV. Personal contact details are **automatically redacted** in the process.



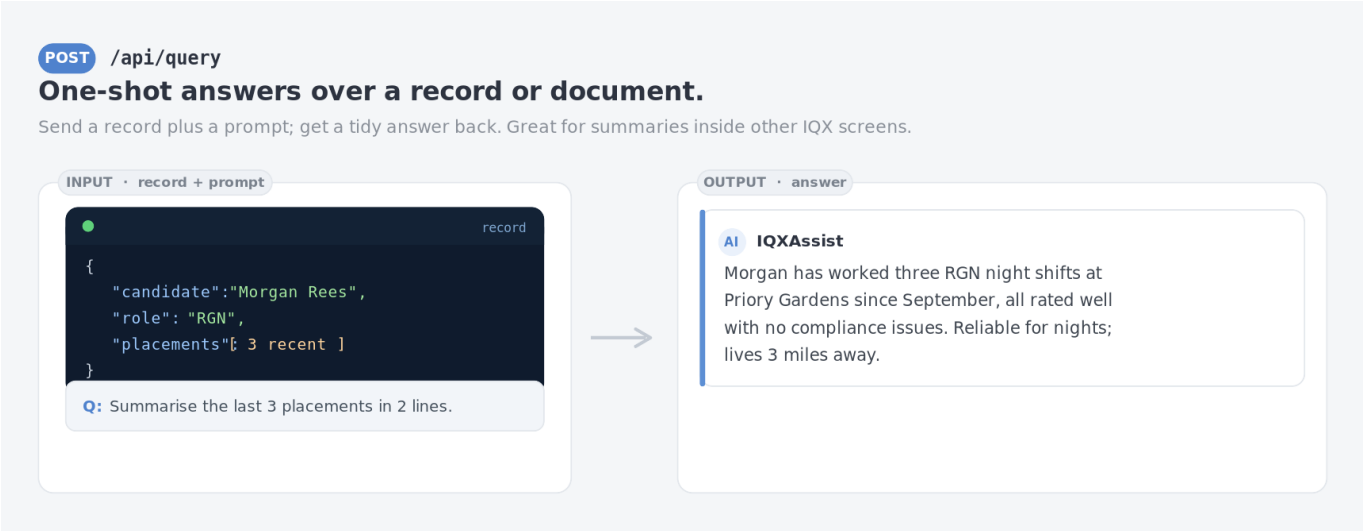
Tighten rough CV wording

Clean up a candidate's own words into polished, professional CV copy.



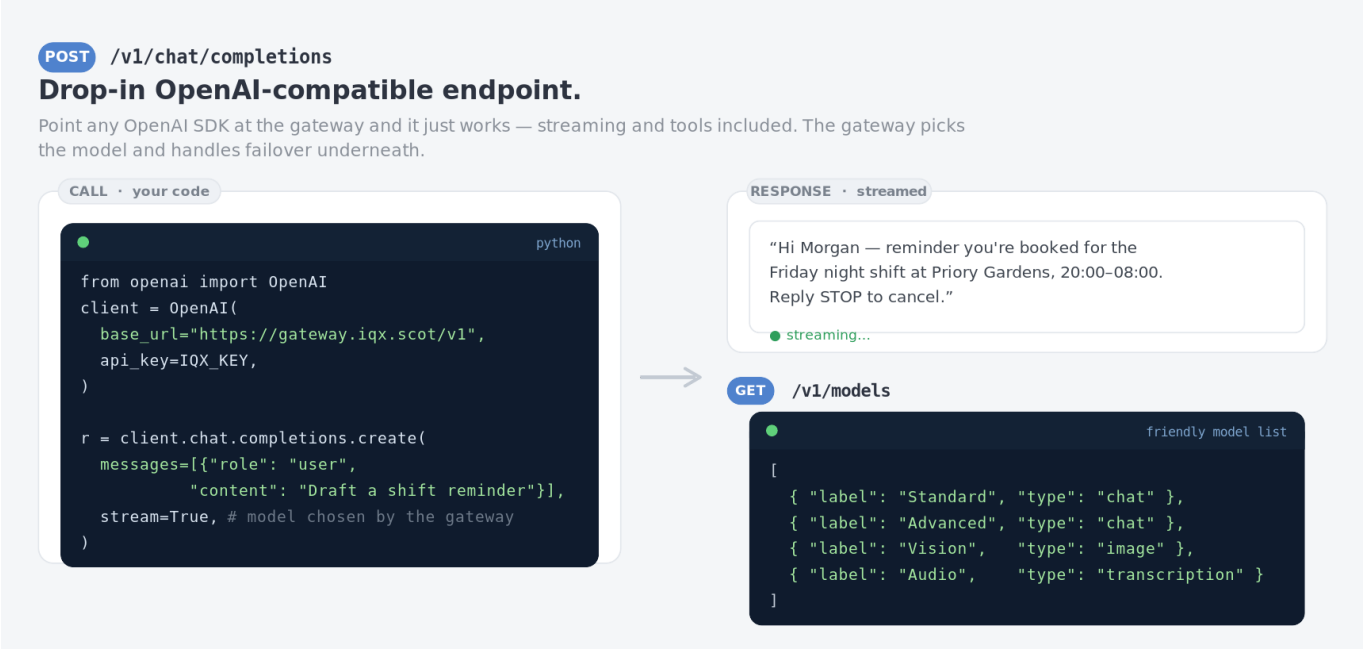
One-shot answers over a record

Send a record (or any document) plus a prompt and get a tidy answer back — great for summaries embedded in other IQX screens.



Drop-in OpenAI-compatible endpoint

Point any OpenAI SDK at `/v1` and it just works — streaming and tools included. The gateway picks the model and handles failover underneath. GET `/v1/models` returns a friendly model list for a picker.



—-

Security & data



Private by design

- Chat content is **individually encrypted at rest** — even database administrators can't read the raw message history.
- Sessions are **context-isolated** per record, so data never crosses between candidates.
- The models **do not train** on your data.
- CV contact details (phone, email, address) are **redacted** automatically.



Where it lives

- Chat history and usage logs are stored in **MongoDB in the UK**.
- Sessions and their messages are **auto-deleted 30 days** after the last interaction; uploaded files are swept within an hour.
- Model inference runs on **EU-hosted** providers and is **transient** — prompts are processed, not stored.
- Usage logs keep **metadata only** (no message content).

—-

Endpoint reference

Endpoint	Method	Purpose
`/api/query`	POST	One-shot text: prompt (+ optional context), returns the completion.
`/api/cv-rewrite`	POST	Rewrite / tidy CV text to a prompt.
`/api/cv/reformat`	POST	Reformat an uploaded CV (PDF/DOCX) into an HTML template, optional logo.
`/api/image`	POST	Vision — ask a question about an image, or extract its fields as JSON.
`/api/extract-photo`	POST	Vision — crop the portrait photo out of an ID image.
`/api/transcribe`	POST	Audio — transcribe an uploaded audio file.
`/v1/chat/completions`	POST	OpenAI-compatible chat. Streaming and tools supported.
`/v1/models`	GET	List the available models (friendly labels) for a model picker.
`/api/model-config`	GET	The preferred default model for the key (so UIs don't hard-code it).
`/api/capabilities`	GET	The calling key's effective tool capabilities.
`/api/chat/…`	—	Stateful chat sessions (new, list, send, stream, rate, export, delete).

—-

Getting access

Keys, capabilities and spend caps are managed by IQX in the gateway admin. If you need a key for a new service or client, or want a capability enabled, contact the IQX development team.

From:
https://iqxusers.co.uk/iqxhelp/ - **IQX**

Permanent link:
https://iqxusers.co.uk/iqxhelp/doku.php?id=ai_gateway

Last update: **2026/11/04 17:09**