SECOND LENS
Assurance Register
Document compliance record

THE ASSURANCE REGISTER API

Check every job advert against UK law before it goes out.

Put the checks inside the systems you already use. The same 50 rules, the same fixes, the same questions and the same signed record, sent and returned by your own software.

Client guide and reference: step by step from your first call to a signed record, then every call, field, limit and error.

Version	1.2, 27 September 2026
API address	`https://ar-api.secondlens.co.uk/v1`
Rule set	`3.8+1.6`
Available	With the Organisation plan. Ask hello@secondlens.co.uk for a key.
Help	hello@secondlens.co.uk. Quote the `x-request-id` of any answer you ask about.

Part of this check is done by AI, and AI can be wrong. Read what it says, do not just take it. This is not legal advice. Never put anything about a candidate into this tool. Every example response in this guide was produced by the API itself, run locally, with the AI's answers stood in for and the key made up.

Contents

Every line below is a link to its section.

1. **Part A: step by step**
    1. What the API does
    2. Before you start
    3. Keep your key safe
    4. Your first call
    5. Check an advert
    6. Read the result
    7. Answer questions and decide on rewrites
    8. Get the signed record
    9. Tell us a finding was wrong
    10. Limits, waiting and retrying
    11. Replacing a key
    12. Data protection
    13. How the API is secured
    14. Getting help
2. **Part B: reference**
    1. Conventions
    2. Endpoints
    3. Request fields
    4. The check answer
    5. A finding
    6. Bands, lenses, sectors and modes
    7. Questions
    8. Errors
    9. Limits
    10. Changes

Part A: step by step

1 What the API does

It checks a UK job advert against 50 rules in ten areas of employment, equality, agency and data protection law. It fixes the routine lines, asks for the facts only you know, such as pay or contract type, and ends with a signed, dated PDF record of what was checked. It is the same check as the website, reached by a program instead of a person.

- **We do not store your advert, the findings, the corrected text or the record.** Anthropic, whose model reads the wording, deletes what it receives within 30 days. What your next call needs travels in a sealed token that you keep.
- **No fact is invented.** Pay, hours and contract type are asked for, never guessed.
- **It is not legal advice.** Part of every check is done by AI, and AI can be wrong. Show people what it says and let them decide.
- **No candidate data.** Never send anything about an applicant.

2 Before you start

1. The API comes with the Organisation plan. Checks made through it come from the same licence as checks on the website.
2. Ask for a key at hello@secondlens.co.uk. Second Lens issues it in its console and gives it to you once. It looks like `sl_live_abcdefghjk_...`, with 43 more characters after the second underscore.
3. Each licence can have 2 keys working at once, so one can be replaced without a gap. A key lasts 12 months, or until the licence ends if sooner.

3 Keep your key safe

> **The key is a password for your organisation's checks.** Keep it on your server, in your secrets store or an environment variable. Never put it in a web page, a mobile app, source code, a support ticket or an address.

- Send it only in the `Authorization: Bearer <key>` header. A key in an address is refused with `key_in_url`, because addresses end up in logs; replace any key you sent that way.
- Requests from a web page on another site are refused, and the API sends no cross-origin headers. Call it from your server.
- We keep only a hash of your key, so we cannot tell you it again. If you lose it, or think it has been seen, ask us to replace it.

4 Your first call

Health needs no key. It says only whether the API can be used, and nothing more:

```
curl -sS https://ar-api.secondlens.co.uk/v1/health
```

```
{
  "ok": true
}
```

Then prove your key works, without spending a check:

```
curl -sS https://ar-api.secondlens.co.uk/v1/me \
  -H "Authorization: Bearer $SL_API_KEY"
```

```
{
  "key": {
    "id": "5u7dgr8quu",
    "name": "Example ATS",
    "prefix": "sl_live_5u7d",
    "state": "live",
    "created_at": "2026-11-17T08:30:28.582Z",
    "expires_at": "2027-11-17T08:30:28.580Z",
    "hourly_limit": 60
  },
  "licence": {
    "plan": "organisation",
    "expires_at": "2027-11-17T08:30:28.580Z"
  },
  "allowance": {
    "used": 0,
    "allowance": null,
    "remaining": null,
    "unlimited": true
  }
}
```

5 Check an advert

Send the advert as plain text, with your details. The details matter: without a contact the line inviting requests for adjustments cannot be written, and without your privacy notice address the line pointing to it cannot be written.

```
curl -sS -X POST https://ar-api.secondlens.co.uk/v1/checks \
  -H "Authorization: Bearer $SL_API_KEY" \
  -H "Content-Type: application/json" \
  -d '{"advert":"Warehouse Operative, Leeds ...","organisationName":"Northgate
Logistics","contactEmail":"jobs@northgate.example","privacyNoticeUrl":"https://northgate.example/privac
y","clientRef":"REQ-1042"}'
```

In JavaScript (Node 18 or later), on your server:

```
const res = await fetch('https://ar-api.secondlens.co.uk/v1/checks', {
  method: 'POST',
  headers: { authorization: `Bearer ${process.env.SL_API_KEY}`, 'content-type': 'application/json' },
  body: JSON.stringify({ advert, organisationName: 'Northgate Logistics', contactEmail:
'jobs@northgate.example' }),
  signal: AbortSignal.timeout(45_000),
});
const result = await res.json();
if (!res.ok) throw new Error(`${result.code}: ${result.detail}`);
```

In Python, with the requests library:

```
import os, requests
res = requests.post(
    "https://ar-api.secondlens.co.uk/v1/checks",
    headers={"Authorization": f"Bearer {os.environ['SL_API_KEY']}"},
    json={"advert": advert, "organisationName": "Northgate Logistics"},
    timeout=45,
)
result = res.json()
if not res.ok:
    raise RuntimeError(f"{result['code']}: {result['detail']}")
```

> A check takes up to about 26 seconds, because the AI reads the wording one rule at a time. Allow at least 45 seconds before your own timeout.

The answer (shortened for the page):

```
{
  "reference": "SL-ZGE3-JDT9",
  "client_ref": "REQ-1042",
  "ruleset_version": "3.8+1.6",
  "model_version": "claude-sonnet-5",
  "prompt_version": "476aa70e9d9b07559de3407e02a10eb0fc7bacbd20eb2b1e59fca2eda3c9c60d",
  "checked_at": "2026-11-17T08:30:28.609Z",
  "mode": "rewrite",
  "context": {
    "jurisdiction": "England",
    "advertiser_type": "own",
    "child_facing": false,
    "regulated": false,
    "depth": 1
  },
  "summary": {
    "open": 3,
    "fixed": 1,
    "byBand": {
      "must_fix": 1,
      "should_fix": 0,
      "have_a_look": 2
    },
    "byLens": {
      "1.0": 2,
      "5.0": 1
    }
  },
  "findings": [
    {
      "id": "model:1.3.2",
      "source": "model",
      "rule_id": "1.3.2",
      "rule_name": "Wording that hints at an age",
      "lens": "1.0",
      "lens_name": "Who can apply",
      "band": "must_fix",
      "span": {
        "start": 232,
        "end": 248
      },
      "original": "young, energetic",
      "explanation": "Describing the ideal person as young suggests older applicants are not wanted.",
      "sources": [
        "Equality Act 2010 ss.5, 13, 19, 39"
      ],
      "confidence": 0.93,
      "status": "open",
      "suggestion": {
        "text": "motivated",
        "applied": false,
        "repairId": "1.3.2-semantic"
      }
    },
    {
      "id": "rules:7.2.1",
      "source": "rules",
      "rule_id": "7.2.1",
      "rule_name": "No privacy notice mentioned",
      "lens": "7.0",
      "lens_name": "Data protection",
      "band": "should_fix",
      "span": null,
      "original": null,
```

```
      "explanation": "No privacy notice mentioned",
      "sources": [
        "UK GDPR Arts 13, 14, as amended by Data (Use and Access) Act 2025 s.77, in force 5 February
2026 (SI 2026/82 reg 2(i))"
      ],
      "confidence": null,
      "status": "fixed",
      "suggestion": {
        "text": "Details of how we handle your personal information are in our privacy notice at
https://northgate.example/privacy.",
        "applied": true,
        "repairId": "7.2.1-A2"
      }
    },
    "... 2 more"
  ],
  "advert": {
    "original": "Warehouse Operative, Leeds LS10\nNorthgate Logistics is hirin...",
    "corrected_text": "Warehouse Operative, Leeds LS10\nNorthgate Logistics is hirin..."
  },
  "questions": [
    {
      "kind": "gap",
      "rule_id": "5.2.4",
      "ask": "Will you consider flexible working for this role?",
      "why": "There is no duty to say. Adverts that do tend to reach more people, and answering this
will not move the score.",
      "type": "choice",
      "options": [
        {
          "id": "yes",
          "label": "Yes, we will discuss it"
        },
        {
          "id": "no",
          "label": "No"
        }
      ],
      "fields": []
    }
  ],
  "pending": [
    {
      "rule_id": "1.3.8",
      "needs": [
        "contact_name",
        "contact_phone"
      ],
      "labels": [
        "A name to contact",
        "Or a phone number"
      ]
    }
  ],
  "assumptions": [],
  "score": {
    "before": 57,
    "now": 67,
    "ceiling": 100
  },
  "to_hundred": "{ ... }",
  "counts": "{ ... }",
  "unchecked": [],
  "degraded": false,
```

```
  "wording": {
    "state": "done",
    "calls": 4
  },
  "allowance": {
    "used": 1,
    "allowance": null,
    "remaining": null,
    "unlimited": true
  },
  "disclaimer": "Part of this check is done by AI, and AI can be wrong. Read what it says, do not just
take it. This is not legal advice. Never put anything about a candidate into this tool.",
  "token": "v1.LB81D9AwqtMTjOfazYey7... (5,234 characters)"
}
```

6 Read the result

- `summary`: what is still open, by band and by lens.
- `findings`: one entry per issue. Each says the rule, the lens, the band, where it is (`span`, as character offsets into `advert.original`), the words, why, the citation from our rule pack, and whether our rules (`source: "rules"`) or the AI (`source: "model"`) found it. AI findings carry a `confidence`; show them as suggestions, because they can be wrong.
- `advert.corrected_text`: the advert as it stands with the fixes applied. Absent in `flag` mode.
- `questions` and `pending`: facts only you know, and standard lines waiting for a detail.
- `unchecked` and `degraded`: rules the AI did not reach in time. **A degraded check is not a clean bill.** Show it.
- `token`: keep it for the next step. It lasts 24 hours.

One finding in full:

```
{
  "id": "model:1.3.2",
  "source": "model",
  "rule_id": "1.3.2",
  "rule_name": "Wording that hints at an age",
  "lens": "1.0",
  "lens_name": "Who can apply",
  "band": "must_fix",
  "span": {
    "start": 232,
    "end": 248
  },
  "original": "young, energetic",
  "explanation": "Describing the ideal person as young suggests older applicants are not wanted.",
  "sources": [
    "Equality Act 2010 ss.5, 13, 19, 39"
  ],
  "confidence": 0.93,
  "status": "open",
  "suggestion": {
    "text": "motivated",
    "applied": false,
    "repairId": "1.3.2-semantic"
  }
}
```

Offsets count UTF-16 code units, as JavaScript strings do, in `advert.original` as the API returns it (Unicode NFC). Count against that string, not your own copy.

7 Answer questions and decide on rewrites

Send the token back with what has changed. Everything in one request applies, or none of it does.

```
curl -sS -X POST https://ar-api.secondlens.co.uk/v1/checks/answers \
  -H "Authorization: Bearer $SL_API_KEY" \
  -H "Content-Type: application/json" \
  -d '{"token":"<token>","answers":{"3.3.2":"permanent"},"repairs":[{"id":"<repair
id>","action":"undo"}],"details":{"contactPhone":"0113 496 0000"}}'
```

The answer has the same shape as a check, with a fresh token. This call is free and uses no AI.

8 Get the signed record

```
curl -sS -X POST https://ar-api.secondlens.co.uk/v1/checks/record \
  -H "Authorization: Bearer $SL_API_KEY" \
  -H "Content-Type: application/json" \
  -d '{"token":"<token>"}' \
  -o record.pdf
```

The answer is the PDF, with `x-record-reference` and `x-record-sealed` headers. A sealed record can be checked by anyone, offline, at https://assurance-register.secondlens.co.uk/verify, or in code against the public keys at `/keys.json`. The seal proves this check happened on this advert with this rule set; it does not prove the advert is lawful.

In the example run for this guide the answer was 200, unsealed, because the machine that made this guide holds no signing key.

9 Tell us a finding was wrong

```
curl -sS -X POST https://ar-api.secondlens.co.uk/v1/reports \
  -H "Authorization: Bearer $SL_API_KEY" \
  -H "Content-Type: application/json" \
  -d '{"reason":"wording_is_fine","ruleId":"1.3.2","excerpt":"energetic"}'
```

This is the one call that stores what you send, so that a person can correct the rules. Send the quote, never the advert or anyone's details.

10 Limits, waiting and retrying

Limit	What it is	What you get
Per key	60 checks started per key per hour, unless Second Lens sets another number (up to 600)	429 `hourly_limit` with `Retry-After`
Across the API	A maximum number of checks a minute for all customers together, set by Second Lens	429 `api_rate_limited` with `Retry-After`
Per licence	The checks your plan includes each month	402 `allowance_spent`

Limit	What it is	What you get
Per advert	40 to 60,000 characters	400 `text_too_short` or `text_too_long`

- **Retry only what can succeed.** Wait the `Retry-After` seconds after a 429, and retry a 503 after a minute or so with a growing wait. Never retry a 400, 401, 402, 403, 413, 415 or 422 unchanged: it will fail the same way.
- **A 200 has cost a check.** Do not send the same advert again to be sure. A result marked `degraded` was still charged, because the rule half of it is a full result.
- **No result, no charge.** An error answer never costs a check. If your connection drops during a check you may have been charged without seeing the answer; `GET /v1/me` shows the count.
- Repeat protection with an `Idempotency-Key` header is not offered in version 1.2. The IETF draft that describes it is not a standard.

Every answer, including every error, carries `x-request-id`.

11 Replacing a key

1. Ask us for a replacement. You get the new key once.
2. The old key keeps working for 7 days, so you can move over without a failed request.
3. Put the new key in place, then tell us, and we revoke the old one at once.

A key that is switched off answers 403 `key_paused`; revoked or expired, 401. If you think a key has been seen by anyone else, ask us to revoke it straight away.

12 Data protection

- **We do not store your advert, the findings or the record**, here or by the API. The token is sealed and held by you.
- We keep a log of API requests for 30 days: the time, the key, what was asked for, the answer's status, how long it took, token counts and the advert's length. No advert, no quote from it, no hash of it and no IP address.
- Your advert goes to our server functions, run by Netlify in the United States, and to Anthropic's AI service, which may run the model in any country it uses. Anthropic's commercial terms say it may not train its models on what we send. Transfers are made under the standard contractual clauses and the UK Addendum.
- Adverts should hold no personal data. Where a contact name or number is in one, it is processed as part of the advert.
- The full terms are the data processing agreement and "What is held, and for how long", at https://assurance-register.secondlens.co.uk/legal/dpa.html and https://assurance-register.secondlens.co.uk/legal/retention.html. A complaint about how we handle personal data goes to hello@secondlens.co.uk; we acknowledge it within 30 days.

13 How the API is secured

In short: every connection is encrypted with TLS, the same encryption banks use to protect data on its way; only a key issued to your organisation gets in; and we keep no copy of your advert, the findings or the record. The detail:

Area	What is done
Encryption in transit	HTTPS only, on `ar-api.secondlens.co.uk`. The connection is TLS 1.2 at the least, and TLS 1.3 where your software supports it. The API runs on Netlify, which states that all traffic on its network is encrypted with at least TLS 1.2 and AES-256. TLS 1.0 and 1.1 are not used.
Strict HTTPS	Every answer carries `Strict-Transport-Security: max-age=63072000; includeSubDomains; preload`, so software that honours it will not use plain HTTP for this domain for two years. Plain HTTP is redirected, but a key sent over plain HTTP has already crossed the network unencrypted: replace it.
Certificate	A publicly trusted certificate, issued and renewed automatically by the host. Check it the normal way. Do not pin it: it is renewed without notice.
Your key	32 random bytes from a cryptographic generator, shown to you once. We store only its SHA-256 hash and compare in constant time, so a copy of our store would not give anyone your key.
Sending the key	Only in the `Authorization: Bearer` header. A key in an address is refused (`key_in_url`). A call from a web page on another site is refused (`cross_origin`), and no cross-origin headers are sent.
Control	Each key can be switched off, limited per hour, replaced or revoked by us at once, and the whole API has one off switch. A licence has at most 2 working keys, and a key expires after 12 months.
Limits	A limit per key per hour and one per minute across the API, so a leaked key or a runaway loop cannot run without end. See Part B, section 9.
Your data	We do not store your advert, the findings or the record. Anthropic, whose model reads the wording, deletes what it receives within 30 days. The conversation travels in a token sealed with AES-256-GCM, bound to your key and good for 24 hours: you cannot read or change it, and another key cannot use it. The request log holds no advert, no hash of one and no IP address, and is deleted after 30 days.
The record	Signed with ECDSA P-256 over SHA-256. The public keys are at `https://assurance-register.secondlens.co.uk/keys.json`, and anyone can check a record at `https://assurance-register.secondlens.co.uk/verify.html`.
Every answer	Carries `Cache-Control: no-store`, a content security policy that allows nothing, `nosniff`, no referrer and no framing. An error never carries internal detail.
The AI step	The advert goes to Anthropic's AI service over an encrypted connection for the wording check. See step 12 for where it runs and what may be done with it.
Standards	Designed against the NCSC's guidance on securing HTTP-based APIs (April 2025).

What it does not do, so you can plan for it:

- It does not use client certificates (mutual TLS), signed requests or OAuth, as UK Open Banking APIs do. Your key is the only credential, so guard it like a password.
- It has no allow-list of IP addresses.

- It has not yet had an independent penetration test. This guide will say when it has.

Your side: use software that supports TLS 1.2 or later and checks certificates, and never turn certificate checking off. Keep the key on your server (step 3), and if you think it has been seen, ask us to revoke it.

Sources: Netlify, Security at Netlify, netlify.com/security (TLS 1.2 minimum and AES-256); NCSC, Securing HTTP-based APIs, April 2025. Everything else in this section is read from the API's own code.

14 Getting help

Write to hello@secondlens.co.uk. Say what you sent, what came back, and the `x-request-id`. Never send your key.

Part B: reference

1. Conventions

- Base address `https://ar-api.secondlens.co.uk/v1`. The API answers on this host only.
- Every request and answer is JSON in UTF-8, except the record, which is `application/pdf`. Send `Content-Type: application/json` on every POST, or the answer is 415.
- Errors are RFC 9457 problem details, `application/problem+json`, with `type`, `title`, `status`, `detail`, `instance` and our own `code`. They never carry internal detail.
- A field a call does not take is refused by name (`unknown_field`), so a misspelt field is found at once rather than ignored.
- Times are ISO 8601 in UTC. Prose in answers is British English.
- HTTPS only, TLS 1.2 or 1.3. Plain HTTP is redirected. See step 13 for how the API is secured.

An error, for example with no key:

```
{
  "type": "https://ar-api.secondlens.co.uk/v1/problems/key_missing",
  "title": "No API key was sent",
  "status": 401,
  "detail": "No API key was sent",
  "instance": "urn:request:ddbf504a2d8f",
  "code": "key_missing"
}
```

And a field the call does not take:

```
{
  "type": "https://ar-api.secondlens.co.uk/v1/problems/unknown_field",
  "title": "The body has a field this call does not take",
  "status": 400,
  "detail": "Not taken by this call: price.",
  "instance": "urn:request:326e0f4a8396",
  "code": "unknown_field",
  "fields": [
    "price"
  ]
}
```

2. Endpoints

Method	Path	Key	What it does
GET	`/v1/health`	No	Whether the API can be used: ok or not ok, and nothing more. No key needed.
GET	`/v1/me`	Yes	Your key, your licence and this month's allowance.
GET	`/v1/rules`	Yes	Every rule, band and lens, with the rule set version, for building your screens.
GET	`/v1/lenses`	Yes	The ten areas of law and how many rules each has.

Method	Path	Key	What it does
POST	`/v1/checks`	Yes	Check an advert. Costs one check.
POST	`/v1/checks/answers`	Yes	Answer questions, decide on rewrites, add missing details. Free.
POST	`/v1/checks/record`	Yes	The signed PDF record of the check. Free.
POST	`/v1/checks/settle`	Yes	What the rules still assume about an edited advert. No AI; free. Answers with { "assumptions": [...] } only: no findings, advert or token, so it is not a check.
POST	`/v1/reports`	Yes	Tell us a finding was wrong. Free.

3. Request fields

POST `/v1/checks`

Field	Meaning
`advert`	The advert as plain text, 40 to 60,000 characters. No HTML and no invisible characters.
`advertiserType`	"own" for your own job, "client" for an agency advertising a client's job. Anything else reads as "own".
`organisationName`	Your organisation's name, at most 200 characters.
`publicAuthority`	true if you are a public authority. Brings the public sector equality duty in.
`businessName`	An agency's trading name, at most 200 characters.
`contactName`	Who applicants contact, at most 200 characters.
`contactEmail`	The contact email address for adjustment requests.
`contactPhone`	The contact phone number, digits, spaces and + ( ) - . only.
`privacyNoticeUrl`	The address of your privacy notice. Never fetched; written into a sentence.
`depth`	1, 2 or 3: how many screening passes the AI makes. The check raises it to 3 itself for child-facing or regulated roles.
`sector`	One of "education", "childcare", "health", "social_care", "financial_services", "security", "other". Every sector except "other" reads the advert at depth 3.
`lenses`	A list of lens ids, such as ["1.0", "5.0"], to return findings for those areas only. Every rule still runs.
`mode`	"flag" returns findings only, with no rewrites; "suggest" proposes rewrites but applies none; "rewrite" (the default) applies the fixes the product is sure of, at depth 1 only; at depth 2 or 3 they are offered, as in "suggest". A "flag" check takes no answers, details or repairs afterwards, only referrals.
`clientRef`	Your own reference, at most 64 characters. Returned as client_ref and never read.

POST `/v1/checks/answers`

Field	Meaning
`token`	The token from the last answer. It carries the conversation and lasts 24 hours.
`answers`	Answers to open questions, by rule id. A choice is the option id; a set of fields is an object of field id to text. null or an empty string takes an answer back.
`referrals`	Answers to referral questions: { "<referral id>": { "<question id>": "<option id>" } }.
`repairs`	Decisions on rewrites: a list of { "id": "<repair id>", "action": "accept" | "undo" }.
`details`	Details that were missing when you checked, such as { "contactEmail": "jobs@example.co.uk" }. The standard lines that were waiting for them are written in.
`reset`	true puts everything back to the advert as it was sent: every change is undone, the ones the check made on its own included, before anything else in the request is applied.

POST `/v1/checks/record`

Field	Meaning
`token`	The token from the last answer. It carries the conversation and lasts 24 hours.

POST `/v1/checks/settle`

Field	Meaning
`advert`	The advert as plain text, 40 to 60,000 characters. No HTML and no invisible characters.
`advertiserType`	"own" for your own job, "client" for an agency advertising a client's job. Anything else reads as "own".
`organisationName`	Your organisation's name, at most 200 characters.
`publicAuthority`	true if you are a public authority. Brings the public sector equality duty in.
`businessName`	An agency's trading name, at most 200 characters.
`contactName`	Who applicants contact, at most 200 characters.
`contactEmail`	The contact email address for adjustment requests.
`contactPhone`	The contact phone number, digits, spaces and + ( ) - . only.
`privacyNoticeUrl`	The address of your privacy notice. Never fetched; written into a sentence.
`today`	The date to read rates against, YYYY-MM-DD. Anything else, or nothing, reads as today; it is not refused.

POST `/v1/reports`

Field	Meaning
`reason`	Why a finding was wrong: one of "rule_does_not_apply", "wording_is_fine", "excerpt_is_wrong", "rewrite_is_wrong", "severity_too_high", "severity_too_low", "citation_is_wrong", "duplicate", "something_else", "wrong_here", "unclear", "did_not_work".
`ruleId`	The rule id of the finding, such as "1.3.2".
`ruleName`	The rule's name, as the finding gave it.

Field	Meaning
`severity`	"block", "advisory", "assumption" or "referral".
`confidence`	The finding's confidence, where it had one.
`excerpt`	The words the finding quoted. Send the quote, not the advert.
`modelReason`	The finding's explanation.
`suggested`	The rewrite it suggested, if any.
`advertiserType`	"own" for your own job, "client" for an agency advertising a client's job. Anything else reads as "own".
`jurisdiction`	The jurisdiction the check found.
`depth`	1, 2 or 3: how many screening passes the AI makes. The check raises it to 3 itself for child-facing or regulated roles.
`rulesetVersion`	The ruleset_version of the check.
`reference`	The check's reference, such as "SL-7K2M-9XQD".

Required: `advert` on checks and settle; `token` on answers and record; `reason` and `ruleId` on reports. Everything else is optional.

4. The check answer

Field	Meaning
`reference`	The check's reference, printed on the record.
`client_ref`	Your clientRef, as sent.
`ruleset_version`	The rule set that ran, as register+pack.
`model_version`	The AI model that read the wording.
`prompt_version`	A fingerprint of the instructions the AI was given.
`checked_at`	When the check started.
`mode`	flag, suggest or rewrite.
`context`	What the check decided about the advert: jurisdiction, advertiser type, child facing, regulated, depth.
`summary`	open, fixed, byBand and byLens counts of what is still open.
`findings`	Every finding, one shape (section 5).
`advert`	original (NFC), and corrected_text (null in flag mode).
`questions`	Open questions and referral questions (section 7).
`pending`	Standard lines waiting for a detail, and which detail.
`assumptions`	What the check assumed because the advert did not say.
`score`	before, now and ceiling: the product's score out of 100.

Field	Meaning
`to_hundred`	The route to a full score, item by item.
`counts`	Fixed, to look at and to answer.
`unchecked`	Rules not reached, with the reason.
`degraded`	true when any rule went unread. Show it.
`wording`	Whether the AI read ran, and how many calls it made.
`allowance`	Where your licence stands this month.
`disclaimer`	The sentence to show beside any result.
`token`	The sealed conversation, for answers and record.

5. A finding

Field	Meaning
`id`	Unique within the answer, such as rules:1.3.8 or model:1.3.2.
`source`	"rules" (our code, the same every time), "model" (the AI; can be wrong) or "house" (your organisation's own house check, set with Second Lens).
`rule_id`	The rule, such as 1.3.2.
`rule_name`	The rule in plain words.
`lens`	The area of law, 1.0 to 10.0 (section 6).
`band`	must_fix, should_fix or have_a_look.
`span`	{ start, end } offsets into advert.original, or null where the finding is about something missing.
`original`	The words at the span.
`explanation`	Why, in plain English.
`sources`	The citation from our rule pack. Never written by the AI.
`confidence`	The AI's confidence from 0 to 1, or null for rules.
`status`	open or fixed.
`suggestion`	A rewrite, with applied and repairId, or null.

6. Bands, lenses, sectors and modes

Band	Shown as	What it means
`must_fix`	Must fix	The law does not allow this.
`should_fix`	Should fix	This may breach the law unless you can justify it.
`have_a_look`	Have a look	There is no legal requirement here, but

Lens	Area of law	Rules
`1.0`	Who can apply	12
`2.0`	Northern Ireland	4
`3.0`	Advertising someone else's job	8
`4.0`	Is the advert accurate	5
`5.0`	Pay and how you are employed	4
`6.0`	Right to work	2
`7.0`	Data protection	3
`8.0`	Working with children, and criminal records	5
`9.0`	Regulated work	4
`10.0`	Telling people the pay	3

Sectors: `education`, `childcare`, `health`, `social_care`, `financial_services`, `security`, `other`. Modes: `flag`, `suggest`, `rewrite`.

Jurisdiction is read from the advert (England, Wales, Scotland or Northern Ireland), from place names, postcodes and wording; where the advert does not say, England is assumed and the answer says so.

7. Questions

Rule	Question	Answer with
`3.3.2`	Is this role permanent, temporary, or for a fixed term?	`permanent`, `temporary`, `fixed_term`
`4.2.1`	The advert states a rate but leaves something out.	An object of the field ids the question lists
`3.3.4`	The advert gives a rate but leaves out something the law wants beside it.	An object of the field ids the question lists
`5.2.4`	Will you consider flexible working for this role?	`yes`, `no`
`10.2.2`	What does this role pay?	An object of the field ids the question lists

Referral questions come as `{ kind: "referral", referral_id, questions: [{ id, ask, options }] }` and are answered in `referrals`. The referrals are `authority`, `pattern`, `level`.

8. Errors

Code	Status	Meaning
`not_found`	404	There is nothing at this address
`method_not_allowed`	405	That method is not allowed here
`cross_origin`	403	Requests from a web page on another site are refused

Code	Status	Meaning
`key_in_url`	400	A key was sent in the address
`unsupported_media_type`	415	The body must be JSON
`body_too_large`	413	The body is too large
`bad_body`	400	The body is not a JSON object
`unknown_field`	400	The body has a field this call does not take
`invalid_field`	400	A field is not in the shape this call takes
`key_missing`	401	No API key was sent
`key_invalid`	401	The API key is not recognised
`key_revoked`	401	The API key has been revoked
`key_expired`	401	The API key has expired
`key_paused`	403	The API key is switched off
`licence_not_live`	403	The licence behind this key is not in force
`service_switched_off`	503	Checking is switched off
`api_off`	503	The API is switched off
`feature_off`	503	This part of Assurance Register is switched off for now. Please try again later.
`hourly_limit`	429	This key has started its checks for the hour
`api_rate_limited`	429	The API is at its maximum rate
`allowance_spent`	402	The licence has used its checks for this month
`text_too_short`	400	The advert is too short
`text_too_long`	400	The advert is too long
`advert_formatting`	400	The advert has formatting code or invisible characters
`token_unavailable`	503	This deployment cannot seal a token
`token_invalid`	400	The token is not one this API made for this key
`token_expired`	400	The token has expired
`answer_refused`	422	An answer could not be used
`record_failed`	500	The record could not be built
`check_failed`	500	The check could not be completed

When the AI part of a check cannot run, the answer is still 200 with the rule findings, `degraded: true` and `wording.code` set to one of these:

Code	Retry?	What it means
`no_api_key`	No	The wording check is not set up yet, so this half of the check could not run. Checking again will not help.

Code	Retry?	What it means
`model_key_rejected`	No	The wording check is not available just now, so this half of the check could not run. Checking again will not help until it is back.
`model_unavailable`	Yes, after a wait	Sorry, something went wrong. Please try again.
`model_rate_limited`	Yes, after a wait	Sorry, something went wrong. Please try again. The service that reads the wording is busy, so wait a minute first.
`model_overloaded`	Yes, after a wait	Sorry, something went wrong. Please try again. The service that reads the wording is busy, so wait a minute first.
`model_spend_capped`	No	The wording check is not available just now, so this half of the check could not run. Checking again will not help until it is back.
`model_unreachable`	Yes, after a wait	Sorry, something went wrong. Please try again. The service that reads the wording could not be reached, so wait a minute first.
`model_timeout`	Yes, after a wait	Sorry, something went wrong. Please try again. The service that reads the wording did not answer in time, so this half of the check stopped rather than keeping you waiting.
`check_failed`	Yes, after a wait	Sorry, something went wrong. Please try again.

9. Limits

What	Limit
Advert	40 to 60,000 characters
Body of a check or settle	120,000 characters
Body of answers or record	400,000 characters
clientRef	64 characters
Checks per key per hour	60 by default, up to 600
Keys per licence	2 working at once
Key life	365 days, or the licence's end if sooner
Old key after a replacement	7 days
Token life	24 hours
Time for one check	About 26 seconds at most

10. Changes

Version	Date	What changed
1.2	27 September 2026	Says what reset, today and settle do: reset goes back to the advert as sent; a today that is not a date reads as today; settle answers with the assumptions only. rewrite applies fixes at depth 1 only. A flag check takes no answers, details or repairs, only referrals. An unknown repair id is refused. An advert with half a character (a lone surrogate) is refused, and names with < or > are dropped.
1.1	26 September 2026	First published version: API keys, the one finding shape, sectors, lenses and modes, corrected text and RFC 9457 errors.

Built from the API's own code on 27 September 2026. Example advert: Warehouse Operative, Leeds LS10.