Environment and debugging

Environment Variables

OpenClaw pulls environment variables from multiple sources. The rule is **never override existing values**.

Precedence (highest → lowest)

1. **Process environment** (what the Gateway process already has from the parent shell/daemon).
2. `.env` **in the current working directory** (dotenv default; does not override).
3. **Global** `.env` at `~/.openclaw/.env` (aka `$OPENCLAW_STATE_DIR/.env`; does not override).
4. **Config** `env` **block** in `~/.openclaw/openclaw.json` (applied only if missing).
5. **Optional login-shell import** (`env.shellEnv.enabled` or `OPENCLAW_LOAD_SHELL_ENV=1`), applied only for missing expected keys.

If the config file is missing entirely, step 4 is skipped; shell import still runs if enabled.

Config `env` block

Two equivalent ways to set inline env vars (both are non-overriding):

```
{
  env: {
    OPENROUTER_API_KEY: "sk-or-...",
    vars: {
      GROQ_API_KEY: "gsk-...",
    },
  },
}
```

Shell env import

`env.shellEnv` runs your login shell and imports only **missing** expected keys:

```
{
  env: {
    shellEnv: {
      enabled: true,
      timeoutMs: 15000,
    },
  },
}
```

Env var equivalents:

- `OPENCLAW_LOAD_SHELL_ENV=1`
- `OPENCLAW_SHELL_ENV_TIMEOUT_MS=15000`

Env var substitution in config

You can reference env vars directly in config string values using `${VAR_NAME}` syntax:

```
{
  models: {
    providers: {
      "vercel-gateway": {
        apiKey: "${VERCEL_GATEWAY_API_KEY}",
      },
    },
  },
}
```

See for full details.

Path-related env vars

Variable	Purpose
`OPENCLAW_HOME`	Override the home directory used for all internal path resolution (`~/.openclaw/` , agent dirs, sessions, credentials). Useful when running OpenClaw as a dedicated service user.
`OPENCLAW_STATE_DIR`	Override the state directory (default `~/.openclaw`).
`OPENCLAW_CONFIG_PATH`	Override the config file path (default `~/.openclaw/openclaw.json`).

`OPENCLAW_HOME`

When set, `OPENCLAW_HOME` replaces the system home directory (`$HOME` / `os.homedir()`) for all internal path resolution. This enables full filesystem isolation for headless service accounts.

Precedence: `OPENCLAW_HOME` > `$HOME` > `USERPROFILE` > `os.homedir()`

Example (macOS LaunchDaemon):

```
<key>EnvironmentVariables</key>
<dict>
  <key>OPENCLAW_HOME</key>
  <string>/Users/kira</string>
</dict>
```

`OPENCLAW_HOME` can also be set to a tilde path (e.g. `~/svc`), which gets expanded using `$HOME` before use.

Related

‹ OpenClaw Lore

Debugging ›