**ChatGPT Routing Helper — Rationale, Release Notes & Install Guide (Chrome & Firefox)**

**Overview**

**ChatGPT Routing Helper** is a lightweight browser extension that exposes model “routing” controls and best-practice meta prompts directly inside ChatGPT’s web UI. It adds a compact panel that can automatically prepend a structured meta line and (optionally) a self-reflection rubric to each message, and records a local log of which options were injected. This helps users drive consistent reasoning depth, complexity, verbosity, push-back behavior, and tool-usage budgets—without manually pasting meta prompts on every turn.

**Why it matters**

* **Consistency at scale:** Complex work benefits from repeatable routing settings (e.g., high reasoning effort with medium verbosity). The extension removes human error from manual prompting and preserves settings across sessions.
* **Transparency & attribution:** The in-panel **View log** tab shows exactly what was injected and when, enabling post-hoc evaluation and reproducibility.
* **Frictionless iteration:** One-click toggles for push-back (Low/Medium/High) and a reusable rubric (private self-critique) improve answer quality without cluttering the chat.
* **Non-intrusive:** No servers, no accounts, no analytics. All data stays in the browser’s local storage.

**Key features**

* **Routing toggles:** Reasoning effort, complexity target, verbosity, tool-call budget, push-back level.
* **Auto-prepend meta:** Injects the selected routing meta line at send time.
* **Self-reflection rubric (optional):** Privately prompts the model to self-critique; not shown in the final answer.
* **Persistent settings:** Stored locally; survive page reloads and new sessions.
* **In-panel log:** Timestamps, the exact meta header used, and a user “Score” field (0–100) for subjective quality tracking.
* **“Log even if not injecting”:** Records a [no-inject] entry when Auto-prepend is off, so attribution isn’t lost during tests.
* **“Test log” button:** Writes a log entry immediately—useful to verify the panel is live without sending a message.
* **Robustness:** Hooks Enter, Send-button clicks, and form submits; includes a debounced SPA observer to re-attach after DOM changes.

**How it works (brief)**

* The extension injects a small content script on ChatGPT’s web pages.
* On **send**, it builds a meta header from dropdown selections (e.g., [meta reasoning\_effort=high; complexity=complex; verbosity=medium; tool\_call\_budget=3]) and optionally appends a push-back policy and a short rubric block.
* If the composer already contains a meta line, injection is skipped (but can still be logged if enabled).
* A local log captures timestamp, selected options, the first 200 characters of the injected header, and an optional user score.
* All processing is local; no network calls are made by the extension.

**Privacy & permissions**

* **Local-only storage:** Settings and log reside in browser storage.
* **No external servers:** The extension does not transmit data.
* **Minimal permissions:** storage, scripting (Chrome), and host access for ChatGPT domains.

**Chrome — Beginner-Friendly Install (Unpacked)**

1. **Download the ZIP**  
   chatgpt\_routing\_ext\_v523\_chrome.zip  
   SHA-256: 5feaee9710c3df2d04420f7e0afeb5882920f9c6efdbeb72c54bab5a5c86edde
2. **Unzip** it to a folder.
3. Open **chrome://extensions** → toggle **Developer mode** ON (top-right).
4. Click **Load unpacked** → select the **unzipped folder** (the one containing manifest.json).
5. Open ChatGPT (chat.openai.com or chatgpt.com). A **Routing Options** panel should appear bottom-right.
6. In the panel:
   * Ensure **Auto-prepend meta** is **ON**.
   * Click **View log** (initially empty).
   * Click **Test log** → a new row should appear (confirms the extension is active).
   * Send a short message via **Send**; the log should record the injected meta line (or [no-inject] … if Auto-prepend is OFF).

**Update later:** Remove the old folder in **chrome://extensions**, then **Load unpacked** with the new one.  
**Uninstall:** **chrome://extensions** → **Remove**.

**Firefox — Beginner-Friendly Install**

**Option A: XPI (one-click install)**

* Download: chatgpt\_routing\_ext\_v523\_firefox.xpi  
  SHA-256: 09b486d0dca968b66eb11c2ae3bb1d60a39070eb66a17cf2c1b3a66581a9aa30
* Open the file with Firefox → click **Add**.  
  *(If Firefox requires a signed add-on and blocks install, use Option B.)*

**Option B: Temporary Add-on (works everywhere; resets on restart)**

1. Go to **about:debugging** → **This Firefox** → **Load Temporary Add-on…**
2. Select the **XPI file** (chatgpt\_routing\_ext\_v523\_firefox.xpi).
3. Open ChatGPT. The **Routing Options** panel should appear bottom-right.
4. Verification:
   * Ensure **Auto-prepend meta** is ON.
   * **View log** → **Test log** (should add a row).
   * Send a short message; a new log row with the meta line confirms injection.

**Uninstall (XPI):** **about:addons** → find the extension → **Remove**.  
**Note:** Temporary add-ons are removed on restart; repeat step 1–2 or install a signed XPI when available.

**Quick troubleshooting**

* **No panel visible:** refresh the page; ensure you’re on **chat.openai.com** or **chatgpt.com**; verify the extension is enabled; allow scripts if using blockers.
* **Log empty:** turn **Auto-prepend meta** and **Log even if not injecting** ON; click **Test log**; send using the **Send** button; toggle **Options** ↔ **View log** once to force a settings read.
* **Performance:** collapse the panel with **Hide**; refresh if needed (the observer re-attaches cleanly).

**Release notes (v5.2.3)**

* **New:** “Log even if not injecting” (ON by default) to preserve attribution in tests.
* **New:** “Test log” button for instant panel verification.
* **Improved:** Send hooks cover Enter, Send-button clicks, and form submits; debounced SPA observer.
* **Fixed:** Safer content-script syntax and guards across Chrome/Firefox.
* **UI:** Compact panel; persistent settings; local Score field (0–100) per log row.

**Intended audience**

* Practitioners who routinely rely on structured prompting (STEM, research, policy, product, law).
* Teams who need **repeatable** routing and traceable meta for audits, QA, or lab notebooks.
* Educators and reviewers who want to **see what was injected** and how it influenced output.

**Security, maintainability, and store readiness**

* **Local-only storage**, **no network access**, **minimal permissions** (limited host scope to ChatGPT domains).
* **Small code surface** with explicit guards for Chrome/Firefox APIs and a debounced DOM observer.
* Clean uninstalls and no persistence outside the browser profile.
* Well-scoped matching rules (ChatGPT domains only) reduce risk of cross-site effects.

**Non-clickable resource placeholders (to be replaced with hosted links)**

* Chrome: chatgpt\_routing\_ext\_v523\_chrome.zip
* Firefox (XPI): chatgpt\_routing\_ext\_v523\_firefox.xpi

SHA-256 hashes are listed above for integrity checks.