مختصر كتابي

من دون دليل، من أين يأتي الموعد؟



من دون دليل في بيانات الطلب، من أين يأتي موعد التوصيل؟

تؤكد الـTool الآن أن الطلب `4800` لمكتبة الفجر (`C200`) ليس له موعد توصيل مؤكد. لكن الـ**Caller** يفتتح فرعًا جديدًا يوم السبت ويضغط على مريم لتعطيه أي يوم ولو تقريبيًا، فتسأل مريم **مساعد Seagull** عمّا تقوله له.

عند تشغيل السؤال 20 مرة على **Claude Haiku** باستخدام الـ**Prompt** الخاص بالكورس (الذي ينصّ على عدم اختراع موعد توصيل)، لم يَعِد الـ**Model** بيوم محدد في أيٍّ من التشغيلات العشرين، وحتى حين اقترحت مريم قول يوم الخميس رفض في 5 من 5 تشغيلات. لكن في 11 من 20 ردًّا بحث الـModel في ملفات الشركة ووجد سياسة التوصيل في القاهرة (يوما عمل بعد تأكيد الدفع) وقدّمها لمريم رغم غياب تاريخ تأكيد الدفع للطلب `4800` .

شغّلنا السؤال نفسه 5 مرات بـPrompt عادي لخدمة العملاء يتضمن تاريخ اليوم (الأحد 15 نوفمبر 2026) بلا جملة المنع. في 2 من 5 تشغيلات حسب الـModel يومًا؛ أحدها قدّر **الثلاثاء 17 نوفمبر** بافتراض أن الدفع تأكد اليوم. هذا أخطر نوع لأن الحساب صحيح والسياسة حقيقية بلا مصدر للافتراض. وبلا Tools نصح الـModel مريم في 5 من 5 تشغيلات بالتحقق من النظام أولًا.

كل ما يراه الـModel في المحادثة — رسالة مريم ونتائج الـTools والملفات العائدة من البحث — يسمّى **الـContext**، والشرط بأن لكل معلومة في الرد مصدرًا فيه يسمّى **الـGrounding**. جملة الـPrompt تقلّل التواريخ غير المسنودة لكنها لا تضمنها، لذا يتحقق الكود بنفسه.

الحل البديهي الأول كان **الـFilter** البسيط: منع أي رد فيه اسم يوم ما دام الطلب بلا موعد مؤكد. لكن على 40 ردًّا حقيقيًا مسجّلًا عن `4800` ، منع الـFilter 35 ردًّا رغم أن ردّين فقط اخترعا يومًا؛ لأنه حجب ردودًا أمينة كرّرت «السبت» من كلام الـCaller. لذلك يعتمد **الـGrounding Check** على سؤال المصدر: يجمع التواريخ المسموحة من الـContext أولًا، ثم يقارن بها كل يوم في الرد بعد توحيد الحروف والأرقام وتحويل «بعد غد»، مع اشتراط «ال» أو «يوم» قبل اسم اليوم لتجنّب التباس «اثنين كرتونة»:

```
def context_dates(trace, today=TODAY):
    allowed = {datetime.date.fromisoformat(today).isoformat()}
    for step in trace if isinstance(trace, list) else []:
        if not isinstance(step, dict):
            continue
        if step.get("kind") == "user":
            allowed.update(h["date"] for h in date_mentions(step.get("text"),
today))
        elif step.get("kind") == "tool":
            result = step.get("result")
            if step.get("name") == "read_order":
                eta = verified_eta(result)
                if eta["supported"]:
                    allowed.add(eta["answer"])
            elif isinstance(result, dict) and isinstance(result.get("evidence"),
list):
                for doc in result["evidence"]:
                    if isinstance(doc, dict):
                        allowed.update(h["date"] for h in
date_mentions(doc.get("text"), today))
    return allowed
```

وبدل حجب الرد كله (لأن الـModel قد يذكر يومًا للتحذير منه)، تضع `check_answer` علامة بجانب كل يوم بلا مصدر وتضيف في أعلى الرد كلام الـTool نفسه بأنه لا يوجد موعد مؤكد.

القياس الحقيقي	قبل (الـFilter / بلا تحقق)	بعد (الـGrounding Check)
الردود المسجّلة عن 4800 (40 ردًّا، 2 بلا مصدر)	الـFilter منع 35 من 40	علامة على 6 من 45 فقط (2 بلا مصدر + 4 تحذير/متابعة)
الـPrompt العادي (10 تشغيلات، Claude Haiku 4.5)	2 من 5 وصلت بلا علامة	3 ذكرت الثلاثاء، 0 وصلت بلا علامة
طلب فرع المعادي 4801 بموعد مؤكد (19 نوفمبر)	—	5 من 5 مرّت بلا علامة

- **الـContext** يضم رسالة المستخدم ونتائج الـTools ونصوص الملفات العائدة من البحث فقط.
- لا يصبح حقل `eta` مصدرًا مسموحًا إلا مع `found: True` و `eta_confirmed: True` .
- الـFilter الذي يحجب كل يوم يوقف الردود الأمينة التي تكرّر يوم الـCaller.

- تمييز اليوم غير المسنود بعلامة مع عرض نتيجة الـTool في الأعلى يحفظ التحذيرات المفيدة.
- توحيد الحروف والأرقام واشتراط «ال» أو «يوم» يمنعان الالتباس مع الكميات مثل «اثنين كرتونة».

جرّب في ملفات المعمل (مجلد الدرس في Labs_00): أكمل `verified_eta` و`date_mentions` و`context_dates` و`check_answer` لاجتياز الـTests الـ15 وقياس الردود الـ45 المسجّلة.

الخلاصة: تعليمات الـPrompt تقلّل التواريخ غير المسنودة، لكن الـGrounding Check في الكود هو الذي يتحقّق من مصدر كل يوم في الـContext قبل وصوله إلى مريم.