

PR #53763 完整报告

vllm-project/vllm

[Bugfix] Handle malformed namespace tools

合并时间: 2026-08-28 02:18

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/53763>

执行摘要

- 一句话: 修复 Responses API 畸形 namespace 工具导致 HTTP 500
- 推荐动作: 该 PR 体量小但值得精读, 适合前端 /API 层开发者学习如何在 Pydantic 前置校验器中做防御性类型判断, 并把不可靠的输入交给 schema 校验兜底。建议关注后续是否补充畸形 namespace 工具的回归测试, 避免类似 500 问题回潮。

功能与动机

PR body 明确指出: **Named tool-choice validation for the Responses API inspects namespace tools before Pydantic validation. A malformed nested entry such as null or a string therefore raises AttributeError and returns HTTP 500.** 即客户端传入畸形 namespace 工具时应返回 HTTP 400 明确拒绝, 而实际却因前置校验在 Pydantic 之前崩溃返回 HTTP 500, 违背 API 错误契约, 需要引入与顶层工具相同的防御性名称查找。

实现拆解

1. 定位校验入口: 改动位于 `vllm/entrypoints/openai/responses/protocol.py` 的 `ResponsesRequest.check_tool_usage` 前置校验器, 它在 Pydantic 字段校验之前预检 tools 与 tool_choice 的组合合法性。
2. 非列表 namespace tools 处理: 原逻辑对 `tool.get('tools', [])` 直接遍历, `null/` 字符串会导致 `AttributeError`。修复先取 `namespaced_tools = tool.get('tools')`, 用 `isinstance(namespaced_tools, list)` 判断; 不是列表时直接 `return data`, 把校验交给 Pydantic, 从而得到 `NamespaceTool.tools: Input should be a valid list` 的 HTTP 400 错误。
3. 列表内元素的防御性名称提取: 循环内先判断元素是否为 dict: dict 用 `namespaced_tool.get('name')`, 非 dict 用 `getattr(namespaced_tool, 'name', None)`, 与顶层工具处理方式保持一致, 彻底消除 `AttributeError`。
4. 测试配套: 本 PR 未新增测试文件, 但作者运行了 `tests/tool_use/test_responses_request_validations.py` (21 通过) 以及 `tests/entrypoints/openai/responses` 下三组测试 (43 通过、1 xfailed), pre-commit 检查通过; CI 经多轮触发最终通过。
5. 提交演进: 首个提交为“Handle malformed namespace tools”, 第二个提交改为“Reject non-list namespace tools”, 即针对维护者提问, 将实现收敛为“非列表直接交由 Pydantic 兜底”, 随后三次 merge main 保持同步。


```

# namespace 工具要求内部 tools 是一个列表;
# 若为 null、字符串等畸形值, 直接返回交给 Pydantic 报 400
namespace = tool.get("name")
namespaced_tools = tool.get("tools")
if not isinstance(namespaced_tools, list):
    return data
for namespaced_tool in namespaced_tools:
    # 与顶层工具一致: dict 用 get, 其他对象用 getattr
    namespaced_name = (
        namespaced_tool.get("name")
        if isinstance(namespaced_tool, dict)
        else getattr(namespaced_tool, "name", None)
    )
    tool_names.add(namespaced_name)
    tool_names.add(f"{namespace}__{namespaced_name}")
else:
    tool_names.add(tool.get("name"))
else:
    tool_names.add(getattr(tool, "name", None))
if not tool_name or tool_name not in tool_names:
    raise VLLMValidationError(
        "Tool choice 'function' not found in 'tools' parameter.",
        parameter="tool_choice",
    )

return data

```

评论区精华

维护者 [yewentao256](#) 在 review 中提出关键边界问题: `{"type": "namespace", "name": "weather", "tools": null}` 这类输入能否被正确处理? 作者回复: 修复后会被 HTTP 400 拒绝, Pydantic 报 `NamespaceTool.description: Field required` 和 `NamespaceTool.tools: Input should be a valid list`。该提问直接促成第二个提交, 将实现从“逐一防御”调整为“非列表提前返回、由 schema 校验兜底”。[claude\[bot\]](#) 说明这是 fork PR, 自动审查被禁用; [yewentao256](#) 最终给出 LGTM 并批准合并。

- namespace tools 为 null 时的处理 (question): 作者回复: 修复后这类输入会以 HTTP 400 被拒绝, Pydantic 报 `NamespaceTool.description: Field required` 与 `NamespaceTool.tools: Input should be a valid list`; 该提问促使提交改为对非列表 tools 提前返回、由 schema 校验兜底。

风险与影响

- 风险:
 1. 前置校验绕行风险: `check_tool_usage` 中 `return data` 会跳过 `tool_choice` 与工具名的匹配检查, 完全依赖 Pydantic 对 `NamespaceTool.tools` 的类型约束兜底; 当前 schema 会报 `Input should be a valid list`, 但后续若放宽该字段类型, 可能绕过命名工

具校验。

2. 测试缺口：本次未新增针对畸形 namespace 工具的回归测试，未来重构存在再次引入 HTTP 500 的可能，建议补一条 tools: null 的单元测试。
3. 兼容性：合法请求（包括 f 与 ns__f 命名选择）行为不变；无效输入从 HTTP 500 变为 HTTP 400，属于契约修正，客户端兼容性风险低。
4. 影响面：改动集中在 Responses API 的命名工具选择路径，单文件 9 行新增，无性能与结构影响。
 - 影响：用户面：使用 Responses API 且传入畸形 namespace 工具的客户端将从收到 HTTP 500 变为 HTTP 400 的明确错误信息，便于快速定位；合法命名工具请求完全不受影响。系统面：校验仍在反序列化阶段完成，无额外性能开销。团队面：该改动为后续 Responses 参数校验提供了“前置校验器与 Pydantic schema 分工兜底”的参考模式，但未附带回归测试是值得后续补强的小隐患。
 - 风险标记：非列表 tools 提前返回依赖 schema 兜底，未新增畸形输入回归测试，错误码行为变更（500→400）

关联脉络

- PR #47815 [Bugfix][OpenAI] Fix streamed completion logprob offsets with echo: 同为 OpenAI 入口层的边界输入修复，将异常行为收敛为明确错误语义；不过改动文件不同，本 PR 在 Responses 协议层，该 PR 在 completion serving 层。
- PR #53999 [Bugfix] Raise clear error on interleaved multimodal placeholder overcount: 同为入口层输入校验加固，把无效请求从 500/ 静默错误改为清晰报错，可归入同一错误处理演进脉络。