PR #39780 完整报告

vllm-project/vllm

[Bugfix] Reject empty tools array with HTTP 400

合并时间：2026-04-16 12:08

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/39780

执行摘要

- 一句话：修复聊天完成请求中空工具数组验证，改为返回 HTTP 400 错误以匹配 OpenAI API。
- 推荐动作：该 PR 值得精读，展示了协议兼容性修复的实践，特别是添加类型守卫和早期验证的设计模式，有助于理解 vLLM 前端验证器的演进。

功能与动机

Issue #39741 报告了空工具数组被错误接受的问题，导致与 OpenAI API 行为不一致。PR body 指出根本原因是 data.get("tools") 返回空列表在 Python 中为 falsy，导致验证逻辑被跳过，这是 PR #8568 意外引入的副作用。

实现拆解

1. 修改核心验证逻辑：在 vllm/entrypoints/openai/chat_completion/protocol.py 的 check_tool_usage 方法中，添加早期空数组验证（if data.get("tools") == []:）和类型守卫（处理 ValueError 和非字典输入），匹配文件中的其他验证器模式。
2. 移除冗余回退块：删除同一方法中处理 tool_choice="required" 且工具为空列表的代码块，因为空工具现在会被早期拒绝，该块变得不可达。
3. 更新主测试：修改 tests/tool_use/test_chat_completion_request_validations.py 中的 test_chat_completion_request_with_no_tools 测试，将原本期望空工具被接受的断言改为期望抛出 ValueError，以覆盖新行为。
4. 修复工具解析器测试：在 tests/tool_parsers/test_ernie45_moe_tool_parser.py 和 tests/tool_parsers/test_xlam_tool_parser.py 中，移除测试代码中构造 ChatCompletionRequest 时传入的 tools=[] 参数，因为这些测试不依赖该字段，避免新验证器导致测试失败。

关键文件：

- vllm/entrypoints/openai/chat_completion/protocol.py（模块 前端协议；类别 source；类型 core-logic；符号 check_tool_usage）：核心验证逻辑变更文件，修复了空工具数组验证的主要 bug。
- tests/tool_use/test_chat_completion_request_validations.py（模块 请求验证；类别 test；类型 test-coverage；符号 test_chat_completion_request_with_no_tools）：主测试文件，更新以覆盖空工具数组被拒绝的新行为。

- tests/tool_parsers/test_ernie45_moe_tool_parser.py（模块 工具解析；类别 test；类型 test-coverage）：工具解析器测试文件，移除不必要的 tools=[] 参数以避免新验证器导致测试失败。
- tests/tool_parsers/test_xlam_tool_parser.py（模块 工具解析；类别 test；类型 test-coverage）：工具解析器测试文件，移除不必要的 tools=[] 参数以避免新验证器导致测试失败。

关键符号：check_tool_usage

关键源码片段

vllm/entrypoints/openai/chat_completion/protocol.py

核心验证逻辑变更文件，修复了空工具数组验证的主要 bug。

```
@model_validator(mode="before")
@classmethod
def check_tool_usage(cls, data):
    # 添加类型守卫：如果data是ValueError实例，直接抛出，处理前一个验证器返回的错误
    if isinstance(data, ValueError):
        raise data
    # 如果data不是字典类型，直接返回，避免后续处理错误
    if not isinstance(data, dict):
        return data

    # 拒绝空工具数组，匹配OpenAI API行为，确保tools字段要么不提供，要么至少有一个工具
    if data.get("tools") == []:
        raise ValueError(
            "`tools` must not be an empty array. "
            "Either provide at least one tool or omit the field entirely."
        )

    # 后续逻辑保持不变：默认tool_choice、验证tool_choice与tools的匹配等
    if "tool_choice" not in data and data.get("tools"):
        data["tool_choice"] = "auto"
    # ... 其余验证步骤
```

评论区精华

gemini-code-assist[bot] 建议添加类型检查以提高验证器健壮性，匹配其他验证器（如 check_structured_outputs_count），这被采纳并实现。DarkLight1337 询问 OpenAI 行为是否改变，jigangz 回应称可能是 OpenAI 更新了行为，原代码是历史工作 around，最终确认变更以匹配当前 OpenAI API。

- 验证器健壮性改进 (correctness): 采纳建议，添加了类型守卫代码（if isinstance(data, ValueError): raise data; if not isinstance(data, dict): return data）。
- OpenAI 行为变化讨论 (design): 确认变更以匹配当前 OpenAI API 行为，移除冗余回退块，添加空数组拒绝。

风险与影响

- 风险：主要风险是 API 行为变更可能影响依赖空工具数组被静默接受的现有客户端，但测试已更新覆盖新行为，且变更仅限于验证逻辑，不影响核心推理路径或性能。此外，添加类型守卫减少了潜在的类型错误风险。
- 影响：用户侧，传入空工具数组的请求现在会收到 HTTP 400 错误和明确错误消息，提升与 OpenAI API 的兼容性和用户体验。系统侧，修复了协议层的一个不一致性问题，增强了前端验证的健壮性。
- 风险标记：API 行为变更，测试覆盖更新

关联脉络

- PR #39217 [Mistral Grammar] Fix tool and reasoning parsing: 同样涉及 tool-calling 和前端验证，展示了工具解析相关修复的延续。