PR #51654 完整报告

vllm-project/vllm

Fix chat completion 500 on non-object JSON bodies

合并时间：2026-08-11 09:31

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/51654

执行摘要

- 一句话：修复 chat completion 非对象 JSON body 返回 500 的问题
- 推荐动作：该 PR 值得快速阅读，适合作为了解 vLLM OpenAI 前端校验器模式的入门案例。它展示了 mode="before" 校验器在面对非 dict 输入时的通用陷阱，以及如何用 isinstance(data, dict) 守卫与既有代码保持一致的解决方案。对于维护 API 兼容层的开发者有借鉴意义，但不涉及复杂架构设计。

功能与动机

PR body 明确指出："While performing contract (negative) testing, it was observed that the server returns an HTTP 500 Internal Server Error instead of an appropriate 4xx client error for invalid request payloads." 同时给出了具体错误堆栈：AttributeError: 'str' object has no attribute 'get'，发生在 check_cache_salt_support 等校验器调用 data.get() 时，导致内部错误被当作 500 返回。

实现拆解

1. 识别缺陷：在 vllm/entrypoints/openai/chat_completion/protocol.py 中，ChatCompletionRequest 和 BatchChatCompletionRequest 的多个 mode="before" 校验器（如 validate_response_format、validate_stream_options、check_logprobs、check_structured_outputs_count、check_generation_prompt、check_batch_mode）直接对 data 调用 .get()，但 data 可能是 JSON 解析后的字符串、列表、数字或 null，导致 AttributeError。
2. 添加守卫：在这些校验器开头统一添加 if not isinstance(data, dict): return data，仿照已有 check_system_message_content_type、check_tool_usage 的写法。对非 dict 输入直接放行，交给 Pydantic 后续字段校验，从而产生 ValidationError（HTTP 400）。
3. 加固 batch 路径：BatchChatCompletionRequest.check_batch_mode 在 response_format 为 None 时也会报错，将该访问封装在 if response_format is not None 中，并保持 rf_type 对 dict 和 dataclass 的兼容处理。
4. 新增测试：在 tests/entrypoints/openai/chat_completion/test_non_object_body_validation.py 中，参数化覆盖字符串、列表、整型和 null 四种非对象 body，断言应抛 ValidationError；同时验证 dict body 下空 cache_salt 仍会正常触发校验，防止守卫误伤合法校验。

5. 回归验证：PR 描述中通过 curl 向运行中的服务器发送非法 JSON 字符串，确认响应从 500 变为 400。

关键文件：

- vllm/entrypoints/openai/chat_completion/protocol.py（模块 请求校验；类别 source；类型 core-logic；符号 validate_response_format, validate_stream_options, check_logprobs, check_structured_outputs_count）：核心修复文件，为 6 个 mode="before" 校验器添加 dict 守卫，并修复 batch 模式中 response_format 为 None 的情况。
- tests/entrypoints/openai/chat_completion/test_non_object_body_validation.py（模块 回归测试；类别 test；类型 test-coverage；符号 test_chat_completion_request_rejects_non_object_body, test_chat_completion_request_cache_salt_still_validated_on_dict）：新增测试，覆盖非对象 body 的回归场景和 dict 下 cache_salt 校验不被误伤。

关键符号：validate_response_format, validate_stream_options, check_logprobs, check_structured_outputs_count, check_generation_prompt, check_batch_mode, test_chat_completion_request_rejects_non_object_body, test_chat_completion_request_cache_salt_still_validated_on_dict

关键源码片段

vllm/entrypoints/openai/chat_completion/protocol.py

核心修复文件，为 6 个 mode="before" 校验器添加 dict 守卫，并修复 batch 模式中 response_format 为 None 的情况。

```
# 来自 vllm/entrypoints/openai/chat_completion/protocol.py
# BatchChatCompletionRequest 的 mode="before" 校验器，
# 在解析非 dict body 时会先被 dict 守卫拦截，确保后续 .get() 安全。
@model_validator(mode="before")
@classmethod
def check_batch_mode(cls, data: Any) -> Any:
    # 若传入的是 BatchChatCompletionRequest 实例，先转成原始 dict 再校验
    if isinstance(data, BatchChatCompletionRequest):
        data = data.model_dump(exclude_unset=True)

    # 非 dict（如 JSON 字符串、数组、数字、null）直接放行，
    # 由 Pydantic 后续字段解析抛出 ValidationError，客户端得到 4xx。
    if not isinstance(data, dict):
        return data

    if data.get("use_beam_search"):
        raise VLLMValidationError(
            "Batch chat completions do not support beam search. "
            "Please set `use_beam_search` to False.",
            parameter="use_beam_search",
        )
    if data.get("logprob_token_ids") and not data.get("logprobs"):
```

```
            raise VLLMValidationError(
                "when using `logprob_token_ids`, `logprobs` must be set to true.",
                parameter="logprob_token_ids",
            )

        response_format = data.get("response_format")
        # 新增 None 检查：此前 response_format 为 None 时 .get() 会 AttributeError
        if response_format is not None:
            rf_type = (
                response_format.get("type")
                if isinstance(response_format, dict)
                else getattr(response_format, "type", None)
            )
            if rf_type == "structural_tag":
                validate_structural_tag_response_format(response_format)

        if (structured_outputs := data.get("structured_outputs")) is not None:
            validate_structured_outputs_structural_tag(structured_outputs)

        n = data.get("n", 1)
        if n is not None and n != 1:
            raise VLLMValidationError(
                "Batch chat completions do not support `n > 1`. Please set `n` to 1.",
                parameter="n",
                value=n,
            )
        return data
```

tests/entrypoints/openai/chat_completion/test_non_object_body_validation.py

新增测试，覆盖非对象 body 的回归场景和 dict 下 cache_salt 校验不被误伤。

```
# tests/entrypoints/openai/chat_completion/test_non_object_body_validation.py
# 验证非对象 JSON body 应返回 4xx 而非 500，
# 同时确保 dict body 的既有校验逻辑不受守卫影响。
"""Non-object JSON bodies must fail validation cleanly (4xx), not AttributeError (500)."""

import pytest
from pydantic import ValidationError

from vllm.entrypoints.openai.chat_completion.protocol import ChatCompletionRequest


@pytest.mark.parametrize(
    "payload",
    [
        "this is not valid json{{", # 字符串
        ["not", "an", "object"], # 数组
        42, # 数字
```

```
        None, # null
    ],
)
def test_chat_completion_request_rejects_non_object_body(payload):
    # 非 dict 输入应被守卫放行，最终由 Pydantic 字段校验抛出 ValidationError
    with pytest.raises(ValidationError):
        ChatCompletionRequest.model_validate(payload)


def test_chat_completion_request_cache_salt_still_validated_on_dict():
    # dict body 仍应正常执行 cache_salt 有效性校验，守卫不能误伤
    with pytest.raises(ValidationError, match="cache_salt"):
        ChatCompletionRequest.model_validate(
            {
                "model": "qwen",
                "messages": [{"role": "user", "content": "hello"}],
                "cache_salt": "",
            }
        )
```

评论区精华

本次 review 几乎没有实质技术讨论。维护者 DarkLight1337 直接批准（"Thanks"）。仅有的插曲是 mergify bot 提示 pre-commit 检查失败，要求作者运行 pre-commit 修复格式，其后作者重新触发 CI 并通过。整体方案简单直接，无设计分歧。

- pre-commit 检查失败 (style): 作者在后续提交中修复（合并时 CI 已通过），无实质设计分歧。
- fork 自动化 review 限制 (other): 维护者 DarkLight1337 批准（"Thanks"），无需进一步讨论。

风险与影响

- 风险：改动局限于 protocol.py 中 6 个校验器的前置守卫，风险极低。主要风险点：
  - 若将来有调用方依赖这些校验器对非 dict 输入抛 AttributeError（几乎不可能），行为会变更为 Pydantic 的 ValidationError，但这是期望修复。
  - check_batch_mode 对 response_format 为 None 的额外处理是独立修复，不会影响原有 dict 场景。
  - 新增测试覆盖了主要非对象类型，但未覆盖如 bool、浮点数等，不过字符串 / 列表 / 整型 /null 已代表主要类别。
  - 无性能影响，因为 isinstance 检查是 O(1) 且仅在请求校验路径。
  - 影响：影响所有使用 OpenAI 兼容 chat/completions 端点的用户：此前发送非对象 JSON body 会得到 500 和混乱的内部错误日志，现在会得到标准的 4xx 客户端错误，便于 API 集成方定位问题。对正常请求无影响。同时也统一了 BatchChatCompletionRequest 与普通请求的错误处理行为。团队维护成本低，代码风格与现有守卫一致。

- 风险标记：仅影响无效请求路径，新增测试覆盖边界输入

关联脉络

- 暂无明显关联 PR