

PR #44686 完整报告

vllm-project/vllm

Fix Harmony tool descriptions for optional fields

合并时间: 2026-06-10 14:29

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/44686>

执行摘要

- 一句话: 修复 Harmony 工具描述为空字段时的 pydantic 错误
- 推荐动作: 建议快速合入。这是明显的兼容性问题, 修复简单且测试覆盖充分。值得关注的是 `create_tool_definition` 函数的后续演进, 可能需要对其他字段做类似的防御性处理。

功能与动机

OpenAI 兼容的 Chat Completions 和 Responses tool schema 允许 function tool 的 `description` 字段省略或为 `null`, 但 `openai_harmony.ToolDescription` 需要字符串。这导致当用户在 `tools` 参数中不提供 `description` 时, vLLM 返回 400 Bad Request。

实现拆解

1. 修改核心转换函数: 在 `vllm/entrypoints/openai/parser/harmony_utils.py` 的 `create_tool_definition` 函数中, 将 `ToolDescription.new()` 的 `description` 参数从直接传入字段值改为 `tool.function.description or ""` (ChatCompletion) 和 `tool.description or ""` (Responses)。
2. 新增测试类: 在 `tests/entrypoints/openai/parser/test_harmony_utils.py` 中新增 `TestCreateToolDefinition` 类, 包含三个测试用例: `test_chat_completion_omitted_description_defaults_to_empty_string`、`test_chat_completion_none_description_defaults_to_empty_string`、`test_response_tool_none_description_defaults_to_empty_string`。
3. 调整导入: 在测试文件中增加了 `from openai.types.responses import FunctionTool` 和 `from vllm.entrypoints.openai.chat_completion.protocol import ChatCompletionToolsParam` 以支持新测试用例。

关键文件:

- `vllm/entrypoints/openai/parser/harmony_utils.py` (模块 前端解析; 类别 `source`; 类型 `core-logic`; 符号 `create_tool_definition`): 核心变更文件, 修改了 `create_tool_definition` 函数, 使 `description` 字段在缺失或为 `None` 时默认转为空字符串。
- `tests/entrypoints/openai/parser/test_harmony_utils.py` (模块 测试; 类别 `test`; 类型 `test-coverage`; 符号 `TestCreateToolDefinition`, `test_chat_completion_omitted_description_defaults_to_empty_string`, `test_chat_completion_none_description_defaults_to_empty_string`, `test_response_tool_none_description_defaults_to_empty_string`): 新增测试类

TestCreateToolDefinition, 覆盖了三种缺失 / 空 description 的场景, 确保修复正确且不回归。

关键符号: create_tool_definition

关键源码片段

vllm/entrypoints/openai/parser/harmony_utils.py

核心变更文件, 修改了 create_tool_definition 函数, 使 description 字段在缺失或为 None 时默认转为空字符串。

```
# vllm/entrypoints/openai/parser/harmony_utils.py

def create_tool_definition(tool: ChatCompletionToolsParam | Tool):
    """将 OpenAI 兼容的 tool 定义转换为 Harmony 的 ToolDescription。

    OpenAI 的 Chat Completions 和 Responses API 允许 description 字段
    省略或为 None, 但 Harmony 的 ToolDescription 要求字符串类型, 因此
    在实例化时使用 `or ""` 做防御性处理。
    """
    if isinstance(tool, ChatCompletionToolsParam):
        return ToolDescription.new(
            name=tool.function.name,
            description=tool.function.description or "", # 修复点
            parameters=tool.function.parameters,
        )
    return ToolDescription.new(
        name=tool.name,
        description=tool.description or "", # 修复点
        parameters=tool.parameters,
    )
```

tests/entrypoints/openai/parser/test_harmony_utils.py

新增测试类 TestCreateToolDefinition, 覆盖了三种缺失 / 空 description 的场景, 确保修复正确且不回归。

```
# tests/entrypoints/openai/parser/test_harmony_utils.py

import pytest
from openai.types.responses import FunctionTool
from openai_harmony import DeveloperContent, Message, Role

from tests.entrypoints.openai.utils import verify_harmony_messages
from vllm.entrypoints.openai.chat_completion.protocol import ChatCompletionToolsParam
from vllm.entrypoints.openai.parser.harmony_utils import (
    auto_drop_analysis_messages,
    create_tool_definition, # 新增导入
    extract_function_from_recipient,
    get_encoding,
```

```

    get_system_message,
    has_custom_tools,
    is_function_recipient,
    parse_chat_input_to_harmony_message,
    parse_chat_output,
)
from vllm.entrypoints.openai.responses.harmony import (
    response_input_to_harmony,
    response_previous_input_to_harmony,
)

```

定义一个通用的参数字典，供所有测试用例复用

```

_TOOL_PARAMETERS = {
    "type": "object",
    "properties": {"status": {"type": "string"}},
    "required": ["status"],
    "additionalProperties": False,
}

```

class TestCreateToolDefinition:

测试 Chat Completion 中 description 字段完全省略

def test_chat_completion_omitted_description_defaults_to_empty_string(self):

```

    tool = ChatCompletionToolsParam(
        function={
            "name": "report_status",
            "parameters": _TOOL_PARAMETERS,
        }
    )

```

tool_definition = create_tool_definition(tool)

assert tool_definition.name == "report_status"

assert tool_definition.description == ""

assert tool_definition.parameters == _TOOL_PARAMETERS

测试 Chat Completion 中 description 字段显式设为 None

def test_chat_completion_none_description_defaults_to_empty_string(self):

```

    tool = ChatCompletionToolsParam(
        function={
            "name": "report_status",
            "description": None,
            "parameters": _TOOL_PARAMETERS,
        }
    )

```

tool_definition = create_tool_definition(tool)

assert tool_definition.name == "report_status"

```
assert tool_definition.description == ""
assert tool_definition.parameters == _TOOL_PARAMETERS

# 测试 Responses API 中 description 字段显式设为 None
def test_response_tool_none_description_defaults_to_empty_string(self):
    tool = FunctionTool(
        name="report_status",
        description=None,
        parameters=_TOOL_PARAMETERS,
        type="function",
    )

    tool_definition = create_tool_definition(tool)

    assert tool_definition.name == "report_status"
    assert tool_definition.description == ""
    assert tool_definition.parameters == _TOOL_PARAMETERS
```

评论区精华

讨论集中在是否需要修复以及如何修复。作者 shenoyvvarun 提供了详细的错误复现信息（pydantic 校验错误），证实了用户会收到 400 错误。作者提出了两条路径：修复为默认空字符串，或者返回 400 错误给客户。最终选择了向下兼容的修复方案。

- 暂无高价值评论线程

风险与影响

- 风险：风险极低。仅对 create_tool_definition 函数中 description 参数做了防御性处理（or ""），不会影响已有显式提供 description 的场景，因为非空字符串不受影响。
- 影响：
 - 用户：修复了使用 Harmony 解析路径时，不传或传 null description 导致的 400 错误。
 - 系统：无性能影响，变更仅 2 行逻辑。
 - 团队：增强了 vLLM 与 OpenAI API 兼容性，降低了用户接入门槛。
 - 风险标记：无显著风险

关联脉络

- 暂无明显关联 PR