

PR #46486 完整报告

vllm-project/vllm

[Bugfix][Tool Parser] PoolsideV1: fix string whitespace and required named tool choice

合并时间: 2026-06-26 14:05

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/46486>

执行摘要

- 一句话: 修复 PoolsideV1 工具解析器的空白处理和工具选择冲突
- 推荐动作: 值得精读——该 PR 展示了在工具解析器中处理两种不同 API 协议下的工具形状差异（嵌套 vs 扁平）的兼容模式，以及如何通过 early return 避免 JSON 引导与原生模板的冲突。测试设计也值得参考（分别覆盖 ChatCompletion 和 Responses 的 required/named 路径）。建议关注 #39870 类似的跨 API 兼容性修复模式，并留意未来迁移到统一 parser engine 的演进方向。

功能与动机

PR body 说明“handle the XML tags properly even if JSON is requested, handle whitespace more intelligently”。测试注释具体指出两个 bug: required/named 工具选择时 JSON 引导与 XML 模板冲突，以及字符串参数空白被破坏。这是内部修复的上游贡献。

实现拆解

1. 添加导入及类属性: 在 `vllm/tool_parsers/poolside_v1_tool_parser.py` 中导入 `ChatCompletionNamedToolChoiceParam` 和 `ToolChoiceFunction`; 在类上设置 `supports_required_and_named = False`。
2. 修改 `adjust_request`: 当 `request.tools` 存在且 `tool_choice` 为 `required` 或 `ChatCompletionNamedToolChoiceParam/ToolChoiceFunction` 时，直接设置 `skip_special_tokens=False` 并返回，跳过基类的 JSON 引导解码。
3. 修改 `_is_string_type`: 使用 `getattr(tool, 'function', tool)` 统一处理 `ChatCompletion` 嵌套工具和 `Responses` 扁平工具，避免 `AttributeError`。
4. 修改 `extract_tool_calls`: 对于字符串类型参数保持原值（不 `strip`），仅对非字符串类型执行 `strip` 和反序列化。
5. 新增测试文件 `tests/tool_parsers/test_poolside_v1_tool_parser.py`，覆盖 `required/named` 跳过结构化输出（`ChatCompletion` 和 `Responses`）、字符串空白保留、以及扁平工具形状的提取场景。

关键文件:

- `vllm/tool_parsers/poolside_v1_tool_parser.py` (模块 工具解析; 类别 `source`; 类型 `core-logic`; 符号 `PoolsideV1ToolParser`, `supports_required_and_named`, `adjust_request`, `extract_tool_calls`): 核心源文件, 修复了 `adjust_request`、

extract_tool_calls 和 _is_string_type 三个关键方法，兼容 ChatCompletion 和 Responses 两种工具形状。

- tests/tool_parsers/test_poolside_v1_tool_parser.py (模块 工具解析; 类别 test; 类型 test-coverage; 符号 _write_file_tool, _responses_write_file_tool, _build_chat_request, _build_responses_request) : 新增测试文件, 覆盖两个 bug (结构化输出跳过、字符串空白保留) 以及 Responses 扁平工具形状, 确保修复正确性。

关键符号: adjust_request, extract_tool_calls, _is_string_type

关键源码片段

vllm/tool_parsers/poolside_v1_tool_parser.py

核心源文件, 修复了 adjust_request、extract_tool_calls 和 _is_string_type 三个关键方法, 兼容 ChatCompletion 和 Responses 两种工具形状。

```
# _is_string_type 方法兼容 ChatCompletion 与 Responses 工具形状
@staticmethod
def _is_string_type(
    tool_name: str,
    arg_name: str,
    tools: list[Tool] | None,
) -> bool:
    if tools is None:
        return False
    for tool in tools:
        # ChatCompletion 工具嵌套在 .function 下,
        # Responses FunctionTool 扁平 (.name 在顶层)
        fn = getattr(tool, "function", tool)
        if getattr(fn, "name", None) != tool_name:
            continue
        params = getattr(fn, "parameters", None)
        if params is None:
            return False
        arg_type = params.get("properties", {}).get(arg_name, {}).get("type", None)
        return arg_type == "string"
    return False

# adjust_request 方法: required/named 时跳过 JSON 引导
def adjust_request(
    self,
    request: ChatCompletionRequest | ResponsesRequest
) -> ChatCompletionRequest | ResponsesRequest:
    """
    当 tool_choice 为 required 或命名时, 直接返回,
    避免 super().adjust_request 安装 JSON 引导解码。
    这些模型使用 XML 模板, JSON 引导会破坏输出。
    """
    if request.tools:
```

```

    tc = request.tool_choice
    if tc == "required" or isinstance(
        tc, (ChatCompletionNamedToolChoiceParam, ToolChoiceFunction)
    ):
        request.skip_special_tokens = False
        return request
    request = super().adjust_request(request)
    if request.tools and request.tool_choice != "none":
        # 原有逻辑：确保工具调用 token 不被跳过
        ... # 未变更部分省略
    return request

# extract_tool_calls 中字符串处理的关键循环
for key, value in pairs:
    arg_key = key.strip()
    # 字符串参数保持原文，空白有意义（如代码 / 文件内容）
    if self._is_string_type(tc_name, arg_key, request.tools):
        arg_val = value
    else:
        arg_val = self._deserialize(value.strip())
    arg_dct[arg_key] = arg_val

```

tests/tool_parsers/test_poolside_v1_tool_parser.py

新增测试文件，覆盖两个 bug（结构化输出跳过、字符串空白保留）以及 Responses 扁平工具形状，确保修复正确性。

构建 ChatCompletion 测试请求的辅助函数

```

def _write_file_tool() -> dict[str, Any]:
    return {
        "type": "function",
        "function": {
            "name": "write_file",
            "description": "Write content to a file",
            "parameters": {
                "type": "object",
                "properties": {
                    "content": {"type": "string"},
                    "mode": {"type": "integer"},
                },
                "required": ["content"],
            },
        },
    }

```

```

def _make_parser(request: ChatCompletionRequest) -> PoolsideV1ToolParser:
    return PoolsideV1ToolParser(_StubTokenizer(), tools=request.tools)

```

测试 required 时跳过结构化输出

```

def test_required_skips_structured_outputs_chatcompletion() -> None:

```

```
request = _build_chat_request(tool_choice="required")
_make_parser(request).adjust_request(request)
```

```
assert request.structured_outputs is None
assert request.skip_special_tokens is False
```

评论区精华

sfeng33 在 review 中指出: `_is_string_type` 直接访问 `tool.function.name` 只适用于 ChatCompletion 工具, Responses API 的 `FunctionTool` 是扁平结构 (`.name` 在顶层), 会导致 `AttributeError`。joerowell 回复已修复, 使用 `getattr(tool, 'function', tool)` 兼容两种形状, 并添加了对应的测试。此外, sfeng33 非阻塞地建议将 Poolside 迁移到新的 parser engine (类似 GLM), 以便受益于增量流式解析和 schema 感知类型转换。

- Responses 工具形状兼容性 (correctness): 已修复: `_is_string_type` 使用 `getattr(tool, 'function', tool)` 统一处理嵌套和扁平工具; 新增 `test_responses_extract_tool_calls_with_flat_tools` 覆盖。

风险与影响

- 风险:
 1. 工具形状兼容性: `_is_string_type` 的 `getattr` 修复假设所有工具对象都有 `.function` 或扁平 `.name`, 若未来出现其他形状可能仍需调整。
 2. 结构化输出跳过逻辑: `adjust_request` 的 `early return` 只在 `request.tools` 非空时触发, 若 `tools` 为空但设置了 `tool_choice` 可能仍走旧路径 (但这种情况被调用方排除)。
 3. 测试覆盖: 新增测试覆盖了 ChatCompletion 和 Responses 的 `required/named` 场景, 但未覆盖 `auto` 或 `none` 等边角情况 (但这些路径未改动)。
 4. 与旧行为的兼容性: 原来 `supports_required_and_named` 默认 `True`, 现设为 `False`, 可能影响依赖此属性的其他调用方 (但该属性仅用于工具解析器选择器)。- 影响: 直接影响使用 PoolsideV1 模型进行工具调用的用户, 修复了字符串参数空白丢失和 `required/named` 工具选择时 JSON 引导破坏输出的问题。ChatCompletion 和 Responses API 均受益。影响范围限于单一模型解析器, 代码变更量小 (+247/-11), 回归风险可控。- 风险标记: Responses 兼容性风险, 结构化输出跳过逻辑风险

关联脉络

- 暂无明显关联 PR