

PR #45346 完整报告

vllm-project/vllm

[Bugfix][Structured Outputs] Reject degenerate `structured\_outputs` that crash EngineCore

合并时间: 2026-07-01 06:41

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/45346>

执行摘要

- 一句话: 拒绝导致 EngineCore 崩溃的退化 structured_outputs
- 推荐动作: 值得精读。该 PR 展示了如何通过前端请求验证快速失败来防止引擎级 DoS, 是安全修复的典范。设计决策清晰: 将验证集中在 `_validate_structured_outputs` 中, 与现有空语法、空 choice 守卫并列, 便于后续扩展。review 中关于空字典的讨论澄清了当前验证范围的边界, 体现了审慎的设计态度。

功能与动机

单个带 `structured_outputs={"json_object": false}` 或 `{"json": ""}` 的请求会触发 EngineCore 内部异常, 因为 `StructuredOutputsParams` 的 `is not None` 检查将它们视为有效约束, 但实际无法映射到结构化输出键 (`get_structured_output_key` 对 `False` 无匹配, 空 `json` 导致 `xgrammar` 编译器异常)。这些 `ValueError` 在引擎 step 循环中无请求隔离, 导致引擎死亡并拒绝所有后续请求 (DoS)。需要在前端快速拒绝这些退化输入。

实现拆解

修复分为三步:

1. 在请求验证中添加两个守卫: 在 `vllm/sampling_params.py` 的 `_validate_structured_outputs` 方法中, 在现有空语法检查之后新增两个判断:
 - 如果 `json` 是字符串且去空格后为空, 则 `raise ValueError`。
 - 如果 `json_object` is `False`, 则 `raise ValueError` (因为 `json_object` 是标志, 只有 `True` 才对应一种约束; 省略 `structured_outputs` 即可关闭结构化输出)。由于这些检查在 `verify()` 中调用, 前端会返回 400, 引擎不受影响。
2. 更新测试文件: 将测试从 `tests/v1/structured_output/test_utils.py` 移到 `tests/v1/structured_output/test_validation.py`, 以适应 `_validate_structured_outputs` 新增的 `model_config` 参数 (来自关联 PR #45436)。使用参数化测试覆盖两个退化场景, 复用已有的 `_StubModelConfig` 辅助类和 `cpu_test` 标记。
3. 确保测试在 CI 中运行: 新测试位于 `v1/structured_output` 目录下, 该目录在 `buildkite/test_areas/misc.yaml` 的 CPU job 中被收集, 与已存在的扩散模型验证测试保持一致。

关键文件:

- `vllm/sampling_params.py` (模块 采样参数; 类别 `source`; 类型 `core-logic`) : 结构化输出验证的核心逻辑所在。在此扩展 `_validate_structured_outputs` 方法, 新增两个守卫以在请求到达引擎之前拒绝退化输入, 是本次修复的主干变更。
- `tests/v1/structured_output/test_validation.py` (模块 验证; 类别 `test`; 类型 `test-coverage`; 符号 `test_degenerate_structured_outputs_rejected`) : 添加了参数化测试 `test_degenerate_structured_outputs_rejected`, 覆盖 `json_object=False` 和 `json=""` 两个退化场景, 确保验证逻辑在单元测试层面得到覆盖。测试使用 `_StubModelConfig` 和新的方法签名, 与扩散模型测试共存。

关键符号: `_validate_structured_outputs`, `test_degenerate_structured_outputs_rejected`

关键源码片段

`vllm/sampling_params.py`

结构化输出验证的核心逻辑所在。在此扩展 `_validate_structured_outputs` 方法, 新增两个守卫以在请求到达引擎之前拒绝退化输入, 是本次修复的主干变更。

```
# vllm/sampling_params.py

# Reject empty string json schema early to avoid engine-side crashes
if (
    isinstance(self.structured_outputs.json, str)
    and self.structured_outputs.json.strip() == ""
):
    raise ValueError("structured_outputs.json cannot be an empty string")

# Reject json_object=False early to avoid engine-side crashes
if self.structured_outputs.json_object is False:
    raise ValueError(
        "structured_outputs.json_object must be True if set; omit "
        "structured_outputs to disable structured outputs"
    )
```

评论区精华

- arpera 提问: 空字典 `{}` 是否也应拒绝? Sunt-ing 回复: `json={}` 是合法的“任意 JSON 值”模式, `get_structured_output_key` 会将其序列化为 `"{}"`, `xgrammar` 能正常编译, 不会崩溃, 因此不应拒绝。修正只针对空字符串。
- vadiklyutiy 建议: 应同步处理 `regex=""` 和 `structural_tag=""` 的类似退化。Sunt-ing 验证后指出: `regex=""` 会匹配空字符串, 不会崩溃; `structural_tag=""` 已在前端被拒绝 (`protocol.py`), 均不属于同一种 DoS 类型。但可考虑在后续 PR 中添加对称性验证以提高一致性。
- arpera 要求同步 main 以修复 CI 测试断裂 (由 #47015 修复)。Sunt-ing 执行了合并。
- 测试位置调整: 由于 #45436 改变了 `_validate_structured_outputs` 签名, 测试从 `test_utils.py` 迁移到 `test_validation.py`, 并更新使用新的参数和 `_StubModelConfig`。
 - 空字典 `{}` 是否也应拒绝? (question): 不需要拒绝空字典。

- 其他退化输入的对称处理 (design): 不阻塞此 PR, 同意后续跟进。
- 测试位置与签名适配 (testing): 测试迁移并适配成功。
- CI 断裂修复 (other): CI 恢复正常。

风险与影响

- 风险:
 - 回归风险低: 新增检查仅在输入为退化值时触发, 不影响正常请求。空字典和 `json_object=True` 等合法输入不受影响。讨论中 `arpera` 和 `Sunt-ing` 已确认空字典是有效的。
 - 性能影响无: 检查是简单的类型和值判断, 开销可忽略。
 - 安全风险消除: 修复了可通过单个请求导致实例级 DoS 的漏洞, 增强了引擎稳定性。
 - 可能遗漏类似退化: 如 `regex=""` 当前不崩溃但可能被其他后端误处理。但现有修复已覆盖入口代码, 其他退化可后续补充。
- 影响:
 - 用户影响: 所有使用结构化输出功能的用户受益。之前发送退化请求会导致 500 错误和引擎死亡, 现在得到明确的 400 错误提示, 引擎可持续服务。
 - 系统影响: 消除了因错误请求导致整个推理实例不可用的风险, 提高了系统的鲁棒性。
 - 团队影响: 修复范围小、风险低, 易于审查和合并。提交历史显示经过多次合并 `main` 以保持同步, 反映出对 CI 质量的重视。
 - 风险标记: DoS 防护, 请求验证增强

关联脉络

- PR #45436 [Bugfix] Structured outputs: pass model_config to `_validate_structured_outputs`: 改变了 `_validate_structured_outputs` 方法签名 (新增 `model_config` 参数), 本 PR 的测试需要据此调整签名及辅助类。