

PR #26433 完整报告

sgl-project/sglang

fix(tool_call): reland schema type normalization

合并时间: 2026-05-28 14:31

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/26433>

执行摘要

- 一句话: 恢复并改进工具参数 JSON Schema 类型归一化
- 推荐动作: 此 PR 解决了工具模式验证中的一个实际兼容性问题, 值得合并。建议关注后续可能出现的边缘类型处理请求。

功能与动机

工具参数模式从数据库 /ORM/ 本机工具导出时, 常使用非标准 JSON Schema 类型名称 (如 `varchar`、`enum`、`int32`)。若不归一化, `Draft202012Validator.check_schema` 会在请求到达模型之前拒绝这些模式, 导致合法的工具体义无法使用。此 PR 恢复并改进了之前被还原的类型归一化逻辑。

实现拆解

1. 定义标准类型集与别名映射 (`python/sglang/srt/function_call/utils.py`): 新建 `_STANDARD_JSON_SCHEMA_TYPES` 和 `_JSON_SCHEMA_TYPE_ALIASES` 字典, 将 DB/ORM 常用类型 (如 `varchar`、`enum`) 映射到 JSON Schema 标准类型。
2. 实现前缀匹配规则 (`python/sglang/srt/function_call/utils.py`): 定义 `_PREFIX_RULES` 元组, 利用 `_matches_type_prefix` 函数精确匹配参数化类型 (如 `int32`、`list[str]`), 通过 `_PREFIX_BOUNDARY_CHARS` 确保前缀边界, 防止误匹配 (如 `int` 不匹配 `internal`)。
3. 归一化单类型与类型列表 (`python/sglang/srt/function_call/utils.py`): `_normalize_single_type` 处理字符串类型, `_normalize_type_list` 处理数组类型 (包括去重)。 `normalize_json_schema_types` 递归遍历 Schema 结构, 重写所有 `type` 字段。
4. 集成到请求验证 (`python/sglang/srt/entrypoints/openai/serving_chat.py`): 在 `_validate_request` 方法中, 于 `Draft202012Validator.check_schema` 前调用 `normalize_json_schema_types`, 并捕获 `RecursionError` 防止循环 Schema 导致服务器崩溃。
5. 添加单元测试 (`test/registered/unit/function_call/test_normalize_json_schema_types.py`): 验证常见别名、参数化类型、前缀边界、递归覆盖及去重逻辑, 共 23 个测试用例, 通过 `register_cpu_ci` 注册到 CI 套件。

关键文件:

- `python/sglang/srt/function_call/utils.py` (模块 工具调用; 类别 `source`; 类型 `core-logic`; 符号 `_matches_type_prefix`, `_normalize_single_type`, `_normalize_type_list`,

normalize_json_schema_types) : 核心实现文件, 定义标准类型、别名映射、前缀匹配规则及递归归一化逻辑。

- test/registered/unit/function_call/test_normalize_json_schema_types.py (模块 测试; 类别 test; 类型 test-coverage; 符号 TestNormalizeJsonSchemaTypes, _assert_accepts, test_enum_alias_becomes_string, test_varchar_alias_becomes_string) : 新增完整单元测试, 覆盖常见别名、参数化类型、前缀边界、递归及去重场景。
- python/sclang/srt/entrypoints/openai/serving_chat.py (模块 请求入口; 类别 source; 类型 dependency-wiring) : 集成点, 在请求验证前调用归一化函数, 并捕获循环引用异常。

关键符号: _matches_type_prefix, _normalize_single_type, _normalize_type_list, normalize_json_schema_types

关键源码片段

python/sclang/srt/entrypoints/openai/serving_chat.py

集成点, 在请求验证前调用归一化函数, 并捕获循环引用异常。

```
# python/sclang/srt/entrypoints/openai/serving_chat.py (_validate_request 部分)
# 导入新增 normalize_json_schema_types
from sclang.srt.function_call.utils import (
    get_json_schema_constraint,
    normalize_json_schema_types, # 新增
)

# 在工具验证循环内
for i, tool in enumerate(request.tools or []):
    if tool.function.parameters is None:
        continue
    try:
        # 先归一化非标准类型, 再标准验证
        normalize_json_schema_types(tool.function.parameters)
        Draft202012Validator.check_schema(tool.function.parameters)
    except SchemaError as e:
        return f"Tool {i} function has invalid 'parameters' schema: {str(e)}"
    except RecursionError: # 新增: 防止循环 Schema 导致 500
        return (
            f"Tool {i} function 'parameters' schema is too deeply nested "
            "or contains a cycle."
        )
```

评论区精华

无实质技术讨论。审查者 ispobock 直接批准, 仅有的评论是作者多次触发 CI 重试。

- 暂无高价值评论线程

风险与影响

- 风险：变更在模式验证路径中增加递归遍历，通过 `RecursionError` 捕获防循环，风险可控。
性能影响：模式通常较小，归一化开销可忽略。兼容性：保留未知类型原样传递，不破坏现有模式。
- 影响：影响所有通过 OpenAI 兼容 API 使用工具定义请求。正常化后，更多非标准类型被接受，提升了与各类工具框架的兼容性。对不使用工具的场景无影响。
- 风险标记：核心路径变更，递归深度已处理，兼容性影响仅限工具模式

关联脉络

- PR #23476 Add tool parameter schema type normalization: 原始实现，后因分支过时被还原
- PR #26379 Revert tool parameter schema type normalization: 还原了 #23476，本 PR 基于当前 main 重新引入并改进