

PR #34777 完整报告

sgl-project/sglang

[Fix] Require JSON booleans for response_format json_schema.strict

合并时间: 2026-08-15 07:46

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/34777>

执行摘要

- 一句话: json_schema.strict 强制要求 JSON 布尔值
- 推荐动作: 值得精读: 这是一个小而精准的类型边界修复, 展示了如何用 pydantic StrictBool 收紧 API 合同, 并配套了正反例测试。

功能与动机

OpenAI wire contract 只接受 JSON 布尔值, 同样的 payload 在 OpenAI 会返回 422。由于 strict 控制 constrained-decoding 的 opt-out (见 to_sampling_params 中的 `strict is not False` 检查), 静默强制转换可能让畸形客户端 payload 切换 schema 执行, 而不是像 OpenAI 那样在验证时失败。

实现拆解

1. 修改 python/sglang/srt/entrypoints/openai/protocol.py: 从 pydantic 导入 StrictBool, 并将 JsonSchemaResponseFormat.strict 字段类型从 Optional[bool] 改为 Optional[StrictBool], 同时补充注释说明理由。
2. 在 test/registered/unit/entrypoints/openai/test_protocol.py 中新增 test_json_schema_strict_requires_json_boolean 测试: 对 True/False/None 验证通过, 对 "yes"/"false"/0/1 断言 ValidationError。
3. 通过 /rerun-test 验证相关测试 (test_protocol.py、test_constrained_decoding.py、test_json_schema_constraint.py) 全部通过, 未引入回归。

关键文件:

- python/sglang/srt/entrypoints/openai/protocol.py (模块 协议层; 类别 source; 类型 core-logic; 符号 JsonSchemaResponseFormat) : 核心变更: 将 strict 字段类型从 Optional[bool] 改为 Optional[StrictBool], 强制 JSON 布尔验证, 与 OpenAI wire contract 对齐。
- test/registered/unit/entrypoints/openai/test_protocol.py (模块 协议测试; 类别 test; 类型 test-coverage; 符号 test_json_schema_strict_requires_json_boolean) : 新增 test_json_schema_strict_requires_json_boolean 测试, 覆盖合法布尔值 / 省略与非布尔值被拒绝的场景, 保障类型收紧不破坏默认行为。

关键符号: JsonSchemaResponseFormat, test_json_schema_strict_requires_json_boolean

关键源码片段

python/sglang/srt/entrypoints/openai/protocol.py

核心变更：将 strict 字段类型从 Optional[bool] 改为 Optional[StrictBool]，强制 JSON 布尔验证，与 OpenAI wire contract 对齐。

从 pydantic 导入 StrictBool，它会拒绝 lax 模式下被隐式转换的值

```
from pydantic import (
    BaseModel,
    ConfigDict,
    Field,
    StrictBool,
    field_serializer,
    field_validator,
    model_serializer,
    model_validator,
)
```

```
class JsonSchemaResponseFormat(BaseModel):
```

```
    """OpenAI 兼容的 JSON Schema 响应格式。"""
```

```
    name: str
```

```
    description: Optional[str] = None
```

```
    # 使用别名规避 pydantic 对 schema 字段的冲突
```

```
    schema_: Optional[Dict[str, object]] = Field(alias="schema", default=None)
```

```
    # OpenAI wire contract 只接受 JSON 布尔值；StrictBool 会拒绝 lax 模式下
```

```
    # 被强制转换的值 ("yes"、"on"、0、1 等)，与 OpenAI 的 422 行为保持一致。
```

```
    # 省略该字段 (None) 时语义不变。
```

```
    strict: Optional[StrictBool] = None
```

test/registered/unit/entrypoints/openai/test_protocol.py

新增 test_json_schema_strict_requires_json_boolean 测试，覆盖合法布尔值 / 省略与非布尔值被拒绝的场景，保障类型收紧不破坏默认行为。

```
def test_json_schema_strict_requires_json_boolean(self):
```

```
    # 构造一个合法的 json_schema 响应格式基础请求
```

```
    base_request = {
```

```
        "model": "test-model",
```

```
        "messages": [{"role": "user", "content": "Hello"}],
```

```
        "response_format": {
```

```
            "type": "json_schema",
```

```
            "json_schema": {"name": "answer", "schema": {"type": "object"}},
```

```
        },
```

```
    }
```

```
    # 合法的 JSON 布尔值 (True/False) 以及省略 (None) 都应通过验证
```

```
    for strict in (True, False, None):
```

```
        with self.subTest(strict=strict):
```

```

response_format = dict(base_request["response_format"])
response_format["json_schema"] = {
    **response_format["json_schema"], "strict": strict,
}
request = ChatCompletionRequest.model_validate(
    **base_request, "response_format": response_format
)
self.assertIs(request.response_format.json_schema.strict, strict)

# 非布尔值（字符串 "yes"/"false"、数字 0/1）必须拒绝，与 OpenAI 的 422 对齐
for strict in ("yes", "false", 0, 1):
    with self.subTest(strict=strict), self.assertRaises(ValidationError):
        response_format = dict(base_request["response_format"])
        response_format["json_schema"] = {
            **response_format["json_schema"], "strict": strict,
        }
        ChatCompletionRequest.model_validate(
            **base_request, "response_format": response_format
        )

```

评论区精华

PR 没有收到实质性 review 评论，合并者直接批准。维护者 gongy 请求重跑 test_protocol.py、test_constrained_decoding.py 和 test_json_schema_constraint.py，重跑结果全部通过。

- 测试重跑确认 (testing): 重跑结果全部通过，未发现回归。

风险与影响

- 风险：主要风险是行为变更：此前接受非布尔值的客户端现在会收到 422。由于默认 None 不变，省略 strict 的请求不受影响；但显式发送 "yes"/0 等的客户端会失败。此变更同时影响 constrained decoding 的 opt-out 路径，更严格地确保只有显式 false 才关闭 schema 强制。
- 影响：影响所有使用 response_format.json_schema 的 OpenAI 兼容 API 客户端，提升与 OpenAI 行为的一致性；对 SGLang 内部逻辑影响小，仅验证层变化。
- 风险标记：验证行为变更，客户端兼容性影响

关联脉络

- 暂无明显关联 PR