

PR #46827 完整报告

vllm-project/vllm

[Rust Frontend] Keep literal "null" string for string-typed tool params

合并时间: 2026-06-29 21:46

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/46827>

执行摘要

本 PR 修复了 Rust 前端工具调用解析器中将 string 类型参数的文字值 "null" 错误转换为 JSON null 的 bug。Blasrodri 在 `convert_with_optional_schema` 中增加了类型保护，并改进 enum 推导以处理 null 成员，使行为与 Python 端一致。影响所有 XML 风格解析器（`deepseek_dsml`、`qwen_coder`、`glm_xml` 等），风险较低且有充分测试覆盖。

功能与动机

Blasrodri 在 PR body 中解释：当工具参数的 schema 类型为 `string` 且值为文字 "null"（例如 `location="null"`）时，Rust 前端会将字符串 "null" 强制转换为 JSON null，导致真实值丢失。这与 Python 端的 `coerce_to_schema_type` 行为不一致——Python 端在声明类型为 `string` 时保留 "null" 字符串。此问题影响所有途经 `convert_with_optional_schema` 的解析器（`qwen_coder`、`glm_xml`、`deepseek_dsml`、`minimax_m2/m3`、`hy_v3`）。

实现拆解

- 核心 null 转换逻辑修改（`parameters.rs: convert_with_optional_schema`）：将原来的 `// For literal null, always convert to JSON null value.` 改为增加 `param_type != Some(&JsonParamType::String)` 条件，即 `string` 类型参数保留文字 "null"；其他类型（`integer/object/array`）或无 schema 时仍转换 JSON null。
- Enum 推导改进（`parameters.rs: from_schema`）：当 schema 省略 `type` 而使用 `enum` 时，原逻辑直接返回 `String`。review 指出若 `enum` 中包含 `null` 成员（如 `{"enum": [null, "auto"]}`），会导致 `string` 类型保护将文字 "null" 转为字符串而非 JSON null。修改后：检测 `enum` 数组是否包含 `null` 值，若包含则返回 `OneOf([String, Null])`，确保文字 "null" 仍可转换为 JSON null，与 Python 端行为一致。
- 测试调整：新增 `string_param_preserves_literal_null_text`（验证 `string` 类型保留 "null"/"NULL" 字符串）和 `nullable_enum_param_coerces_literal_null`（验证含 `null` 的 `enum` 仍将文字 "null" 转为 JSON null）。同时调整 `deepseek_v32.rs` 中现有测试的期望值：`"empty": null` → `"empty": "null"`。

`rust/src/parser/src/tool/parameters.rs`

核心实现文件，包含 `convert_with_optional_schema` 的 null 转换逻辑修改和 `JsonParamType::from_schema` 的 enum 推导改进，以及新增的两个测试用例。

```
// rust/src/parser/src/tool/parameters.rs
```

```
/// 将参数输入转换为规范化的 JSON 值。
```

```

/// 对于文字 `null`，除了 string 类型的参数外，都转换为 JSON null。
/// string 类型参数必须保留文字 "null" 字符串，因为模型输出文字 "null"
/// 表示字符串本身，而非缺失值。
fn convert_with_optional_schema(param_type: Option<&JsonParamType>, input: &ParamInput) -
> Value {
    // 如果输入为文字 "null"，且参数类型不是 String，则转为 JSON null
    if let ParamInput::Text(value) = input
        && value.eq_ignore_ascii_case("null")
        && param_type != Some(&JsonParamType::String)
    {
        return Value::Null;
    }

    // 如果有 schema，尝试根据类型转换
    if let Some(param_type) = param_type
        && let Some(value) = try_convert_value(param_type, input)
    {
        return value;
    }
    // 无 schema 或转换失败，回退为字符串
    match input {
        ParamInput::Text(value) => Value::String(value.clone()),
        ParamInput::Elements(elements) => {
            Value::Object(convert_elements_to_object(elements, &BTreeMap::new(), None))
        }
    }
}

// 在 from_schema 中，对于 enum 类型的处理：
// 如果 enum 包含 null 成员，则推导为 OneOf([String, Null])，
// 使得文字 "null" 仍能转换为 JSON null（与 Python 行为一致）
if let Some(values) = schema.get("enum").and_then(Value::as_array) {
    if values.iter().any(Value::is_null) {
        return Some(Self::one_of(vec![Self::String, Self::Null]));
    }
    return Some(Self::String);
}

```

rust/src/parser/src/tool/deepseek_dsml/deepseek_v32.rs

测试文件，调整了一个现有测试的期望值以反映修复后的正确行为。

```

// rust/src/parser/src/tool/deepseek_dsml/deepseek_v32.rs
#[test]
fn deepseek_v32_parse_complete_converts_schema_types() {
    // ... 测试输入包含 <parameter name="empty" string="false">null</parameter>
    // 现在期望 empty 字段保留为字符串 "null"，因为它在 schema 中被定义为 string 类型
    assert_eq!(
        serde_json::from_str::<Value>(&output.calls()[0].arguments).unwrap(),
        json!({

```

```
        "whole": 5.0,  
        "flag": true,  
        "payload": { "nested": true },  
        "items": [1, 2],  
        "empty": "null", // 修复前为 null, 修复后为 "null"  
    })  
);  
}
```

评论区精华

chatgpt-codex-connector[bot] (review 评论) : "When a schema omits `type` and uses an enum such as `{\"enum\": [null, \"auto\"]}`, `JsonParamType::from_schema` currently normalizes it to `String` just because `enum` is present. This new `exact-String` guard therefore makes XML-style parsers serialize a literal `null` argument as the string `\"null\"` instead of JSON `null`."

Blasrodri 随后回复并修复: "Good catch, fixed. A `null` member in an `enum` now normalizes the param to `OneOf([String, Null])` instead of `String`, so a literal `\"null\"` coerces to JSON `null`."

风险与影响

风险: 低。变更集中 (两个函数), 测试覆盖充分 (新增两个测试 + 调整一个现有测试)。唯一潜在风险是若未来有其他代码依赖“总是将文字 `null` 转为 JSON `null`”的行为, 可能在 `string` 类型参数上遇到行为变化——但这正是本 PR 要修复的正确行为, 且与 Python 端对齐。

影响: 影响所有使用 `convert_with_optional_schema` 的 XML 风格解析器 (`deepseek_dsml`、`qwen_coder`、`glm_xml`、`minimax_m2/m3`、`hy_v3`)。用户在使用这些模型时, `string` 类型的工具参数值中的文字 `\"null\"` 现在将正确保留为字符串, 而非丢失。其他类型参数行为不变。

关联脉络

无直接关联的历史 PR。本 PR 是 Rust 前端工具调用功能的一个独立修复, 与 #44512 (整合 `scale-out` 入口点) 和 #46800 (`Harmony Renderer`) 属于同一 Rust 前端开发线, 但无直接代码依赖。