PR #7307 完整报告

PaddlePaddle/FastDeploy

[DataProcessor] add strict

合并时间：2026-04-14 17:25

原文链接：http://prhub.com.cn/PaddlePaddle/FastDeploy/pull/7307

执行摘要

本次 PR 向 FastDeploy 的 OpenAI 兼容协议中添加了 strict 字段，支持函数调用的严格模式。变更仅涉及 protocol.py 中 FunctionDefinition 类的一个字段添加，经过讨论优化为 Optional[bool] = None 以确保向后兼容性。这是一个低风险、影响有限的 API 扩展，主要价值在于对齐 OpenAI 官方规范。

功能与动机

根据 review 讨论，本次变更的目的是 " 向 OpenAI Function Calling API 规范对齐，支持 strict 模式，用于强制模型遵循工具参数的 schema 定义 "。fastdeploy-bot 在 review 中明确提到 " 支持 OpenAI API 中 Function Calling 的严格模式（strict mode），确保模型生成的 JSON 输出严格遵循参数定义的 JSON Schema"，并参考了 OpenAI 官方文档。这增强了 FastDeploy 在函数调用场景下的控制能力。

实现拆解

实现非常简单，仅在 fastdeploy/entrypoints/openai/protocol.py 文件的 FunctionDefinition 类中添加了一个字段：

```
class FunctionDefinition(BaseModel):
    name: str
    description: Optional[str] = None
    parameters: Optional[dict[str, Any]] = None
    strict: Optional[bool] = None # 新增字段
```

关键设计决策：

1. 字段类型：从最初的 bool = False 改为 Optional[bool] = None
2. 兼容性考虑：使用 Optional 确保未显式指定时不会序列化该字段，保持与现有请求负载的一致性
3. 对齐参考：与同文件中的 JsonSchemaResponseFormat.strict 字段保持相同设计模式

评论区精华

review 讨论主要集中在两个技术点上：

> Copilot：" 这里把 strict 设为 bool = False 会导致在 model_dump() 时，总是把 strict: false 序列化并透传到 engine/request，从而改变未显式指定 strict 的请求负载。为保持与现有行为 / 文档示例一致，建议将其改为 Optional[bool] = None"

> fastdeploy-bot: " 建议添加单元测试验证 strict 字段的序列化 / 反序列化行为，确保与 JsonSchemaResponseFormat.strict 行为一致 "

最终采纳了 Copilot 的建议，将字段改为 Optional[bool] = None，但测试建议未被立即实施。

风险与影响

风险分析：

1. 兼容性风险：低。字段为 Optional 且默认 None，未指定时不序列化，完全向后兼容。
2. 序列化风险：已解决。最初设计可能意外添加 strict: false 到所有请求，优化后避免了此问题。
3. 测试风险：中等。缺少针对新字段的单元测试，可能影响未来重构信心。

影响评估：

- 用户影响：为需要严格模式函数调用的用户提供了官方支持
- 系统影响：仅扩展 API 协议定义，不改变核心推理逻辑
- 团队影响：变更简单，易于维护和理解

关联脉络

从近期历史 PR 看，FastDeploy 持续增强其 OpenAI 兼容 API 能力：

- PR #7352 在 api_server.py 中添加了 IP 白名单检查，增强安全性
- 本次 PR #7307 在 protocol.py 中扩展协议支持

这两次变更都集中在 fastdeploy/entrypoints/openai/ 目录下，体现了团队对 API 层完整性和规范对齐的持续投入。与 OpenAI 官方规范保持同步有助于提升 FastDeploy 作为推理服务的兼容性和易用性。