

PR #56369 完整报告

vllm-project/vllm

[Frontend][last/N] Move all non-OpenAI content out of the OpenAI folder.

合并时间: 2026-09-11 13:33

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/56369>

执行摘要

- 一句话: SSE keep-alive 模块迁出 OpenAI 目录并删除冗余截断测试
- 推荐动作: 值得花时间看的不是搬迁本身 (机械改路径), 而是两件事: 一是删除 `test_async_tokenization.py` 的覆盖归属, 建议在合并后确认 `truncate_prompt_tokens` 是否已有等价测试在其他文件接管, 必要时补一个轻量用例; 二是系列重构「不留兼容别名、仓库内全量替换」的策略, 接触者应意识到 `vllm.entrypoints` 属于半内部 API, 升级前需自行检查 `import`。值得借鉴的工程实践是: 搬迁时保持文件内容零改动, 让 review 只需核对路径与引用点, 把重构风险压到最低。

功能与动机

PR body 给出了明确的动机: 『In the early days of vLLM (2023), there was only the OpenAI, so vLLM online serving was referred to as the OpenAI-Compatible Server. As we have entered 2026, generative AI has grown significantly, and more and more features have been added to vLLM online serving. We'd better move all non-OpenAI content out of the OpenAI folder.』并点名 #41907、#52131、#53500、#53659、#54492 为同一系列的前序 PR, 本 PR 负责最后一块非 OpenAI 内容 (SSE keep-alive 工具) 与一处冗余测试的清理。

实现拆解

变更入口: 延续目录重构系列, 把 `vllm/entrypoints/openai` 里不具备 OpenAI 协议语义的工具模块迁往 `vllm/entrypoints/serve/utils`, 使 `openai` 包只保留 OpenAI 协议实现。

1. 模块搬迁 (内容零改动): `vllm/entrypoints/serve/utils/sse_keep_alive.py` 由 `vllm/entrypoints/openai/sse_keep_alive.py` 直接改名而来 (+0/-0), `SSE_KEEP_ALIVE_COMMENT`、`with_sse_keep_alive`、`_keep_alive_stream` 的实现与 `docstring` 一字未改。这样改的目的是让 `serve` 层成为跨协议共享的流式工具归属地; 由于内容不变, SSE 保活语义、默认关闭路径 (`interval` 非正 / NaN / inf 时原样返回同一个 `generator`) 以及 `asyncio.wait` 而非 `wait_for` 的实现细节全部保持等价。
2. 调用方导入改写: `vllm/entrypoints/openai/chat_completion/api_router.py` 与 `vllm/entrypoints/openai/completion/api_router.py` 各改 1 行 `import`, 由 `vllm.entrypoints.openai.sse_keep_alive` 指向 `vllm.entrypoints.serve.utils.sse_keep_alive`。`create_chat_completion` / `create_completion` 内部对 `with_sse_keep_alive` 的调用方式与参数未动, 因此 `/v1/chat/completions`、`/v1/completions` 的 SSE 帧内容、保活注释节

奏与代理侧读超时表现零变化。

3. 测试同步搬迁: tests/entrypoints/openai/test_sse_keep_alive.py → tests/entrypoints/serve/utils/test_sse_keep_alive.py, 只替换 import 行; test_disabled_returns_same_generator (关闭路径用 is 断言对象同一性)、test_keep_alive_during_silent_start 等用例的断言集合不变, 保证搬迁后保活逻辑仍有单测兜底。
4. 删除端到端用例: 整体删除 tests/entrypoints/openai/test_async_tokenization.py (-82 行), 移除 test_with_and_without_truncate —— 该用例在 completion / chat 两条路由上把带与不带 truncate_prompt_tokens 的请求随机混排后并发提交, 并断言 10 个响应中没有 500。PR body 以 rm tests/entrypoints/openai/test_async_tokenization.py #45759 标注依据, 意图是把这层覆盖交给更合适的归属位置。
5. 无配置 / schema / 部署配套: 未改动 pyproject.toml、__init__.py、CI 配置或文档; 目标目录 vllm/entrypoints/serve/utils/ 已存在 (同目录已有 api_utils.py、orca_metrics.py), 无需新增包初始化文件, 与仓库已启用的 ruff INP (要求 vllm/ 下目录含 __init__.py) 规则兼容。CI 由维护者 DarkLight1337 以 /ci run 手动触发 (Buildkite #88246, commit 112eef51be26)。

关键文件:

- vllm/entrypoints/serve/utils/sse_keep_alive.py (模块 流式保活; 类别 source; 类型 rename-or-move; 符号 with_sse_keep_alive, _keep_alive_stream, SSE_KEEP_ALIVE_COMMENT): 本 PR 被搬迁的核心模块, 从 openai 包迁入 serve 层工具目录; 其 SSE 保活实现 (禁用路径零开销、asyncio.wait 避免取消进行中的 anext) 是这次搬迁唯一承载真实逻辑的文件。
- tests/entrypoints/openai/test_async_tokenization.py (模块 接口测试; 类别 test; 类型 deletion; 符号 server, client, test_with_and_without_truncate, get_status_code): 本 PR 唯一被删除的文件 (-82 行)。理解它测什么、覆盖面被谁接管, 是评估本次改动风险的关键。
- vllm/entrypoints/openai/chat_completion/api_router.py (模块 聊天接口; 类别 source; 类型 endpoint; 符号 with_sse_keep_alive): chat 路由的调用方, 唯一改动是把 with_sse_keep_alive 的 import 指向新模块; 代表本次搬迁对线上路径的全部影响面。
- vllm/entrypoints/openai/completion/api_router.py (模块 补全接口; 类别 source; 类型 endpoint; 符号 with_sse_keep_alive): completion 路由的调用方, 同样只改 1 行 import, 与 chat 路由一起构成本次搬迁的全部生产代码改动。
- tests/entrypoints/serve/utils/test_sse_keep_alive.py (模块 单元测试; 类别 test; 类型 rename-or-move; 符号 test_disabled_returns_same_generator, test_keep_alive_during_silent_start): keep-alive 单测随源码搬迁, 只改 import; 它是搬迁后保住 with_sse_keep_alive 语义的唯一测试。

关键符号: with_sse_keep_alive, _keep_alive_stream, test_with_and_without_truncate, create_chat_completion, create_completion

关键源码片段

vllm/entrypoints/serve/utils/sse_keep_alive.py

本 PR 被搬迁的核心模块，从 openai 包迁入 serve 层工具目录；其 SSE 保活实现（禁用路径零开销、asyncio.wait 避免取消进行中的 anext）是这次搬迁唯一承载真实逻辑的文件。

```
# path: vllm/entrypoints/serve/utils/sse_keep_alive.py
# 由 vllm/entrypoints/openai/sse_keep_alive.py 原样搬迁而来，文件内容零改动，
# 唯一变化是它不再属于 OpenAI 协议层，而成为 serve 层的共用流式工具。

import asyncio
import contextlib
from collections.abc import AsyncGenerator

# SSE 规范允许以 ":" 开头的注释行，合规客户端会忽略它；
# 但反向代理（Cloudflare Tunnel / NGINX proxy_read_timeout / AWS ALB）
# 仍会把它算作「连接还有活动」的字节，从而不在读超时后切断连接。
SSE_KEEP_ALIVE_COMMENT = ": keep-alive"

"

def with_sse_keep_alive(
    generator: AsyncGenerator[str, None],
    interval: float,
) -> AsyncGenerator[str, None]:
    """Emit an SSE keep-alive comment when generator is idle."""
    # 关闭路径：interval 非正 / NaN / inf 时直接返回同一个对象，
    # 保证默认路径零封装、零 per-chunk 开销；
    # tests/entrypoints/serve/utils/test_sse_keep_alive.py 里的
    # test_disabled_returns_same_generator 用 `is` 断言守住这条不变量。
    if not 0 < interval < float("inf"):
        return generator
    return _keep_alive_stream(generator, interval)

async def _keep_alive_stream(
    generator: AsyncGenerator[str, None],
    interval: float,
) -> AsyncGenerator[str, None]:
    # 只保留一个 anext() 任务推进上游：asyncio 禁止对同一异步生成器并发 __anext__，
    # 而 asyncio.wait_for 会在超时时取消任务，把上游的推进打断在半途；
    # 这里用 asyncio.wait 只观察「超时 / 完成」两种结局，不取消正在进行的推进。
    next_task: asyncio.Task[str] = asyncio.create_task(anext(generator))
    try:
        while True:
            done, _ = await asyncio.wait({next_task}, timeout=interval)
            if not done:
                # 排队、prefill 或 token 间隔超过 interval：补一条保活注释后继续等待。
                yield SSE_KEEP_ALIVE_COMMENT
```

```

        continue
    try:
        chunk = next_task.result()
    except StopAsyncIteration:
        return # 上游正常结束
    yield chunk
    next_task = asyncio.create_task(anext(generator))
finally:
    # 清理阶段：先 cancel 再吞掉结果（与 merge_async_iterators 的处理方式一致），
    # 避免清理动作覆盖掉循环里已经处理过的真实结局。
    next_task.cancel()
    with contextlib.suppress(BaseException):
        await next_task
    # 显式 aclose() 上游，让其 finally / 资源释放确定性执行，而不是等 GC 触发。
    with contextlib.suppress(BaseException):
        await generator.aclose()

```

tests/entrypoints/openai/test_async_tokenization.py

本 PR 唯一被删除的文件（-82 行）。理解它测什么、覆盖面被谁接管，是评估本次改动风险的关键。

```

# path: tests/entrypoints/openai/test_async_tokenization.py
# 本 PR 整体删除该文件（-82 行）。删除前，它是唯一同时覆盖 completion 与 chat
# 两条路由、并把「带 truncate_prompt_tokens」与「不带」的请求随机混排后并发提交的用例。

```

```

@pytest.mark.asyncio
@pytest.mark.parametrize(
    ids=["completion", "chat"],
    argnames=["create_func_gen", "content_body"],
    argvalues=[
        (lambda x: x.completions.create, {"prompt": " ".join(["A"] * 10_000)}),
        (
            lambda x: x.chat.completions.create,
            {"messages": [{"role": "user", "content": " ".join(["A"] * 10_000)}]},
        ),
    ],
)
async def test_with_and_without_truncate(server, client, create_func_gen, content_body):
    create_func = create_func_gen(client)
    body = {"model": MODEL_NAME, **content_body, "max_tokens": 10}

    # 一半请求显式给 truncate_prompt_tokens=1000（超长 prompt 被截到 1000），
    # 另一半不给（走全量 prompt 路径）；两者随机打乱后用 asyncio.gather 并发发出。
    num_requests = 10
    truncate_prompt_tokens = [1000] * (num_requests // 2) + [None] * (
        num_requests - num_requests // 2
    )
    random.shuffle(truncate_prompt_tokens)
    bodies = [

```

```

    {**body, "extra_body": {"truncate_prompt_tokens": t}}
    for t in truncate_prompt_tokens
]

async def get_status_code(**kwargs):
    try:
        await create_func(**kwargs)
        return 200
    except openai.APIStatusError as e:
        return e.status_code

# 原断言很弱：只要求 10 个并发请求都不返回 500，据此说明混合截断请求
# 不会把服务端打挂。删除后若没有等价测试接管，这条链路的覆盖需要重新确认。
responses = await asyncio.gather(*[get_status_code(**b) for b in bodies])
assert 500 not in responses

```

评论区精华

本 PR 的讨论极其轻量，几乎没有实质性技术交锋：

claude[bot]: 「This pull request is from a fork — automated review is disabled. A repository maintainer can comment @claude review to run a one-time review.」

自动化审查因 PR 来自 fork 被跳过，需要维护者手动发起一次性审查，实际并未发生。

DarkLight1337: `/ci run` (issue 评论)，随后给出正文为空的 APPROVED。

维护者用「触发一次 CI + 一次批准」完成放行，对搬迁路径、导入改写或测试删除都没有提出质疑或补充。

未被讨论但值得注意的两点：其一，旧路径 `vllm.entrypoints.openai.sse_keep_alive` 被物理删除且没有留兼容别名，评审中无人提到下游兼容问题；其二，删除 `test_async_tokenization.py` 的依据只有 body 里的一行 [#45759](#) 引用，Test Plan 与 Test Result 均为空，覆盖是否被接管在评审层面是隐式决定。

- fork PR 自动审查被禁用，改由维护者手动触发 CI 与批准 (other): 自动化审查缺位，本 PR 完全依赖维护者的空批准与 CI 结果放行；改动本身争议度低，但也意味着决策缺少显式讨论记录。
- 删除 `test_async_tokenization.py` 后 `truncate_prompt_tokens` 覆盖归属未明 (testing): 讨论未展开，替代覆盖是否存在无法从本次材料确认；建议合并后核对 [#45759](#) 或其他测试是否接管该场景。
- 模块搬迁不保留旧导入路径的兼容别名 (design): 沿用该重构系列「仓库内全量替换、不留兼容层」的一贯策略，实际上把 `vllm.entrypoints` 视为半内部 API；下游 fork 与外部插件升级时需自行改 import。

风险与影响

- 风险：

1. 测试覆盖缺口（首要风险）：tests/entrypoints/openai/test_async_tokenization.py 被整体删除（-82 行）。删除前它是唯一同时覆盖 /v1/completions 与 /v1/chat/completions、并把 truncate_prompt_tokens=1000 与「不截断」请求随机混排后并发提交（10 路 asyncio.gather）的用例，用于验证共享的引擎侧截断路径不会互相污染。PR body 只给出 #45759 作为依据且未附 Test Plan / Test Result，若没有等价用例接管，截断相关的回归（截断长度计算、并发请求状态串扰、超长 prompt 的 500）将不再被 CI 捕获。
2. 导入路径 breaking，无兼容别名：vllm/entrypoints/openai/sse_keep_alive 被直接移除，仓库内 2 个调用点与 1 个测试文件已全部替换，但下游 fork、内部依赖包或示例脚本若仍写 from vllm.entrypoints.openai.sse_keep_alive import ...，升级后会立即 ImportError。vLLM 对 vllm.entrypoints 的实际定位是半内部 API，这一取舍与系列前序 PR 一致。
3. 行为回归面极小：搬迁文件 +0/-0，with_sse_keep_alive 的关闭路径与 _keep_alive_stream 的 asyncio.wait / cancel / aclose 清理顺序都未改动，SSE 输出与代理超时行为无回归点。
4. 流程性风险：fork PR 自动审查禁用 + 空 APPROVED，意味着这次删除测试的决策缺少显式评审记录；一旦后续发现覆盖缺口，追溯成本更高。

• 影响：

- 用户 / 线上服务：无感知。HTTP 端点、SSE 帧格式、`: keep-alive`

"**注释的发送时机**与`interval` 语义都不变。

- 开发者 / 下游：vllm.entrypoints.openai 的公开面进一步收窄，非 OpenAI 语义的工具统一收敛到 vllm.entrypoints.serve.utils；外部引用旧路径需一次性改 import。
- 团队 / 架构：这是「OpenAI 目录只留 OpenAI 协议内容」路线的最后一步，为 Anthropic 等其他协议入口（如 vllm/entrypoints/anthropic/serving.py）复用 serve 层工具扫清了命名障碍，后续新增协议不再需要往 openai 包里塞公共工具。
- 测试资产：净减少 82 行端到端用例，测试面收窄的收益与代价需维护者确认；SSE keep-alive 自身的单测在搬迁后保持完整。
- 风险标记：删除端到端测试覆盖，旧导入路径无兼容别名，Test Plan 为空

关联脉络

- PR #41907 非 OpenAI 内容迁出 OpenAI 目录（系列前序 PR，材料未提供标题）：PR body 明确列出 #41907 #52131 #53500 #53659 #54492 为同一目录重构系列的前序 PR，本 PR 是其 [last/N] 收尾，方向与做法完全延续。
- PR #54492 非 OpenAI 内容迁出 OpenAI 目录（系列前序 PR，材料未提供标题）：同一系列的直接前置 PR，决定了本 PR 的搬迁目标目录与「不留兼容别名」的处理方式。
- PR #55325 [Bugfix] Set stop_sequence explicitly in streaming message_delta event: 同属 entrypoints 层的协议实现改动（vllm/entrypoints/anthropic/serving.py），体现 vllm/entrypoints 下多协议并行、OpenAI 包不再是唯一归属的演进趋势，与本 PR 的目录收敛方向一致。