PR #33845 完整报告

sgl-project/sglang

[diffusion] centralize entrypoint API hygiene

合并时间：2026-08-06 21:53

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/33845

执行摘要

- 一句话：集中扩散入口 API 卫生与共享游标分页
- 推荐动作：值得精读，重点看 stores.py 中 AsyncDictStore.list_page 与 common_api.py 中 _build_model_card 的集中化方式，可作为入口 API 卫生重构的参考样例。建议后续补充针对 list_page 游标语义与非法 order 兜底的单元测试，以覆盖本次重构的关键行为。

功能与动机

PR body 指出：导入 entrypoints 包时曾修改全局 logger 状态，而并行的 OpenAI 端点各自维护重复的响应与分页逻辑。目标是让副作用只发生在运行时启动，并让共享行为拥有单一负责人；同时声明现有排序、invalid-order fallback 与游标行为均被保留。

实现拆解

1. 日志抑制与日志卫生：在 runtime/entrypoints/init.py 中删除 import 期的 globally_suppress_loggers() 调用，只保留 SPDX 许可证头；在 runtime/entrypoints/http_server.py 的 create_app() 开头调用 globally_suppress_loggers()，使副作用只在真正启动 HTTP 服务时发生；在 runtime/entrypoints/diffusion_generator.py 的 from_server_args() 中也加入该调用以覆盖本地 CLI 场景，并将 Local mode 日志改为 %s 惰性格式化。runtime/scheduler_client.py 中 run_zeromq_broker 每次收到离线 job 的 logger.info 降为 logger.debug。
2. 模型卡片组装集中化：在 runtime/entrypoints/openai/common_api.py 新增 _build_model_card(server_args, model_id) 函数，统一承担 get_model_info 查询、card_kwargs 组装与 DiffusionModelCard 构造；available_models() 和 retrieve_model() 删除各自重复代码并改为调用该函数。
3. 游标分页集中化：在 runtime/entrypoints/openai/stores.py 中新增 AsyncDictStore.list_page(after, limit, order)，在 store 内部完成按 created_at 排序、after 游标切片与 limit 截断，替代 list_values()；runtime/entrypoints/openai/video_api.py 的 list_videos() 与 mesh_api.py 的 list_meshes() 删除各自重复的排序、兜底与游标逻辑，改为透传参数调用 list_page；两处后台任务失败的 logger.error(f"{e}") 改为 logger.exception(...)。
4. 测试与 CI 配套：本 PR 未新增或修改测试文件；run-ci 与 run-ci-extra 均通过，回归主要依赖现有集成测试。

关键文件：

- python/sglang/multimodal_gen/runtime/entrypoints/openai/common_api.py（模块 入口层；类别 source；类型 entrypoint；符号 _build_model_card）：新增 _build_model_card 统一模型卡片组装，available_models 与 retrieve_model 两个端点都改为复用它，是本次 API 卫生重构的核心。
- python/sglang/multimodal_gen/runtime/entrypoints/openai/stores.py（模块 共享存储；类别 source；类型 dependency-wiring；符号 list_page, list_values）：AsyncDictStore.list_values 升级为带 OpenAI 游标语义的 list_page，视频与网格列表接口得以删除重复排序与游标切片代码，是分页逻辑的单一归属点。
- python/sglang/multimodal_gen/runtime/entrypoints/openai/video_api.py（模块 视频接口；类别 source；类型 entrypoint；符号 list_videos）：list_videos 改用 list_page 并删除重复游标逻辑，任务失败日志改为 logger.exception 保留 traceback，是可观测性改进在视频接口上的落地。
- python/sglang/multimodal_gen/runtime/entrypoints/openai/mesh_api.py（模块 网格接口；类别 source；类型 entrypoint；符号 list_meshes）：与 video_api 对称，list_meshes 切换到 list_page，任务失败日志改为 logger.exception，是本次重构在网格接口上的落地。
- python/sglang/multimodal_gen/runtime/entrypoints/http_server.py（模块 启动装配；类别 source；类型 dependency-wiring；符号 create_app）：globally_suppress_loggers 从包导入期移入 create_app() 启动路径，使日志抑制副作用只在真正启动 HTTP 服务时发生。
- python/sglang/multimodal_gen/runtime/entrypoints/__init__.py（模块 包初始化；类别 source；类型 dependency-wiring）：移除模块导入期的日志抑制副作用，只保留 SPDX 许可证头，是本 PR 消除 import 副作用的起点。
- python/sglang/multimodal_gen/runtime/scheduler_client.py（模块 调度客户端；类别 source；类型 core-logic；符号 run_zeromq_broker）：run_zeromq_broker 收到离线 job 的日志从 logger.info 降为 logger.debug，降低离线负载下高频日志噪音。
- python/sglang/multimodal_gen/runtime/entrypoints/diffusion_generator.py（模块 生成器；类别 source；类型 core-logic；符号 from_server_args）：独立生成器入口也调用 globally_suppress_loggers，并用惰性格式化日志，保证不使用 HTTP 服务的 CLI 场景日志行为一致。

关键符号：_build_model_card, AsyncDictStore.list_page, AsyncDictStore.list_values, list_videos, list_meshes

关键源码片段

python/sglang/multimodal_gen/runtime/entrypoints/openai/common_api.py

新增 _build_model_card 统一模型卡片组装，available_models 与 retrieve_model 两个端点都改为复用它，是本次 API 卫生重构的核心。

```
# python/sglang/multimodal_gen/runtime/entrypoints/openai/common_api.py
# 集中组装 OpenAI 兼容的扩散模型卡片，供多个端点复用。
```

```
def _build_model_card(server_args: ServerArgs, model_id: str) -> DiffusionModelCard:
    # 统一从全局 server_args 获取模型信息，避免两个端点各自重复 get_model_info。
    model_info = get_model_info(
        server_args.model_path,
        backend=server_args.backend,
        model_id=server_args.model_id,
    )
    card_kwargs: dict[str, Any] = {
        'id': model_id,
        'root': model_id,
        # 扩散专属字段：GPU 数量、任务类型与 DIT/VAE 精度。
        'num_gpus': server_args.num_gpus,
        'task_type': server_args.pipeline_config.task_type.name,
        'dit_precision': server_args.pipeline_config.dit_precision,
        'vae_precision': server_args.pipeline_config.vae_precision,
    }
    if model_info:
        card_kwargs['pipeline_name'] = model_info.pipeline_cls.pipeline_name
        card_kwargs['pipeline_class'] = model_info.pipeline_cls.__name__
    return DiffusionModelCard(**card_kwargs)


# 列表端点：组装卡片后直接返回 dict，保留扩展字段。
@router.get('/models')
async def available_models():
    return {'object': 'list', 'data': [_build_model_card(server_args, server_args.model_path).
    model_dump()]}


# 详情端点：同样复用 _build_model_card。
@router.get('/models/{model:path}')
async def retrieve_model(model: str):
    ...
    return _build_model_card(server_args, model).model_dump()
```

python/sglang/multimodal_gen/runtime/entrypoints/openai/stores.py

AsyncDictStore.list_values 升级为带 OpenAI 游标语义的 list_page，视频与网格列表接口得以删除重复排序与游标切片代码，是分页逻辑的单一归属点。

```
# python/sglang/multimodal_gen/runtime/entrypoints/openai/stores.py
# 把取全部加外部排序、游标切片下沉为 store 自身能力。
async def list_page(
    self,
    *,
    after: Optional[str] = None,
    limit: Optional[int] = None,
    order: Optional[str] = 'desc',
) -> list[Dict[str, Any]]:
    # 兼容旧行为：非 asc 之外的值一律按降序处理。
```

```
    normalized_order = (order or 'desc').lower()
    reverse = normalized_order != 'asc'
    async with self._lock:
        # 持锁排序，保证并发安全下的稳定顺序。
        items = sorted(
            self._items.values(),
            key=lambda item: item.get('created_at', 0),
            reverse=reverse,
        )

    if after is not None:
        try:
            # 游标定位：返回 after 之后的任务，找不到则返回空页。
            index = next(i for i, item in enumerate(items) if item['id'] == after)
            items = items[index + 1:]
        except StopIteration:
            return []

    return items[:limit] if limit is not None else items
```

python/sglang/multimodal_gen/runtime/entrypoints/openai/video_api.py

list_videos 改用 list_page 并删除重复游标逻辑，任务失败日志改为 logger.exception 保留 traceback，是可观测性改进在视频接口上的落地。

```
# python/sglang/multimodal_gen/runtime/entrypoints/openai/video_api.py

# 列表端点：直接透传分页参数，共享 store 的排序与游标逻辑。
@router.get('', response_model=VideoListResponse)
async def list_videos(
    after: Optional[str] = Query(None),
    limit: Optional[int] = Query(None, ge=1, le=100),
    order: Optional[str] = Query('desc'),
):
    jobs = await VIDEO_STORE.list_page(after=after, limit=limit, order=order)
    items = [VideoResponse(**job) for job in jobs]
    return VideoListResponse(data=items)


# 后台任务失败路径：logger.exception 保留完整 traceback，便于定位失败根因。
except Exception as e:
    logger.exception('Video job %s failed', job_id)
    await VIDEO_STORE.update_fields(job_id, {...})
```

评论区精华

该 PR 没有实质 review 讨论。评论区的两条记录均为作者触发的 /tag-and-rerun-ci 重跑指令，Review 审核与行内评论均为空。设计决策主要依靠 PR body 与代码形态呈现，缺少外部评审对分页语义或日志抑制范围的独立质疑。

- 暂无高价值评论线程

风险与影响

- 风险：分页语义：list_page 对非法 order 的降级由原来的显式判断改为隐式的 reverse = normalized_order != asc，当前行为一致但可读性下降；after 游标找不到时返回空列表与旧实现一致。 锁粒度：list_page 在 asyncio.Lock 内完成排序，任务数大时锁持有时间比旧的锁内取值、锁外排序更长，高并发列表请求下可能加剧争用。 日志行为：失败任务改用 logger.exception 会输出完整 traceback，批量失败时日志量明显增加；run_zeromq_broker 的 info 日志降为 debug，可能影响依赖该日志的运维脚本。 抑制位置：任何绕过 create_app() 或 from_server_args() 的导入路径将不再静音第三方 logger，可能改变工具脚本或测试的日志输出。 测试覆盖：重构未配套单测，游标与模型卡片两条关键路径的回归主要依赖现有 CI 集成测试。
- 影响：影响范围集中在 sglang/multimodal_gen 运行时。用户侧：OpenAI 兼容端点的 /v1/models、/v1/models/{model}、/v1/videos、/v1/meshes 响应行为保持不变。系统侧：日志量与日志内容发生变化，启动路径副作用收敛。团队侧：API 行为获得单一归属点，后续新增视频或网格类接口时可复用 list_page 与 _build_model_card，减少重复代码。该改动不触及 SRT 核心推理路径。
- 风险标记：缺少测试覆盖，分页锁内排序，日志抑制位置变更

关联脉络

- PR #33725 [diffusion] feat: data-parallel serving (--dp-size): 同属 multimodal_gen 服务化演进，且同样改动 runtime/scheduler_client.py，与本 PR 的 broker 日志调整落在同一文件。
- PR #33775 [diffusion] feat: capture-safe pynccl all-to-all: 同为 diffusion 运行时基础能力建设，与本 PR 一起构成 SGLang-Diffusion 服务入口与运行时的持续演进。
- PR #32999 [diffusion] Fix GLM-Image resolution alignment: 此前改动过 entrypoints/openai/image_api.py 与 protocol，属于同一 OpenAI 端点域，本次 _build_model_card 与 list_page 的集中化进一步巩固该域边界。