

PR #37226 完整报告

sgl-project/sglang

[Rust] Simplify request defaults and document batch header ABI

合并时间: 2026-09-01 03:31

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/37226>

执行摘要

本 PR 是 rust-server-cleanup 栈的收尾步骤 (4/4)，核心变更集中在 rust/sglang-server 的请求 / 响应消息层：删除 `GenerateBody` 中所有 `Option` 字段上冗余的 `#[serde(default)]` 注解（仅保留 `stream: bool` 上的），同时把 `BatchHeader` 中含义模糊的 `tid` 字段族重命名为 `tokids_lp`，并补齐 `direction_family_shape` 命名语法与 Python 生产端 `header_cols` 对齐的 ABI 文档。PR body 明确声明不改变任何运行时或线上格式行为，cargo test 251 项全部通过，rust 原生 e2e 测试经 `/rerun-test` 复核通过。

功能与动机

该 PR 的动机是“去除冗余请求反序列化注解，并把紧凑 batch header 契约放到其定义旁边”（PR body 原文：Remove redundant request deserialization annotations and put the compact batch-header contract next to its definition）。具体有三个意图：

1. `Option` 字段在 `serde` 缺省时自动反序列化为 `None`，`#[serde(default)]` 是噪音，删除后代码更贴近类型语义。
2. `BatchHeader` 旧文档声称“所有数值字段都 `#[serde(default)]`”，实际上前四列（`rids`、`finish_reasons`、`prompt_tokens`、`tok_lens`）是必填的，只有 `extras` 后缀列默认空，文档与实际契约不符。
3. `tid` 字段族名称含混，容易与 `prefill input` 混淆，重命名为 `tokids_lp` 后与 Python 侧 `token_ids_logprob` 语义对齐。

实现拆解

1. 清理 `GenerateBody` 注解（rust/sglang-server/src/message/request.rs）删除 `rid`、`text`、`input_ids`、`sampling_params`、`return_logprob`、`logprob_start_len`、`top_logprobs_num`、`token_ids_logprob`、`return_hidden_states`、`return_text_in_logprobs`、`bootstrap_host`、`bootstrap_port`、`bootstrap_room`、`bootstrap_pair_key`、`decode_tp_size`、`routed_dp_rank`、`disagg_prefill_dp_rank`、`image_data`、`mm_hashes`、`video_data`、`audio_data` 等全部 `Option` 字段上的 `#[serde(default)]`；仅保留非可选字段 `stream: bool` 上的注解。serde 对 `Option` 缺省即反序列化为 `None`，行为完全等价。
2. 修正 `BatchHeader` 文档契约（rust/sglang-server/src/message/response.rs）将文档改为“前四个字段必填；`tok_lens` 之后的每个字段默认空，因此热路径只发四元素 header”，并明确字段顺序即线上 ABI，必须与 python/sglang/srt/rust_server/server.py 中 `RustTokenizerManager.push_generation` 的 `header_cols` 保持一致。

3. 新增命名语法文档 字段名遵循 `direction_family_shape`: `direction` 分 `out` (decode 输出) 与 `in` (prefill 输入); `family` 分 `lp` (token logprobs)、`top` (top-k logprobs)、`tokids_lp` (请求指定 token 的 logprobs) 与 `hidden` (隐藏状态); `shape` 分 `lens` (每请求元素数)、`reqlens` (每请求位置 / 行数) 与 `poslens` (每位置 / 行元素数)。
4. 重命名 `tid` 字段族 `out_tid_reqlens` → `out_tokids_lp_reqlens`、`out_tid_poslens` → `out_tokids_lp_poslens`、`in_tid_reqlens` → `in_tokids_lp_reqlens`、`in_tid_poslens` → `in_tokids_lp_poslens`。同步更新 `for_each_chunk` 中 `per_req_ok` 校验、`sum` 对照检查、`n_od/n_id` 列元素计数、`has_extras` 守卫、`take_ragged` 读取处, 以及测试内 `header` 数组注释。字段顺序不变, `wire` 布局不变。
5. 测试与构建验证 本 PR 未新增独立测试文件, 测试覆盖依赖 `response.rs` 内已有的单元测试模块 (含 `decodes_all_extras_families_without_transposition` 等重命名后的用例)。PR body 给出 `cargo test` (251 passed)、`cargo clippy -D warnings`、`cargo fmt --check` 验证结果; Issue 评论中 `/rerun-test` 触发 rust 测试、`srt_endpoint`、`rust-native mm e2e/mmmu` 与 `openai completion rust` 测试, 全部通过。

rust/sclang-server/src/message/response.rs

BatchHeader 文档化 ABI 契约、字段重命名 `tid`→`tokids_lp` 并同步 `for_each_chunk` 解码校验与测试注释, 是理解 batch header wire 格式的关键文件。

```
/// Columnar scalar header for a whole decode batch. The first four fields are
/// required; every field after `tok_lens` defaults empty, so the hot path emits
/// a four-element header. Field order is the wire ABI and must match
/// `RustTokenizerManager.push_generation`'s `header_cols` in
/// `python/sclang/srt/rust_server/server.py`.
///
/// Field names follow `direction_family_shape`:
/// - direction: `out` = decode output, `in` = prefill input;
/// - family: `lp` = token logprobs, `top` = top-k logprobs, `tokids_lp` =
///   requested-token logprobs, and `hidden` = hidden states;
/// - shape: `lens` counts elements per request, `reqlens` counts positions or
///   rows per request, and `poslens` counts elements per position or row.
///
/// 关键点: 字段顺序即线上 ABI, msgpack 按声明顺序输出; 前四列 (rids、
/// finish_reasons、prompt_tokens、tok_lens) 必填, 其后每个列都有
/// #[serde(default)], 缺省为空数组, 热路径因此只发四元素 header。
#[derive(Debug, Default, Serialize, Deserialize)]
pub struct BatchHeader {
    /// Request ids, as the same strings Python holds (`Req.rid`, uuid hex) —
    /// hashed back to the internal routing key in `decode_one`
    /// (`Rid::shard`), mirroring the control path. The wire has no
    /// rid-shape coupling; any string is a valid rid.
    pub rids: Vec<String>,
    pub finish_reasons: Vec<Option<FinishReason>>,
    pub prompt_tokens: Vec<u32>,
    pub tok_lens: Vec<u32>,

    // 以下 extras 全部 `#[serde(default)]`: 不存在整列时为空数组,
```

```

// `has_extras` 只检查 reqlens 族即可跳过整套 extras 解码。
#[serde(default)]
pub out_lp_lens: Vec<u32>,
#[serde(default)]
pub in_lp_lens: Vec<u32>,
#[serde(default)]
pub out_top_reqlens: Vec<u32>,
#[serde(default)]
pub out_top_poslens: Vec<u32>,
#[serde(default)]
pub in_top_reqlens: Vec<u32>,
#[serde(default)]
pub in_top_poslens: Vec<u32>,
// tid 族重命名为 tokids_lp, 语义对齐 Python 的 token_ids_logprob,
// 字段顺序不变, wire 格式不变。
#[serde(default)]
pub out_tokids_lp_reqlens: Vec<u32>,
#[serde(default)]
pub out_tokids_lp_poslens: Vec<u32>,
#[serde(default)]
pub in_tokids_lp_reqlens: Vec<u32>,
#[serde(default)]
pub in_tokids_lp_poslens: Vec<u32>,
#[serde(default)]
pub hidden_reqlens: Vec<u32>,
#[serde(default)]
pub hidden_poslens: Vec<u32>,
}

```

rust/sglang-server/src/message/request.rs

删除 GenerateBody 所有 Option 字段上冗余的 serde(default) 注解, 清理反序列化配置噪音, 体现 Option 类型自身语义。

```

/// The `/generate` wire body before batch splitting: `text`/`input_ids`/
/// `sampling_params` each scalar-or-list, fanned into per-request
/// [`GenerateRequest`]s by [`into_requests`](GenerateBody::into_requests).
///
/// Unknown keys are IGNORED, matching Python: FastAPI builds `GenerateReqInput`
/// as a pydantic dataclass, which drops extras. `deny_unknown_fields` here
/// turned every `GenerateReqInput` field this server has not ported —
/// `priority`, `extra_key`, `session_id`, `session_params`,
/// `return_sampling_mask`, `custom_logit_processor`, and ~40 more — into a
/// 400, so a client that worked against the Python server broke against this
/// one. The cost of dropping it is that a typo (`temperature`) is silently
/// ignored rather than reported; that is the same trade Python already makes.
///
/// 清理说明: Option 字段缺省时 serde 自动反序列化为 None, 无需重复标注
/// #[serde(default)]; 仅非可选字段 (stream) 需要显式默认值。
#[derive(Debug, Clone, Default, Deserialize)]

```

```

pub struct GenerateBody {
  /// Optional client-supplied request id(s): a single string (a batch fans it
  /// out as `{rid}_{i}`, mirroring Python `_normalize_batch`) or one per item.
  pub rid: Option<OneOrMany<String>>,
  pub text: Option<OneOrMany<String>>,
  pub input_ids: Option<OneOrMany<TokenIds>>,
  #[serde(default)]
  pub stream: bool,
  /// One params object (broadcast) or a list of them (per item); see
  /// [ `SamplingParamsInput` ].
  pub sampling_params: Option<SamplingParamsInput>,
  /// Logprob / hidden-state options: a scalar broadcasts to every prompt, a
  /// list is per-prompt (Python `_normalize_logprob_params`).
  pub return_logprob: Option<OneOrMany<bool>>,
  pub logprob_start_len: Option<OneOrMany<i64>>,
  pub top_logprobs_num: Option<OneOrMany<i64>>,
  /// Token ids to report logprobs for: one list (broadcast to every prompt) or
  /// one list per prompt, mirroring Python's
  /// `Union[List[int], List[List[int]]` fan-out in `_normalize_batch`.
  pub token_ids_logprob: Option<OneOrMany<TokenIds>>,
  pub return_hidden_states: Option<OneOrMany<bool>>,
  /// Scalar-only in Python too (`return_text_in_logprobs: bool`).
  pub return_text_in_logprobs: Option<bool>,
  /// PD-disaggregation routing, injected per request by the PD router
  /// (mini_lb / sgl-model-gateway): a scalar for a single prompt, one-per-item
  /// lists for a batch. Elements are nullable (`List[Optional[...]]` in
  /// Python) — the router sends `bootstrap_port: [null, ...]` when deferring to
  /// the scheduler's `--disaggregation-bootstrap-port` default.
  pub bootstrap_host: Option<OneOrMany<Option<String>>>,
  pub bootstrap_port: Option<OneOrMany<Option<i64>>>,
  /// `bootstrap_room` fits in i64: the PD routers draw it from `[0, 2^63)`.
  pub bootstrap_room: Option<OneOrMany<Option<i64>>>,
  pub bootstrap_pair_key: Option<OneOrMany<Option<String>>>,
  pub decode_tp_size: Option<OneOrMany<Option<i64>>>,
  /// DP routing hints — per-request scalars even for batches, as in Python.
  pub routed_dp_rank: Option<i64>,
  pub disagg_prefill_dp_rank: Option<i64>,
  /// Multimodal inputs, permissive `Value` so any shape Python's
  /// `GenerateReqInput` accepts (URL / base64 / list / list-of-lists) parses.
  /// `into_requests` fans them out per the Python
  /// `_normalize_{image,video,audio}_data` batch rules.
  pub image_data: Option<rmvp::Value>,
  /// Caller-supplied per-item content hashes (hex) overriding the computed
  /// ones, so an external router's keys align with the prefix cache. Single
  /// requests only: Python declares the batched shapes but `__getitem__` never
  /// forwards them, so a batch is rejected here rather than answered with
  /// hashes it did not ask for.
  pub mm_hashes: Option<rmvp::Value>,
  pub video_data: Option<rmvp::Value>,

```

```
pub audio_data: Option<rmvpv::Value>,  
}
```

评论区精华

- merrymercy 在 review 中自评 approve, rainj-me 给出 APPROVED 空评论, 无实质反对意见。
- Issue 评论中通过 /rerun-test 触发多组测试确认: ubuntu-latest 上 rust 测试、1-gpu-5090 上 test_srt_endpoint.py、1-gpu-h100 上 test_rust_native_mm_e2e.py 与 test_rust_native_mm_mmmu.py 全部通过; 随后又补跑 test_openai_completion_rust.py 通过。这些 rerun 表明该 PR 对 rust 原生 server 的核心路径有回归验证需求, 验证结果均绿。

风险与影响

具体风险:

1. 跨语言 ABI 仅文档约束: 文档写明字段顺序必须与 Python header_cols 一致, 但没有任何编译期或运行期校验。若未来 Python 侧调整 header_cols 顺序而 Rust 侧未同步, 会产生难以排查的跨语言错位。
2. 重命名散落面广: tokids_lp 重命名涉及 for_each_chunk 中校验、计数、守卫与 take_ragged 读取多处, 且测试注释中大量出现字段名; 虽然本次全部同步, 但后续手写 msgpack 测试若按字段名而非顺序构造, 容易漏改。
3. 合并噪音: 21 个 commit 含多次 merge main, 最终 diff 收敛到 2 个文件; 中间分支若有人基于旧分支开发, 可能引入基于 tid 名称的新代码。

影响评估: 对客户端用户完全无感知; 对团队而言, `direction_family_shape` 命名契约与 ABI 文档为后续 Rust server 演进 (如新增 extras 族) 提供了明确基线, 降低跨语言维护成本。整体风险低。

关联脉络

本 PR 是 rust-server-cleanup 4 连发栈的最后一环: 前置 #37220 (结构拆分)、#37221 (配置整理)、#37222 (sampling 与 wire schema 对齐), 本 PR 在 schema 对齐基础上做最终命名与文档收尾。从同仓库更宏观的脉络看, sglang 正在持续把核心 server 路径迁入 Rust (如 #32710 的 Rust TreeCore 缓存后端、#37249 的 sgl-model-gateway 等), 这类“先功能、后清理”的演进模式说明 Rust server 已进入稳定维护期, 文档与命名规范化是长期可维护性的关键投资。