

# PR #35753 完整报告

sgl-project/sglang

[docs] Tell Qwen3.8-27B DFLASH2 users to build from main

合并时间: 2026-08-21 07:34

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/35753>

## Qwen3.8-27B DFLASH2 文档修正：构建要求随命令走，徽章支持按选项状态

### 执行摘要

本 PR 修正 #35663 引入的两处文档缺陷：Qwen3.8-27B cookbook 固定的 `lmsysorg/sglang:qwen38-27b` 镜像早于 DFLASH2 支持 (#35371) 与 NVFP4 量化 `lm_head` 修复 (#35496)，读者按页面操作会在 NVFP4 cell 上启动失败且无人告知原因；验证状态又曾被写成命令内 `#` 注释行而非徽章。改动在文档中新增两个 `<Warning>` 和命令内注释，把“使用 DFLASH2 需从 main 构建”的要求带到复制命令旁；同时让 `_deployment.jsx` 支持函数型 `verificationStatus`，使 H200、DGX Spark 等平台只在选中 DFLASH2 时显示 in-progress 徽章，并同步为 `check_cookbook_configs.mjs` 增加探针校验。整体是纯文档域修正外加 docs 渲染基础设施的小型演进，不触碰运行时路径。

### 功能与动机

PR body 明确这是对 #35663 的 follow-up，解决两个问题：

- 构建要求未文档化：页面固定使用 `lmsysorg/sglang:qwen38-27b`，但 DFlash2 选择器支持到 #35371 才落地，NVFP4 量化 `lm_head` 路径还需要 #35496，二者都晚于该 tag 与最新 release。读者照页面操作会在 NVFP4 上遭遇启动失败 requires a dense FP16/BF16/FP32 target `lm_head`，且没有任何提示解释原因。
- 提示行写进了命令：`# DFLASH2 on this platform: final verification in progress` 被当作注释粘进复制粘贴命令，既难看又容易被忽略；面板本就有带 in-progress 状态的验证徽章机制，#35663 应当直接使用徽章。

### 实现拆解

1. 文档层 (`docs/cookbook/autoregressive/Qwen/Qwen3.8-27B.mdx`)：在 Install 手风琴与 Deploy 命令面板后各加一个 `<Warning>`，说明 DFLASH2 需要跟踪 main 的构建（引用 #35371 与 #35496），NVFP4 上缺 #35496 会在启动时报 requires a dense FP16/BF16/FP32 target `lm_head`；同时给 `pip install sglang` 与 `docker pull lmsysorg/sglang:qwen38-27b` 命令追加注释，提示 DFLASH2 cells 应改从源码构建或拉取 nightly 镜像。两个 `<Warning>` 都点明 None / EAGLE / DSPARK 仍可在固定 tag 上运行，避免读成整页版本升级。

2. 配置层 ([docs/src/snippets/configs/Qwen/qwen3.8-27b.jsx](#)) : 删除 dflash 选项上的 hints 函数, 改为在 H200 (FP8 / BF16)、DGX Spark 等多个 cell 上添加函数型 verificationStatus: (sel) => sel.spec === "dflash" ? "in-progress" : "verified", 使徽章只对 DFLASH2 选择降级为 in-progress, 其余选择保持 Verified。
3. 渲染引擎 ([docs/src/snippets/\\_deployment.jsx](#)) : cellVerifyStatus 签名从 (c) 扩展为 (c, sel), 函数型 verificationStatus 用当前选择求值, 静态值沿用旧路径; 回退链保持 v ? c.verified, 未知字符串继续回退 unverified。唯一调用点同步传入 sel; kimi-k3.jsx 的 49 处静态字符串走未变路径。
4. 校验与作者配套: docs/scripts/check\_cookbook\_configs.mjs 新增对函数型 cells[].verificationStatus 的探针, 在完整选择空间上要求返回值属于 verified / in-progress / unverified 或 undefined; .claude/skills/cookbook-add-model/references/authoring-reference.md 与 templates/config.jsx.tmpl 同步记录该字段的三态与函数语义。
5. 测试配套: 本 PR 无自动化测试 (docs only), 但作者用变异验证了检查器——拼错 "in-progres" 或让谓词抛异常都会使检查失败, 真实配置通过。改动前检查器不覆盖 cells 的函数型字段, 本次一并闭合该缺口。

## [docs/src/snippets/\\_deployment.jsx](#)

cookbook 渲染引擎的核心改动: `cellVerifyStatus` 从静态字段读取扩展为支持按 selection 求值的函数型 verificationStatus, 是所有 cell 徽章渲染的公共路径。

```
// `cellVerifyStatus` 是 cookbook 渲染引擎中读取验证徽章的唯一入口。
// 旧实现只读静态字段; 新实现允许 `verificationStatus` 是 selection 的函数,
// 这样同一个 cell 可以按当前选项显示不同状态 (例如只有 DFLASH2 尚在验证)。
const cellVerifyStatus = (c, sel) => {
  if (!c) return "unverified"; // 没有匹配 cell 时保持“未验证”
  const v =
    typeof c.verificationStatus === "function"
      ? c.verificationStatus(sel) // 函数形式: 按当前选择求值
      : c.verificationStatus; // 静态形式: 字符串或布尔值
  return verifyStatusOf(v ?? c.verified);
};

// 唯一调用点: `sel` 是当前完整选择 (hw x 各维度选项), 与 `findCell`
// 的查找维度一致, 保证按选项展示 in-progress 徽章。
// `verifyStatusOf` 负责三态归一: 白名单字符串原样返回, 未知字符串
// 回退 `unverified`, 布尔值维持 verified 基线语义。
const verifyStatus = cellVerifyStatus(cell, sel);
```

## [docs/src/snippets/configs/Qwen/qwen3.8-27b.jsx](#)

配置主体改动: 移除命令内 hints 提示行, 改为在 H200、DGX Spark 等 cell 上按 `sel.spec === "dflash"` 返回 in-progress, 驱动本 PR 的徽章语义。

```
// DFLASH2 选项删除了原来的 `hints` (它把“验证中”注释渲染进复制命令),
// 验证状态改由 cell 级徽章表达。下面以 H200 / FP8 cell 为例: 该 cell
// 整体已验证 (verified: true), 但 DFLASH2 尚未在此平台端到端跑过,
// 所以 `verificationStatus` 写成 selection 的函数, 只在选到 dflash 时
```

```
// 把徽章降为 in-progress, 其余 Speculative Decoding 选项保持 Verified。
{
  match: { hw: "h200", variant: "default", quant: "fp8", nodes: "single" },
  verified: true,
  verificationStatus: (sel) =>
    sel.spec === "dflash" ? "in-progress" : "verified",
  env: [],
  flags: [
    "--trust-remote-code",
    // ... 其余启动参数由部署命令生成器按选择展开
  ],
}
```

## docs/scripts/check\_cookbook\_configs.mjs

CI 校验配套：新增对函数型 `cells[].verificationStatus` 的探针，防止拼写错误或抛异常的谓词流入渲染页面。

```
// 函数型 verificationStatus 会随选择渲染，因此必须像其他函数型字段一样
// 在完整选择空间上探测：返回值必须是三态白名单之一或 undefined，
// 否则打字错误会静默回退 "unverified"，直到浏览器渲染才暴露。
const VERIFY_STATES = ["verified", "in-progress", "unverified"];
for (const [i, cell] of (config.cells || []).entries()) {
  if (typeof cell.verificationStatus !== "function") continue;
  probe(
    (sel) => {
      const out = cell.verificationStatus(sel);
      if (out !== undefined && !VERIFY_STATES.includes(out)) {
        throw new Error(
          `returned ${JSON.stringify(out)}, expected one of [${VERIFY_STATES}]`,
        );
      }
    },
    `cells[${i}].verificationStatus`, // probe 的 label，失败时定位到具体 cell
  );
}
```

## 评论区精华

本 PR 没有 review 评论线程，唯一的 issue 评论来自 mintlify[bot] 的预览部署通知，reviewer zijiexia 直接批准（APPROVED）。设计权衡集中写在 PR body：

`cellVerifyStatus` 原本只读 `cell` 的静态字段，无法表达同一 `cell` 在不同选项下验证状态不同；静态 `in-progress` 会把其他平台的其他选项误标为未验证。

影响面论证：`verificationStatus` 只有 `cellVerifyStatus` 一个读取点，`_playground.jsx` 与 `check_cookbook_configs.mjs` 均不读取；`kimi-k3.jsx` 的 49 处静态字符串走未变路径；布尔 / 字符串 / `undefined` 行为与旧版一致，未知字符串仍回退 `unverified`，因此拼写错误不可能误显为绿色 Verified 徽章。

## 风险与影响

- 渲染引擎回归面：cellVerifyStatus 签名变化影响所有 cookbook 页面。当前调用点只有一处且已同步传入 sel，但未来新增调用点若漏传 sel，函数型 verificationStatus 会在 sel.spec 处抛 TypeError 导致页面崩溃；CI 探针覆盖完整选择空间，能兜住配置侧错误，却兜不住渲染侧调用遗漏。
- 拼写与回退语义：未知字符串回退 unverified 保证不会误显绿色徽章，但也会让 "in-progres" 这类拼写错误静默降级为“未验证”；此次新增探针会在 CI 阶段拦截，属于配套缓解。
- 文档时效性：两个 <Warning> 引用了具体 PR 号与失败信息，若 DFlash2 / NVFP4 要求被回滚或合入 release，页面需要同步跟进；命令注释里的 dev-nightly-0820 示例会过期（后续 #35767 即把该注释指回滚动 dev 镜像 tag）。
- 影响面：对用户是文档可读性与正确性的提升；对系统只影响 docs 渲染引擎与检查脚本，不触碰 SGLang 运行时；对团队新增了一个可函数化的文档配置字段，作者参考与模板同步更新，降低后续踩坑概率。

## 关联脉络

- 35663 引入 DFLASH2 cells，但把验证状态写成命令内 hints 行且未文档化构建要求，本 PR 是它的直接修正。
- 35371（DFlash2 选择器支持）与 #35496（NVFP4 量化 lm\_head）是文档中声明的最低版本前提，二者都晚于固定 tag 和最新 release。
- 35767 紧接本 PR 修正同页注释，把 dev-nightly-0820 示例改回滚动 dev 镜像 tag，两者构成同一 cookbook 文档的连续演进。
- 更大趋势：cookbook 部署矩阵正在把“验证状态”从静态标注升级为按选择动态表达（函数型 verificationStatus），并用 CI 探针在完整选择空间上兜底，docs 配置的工程化程度在持续提高。