

# PR #48973 完整报告

vllm-project/vllm

[Bugfix] Accept RFC 2397 parameters in base64 data URLs

合并时间: 2026-07-24 23:23

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/48973>

## 执行摘要

- 一句话: 修复带 RFC 2397 参数的 base64 数据 URL 被拒绝的问题
- 推荐动作: 建议在遇到类似多模态输入问题时参考此 PR 的方法论: 优先对齐 RFC 规范, 同时在测试中覆盖常见偏差。该 PR 的解析改进和错误消息清晰化值得借鉴。

## 功能与动机

PR body 指出 RFC 2397 明确允许在 mediatype 和 base64 标记间包含参数, 而当前实现只识别第一个分号, 导致合法数据 URL 被拒绝。客户端 (如 charset 参数) 无法发送多模态内容。

## 实现拆解

1. 拆分数据 URL 头部: 将 `split(",", 1)` 改为 `partition(",", 1)`, 以正确检测缺失逗号的情况, 并抛出 `ValueError`。
2. 识别 base64 标记: 使用 `rpartition(";")` 从右向左查找 `;`, 确保 `data:image/png;charset=utf-8;base64,...` 中的 `;base64` 被正确识别, 中间参数被忽略。
3. 清理媒体类型: 使用 `media_type.partition(";")[0]` 去除媒体类型后的参数部分, 避免将 `image/png;charset=utf-8` 等字符串传递给 `load_base64`。
4. 保留非 base64 路径: 当 `;` 不存在或 `encoding != "base64"` 时, 仍抛出 `NotImplementedError`, 保证不改写原有逻辑。
5. 新增测试覆盖: 在 `test_connector.py` 中添加 `test_fetch_image_data_url_with_params` 验证带参数 URL 的加载, 以及 `test_fetch_image_data_url_malformed` 验证缺失逗号、非 base64、大小写不匹配等异常情况。

关键文件:

- `vllm/multimodal/media/connector.py` (模块 多模态; 类别 source; 类型 core-logic) : 核心变更文件, 修改了 `_load_data_url` 方法以支持 RFC 2397 参数并改进错误处理。
- `tests/multimodal/media/test_connector.py` (模块 测试; 类别 test; 类型 test-coverage ; 符号 `test_fetch_image_data_url_with_params`, `test_fetch_image_data_url_malformed`) : 新增测试覆盖带参数 base64 URL 和格式错误处理, 确保修改的正确性和鲁棒性。

关键符号: `_load_data_url`, `test_fetch_image_data_url_with_params`, `test_fetch_image_data_url_malformed`

## 关键源码片段

### vllm/multimodal/media/connector.py

核心变更文件，修改了 `_load_data_url` 方法以支持 RFC 2397 参数并改进错误处理。

```
def _load_data_url(
    self,
    url: str,
    media_io: MediaIO[M],
) -> _M: # type: ignore[type-var]
    # Format per RFC 2397:
    # data:[<mediatype>][;<param>=<value>]*[;base64],<data>
    # 使用 partition 而不是 split，以检测缺少逗号的格式
    data_spec, sep, data = url[5:].partition(",")
    if not sep:
        msg = f"Invalid data URL {url[:32]!r}: missing ',' separator."
        raise ValueError(msg)

    # 使用 rpartition 取最后一个；为 base64 标记，
    # 忽略中间可能存在的 RFC 2397 参数
    media_type, sep, encoding = data_spec.rpartition(";")
    if not sep or encoding != "base64":
        msg = "Only base64 data URLs are supported for now."
        raise NotImplementedError(msg)

    # 移除媒体类型中的参数部分（例如 charset=utf-8），
    # 避免将含参数的字符串传给 load_base64
    media_type = media_type.partition(";")[0]
    return media_io.load_base64(media_type, data)
```

### tests/multimodal/media/test\_connector.py

新增测试覆盖带参数 base64 URL 和格式错误处理，确保修改的正确性和鲁棒性。

```
def test_fetch_image_data_url_malformed():
    # 测试多种不符合 RFC 2397 的 data URL 格式
    connector = MediaConnector()

    # 缺少逗号的分隔符
    with pytest.raises(ValueError, match="missing ','"):
        connector.fetch_image("data:image/png;base64")

    # 非 base64 的 data URL（普通百分号编码）
    with pytest.raises(NotImplementedError, match="base64"):
        connector.fetch_image("data:text/plain,hello")

    # ； 前面的 base64 标记缺少分号，被视为格式错误
    # "data:base64,..." 中 "base64" 位于媒体类型位置
    with pytest.raises(NotImplementedError, match="base64"):
        connector.fetch_image("data:base64,aGVsbG8=")
```

```
# RFC 2397 规定 base64 标记必须为小写，无空格
with pytest.raises(NotImplementedError, match="base64"):
    connector.fetch_image("data:image/png;BASE64,aGVsbG8=")

with pytest.raises(NotImplementedError, match="base64"):
    connector.fetch_image("data:image/png; base64,aGVsbG8=")
```

## 评论区精华

review 中 [DarkLight1337](#) 询问 RFC 是否允许空白字符。[cjackal](#) 引用 RFC 指出不允许空白，且仅允许小写 base64。[thomas-fahrner-parasail](#) 补充了 WHATWG Fetch Standard 的宽松处理，但最终团队决定采用严格 RFC 语义以保持简单，[thomas-fahrner-parasail](#) 据此更新了实现与测试。

- 是否应允许 base64 标记周围的空白字符？(design): 采用严格 RFC 语法：base64 标记必须为小写且无周围空白，不符合的均抛出 `NotImplementedError`。

## 风险与影响

- 风险：本次变更仅影响多模态输入中的数据 URL 解析路径，非核心推理逻辑，风险较低。主要风险包括：边缘格式（如大小写、空格）可能被拒绝，但测试已明确覆盖这些场景，且严格语法符合 RFC 规范，避免了因接受非标准格式导致的潜在兼容性问题。没有引入新的依赖或性能开销。
- 影响：影响范围仅限于通过 `data: scheme` 传递多模态内容的用户，之前被拒绝的带参数 base64 URL 现在可用。对系统性能无影响，对团队协作无影响，因为变更独立且测试充分。
- 风险标记：暂无

## 关联脉络

- 暂无明显关联 PR