Museのスペースに大きな画像を登録する方法

文字データ転送の制限と、分割アップロードによる解決

起きた問題

Museに作ってもらったWebアプリ（スペース）に、生成した画像をアプリ内のギャラリーへ登録しようとしたところ、実サイズの画像を登録できませんでした。登録できたのは生成条件などの文字情報だけで、画像そのものは渡せなかったのです。

原因

Museがアプリの機能（アクション）へデータを渡すとき、画像のようなバイナリデータも、Base64という形式の文字列に変換して「文字データ」としてやり取りします。

- Base64に変換すると、データ量は元の約1.33倍に増えます
- 1回の送信で扱える文字量には限りがあります
- そのため、数MB級の画像は数百万文字の文字列になり、そのままでは渡しきれません。プレビュー用に縮小しても数万〜十数万文字になり、一度に正確に転送するのは現実的ではありませんでした

なお、生成された画像の原画ファイル自体は、Museの環境内にファイルとして保存され、ライブラリからいつでも開けます。失敗していたのは、あくまでアプリ内ギャラリーへの「控え」の登録だけです。

解決方法：分割アップロード

大きなデータを小さな断片に分けて送り、受け取る側で結合して元に戻す「分割アップロード（chunked upload）」という、ファイル転送では一般的な手法を使います。

1. アプリ側に、データの受け取りを「開始 → 断片の追記 → 完了（結合）」の3段階で行う仕組みを用意する
2. Museは画像をプレビュー用に軽量化（縮小やWebP化など）したうえで、Base64文字列を一定の文字数ごとに分割し、順番に送信する
3. 完了時に、送信前と結合後のデータの指紋（SHA-256ハッシュ）を照合し、欠落や文字化けがないことを確認する

運用のポイント

- **住み分けが現実的**：ギャラリーにはプレビュー版、原画はファイルとして保存、という形が扱いやすいです。原画まで分割転送しようとすると断片が100個以上になり、作業量が膨らみます

- **画質とのバランス**：プレビューは多少の画質劣化を許容し、軽く作るのが向いています
- **トークン消費について**：保存データが増えてもトークン使用量は減りません。消費は保存量ではなく、読み書きや転送など「処理した量」に応じて増えます。分割転送の作業自体は、文字として経由する量ぶんのトークンを消費しますので、その意味でもプレビューは軽いほど有利です

まとめ

- Museからアプリへ画像を渡すときは、文字データ（Base64）としてやり取りされるため、大きな画像は一度に渡せない
- 対策は、受け側に分割受け取りの仕組みを用意し、軽量化したプレビューを断片に分けて送り、ハッシュで完全性を確認すること
- 原画はファイルとして別に保存されるため、画質面の心配はない