

Blueskyに伏せ字投稿ができるWebアプリの作り方

2024-02-21 [Yasuyuki ENDO](#)

GitHubリポジトリ

<https://github.com/eyasuyuki/bluespoiler/>

説明すること

- どうやって伏せ字投稿するのか
- なぜ作ろうと思ったのか
- なぜWebサービスやモバイルアプリとしてリリースしなかったのか
- なぜFlutterで書いたのか
- 画像サイズチェック
- Blueskyへの接続とテスト
- 記事URLの推測
- 今後の展開

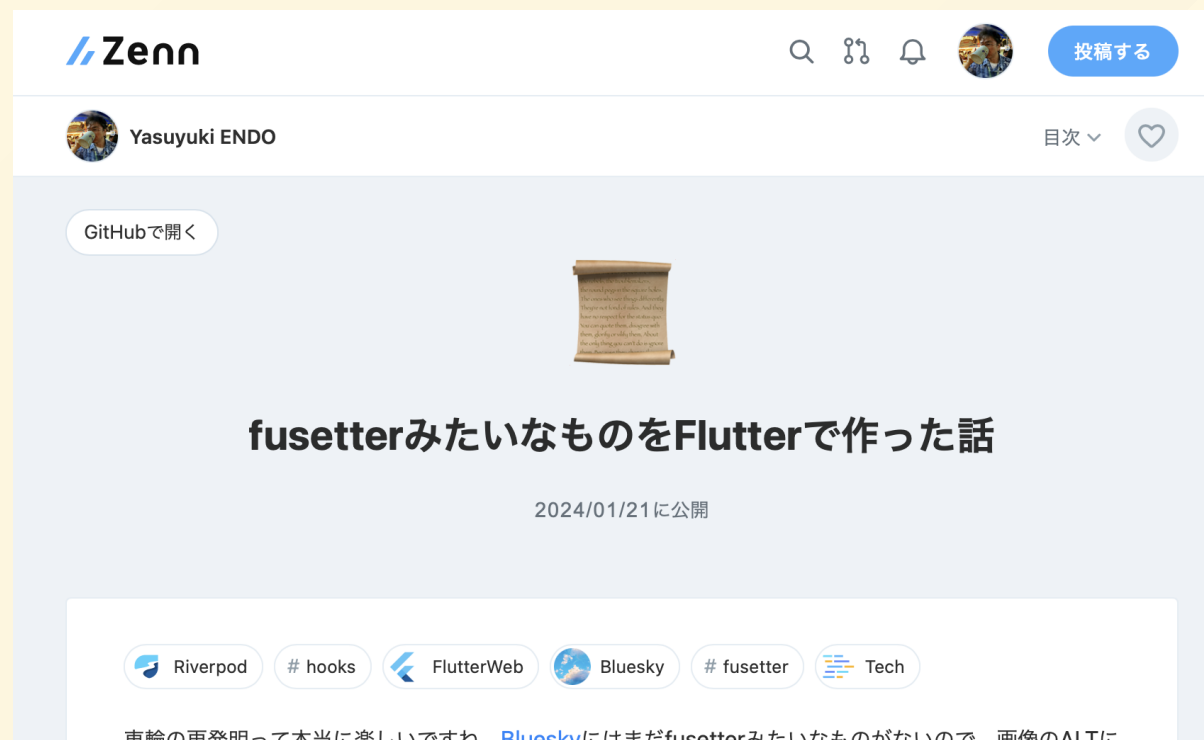
説明しないこと

- fusetterの伏せ字アルゴリズムを勝手に推測する
- Flutterのライブラリの使い方
 - Riverpod
 - Freezed
 - `go_router`
 - `flutter_hooks`
 - `image_picker_web`
- Rich TextでURLをリンクさせる

詳しくはZennに書きました。

<https://zenn.dev/eyasuyuki/articles/825b28b0ec0a4c>

※ イイネください....



どうやって伏せ字投稿するのか

- <https://eyasuyuki.github.io/bluespoiler/>
- [と]の間の文が○になる(fusetterの再発明)
- ネタバレの本文は画像のALTに入る

伏せ字投稿

ネタバレALT



投稿



ENDO Yasuyuki

@javaopen.org

シベリア少女鉄道 vol.37『持続可能彼女』シアターサンモール

一応表向きには地球に帰還できなくなった宇宙船での4人の女性乗組員の会話劇なのだが、
oooooooooooooooooooooooooooo
oooooooooooooooooooooooooooo舞台上の

oooooooooooooooooooooooooooo
oooo。舞台上の俳優たちも
ooooooooooooooooooooooooooooしたりする。メタ演劇と呼ぶにはあまりにも素朴な不思議な舞台だった。ooooは
ooooooooooooooooooooooooooooで彼女のooooを見てもこの
ooooooooooooooooooooさされていてさすがだと思った。



返信を書く



シベリア少女鉄道 vol.37『持続可能彼女』シアターサンモール

一応表向きには地球に帰還できなくなった宇宙船での4人の女性乗組員の会話劇なのだが、この劇場には風間さなえ演じる風間さなえが住み着いており本番中に舞台を横切ったり舞台上の小道具を拝借して日常生活を行ったりとやりたい放題。舞台上の俳優たちも借り暮らしの彼女に気づいて彼女の生活を応援したりする。メタ演劇と呼ぶにはあまりにも素朴な不思議な舞台だった。風間さなえは配役表にも書かれていない完全なサプライズで彼女のインスタを見てもこの舞台の稽古のことは完全に隠蔽されていてさすがだと思った。

📷 保存

🔗 共有

デモ

<https://youtu.be/eZX9s03MDQc?si=bgW2A1EWXn87TGWZ>

なぜ作ろうと思ったのか

- Blueskyにはfusetterがないので
- だがfusetterみたいに自前サーバーを持つのは面倒
- 画像のALTなら1000文字まで入れられる(邪道)
- 自前バックエンドは持たずにBlueskyとだけ通信すればいい

なぜWebサービスやモバイルアプリとしてリリースしなかったのか

- Webサービスは維持費がかかるし無料枠だとしても維持するのが大変
- 当初はモバイルアプリ版も作ろうと思った
- Webアプリとして公開してiPhoneのホーム画面に追加してみた。アプリ必要ないじゃん
- 失敗したのは何もログを取っていないこと(ユーザーにとってはメリット)

なぜFlutterで書いたのか

- 当初はモバイルアプリ版も出そうと思っていた
- ReactやVueなどWebクライアントサイドの流行を追うのは大変
- FlutterならモバイルもWebもデスクトップも同時開発できる
- Flutterは2年ぐらい触ってないのでキャッチアップしたかった
- Bluesky.dartの情報がすぐに見つかった

画像サイズチェック

- 976.56KBを超える画像をアップロードするとエラーになる
- 976.56KB以下に圧縮を試みたがWebクライアントサイドでは無理そう
- 単に警告するだけにした

画像選択部分のコード

```
Future<void> pickImage() async {  
  const maxImageSize = 999997;  
  try {  
    Uint8List? uint8list = await ImagePickerWeb.getImageAsBytes();  
    if (uint8list!.lengthInBytes > maxImageSize) {  
      ScaffoldMessenger.of(context).showSnackBar( // 画像サイズ超過はSnackBarで警告する  
        SnackBar(  
          content: Text(AppLocalizations.of(context)!.image_size_text),  
          backgroundColor: Colors.redAccent,  
        ), // SnackBar  
      );  
    } else {  
      imageBytes.value = uint8list; // ImagePickerWebで選択した画像データをとっておく  
    }  
  } catch (e) {  
    print(e);  
  }  
}
```

※ Dartは **new** を省略できます。

Blueskyへの接続とテスト

Articleクラス

```
@freezed
class Article with _$Article {
  const factory Article({
    required String id,
    required String password,
    required String body,
    required String lang,
    required String alt,
    @Uint8ListConverter() Uint8List? image,
  }) = _Article;

  factory Article.fromJson(Map<String, dynamic> json) => _$ArticleFromJson(json);
}
```

Blueskyへの投稿

```
@riverpod
Future<PostResult> postArticle(PostArticleRef ref, {required Article article}) async {
  try {
    final session = await bsky.createSession(identifier: article.id, password: article.password);
    final bluesky = bsky.Bluesky.fromSession(session.data);
    final uploaded = await bluesky.repo.uploadBlob(article.image!);
    final post = await bluesky.feed.post(
      text: article.body,
      languageTags: [
        article.lang,
      ],
      embed: bsky.Embed.images(
        data: bsky.EmbedImages(
          images: [
            bsky.Image(
              alt: article.alt,
              image: uploaded.data.blob,
            ), // bsky.Image
          ],
        ), // bsky.EmbedImages, bsky.Embed.images
      ),
      facets: createFacets(article.body, extractUrl(article.body)),
    );
  }
```

テスト

- とりあえず本当に接続してテストする
- 本物のID/パスワードを隠蔽するために `flutter_dotenv` を使う
- `.gitignore` に `.env` を追加する
- プロジェクトルートに `.env` ファイルを作成

```
BLUESKY_IDENTIFIER=eyasuyuki@gmail.com  
BLUESKY_PASSWORD=
```


テストmain

```
void main() async {  
  await dotenv.load(fileName: '.env');  
  final id = dotenv.get( 'BLUESKY_IDENTIFIER' );  
  final password = dotenv.get( 'BLUESKY_PASSWORD' );  
}
```

画像投稿のテスト

```
test('test postArticle', () async {
  final container = ProviderContainer();
  // image data
  final file = File('assets/test.jpg');
  final img = await file.readAsBytes();
  // test loading
  Article article = Article(
    id: id,
    password: password,
    body: '画像投稿テスト',
    lang: 'ja',
    alt: 'テストの文字列',
    image: img);

  // test loading
  expect(
    container.read(postArticleProvider.call(article: article)),
    const AsyncValue<PostResult>.loading()
  );
  // test post
  final result = await container.read(postArticleProvider.call(article: article).future);
  expect(result, isNotNull);
  expect(result.status, equals('ok'));
  expect(result.url, isNotNull);
});
```

※ 本当に投稿されます。

記事URLの推測

BlueskyにPostすると例えばこんな結果が返る。

```
{"uri": "at://did:plc:dptps7rgxju4nrg6qskop2wz/app.bsky.feed.post/3kjmvo7ocl62e",  
  "cid": "bafyreihxcb3n6hxucw3hdeb3kylhzkoumitkx4hygoy24tyxt4hemajdde"}
```

Blueskyの記事URLはこんな感じ。

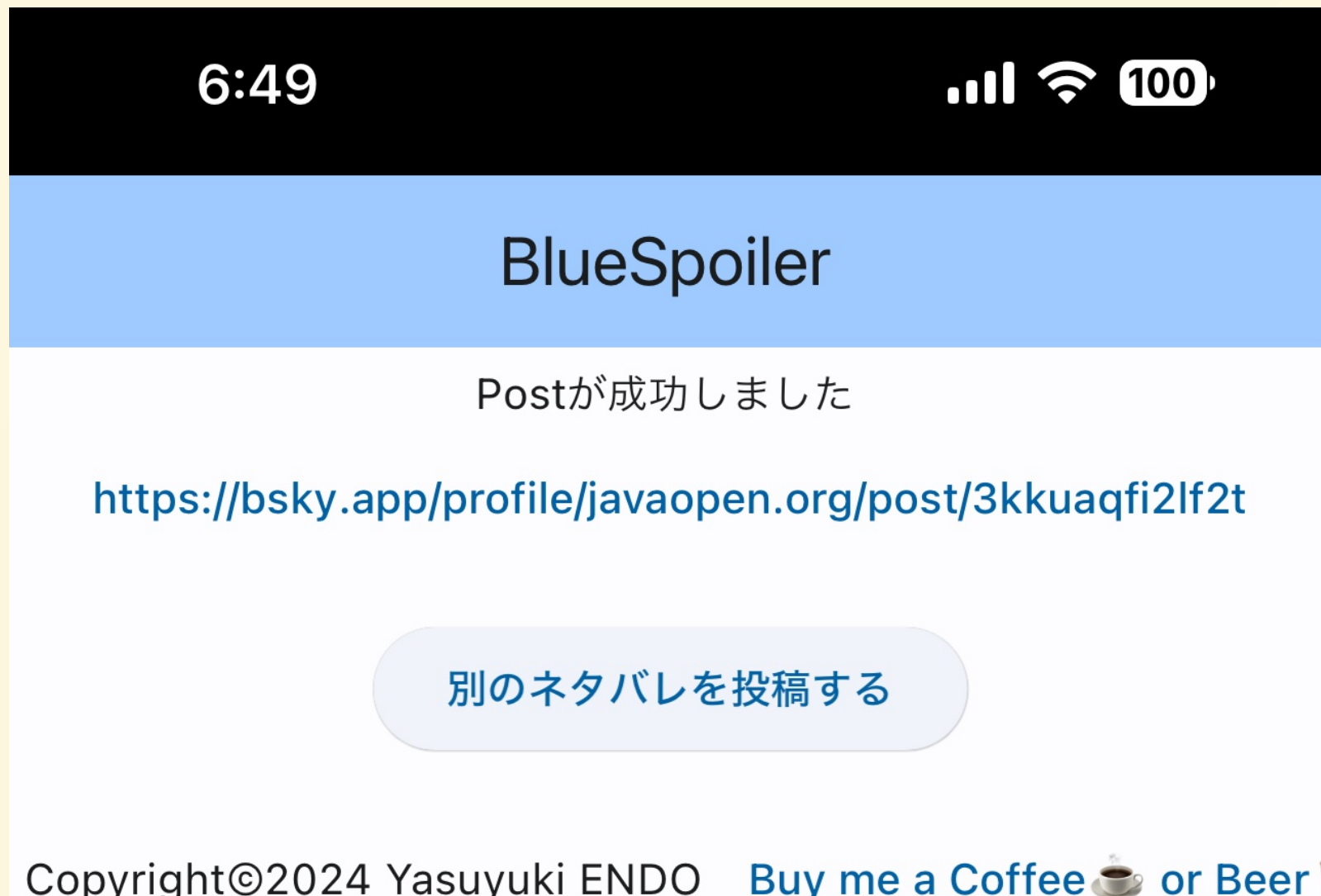
<https://bsky.app/profile/javaopen.org/post/3kjmvo7ocl62e>

- `3kjmvo7ocl62e` が記事のIDっぽい
- `javaopen.org` の部分は `session.data.handle` で取得できる

記事URLの生成

```
final articleId = post.data.uri.href.split('/').last;
final url = Uri(
  · scheme: 'https',
  · host: 'bsky.app',
  · pathSegments: <String>[
    · · 'profile',
    · · session.data.handle,
    · · 'post',
    · · articleId,
  · ], // <String>[]
); // Uri
```

投稿後の画面



今後の展開

- 認証方法の改善(Bluesky待ち)
- ログをとる(Google Analytics?)
- 伏せ字投稿ランキング?

発表させていただきありがとうございました。