エラーから原因を特定する思考プロセス

📋 発生したエラー

× [TypeError: Failed to parse URL from undefined/customers]

digest: '2285087383',

[cause]: [TypeError: Invalid URL]

code: 'ERR_INVALID_URL',

input: 'undefined/customers'

🤔 Step 1: エラーメッセージから仮説を立てる

エラーから読み取れること

1. **`Failed to parse URL`** → URLの解析に失敗している
2. **`undefined/customers`** → 何かが`undefined`になっている
3. **`Invalid URL`** → 不正なURL形式

🔍 初期仮説の立て方

エラーの種類を分類:

├─ GETリクエストが飛んでない → そもそも接続できていない？

├─ 到達はしているけどエラー → データ型や定義の問題？

└─ データは取得できているけど表示されない → データの受け渡し問題？

✅ 今回の仮説

「**`undefined`**という文字列が入っているということは、何かの変数が未定義のまま文字列結合されている」

🔎 Step 2: エラーが出た場所を特定する

エラーログの詳細を見る

POST /customers/create 500 in 43ms

→ **`/customers/create`** のPOSTリクエストで500エラー

該当ファイルを探す

frontend/src/app/customers/create/

├── page.jsx ← フォームのUI

├── createCustomer.js ← API呼び出しロジック

└── confirm/ ← 確認画面

📂 Step 3: 関係しそうな関数を確認

3-1. エラーが出たファイルを開く

`createCustomer.js` を確認:

```
const res = await fetch(
  process.env.NEXT_PUBLIC_API_ENDPOINT + `/customers`,
  {
    method: "POST",
    headers: { "Content-Type": "application/json" },
    body: body_msg,
  }
);
```

🎯 怪しいポイント発見!

process.env.NEXT_PUBLIC_API_ENDPOINT + `/customers`

↑

これが undefined になってる？

🧪 Step 4: 仮説を検証する

検証方法1: console.log で確認

```
console.log("API_ENDPOINT:", process.env.NEXT_PUBLIC_API_ENDPOINT);

// 出力: API_ENDPOINT: undefined

const url = process.env.NEXT_PUBLIC_API_ENDPOINT + `/customers`;

console.log("完成したURL:", url);

// 出力: 完成したURL: undefined/customers
```

ビンゴ! 環境変数が設定されていない

検証方法2: VSCodeの「定義へ移動」を使う

1. `process.env.NEXT_PUBLIC_API_ENDPOINT` の上で右クリック
2. 「定義へ移動」を選択
3. → 定義が見つからない = 環境変数が未設定

検証方法3: 他のファイルでも使われているか確認

`fetchCustomers.js` を見る:

```
const res = await fetch(

  process.env.NEXT_PUBLIC_API_ENDPOINT + "/allcustomers",
```

{

cache: "no-cache",

}

);

→ 同じ環境変数を使っている = 全体的な設定の問題

💡 Step 5: 原因の特定

根本原因

環境変数 **`NEXT_PUBLIC_API_ENDPOINT`** が設定されていない

なぜこれが問題？

Next.jsでは:

- 環境変数は `.env.local` や `.env` ファイルで設定する
- `NEXT_PUBLIC_` プレフィックスはクライアント側で使える変数
- 設定されていない変数は `undefined` になる

エラーの流れ

1. createCustomer.js で fetch() を実行

↓

2. process.env.NEXT_PUBLIC_API_ENDPOINT が undefined

↓

3. undefined + "/customers" = "undefined/customers"

↓

4. new URL("undefined/customers") が失敗

↓

5. TypeError: Invalid URL

🔧 Step 6: 解決策の実装

解決方法: `.env.local` ファイルを作成

```
# frontend/.env.local を作成

NEXT_PUBLIC_API_ENDPOINT=http://localhost:8000
```

ファイル配置

```
frontend/

├── .env.local        ← ここに作成

├── src/

├── package.json

└── ...
```

確認方法

1. サーバーを再起動 (環境変数を読み込むため)

```
npm run dev
```

2. ブラウザのコンソールで確認

```
console.log(process.env.NEXT_PUBLIC_API_ENDPOINT);

// 出力: http://localhost:8000
```

📊 思考プロセスのフローチャート

エラー発生

↓

[Step 1] エラーメッセージを読む

├─ 何が失敗した？ → URL解析

├─ どこで失敗した？ → undefined/customers

└─ なぜ失敗した？ → 変数が未定義

↓

[Step 2] 仮説を立てる

「環境変数が設定されていないのでは？」

↓

[Step 3] 該当箇所を確認

├─ エラーログから該当ファイルを特定

│ → createCustomer.js

├─「定義へ移動」で変数の定義を探す

│ → 見つからない = 未設定

└─ 他のファイルでも使われているか確認

→ fetchCustomers.js でも同じ変数

↓

[Step 4] 原因を特定

「.env.local ファイルが存在しない」

↓

[Step 5] 解決策を実装

.env.local を作成して環境変数を設定

↓

[Step 6] 検証

サーバー再起動 → 動作確認 → ✅ 解決!

🎯 効率的なデバッグのポイント

1. エラーメッセージを丁寧に読む

- 何が 失敗したか (`Failed to parse URL`)
- どこで 失敗したか (`undefined/customers`)
- なぜ 失敗したか (推測: 変数が未定義)

2. 仮説を立ててから調べる

❌ 悪い例: とりあえず全ファイルを見る

✅ 良い例: エラーから仮説を立てて、関連ファイルだけ見る

3. VSCodeの機能を活用

- 「定義へ移動」: 変数や関数の定義元を探す
- 「すべての参照を検索」: その変数がどこで使われているか確認
- 「ファイル内検索」: 特定の文字列を含むファイルを探す

4. 最小限のエネルギーで進める

エラー箇所を特定

↓

関連する関数だけ見る

↓

定義へ移動で依存関係を確認

↓

原因を特定

↓

ピンポイントで修正

5. console.log を活用

// デバッグ用のログを入れる

console.log("変数の値:", 変数);

console.log("ここまで到達した");

🔄 同じ思考プロセスを他のエラーにも適用

例: GETリクエストが飛ばない場合

仮説1: そもそも接続できていない？

→ バックエンドが起動しているか確認

→ ポート番号は合っているか確認

仮説2: リクエストは飛んでいるけどエラー？

→ ネットワークタブでステータスコード確認

→ 4xx → クライアント側の問題

→ 5xx → サーバー側の問題

仮説3: データは取得できているが表示されない？

→ console.log でレスポンスを確認

→ データの型が期待と違う？

→ 表示ロジックの問題？

🎓 今回の学び

技術的な学び

- Next.jsの環境変数は `.env.local` で設定
- `NEXT_PUBLIC_` プレフィックスはクライアント側で必要
- 環境変数の変更後はサーバー再起動が必要

思考プロセスの学び

- エラーメッセージは宝の山 → 丁寧に読む
- 仮説駆動 → 闇雲に調べない
- ツールを活用 → VSCodeの機能で効率化
- 最小限の労力 → 関係ない部分は見ない