

## 概要 (What / Why)

スプレッドシート（以下「スプシ」）の重い日別集計を **PHP側でオフロード**し、スプシはデータ倉庫として利用。集計結果はJSONでキャッシュし、表示は軽量ページで行う構成。併せて、**演者（動画担当）/セールス担当のマスタ**をWebから安全に編集できる小さなCRUDを提供。

- **目的:** スプシ表示の体感速度を改善（式ゼロ化/値化）、運用の自動化、将来的な拡張を容易に。
- **方針:** vendor不使用（Google公式SDKなし）。JWTでアクセストークン取得→Sheets REST APIを生でもつ。
- **状態:** 取り込み・マスタ管理は完成。表示ページは後続（保留）。

## 全体アーキテクチャ (How)

Google Sheets（原データ）

└─ [REST API]

└─ PHP（取り込み import / 集計 aggregate）

└─ cache/\*.json（集計/取り込みキャッシュ）

└─ data/\*.json（マスタ：actors/sales）← WebでCRUD

└─ display.php（表示：保留中）

- **取り込み:** `sheets.values.get` をA1レンジで一括取得。1行目=ヘッダ、2行目~データ（configで行番号可変）。- **集計:** PHPで日付正規化/数値化→日別に合算・件数化→ `cache/daily_summary.json` に出力。- **表示:** JSONを読むだけの超軽量ページ（Chart.js想定）。※現時点は実装保留。- **マスタ管理:** `actors.json`（演者）/`sales.json`（セールス）をWebでCRUD。APIはトークン必須+原子的保存。

## ファイル構成 (現状)

/your-app

└─ config.php	# 基本設定（ID/鍵/パス/行開始など）
└─ google_jwt.php	# JWT生成&トークン取得（OpenSSL, cURL）
└─ sheets_rest.php	# Sheets REST呼び出し（values.get）
└─ import_sheets.php	# 取り込み（複数シート→ヘッダ付き連想配列）
└─ pull_raw.php	# 動作確認用：取り込み→raw_rows.json保存
└─ aggregate_daily.php	# 集計（セールス日ベース）→ daily_summary.json
└─ display.php	# 表示（※今は保留。JSONを読む想定）
└─ api_people.php	# actors/sales CRUD API（GET/POST）
└─ people_store.php	# JSONローダ/セーブ（flock/renameで原子的）
└─ admin_people.php	# 管理UI（ヘッダ/フッタ/静的資産分離）
└─ partials/	
└─ header.php	# 共通ヘッダ（\$extraHead/\$extraFoot対応）
└─ footer.php	# 共通フッタ
└─ public/	
└─ styles.css	# 共通スタイル

```

├── admin_people.js      # 管理UIのJS
├── data/                # ※Web直アクセス不可に（.htaccess等）
│   ├── actors.json     # 演者マスタ（schema下記）
│   └── sales.json      # セールスマスタ（schema下記）
└── cache/
    ├── raw_rows.json   # 取り込みキャッシュ
    └── daily_summary.json # 日別集計キャッシュ

```

## 設定（config.php）

- SPREADSHEET\_ID : 原データのスプレッドシートID
- SOURCES[] : 取り込み対象（sheet, range）
- TIMEZONE : 例 Asia/Tokyo
- SERVICE\_ACCOUNT\_JSON : サービスアカウント鍵のパス（Web外保管）
- RAW\_CACHE\_FILE : /cache/raw\_rows.json
- TOKEN\_CACHE\_FILE : /cache/oauth\_token.json（任意）
- HEADER\_ROW : 1（ヘッダ行）
- DATA\_START\_ROW : 2（データ開始行）
- DATA\_DIR : /data（マスタJSON置き場）
- ADMIN\_TOKEN : 管理API用の長いトークン

## データ仕様（JSONスキーマ）

### マスタ：actors.json / sales.json（共通）

```

{
  "kind": "actors" | "sales",
  "updated_at": "ISO-8601",
  "items": [
    {
      "id": "uuid-v4",
      "name": "氏名(必須)",
      "kana": "かな",
      "email": "",
      "tags": ["string"],
      "active": true,
      "note": "備考"
    }
  ]
}

```

- 保存は people\_store.php の ds\_save\_atomic() が flock + rename で原子的に行う。

## 取り込みキャッシュ：raw\_rows.json

```
{
  "generated_at": "ISO-8601",
  "headers": ["..."],
  "rows": [{"ヘッダ": "値", "...": "..."}]
}
```

## 集計キャッシュ：daily\_summary.json（セールス日ベース）

```
{
  "generated_at": "ISO-8601",
  "headers": ["日付", "セールス件数", "成約数", "失注数", "保留数", "入金合計"],
  "rows": [
    {"日付": "2025-09-01", "セールス件数": 10, "成約数": 3, "失注数": 5, "保留数": 2, "入金合計": 150000}
  ],
  "base": "sales_date"
}
```

## API仕様（admin用・同一オリジン想定）

共通: ヘッダ `X-Admin-Token: <ADMIN_TOKEN>` 必須。

**GET** `/api_people.php?kind=actors|sales`

• 200 OK `{ kind, updated_at, items: [...] }`

**POST** `/api_people.php?kind=actors|sales`

• Content-Type: application/json

• Body:

• Upsert: `{ "action": "upsert", "item": { id?, name*, kana, email, tags[], active, note } }`

• Delete: `{ "action": "delete", "id": "uuid" }`

• 200 OK `{ ok: true, item? }`

認証エラー: 401 `{ error: "unauthorized" }` / パラメータ不正: 400

## 画面仕様（admin\_people.php）

- タブ：演者 / セールス
- 各タブに「新規追加/編集フォーム」＋「一覧テーブル」
- 保存 → Upsert（新規は先頭に挿入／既存は上書き）
- 削除 → Confirm後にDelete
- トークンはJSに直書きせず、`.container` の `data-admin-token` 属性からJSへ渡す

- CSS/JSは分離： `public/styles.css`、 `public/admin_people.js`
- ヘッダ/フッタ分離： `partials/header.php` / `partials/footer.php`。共通CSSはheaderに固定、ページ固有は `$extraHead/$extraFoot` で追加可。

---

## セキュリティ/運用

- `/data` と `/secrets` は **Web直アクセス禁止**（.htaccess or サーバ設定）
- `ADMIN_TOKEN` は十分に長い乱数に。**Basic認証/IP制限**の併用が望ましい
- サーバ時刻ズレはトークン取得失敗要因（NTP同期）
- サービスアカウントのメールを対象スプシに共有（閲覧者 or 編集者）

---

## セットアップ/運用手順

1. GCPでプロジェクト作成→**Sheets API有効化**→**サービスアカウント作成**→JSON鍵DL
2. スプシをサービスアカウントのメールで共有（閲覧/編集権）
3. `config.php` を設定、`/data` と `/cache` を作成（書込権限）
4. 取り込みテスト: `php pull_raw.php` (`raw_rows.json` 生成)
5. 集計: `php aggregate_daily.php` (`daily_summary.json` 生成)
6. 管理UI: `admin_people.php` を開き、演者/セールスをCRUD
7. 定時実行（例：5分ごと or 00:05）

```
*/5 * * * * /usr/bin/php /var/www/your-app/aggregate_daily.php >> /var/log/agg.log
2>&1
```

---

## 今後の拡張（ToDo）

- **流入json**（媒体/キャンペーン/キーワード等）を同じCRUD枠組みで追加
- **display.php 実装**（`daily_summary.json` をグラフ/テーブル表示）
- **差分集計**（前回の最終日以降のみ集計し追記）
- **集計定義の外出し**（合計/平均/最大/ユニーク等を `metrics.php` に）
- **API公開** `/api/daily?from=YYYY-MM-DD&to=...`（外部/別アプリから再利用）

---

## 新しいチャットに渡す“導入プロンプト”例

これは社内のPHPツール（vendorなし）で、スプシの取り込み～日別集計～マスタ管理（演者/セールスCRUD）までを行うミニアプリです。下に仕様書があります。これを踏まえて、①流入jsonのCRUD追加、②`daily_summary.json`の指標を追加（平均単価、担当別成約数など）、③`display.php`の実装（`Chart.js`）を一緒に進めたいです。必要なファイルとコードを差分で提示してください。

— 仕様書ここから —（このドキュメント全文を貼り付け）— 仕様書ここまで —

---

## 変更履歴

- v1.0: 初版（取り込み/集計/CRUD/分割レイアウトの仕様を確定）