

Données statiques et pipeline import/export

Vue complète

La static data n'est pas maintenue uniquement à la main dans Unity. Le dépôt contient un pipeline qui peut lire Google Sheets, générer les JSON du client puis publier la même donnée dans BrainCloud.

```
flowchart LR
  S[Google Sheets] --> I[exports/importer.js]
  I --> J[Assets/GameSpecific/.../Resources/data_*/*.json]
  J --> E[exports/exportAllData.js]
  E --> BCE[BrainCloudExporter.js]
  BCE --> CMD[_command/update_static_data]
  CMD --> BC[(Static Data BrainCloud)]
  J --> U[Unity Resources]
```

importer.js

`exports/importer.js` télécharge les feuilles via Google Sheets API, nettoie les headers, infère ou force les types, convertit `string/double/long/bool/json` et écrit les JSON. Un header peut forcer son type avec une notation comme `(str)`, `(long)`, `(double)`, `(bool)` ou `(json)`.

exportAllData.js

`exports/exportAllData.js` contient la carte des jeux, Spreadsheet IDs, GIDs et fichiers additionnels. Il connaît les jeux By Year, Acc1/Acc2, le dataset partagé `byModule`, les tables fiscales annuelles et les données de bots.

Pour un jeu Accounting, la pipeline met à jour les données propres au jeu puis le dataset partagé `byModule`. Les données sont écrites sous `../game/Assets/GameSpecific/<gameUid>/Resources/data_<gameUid>/`.

Publication BrainCloud

`BrainCloudExporter.exportJson()` initialise un wrapper BrainCloud et appelle `_command/update_static_data`. Certaines sources utilisent `separateTabs` et un `collectionType` dédié, notamment les chunks fiscaux.

Données By Module partagées

Le dataset `byModule` inclut notamment accounts, account types, taxes, employer taxes, interests, immobilizations, contract regularizations, ainsi que des fichiers additionnels comme `modules`, `module_pages`, `regions_settings`, `act_settings`, `modules_settings`, `pages_settings` et `modules_contexts`.

Checklist de modification

1. Identifier si la source officielle est Google Sheets ou un JSON additionnel.
2. Modifier la source correcte.
3. Importer/générer le JSON.
4. Inspecter le diff du JSON Unity.
5. Publier vers le bon environnement BrainCloud si nécessaire.
6. Tester le chargement Unity et le runtime serveur.
7. Pour By Module, vérifier aussi les filtres régionaux et settings décrits dans la page de configuration Accounting.

Ajouter un nouveau dataset Google Sheets de A à Z

1. Définir la source dans exportAllData.js

Ajouter le mapping du jeu vers la Spreadsheet et le GID/tab attendu. Si la donnée appartient au socle Accounting partagé, déterminer si elle doit vivre sous `byModule` plutôt que sous `acc1/acc2`.

2. Définir le nom du JSON Unity

Le nom doit correspondre à ce que `StaticData*.Initialize()` charge. Une nouvelle feuille importée mais jamais référencée par le loader ne sera jamais utilisée.

3. Définir les types de colonnes

Dans Google Sheets, utiliser les suffixes forcés `(str)`, `(long)`, `(double)`, `(bool)` ou `(json)` lorsque l'inférence est ambiguë. Une erreur ici peut produire un JSON valide mais un mauvais type runtime.

4. Importer localement

Depuis `node.js/exports/` :

```
importDataGoogleSheet.bat  
# ou directement  
node exportAllData.js importAllGoogleSheet <gameUid>
```

Le batch valide une liste de GameUid; si un nouveau GameUid/dataset n'y apparaît pas, mettre aussi ce batch à jour.

5. Inspecter le diff JSON

Ne jamais pousser l'export sans regarder le diff. Vérifier notamment UID, types numériques, booléens, tableaux JSON, lignes supprimées et ordre des données importantes.

6. Publier dans BrainCloud

```
exportToBrainCloud.bat  
# ou  
node exportAllData.js exportAllBrainCloud <gameUid>
```

Le chemin passe par `BrainCloudExporter.js` puis `_command/update_static_data`.

7. Valider les deux côtés

- Unity : le JSON est présent dans le bon `Resources/data_*` et le contrôler le charge.
- Serveur : `utils_game_specific_base/``utilsByGameUid` retourne la nouvelle donnée.
- Accounting : tester les overrides région/module/page si le dataset est filtré par settings.

Ne pas éditer le JSON généré comme source officielle

Si le fichier vient d'une Google Sheet, une modification manuelle du JSON sera écrasée au prochain import. Modifier la Sheet puis régénérer. Les seuls JSON maintenus directement sont ceux explicitement traités comme fichiers additionnels par `exportAllData.js`.

Revision #2

Created 2026-09-18 19:10:06 UTC by Mathis

Updated 2026-09-18 20:09:24 UTC by Mathis