

Référence des utils serveur BrainCloud

Carte générale

Fichier/dossier	Responsabilité
<code>utils.js</code>	Helpers transversaux, UUIDs, calculs simples, copie/merge, équipes/étudiants, rôles, CRC, logging.
<code>utils_documents.js</code>	Lecture, query, création, sauvegarde, suppression, increments et versioning des Custom Entities.
<code>utils_message.js</code>	Envoi de messages RTT à user/team/sim/activity et messages spécialisés.
<code>utils_eval.js</code>	Stubs et scores d'évaluation communs.
<code>utils_rankings.js</code>	Scores, tops et mise à jour des rankings.
<code>utils_RNG.js</code>	Random déterministe utilisé par la génération serveur.
<code>utils_redis.js</code>	Helpers d'accès Redis lorsque la feature l'utilise.
<code>utils_game_specific_base.js</code>	Base des utils game-specific et accès aux static data/chunks.
<code>utils_by_module.js</code>	Charge/initialise le socle By Module.
<code>utils_by_year.js</code>	Cycle annuel, jobs, bots et transitions By Year.
<code>by_module/</code>	Moteurs Accounting : création data, bills, answer key, grading, results, inventory.
<code>command/</code>	Implémentations réutilisables appelées par les wrappers command.
<code>factory/</code>	Sélection game-specific des utils, évaluations et générateurs de résultats.
<code>version/</code>	Migrations par type de document.

utils.js — fonctions structurantes

Parmi les helpers importants : `getTeamActUid`, `getActUid`, `dates/mois/années`, `deleteTeam`, `getDisplayName`, `getActEvalTeamData`, `getTeamDecisionsData`, `getTeamsOfActivity`, `getStudentsOfActivity`, `hasRole`, `getUserUid`, `mergeObjects`, `copyOf/deepCopyOf`, logging Discord et validation CRC avant sauvegarde équipe.

command/

- `utils_create_activity.js` : logique partagée de création d'activité;
- `utils_create_team.js` : création/initialisation d'équipe;
- `utils_custom_command_by_module.js` : routeur By Module;
- `utils_custom_command_by_year.js` : routeur By Year;
- `utils_join_simulation.js` : logique de join simulation;
- `utils_reopen_module.js` : rollback/réouverture By Module;
- `utils_reopen_year.js` : rollback/réouverture By Year;
- `utils_start_next_activity.js` : transition d'activité.

factory/

- `utils_factory_game_specific_utils.js` : instancie `utilsByGameUid`;
- `utils_factory_generate_results.js` : moteur de résultats par jeu;
- `utils_factory_eval.js` : moteur d'évaluation par jeu.

Principe de placement

Un wrapper sous `_command` doit rester mince. Dès qu'une logique est partagée entre plusieurs entrées, elle doit être placée dans `_utils` ou dans la famille métier adéquate. Les règles propres à ENT1/MK1/etc. restent dans leurs dossiers game-specific.

Revision #1

Created 2026-09-18 19:26:49 UTC by Mathis

Updated 2026-09-18 19:26:51 UTC by Mathis