

Créateur d'équipes - Tournois Valorant

Petit projet frontend (HTML/CSS/JS) pour composer des équipes équilibrées pour des tournois Valorant.

Fonctionnalités

- Saisie simple des joueurs (une ligne par joueur).
- Option: ajouter "|score" ou "|rankName" après le nom, ex: `Joueur#tag|75` ou `Joueur#tag|Gold`.
- Optionnel: récupérer le rang depuis Tracker.gg si vous fournissez une clé API (TRN-API-Key).
- Algorithme de répartition simple pour équilibrer les scores des équipes.

Usage

1. Ouvrez `index.html` dans un navigateur (double-clic ou via un serveur local).
2. Collez la liste de joueurs dans le champ prévu (une ligne par joueur).
3. Choisissez le nombre d'équipes.
4. Si vous avez une clé Tracker.gg (TRN-API-Key), collez-la dans le champ et cochez "Utiliser Tracker.gg".
5. Cliquez sur "Créer les équipes".

Obtenir une clé Tracker.gg

- Allez sur <https://tracker.gg/> et créez un compte ou connectez-vous pour demander une clé API (note: procédure et disponibilité peuvent changer).
- Collez la clé dans le champ "Tracker.gg API Key".

Utiliser le proxy local (recommandé pour contourner CORS)

- Installez les dépendances et lancez le proxy local (Node.js requis) :

```
cd <chemin_du_projet>
npm install
# définir la variable d'environnement TRN_API_KEY (ou créer un fichier
.env contenant TRN_API_KEY=your_key)
$env:TRN_API_KEY = 'votre_cle'
npm start
```

- Le proxy écoute par défaut sur `http://localhost:3000` et expose `/api/valorant/profile?tag={tag}`.
- Le frontend essaiera automatiquement d'appeler d'abord `http://localhost:3000` (évite les erreurs CORS). Si le proxy n'est pas disponible mais que vous avez fourni une clé dans l'UI, il tentera l'API publique.

Déploiement sur Vercel (recommandé)

Pour éviter les problèmes CORS en production et protéger votre clé API, déployez la fonction serverless incluse sous Vercel.

1. Créez un projet sur Vercel et connectez votre dépôt Git.
2. Dans les Settings → Environment Variables, ajoutez :
 - Key: TRN_API_KEY
 - Value: <votre_tracker_gg_api_key>
3. Poussez vos changements et déployez. Vercel publiera automatiquement la route serverless disponible sous `/api/valorant/profile`.
4. Le frontend appelle ensuite `/api/valorant/profile?tag=Nom%231234` (ou ? `player=Nom&tag=1234`) — aucune clé n'est exposée côté client.

Tester localement avec Vercel CLI

Vous pouvez tester la fonction localement avec la CLI Vercel :

```
npm i -g vercel
copy .env.example .env
# remplir .env avec TRN_API_KEY=...
vercel dev
```

`vercel dev` montera l'API serverless sous `http://localhost:3000` (ou autre port) et chargera les variables d'environnement depuis `.env`.

Notes techniques et limites

- Le script tente d'appeler l'endpoint public `https://public-api.tracker.gg/v2/valorant/standard/profile/riot/{tag}` et utilise l'en-tête `TRN-API-Key`. La structure des réponses peut varier et le parsing est conservateur. Si la récupération échoue, on retombe sur une valeur par défaut (50).
- Pour éviter les problèmes de CORS ou de quota, vous pouvez exécuter un petit proxy côté serveur ou limiter la fréquence des requêtes.
- Ce dépôt est volontairement minimal pour servir de point de départ.

Améliorations possibles

- Ajouter vérification et choix de la plateforme (riot, steam, etc.).
- Mieux parser les réponses de Tracker.gg selon la version de l'API.
- Ajouter un mode serveur pour cacher la clé API et contourner les problèmes CORS.
- Ajouter des tests unitaires pour l'algorithme d'équilibrage.

Licence: MIT (exemple)