🚀 Simulateur de Trafic SimplyFoot

Script de simulation de trafic pour tester les performances de votre application SimplyFoot avec Supabase.

🔧 Installation

```
cd scripts
npm install
```

⚙️ Configuration

1. **Modifiez les constantes Supabase** dans `traffic-simulator.js`:

   ```
   const SUPABASE_URL = 'https://votre-projet.supabase.co'
   const SUPABASE_ANON_KEY = 'votre-clé-anon'
   ```

2. **Créez des comptes de test** (optionnel):
   - `test-joueur-1@simplyfoot.test`
   - `test-coach-1@simplyfoot.test`
   - etc.

🏃 Utilisation

Simulation rapide (5 utilisateurs, 30s)

```
npm run simulate:light
```

Simulation normale (20 utilisateurs, 2min)

```
npm run simulate:medium
```

Simulation intensive (50 utilisateurs, 5min)

```
npm run simulate:heavy
```

Test de stress (100 utilisateurs, 10min)

```
npm run simulate:stress
```

Simulation personnalisée

```
node traffic-simulator.js [utilisateurs] [durée_ms] [montée_ms] [--verbose]
```

Exemples:

```
# 30 utilisateurs pendant 3 minutes avec montée progressive de 20s
node traffic-simulator.js 30 180000 20000

# Mode verbose pour voir tous les détails
node traffic-simulator.js 10 60000 10000 --verbose
```

📊 Scénarios Simulés

👤 Utilisateur Joueur

- **Dashboard** - Récupération données tableau de bord
- **Convocations** - Consultation événements et réponses
- **Messages** - Lecture messages privés et groupe
- **Équipe** - Consultation coéquipiers
- **Évaluations** - Consultation évaluations mentales/techniques
- **Participation** - Mise à jour statut événement

👨‍💼 Utilisateur Coach

- **Dashboard Coach** - Vue d'ensemble équipes et événements
- **Gestion Équipes** - Consultation joueurs et équipes
- **Événements** - Gestion matchs et entraînements
- **Composition** - Mise à jour compositions d'équipe
- **Messages** - Envoi messages groupe et privés
- **Joueurs** - Consultation fiches détaillées

📈 Métriques Collectées

Performance

- **Requêtes/seconde (RPS)**
- **Temps de réponse** (min/max/moyenne)
- **Taux d'erreur**
- **Throughput total**

Détails

- **Endpoints les plus sollicités**
- **Types d'erreurs rencontrées**
- **Répartition par utilisateur**
- **Progression temporelle**

🗀 Rapports

Les rapports sont automatiquement sauvegardés dans `scripts/reports/`:

```
traffic-report-[timestamp].json
```

Structure du rapport

```
{
  "simulation": {
    "users": 20,
    "duration": 120.5,
    "scenarios": ["joueur", "coach"]
  },
  "performance": {
    "totalRequests": 2456,
    "requestsPerSecond": 20.38,
    "errorRate": 1.2,
    "responseTime": {
      "average": 145,
      "min": 23,
      "max": 2341
    }
  },
  "endpoints": { ... },
  "errors": { ... }
}
```

🎯 Cas d'Usage

Test de Montée en Charge

```
# Progression: 5 → 20 → 50 utilisateurs
npm run simulate:light
npm run simulate:medium
npm run simulate:heavy
```

Test de Limite Supabase

```
# Augmentez progressivement jusqu'à voir des erreurs
node traffic-simulator.js 100 300000 30000
node traffic-simulator.js 200 300000 60000
```

Test de Stabilité

```
# Simulation longue avec charge modérée
node traffic-simulator.js 25 1800000 30000  # 30 minutes
```

⚠️ Limitations Supabase

Plan Gratuit

- **25 000 requêtes/mois**
- **2 connexions simultanées max**
- **500 MB base de données**

Plan Pro

- **500 000 requêtes/mois**
- **15 connexions simultanées**
- **8 GB base de données**

🔍 Analyse des Résultats

Performances Acceptables

- **RPS**: 10-50 req/sec selon plan
- **Temps réponse**: < 500ms en moyenne
- **Taux erreur**: < 5%

Signaux d'Alarme

- **429 Too Many Requests** → Limite de taux atteinte
- **503 Service Unavailable** → Surcharge serveur
- **Timeouts fréquents** → Connexions insuffisantes

🛠️ Personnalisation

Ajouter de nouveaux scénarios

```
async simulateNouveauScenario(userClient) {
  return this.makeRequest(userClient, 'mon-endpoint', async () => {
    const { data } = await userClient.client
      .from('ma_table')
```

```
        .select('*')
      return data
    })
  }
```

Modifier la répartition utilisateurs

```
const userType = ['joueur', 'coach', 'admin'][i % 3]  // 33% chaque type
```

Précautions

1. **Ne pas surcharger** votre environnement de production
2. **Utilisez des données de test** pour éviter la pollution
3. **Surveillez vos quotas** Supabase pendant les tests
4. **Commencez petit** puis augmentez progressivement

Documentation SimplyFoot Traffic Simulator v1.0