

Guide de préparation pour la migration web - Application Soakoja

1. Ajouts critiques au développement initial

Configuration d'environnement

javascript

// Ajouter dans le développement

```
const config = {
  development: {
    database: './database/soakoja_dev.sqlite',
    host: 'localhost',
    port: 3000
  },
  production: {
    database: process.env.DB_PATH || './database/soakoja_prod.sqlite',
    host: process.env.HOST || '0.0.0.0',
    port: process.env.PORT || 8080,
    secure: true
  }
}
```

Variables d'environnement à prévoir

env

```
# Fichier .env pour la production
NODE_ENV=production
PORT=8080
HOST=0.0.0.0
DB_PATH=/home/soakoja/database/soakoja.sqlite
SESSION_SECRET=your_strong_session_secret_here
DIAMEX_SYNC_PATH=/path/to/diamex/sync
BACKUP_PATH=/home/soakoja/backups
```

2. Adaptations spécifiques pour l'hébergement web

Structure de fichiers adaptée

```
soakojaweb/  
├── app/  
│   ├── controllers/  
│   ├── models/  
│   ├── views/  
│   └── middleware/  
├── public/  
│   ├── css/  
│   ├── js/  
│   ├── images/  
│   └── uploads/  
├── database/  
│   ├── migrations/  
│   ├── seeders/  
│   └── soakojasqlite  
├── config/  
├── logs/  
├── backups/  
├── package.json  
├── server.js  
└── .env
```

Sécurité renforcée pour le web

- **HTTPS obligatoire** : Configuration SSL/TLS
- **Helmet.js** : Protection headers HTTP
- **Rate limiting** : Limitation des requêtes par IP
- **CORS** : Configuration pour les domaines autorisés
- **File upload** : Validation stricte des fichiers uploadés

Optimisations pour la bande passante limitée

javascript

```
// Compression des réponses
app.use(compression());

// Cache des ressources statiques
app.use(express.static('public', {
  ..maxAge: '1d',
  ..etag: true
}));

// Pagination obligatoire pour les grandes listes
const ITEMS_PER_PAGE = 20;
```

3. Fonctionnalités spécifiques au web

Gestion des sessions web

```
javascript

app.use(session({
  ..secret: process.env.SESSION_SECRET,
  ..resave: false,
  ..saveUninitialized: false,
  ..cookie: {
    ..secure: process.env.NODE_ENV === 'production',
    ..maxAge: 8 * 60 * 60 * 1000 // 8 heures
  }
}));
```

API REST pour intégrations futures

```
javascript

// Routes API séparées
app.use('/api/v1', apiRoutes);

// Réponses JSON standardisées
{
  ..success: true,
  ..data: {...},
  ..message: "Opération réussie",
  ..timestamp: "2025-01-15T10:30:00Z"
}
```

Upload et gestion des fichiers

- **Photos d'ouvrages** : Redimensionnement automatique

- **Documents DiameX** : Validation et stockage sécurisé
- **Exports** : Génération PDF/Excel côté serveur

4. Sauvegarde et synchronisation

Système de backup automatique

javascript

```
// Backup quotidien de la base SQLite
const backupSchedule = cron.schedule('0 2 * * *', () => {
  const timestamp = new Date().toISOString().slice(0, 19);
  const backupPath = `./backups/soakoja_${timestamp}.sqlite`;
  fs.copyFileSync('./database/soakoja.sqlite', backupPath);
});
```

Synchronisation DiameX adaptée

- **Mode déconnecté** : Queue des opérations en attente
- **Retry automatique** : En cas d'échec réseau
- **Logs détaillés** : Traçabilité des synchronisations

5. Interface utilisateur web

Design responsive obligatoire

CSS

```
/* CSS adaptatif pour tous les écrans */
@media (max-width: 768px) {
  .dashboard-grid {
    grid-template-columns: 1fr;
  }
  .table-responsive {
    overflow-x: auto;
  }
}
```

Progressive Web App (PWA)

- **Service Worker** : Cache des ressources critiques
- **Manifest** : Installation possible sur mobile
- **Mode offline** : Fonctionnalités de base sans réseau

Interface adaptée à Madagascar

- **Langue française** : Tous les textes
- **Formats locaux** : Dates, monnaies (Ariary)
- **Timezone** : GMT+3 (Madagascar)

6. Monitoring et maintenance

Logs applicatifs

javascript

```
const winston = require('winston');

const logger = winston.createLogger({
  level: 'info',
  format: winston.format.json(),
  transports: [
    new winston.transports.File({ filename: 'logs/error.log', level: 'error' }),
    new winston.transports.File({ filename: 'logs/combined.log' })
  ]
});
```

Health checks

javascript

```
app.get('/health', (req, res) => {
  res.json({
    status: 'OK',
    timestamp: new Date().toISOString(),
    uptime: process.uptime(),
    database: 'connected' // Vérifier la DB
  });
});
```

7. Déploiement et migration

Script de déploiement

bash

```
#!/bin/bash
# deploy.sh
echo "Déploiement Soakoja..."
npm install --production
npm run migrate
npm run seed
pm2 start server.js --name soakoja
pm2 save
```

Migration des données existantes

```
javascript

// Script de migration depuis la version locale
const migrateData = async () => {
  // Exporter les données locales
  // Valider l'intégrité
  // Importer sur le serveur web
  // Vérifier la cohérence
};
```

8. Considérations spécifiques Madagascar

Optimisations réseau

- **Compression gzip** : Réduction de 60-80% du trafic
- **Minification** : CSS/JS optimisés
- **Images optimisées** : WebP avec fallback JPEG
- **Cache agressif** : Ressources statiques

Gestion des pannes électriques

- **Auto-restart** : PM2 ou équivalent
- **Sauvegarde continue** : Transactions atomiques
- **Mode dégradé** : Fonctionnement minimal

Accès multi-sites

- **VPN ou tunnel** : Accès sécurisé entre sites
- **Synchronisation différée** : Réplication de données
- **Cache local** : Données critiques disponibles offline

9. Questions importantes pour Mada Hosting

Spécifications serveur nécessaires

- **Node.js version** : 16+ supportée ?
- **SQLite** : Autorisé ou migration MySQL nécessaire ?
- **Espace disque** : Pour DB + backups + logs
- **Bande passante** : Limites mensuelles
- **SSL/TLS** : Certificats inclus ?

Accès et maintenance

- **SSH** : Accès terminal disponible ?
- **Cron jobs** : Pour les tâches automatiques
- **Process manager** : PM2 ou équivalent autorisé ?
- **Monitoring** : Outils de surveillance disponibles ?

Sécurité et conformité

- **Firewall** : Configuration possible ?
- **Backups** : Politique de sauvegarde de l'hébergeur
- **Conformité** : Réglementations malgaches respectées ?

10. Checklist avant migration

- ☐ Variables d'environnement configurées
- ☐ Base de données migrée et testée
- ☐ SSL/HTTPS fonctionnel
- ☐ Sauvegardes automatiques configurées
- ☐ Logs de surveillance actifs
- ☐ Tests de charge réalisés
- ☐ Documentation de déploiement à jour
- ☐ Formation utilisateurs web planifiée
- ☐ Plan de rollback préparé

11. Plan de migration par étapes

Phase 1 : Développement et tests locaux

- Application fonctionnelle en local
- Tests complets des fonctionnalités
- Optimisations performances

Phase 2 : Préparation hébergement

- Configuration serveur chez Mada Hosting
- Tests de connectivité et performances
- Setup des sauvegardes

Phase 3 : Migration progressive

- Déploiement en mode test
- Migration des données historiques
- Formation des utilisateurs

Phase 4 : Mise en production

- Bascule définitive
- Monitoring intensif
- Support utilisateurs