

# Documentation Technique - HabitatsConnect

---

## 1. Architecture du Code

### 1.1 Structure du projet

```
habitatsconnect/  
├── src/  
│   ├── app/           # Next.js App Router  
│   ├── components/    # Composants React  
│   ├── hooks/         # Hooks personnalisés  
│   ├── lib/           # Utilitaires et configurations  
│   ├── store/         # State management Zustand  
│   └── types/         # Définitions TypeScript  
├── public/            # Assets statiques  
├── docs/              # Documentation  
└── tests/             # Tests unitaires/intégration
```

### 1.2 Patterns utilisés

- **Atomic Design** : Composants organisés par taille
- **Custom Hooks** : Logique métier réutilisable
- **Compound Components** : Composants complexes
- **Error Boundaries** : Gestion erreurs React

## 2. API Documentation

### 2.1 Endpoints REST

#### Propriétés

```
GET /api/properties  
// Liste des propriétés avec filtres  
Query params:  
- location: string  
- checkIn: date  
- checkOut: date  
- guests: number  
- minPrice: number  
- maxPrice: number  
- type: PropertyType  
  
Response: {  
  properties: Property[],  
  total: number,  
  page: number,  
}
```

```
    limit: number
  }
```

## Réservations

```
POST /api/reservations
// Créer une réservation
Body: {
  propertyId: string,
  checkIn: Date,
  checkOut: Date,
  guests: number,
  totalPrice: number,
  depositAmount: number
}

Response: {
  id: string,
  status: 'pending',
  ...reservationData
}
```

## 2.2 WebSocket Events

- `reservation:created` : Nouvelle réservation
- `message:received` : Nouveau message
- `property:updated` : Mise à jour annonce

## 3. Base de Données

### 3.1 Schéma Firestore

```
// Collections
interface FirestoreSchema {
  users: Collection<User>
  properties: Collection<Property>
  reservations: Collection<Reservation>
  messages: Collection<Message>
  conversations: Collection<Conversation>
  transactions: Collection<Transaction>
}
```

### 3.2 Index requis

- `properties.ownerId` (asc)
- `properties.location` (asc)
- `reservations.propertyId` (asc)

- `reservations.clientId` (asc)
- `messages.conversationId` (asc)

## 4. Configuration et Déploiement

### 4.1 Variables d'environnement

```
# Firebase
NEXT_PUBLIC_FIREBASE_API_KEY=...
NEXT_PUBLIC_FIREBASE_AUTH_DOMAIN=...
NEXT_PUBLIC_FIREBASE_PROJECT_ID=...

# Stripe
STRIPE_PUBLIC_KEY=...
STRIPE_SECRET_KEY=...
STRIPE_WEBHOOK_SECRET=...

# Email
SENDGRID_API_KEY=...

# Google Maps
GOOGLE_MAPS_API_KEY=...
```

### 4.2 Commandes de déploiement

```
# Installation
npm install

# Développement
npm run dev

# Build production
npm run build

# Tests
npm run test

# Lint
npm run lint
```

## 5. Maintenance et Monitoring

### 5.1 Logs

- **Application** : Console + Sentry
- **Performance** : Vercel Analytics
- **Erreurs** : Sentry error tracking

## 5.2 Métriques

- Response times
- Error rates
- User sessions
- Conversion rates

## 5.3 Alertes

- Downtime > 5 minutes
- Error rate > 5%
- Performance degradation