Fiche récap : Communiquez avec un serveur HTTP grâce à Angular

Développement

1. Configurer l'application Dwelio pour tester les interactions frontend/backend.
2. Envoyer des requêtes GET pour afficher dynamiquement les biens disponibles.
3. Afficher les détails d'un bien en récupérant ses informations via l'URL.
4. Soumettre une offre d'achat en vérifiant les règles métier du backend.
5. Intercepter les requêtes HTTP pour gérer l'authentification et les erreurs globales.

Code 💻

typescript

```typescript
// Activer HttpClient avec intercepteurs
provideHttpClient(
  withInterceptors([tokenInterceptor, loggerInterceptor])
);

// Service Angular pour récupérer tous les biens
getAllProperties() {
  return this.http.get<HousingPropertyPreview[]>('http://localhost:3030/api/properties');
}

// Intercepteur pour ajouter un header Authorization
export const tokenInterceptor: HttpInterceptorFn = (req, next) => {
  const tokenService = inject(TokenService);
  const authReq = req.clone({
    setHeaders: { Authorization: `Bearer ${tokenService.userToken()}` }
  });
  return next(authReq);
};
```

Bonnes pratiques 👍

- ✅ Importer provideHttpClient() pour activer les requêtes HTTP.
- ✅ Utiliser le pipe async pour afficher le résultat des Observables.
- ✅ Spécifier le type des réponses pour éviter les erreurs de runtime.
- ✅ Souscrire aux Observables pour déclencher les requêtes HTTP.
- ✅ Créer des services Angular pour centraliser les appels HTTP.
- ✅ Gérer les erreurs avec un intercepteur pour afficher un modal.
- ✅ Cloner les requêtes dans les intercepteurs pour ajouter des headers.

Erreurs classiques 👎

- ❌ Oublier de souscrire aux Observables et ne pas envoyer de requêtes.
- ❌ Ignorer les types de réponse et provoquer des erreurs dans les templates.
- ❌ Confondre les opérateurs RxJS comme tap, filter et switchMap.
- ❌ Oublier d'enregistrer les intercepteurs dans app.config.ts.
- ❌ Négliger la gestion des erreurs dans les services HTTP.
- ❌ Ne pas utiliser les routes dynamiques pour afficher les détails d'un bien.
- ❌ Utiliser un token sans intercepteur pour les appels sécurisés.

Définitions 🔎

Observable
Flux de données asynchrones auquel on peut s'abonner pour réagir à chaque émission.

HttpClient
Service Angular permettant d'effectuer des requêtes HTTP vers un serveur.

Intercepteur HTTP
Fonction interceptant requêtes ou réponses pour y appliquer un traitement global.

pipe()
Méthode qui permet de chaîner les opérateurs sur un Observable (comme tap, filter, etc.).

switchMap()
Opérateur RxJS qui annule les requêtes précédentes et déclenche une nouvelle requête basée sur la dernière valeur reçue.