

DOCUMENTATION

Technique

Innov'Events

ECF — Concepteur Développeur d'Applications
Février 2026

1. Présentation et périmètre

Innov'Events est une application de gestion d'événements composée de :

- **Backend API** en PHP 8.2 (Apache) — dossier backend/
- **Frontend web** en Angular 20 — dossier frontend/
- **Application mobile** en Ionic Angular 7 / Angular 17 — dossier mobile/

Cette documentation décrit : l'architecture et le fonctionnement technique du projet, l'environnement de développement (Docker + outils), les principaux composants (API, BDD, logs, emails) et l'état de la CI/CD.

2. Organisation du dépôt

2.1 Arborescence principale

Dossier / Fichier	Contenu
backend/	API REST PHP (scripts par domaine)
frontend/	Application web Angular
mobile/	Application mobile Ionic/Angular
docker-compose.yml	Environnement dev (services DB, outils, mail)
innovevents.sql	Export SQL pour initialiser MySQL
.github/workflows/ci-cd.yml	Pipeline CI/CD

2.2 Convention d'API (script-based)

Le backend est structuré par scripts PHP : 1 fichier = 1 endpoint, regroupement par domaine sous backend/api/<domaine>/...

Exemples :

```
backend/api/auth/login.php
backend/api/events/read_public.php
backend/api/quotes/generate_pdf.php
```

3. Stack technique

3.1 Backend (PHP)

Élément	Détail
Langage	PHP 8.2 via Apache (Dockerfile)
Extensions	pdo, pdo_mysql, mongodb
mongodb/mongodb	Driver MongoDB pour PHP
tecnickcom/tcpdf	Génération de PDF (devis)
phpmailer/phpmailer	Envoi d'emails

Références : backend/Dockerfile, backend/composer.json, backend/config/*, backend/services/*

3.2 Frontend web

- Angular 20

- RxJS 7

Référence : *frontend/package.json*

3.3 Mobile

- Ionic Angular 7 / Angular 17
- Cypress (E2E)

Références : *mobile/package.json, mobile/proxy.conf.json, mobile/src/environments/**

3.4 Environnement dev (Docker Compose)

Services dev : API backend, MySQL 8, MongoDB, phpMyAdmin, Mongo Express, MailHog.

Référence : *docker-compose.yml*

4. Environnement de développement

4.1 Prérequis

- Docker et Docker Compose
- Node.js 20+ et npm (pour lancer le frontend et le mobile en local)

4.2 Ports et URLs

Service	URL	Port
Backend API	http://localhost:8080	8080
phpMyAdmin	http://localhost:8081	8081
Mongo Express	http://localhost:8082	8082
MailHog UI	http://localhost:8090	8090
MongoDB	localhost:27017	27017
SMTP MailHog	localhost:1025	1025

5. Installation et initialisation (dev)

5.1 Démarrer l'environnement (Docker)

Depuis la racine du projet :

```
docker compose up -d --build
docker compose ps
```

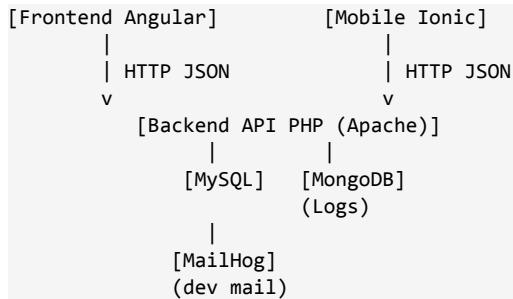
5.2 Importer la base MySQL via phpMyAdmin

1. Ouvrir <http://localhost:8081>
2. Se connecter : utilisateur root / mot de passe root
3. Sélectionner la base innovevents
4. Onglet Importer → Importer innovevents.sql
5. Valider l'import

Références : *innovevents.sql, service phpMyAdmin (docker-compose.yml)*

6. Architecture globale

Le frontend et le mobile consomment l'API via HTTP/JSON. Le backend persiste les données dans MySQL et utilise MongoDB pour les logs. Les emails en dev sont capturés par MailHog.



Voir le document « Architecture logicielle » pour les schémas détaillés.

7. Backend API (PHP)

7.1 Configuration MySQL

Paramètre	Valeur (dev)
host	db
database	innovevents
user	root
password	root

Référence : [backend/config/database.php](#)

7.2 Configuration MongoDB

Paramètre	Valeur (dev)
host	mongo
port	27017
database	innovevents_nosql
user	root
password	root

Référence : [backend/config/database_mongo.php](#)

7.3 JWT (authentification)

Le backend lit JWT_SECRET. Si non défini, une valeur par défaut est utilisée (à éviter en production).

Références : [backend/config/jwt.php](#), [backend/services/JwtService.php](#)

7.4 Email

- En dev : envoi vers MailHog (mailhog:1025)
- En prod : SMTP Brevo (configuration côté production)

Référence : [backend/config/mail.php](#)

7.5 Logs d'activité (MongoDB)

Le backend insère des logs (documents) dans MongoDB pour tracer certaines actions métier. Des endpoints dédiés permettent de consulter les logs/stats :

```
backend/api/logs/read.php
backend/api/logs/stats.php
```

Référence : [backend/services/MongoLogger.php](#)

7.6 Upload d'images

- Contrôle extension autorisée (images)
- Contrôle taille maximale
- Génération de nom unique
- Stockage dans backend/uploads/<domaine>/
- Chemin renvoyé sous la forme /uploads/<domaine>/<fichier>

Références : [backend/api/events/upload_image.php](#), [backend/api/prospects/upload_image.php](#)

8. Base de données MySQL

Le dump innovevents.sql contient la structure et les données nécessaires au fonctionnement.

Table	Rôle
users	Authentification + rôle (admin, employee, client), activation, vérification email, statut changement mdp
clients	Profil client (lié à users)
events	Événements (liés aux clients, assignation possible à un employé)
quotes / quote_items	Devis + lignes de devis
prospects	Prospects (convertibles en clients)
reviews	Avis (publication selon validation)
tasks	Tâches liées aux événements
event_notes	Notes liées aux événements
services, themes, event_types	Référentiels
app_settings	Paramètres applicatifs

Référence : [innovevents.sql](#) — Voir le document « Documentation BDD » pour le dictionnaire complet.

9. Frontend web (Angular)

Consommation de l'API via HTTP. Le token JWT est stocké côté client et réutilisé pour les appels authentifiés (via interceptor côté Angular).

Commandes :

```
cd frontend
npm install
npm start
```

Références : [frontend/package.json](#), [frontend/src/app/*](#)

10. Mobile (Ionic Angular)

Le mobile utilise un proxy en dev (évite d'avoir à gérer CORS côté navigateur) : appels vers /api/... proxy vers <http://localhost:8080>.

Commandes :

```
cd mobile
npm install
npm start
```

Références : *mobile/proxy.conf.json, mobile/src/environments/environment.ts, mobile/src/app/**

11. CORS (dev vs prod)

11.1 En développement (local)

En local, la configuration CORS n'est pas strictement uniformisée : elle est gérée principalement pour les usages localhost (notamment 4200 et 4300) et peut varier selon les endpoints.

Remarque : côté mobile, le proxy /api permet de limiter les impacts CORS en environnement de développement.

11.2 En production

En production, la configuration CORS est bien configurée, uniformisée et restreinte aux domaines autorisés.

12. CI/CD (GitHub Actions) — état actuel

Le pipeline CI exécute les étapes principales (installation, lint, tests, build) sur le frontend et le mobile. La partie CD (déploiement automatisé) n'a pas été finalisée : la configuration n'est pas complète et le déploiement automatique n'est pas opérationnel à ce stade.

Référence : *.github/workflows/ci-cd.yml*

13. Tests et qualité

Frontend :

```
cd frontend
npm test
npm run lint
npm run build
```

Mobile :

```
cd mobile
npm test
npm run lint
npm run e2e
```

Références : *frontend/src/**/*spec.ts, mobile/src/**/*spec.ts, mobile/cypress/**

14. Exploitation et maintenance (dev)

Logs :

```
docker compose logs -f backend
docker compose logs -f db
docker compose logs -f mongo
docker compose logs -f mailhog
```

Shell conteneur :

```
docker exec -it ecf_backend bash
```

Reset complet (⚠ destructif) :

```
docker compose down -v
```

15. Dépannage rapide (FAQ)

15.1 API inaccessible (8080)

- Vérifier docker compose ps
- Consulter docker compose logs -f backend

15.2 Erreur MySQL

- Vérifier que db est Up
- Vérifier que innovevents.sql a bien été importé (phpMyAdmin)

15.3 Emails dev

- Vérifier mailhog Up
 - Ouvrir <http://localhost:8090>
-

Auteur : Florian Gourinat — Projet ECF 2026 — Innov'Events