

PLAN DE TESTS

Innov'Events

ECF — Concepteur Développeur d'Applications
Février 2026

1. Environnement de tests

L'application mobile Innov'Events est pourvue d'un environnement de test complet couvrant les trois niveaux de tests requis : tests unitaires, tests fonctionnels et tests end-to-end.

Élément	Technologie / Outil
Framework tests unitaires/fonctionnels	Jasmine 5.1 + Karma 6.4
Framework tests E2E	Cypress 13.17
Couverture de code	karma-coverage (Istanbul)
Navigateur de test	Chrome Headless 144
Application testée	Application Mobile — Ionic Angular 17
Fonctionnalité vérifiée en intégralité	Authentification administrateur

Commandes d'exécution

Tests unitaires + fonctionnels avec coverage :

```
cd mobile && npm test
```

Tests E2E (interface graphique) :

```
cd mobile && npx cypress open
```

Tests E2E (mode headless CI/CD) :

```
cd mobile && npx cypress run
```

2. Fonctionnalité testée : Authentification administrateur

La fonctionnalité d'authentification est testée en intégralité car elle constitue le point d'entrée sécurisé de l'application mobile. Elle couvre :

- Validation du formulaire de connexion (champs vides, email seul, mot de passe seul)
- Appel API POST /api/auth/login.php avec les identifiants
- Contrôle du rôle utilisateur — seuls les administrateurs sont autorisés
- Rejet des employés et clients (test de sécurité)
- Stockage de la session (token + données utilisateur) dans le localStorage
- Déconnexion complète et nettoyage de la session
- Gestion des erreurs serveur (401 Unauthorized, 500 Server Error)

3. Tableau des tests par fonctionnalité

3.1 Tests unitaires — AuthService

Fichier : src/app/services/auth.service.spec.ts

#	Test	Type	Résultat attendu
TU-01	Le service est créé correctement	Unitaire	service est truthy
TU-02	Aucun utilisateur connecté par défaut	Unitaire	currentUserValue est null
TU-03	isLoggedIn() retourne false sans utilisateur	Unitaire	Retourne false

3.2 Tests fonctionnels — Authentification (AuthService)

Fichier : src/app/services/auth.service.spec.ts

#	Test	Type	Résultat attendu
TF-01	Connexion réussie avec compte admin	Fonctionnel	role=admin, isLoggedIn=true, localStorage rempli
TF-02	Rejet connexion employé	Fonctionnel (sécurité)	Erreur "administrateurs", isLoggedIn=false
TF-03	Rejet connexion client	Fonctionnel (sécurité)	Erreur "administrateurs", localStorage vide
TF-04	Gestion erreur serveur (401)	Fonctionnel	Erreur truthy, isLoggedIn=false
TF-05	Déconnexion complète	Fonctionnel	currentUserValue=null, localStorage vidé
TF-06	Token stocké dans localStorage	Fonctionnel	Token et rôle présents dans le stockage
TF-07	Corps de requête POST correct	Fonctionnel	Body contient {email, password}

3.3 Tests unitaires — Page de connexion (LoginPage)

Fichier : src/app/pages/login/login.page.spec.ts

#	Test	Type	Résultat attendu
TU-04	Le composant est créé	Unitaire	Composant truthy
TU-05	Champs initialisés vides	Unitaire	email="", password="", loading=false
TU-06	Erreur si champs vides	Unitaire	"Veuillez remplir tous les champs"
TU-07	Erreur si email seul renseigné	Unitaire	"Veuillez remplir tous les champs"
TU-08	Erreur si mot de passe seul	Unitaire	"Veuillez remplir tous les champs"
TU-09	Loading activé avec champs valides	Unitaire	loading=true, error=""

3.4 Tests fonctionnels — Liste événements (EventsPage)

Fichier : src/app/pages/events/events.page.spec.ts

#	Test	Type	Résultat attendu
TU-10	Le composant est créé	Unitaire	Composant truthy
TU-11	Initialisation loading=true	Unitaire	loading=true, events=[]
TU-12	Labels de statut définis	Unitaire	statusLabels[draft]=Brouillon
TU-13	Couleurs de statut correctes	Unitaire	in_progress→success, cancelled→danger
TU-14	Formatage de date	Unitaire	Contient "2026", vide retourne "_"
TF-08	Chargement et filtrage	Fonctionnel	Exclut completed/cancelled
TF-09	Tri par date ascendant	Fonctionnel	Dates croissantes
TF-10	Gestion réponse vide	Fonctionnel	events.length=0
TF-11	Gestion erreur API 500	Fonctionnel	loading=false, pas de crash

3.5 Tests fonctionnels — Détail événement (EventDetailPage)

Fichier : src/app/pages/event-detail/event-detail.page.spec.ts

#	Test	Type	Résultat attendu
TU-15	Le composant est créé	Unitaire	Composant truthy
TU-16	Initialisation loading=true	Unitaire	loading=true
TU-17	Formatage de dates	Unitaire	Contient "avril", vide retourne "-"
TF-12	Chargement détail événement	Fonctionnel	Nom, client, notes chargés
TF-13	Note vide non envoyée	Fonctionnel	Pas d'appel API

TF-14	Ajout note via API	Fonctionnel	POST envoyé, note ajoutée, champ vidé
-------	--------------------	-------------	---------------------------------------

3.6 Tests fonctionnels — Fiche client (ClientDetailPage)

Fichier : src/app/pages/client-detail/client-detail.page.spec.ts

#	Test	Type	Résultat attendu
TU-18	Le composant est créé	Unitaire	Composant truthy
TU-19	Initialisation loading=true	Unitaire	loading=true
TF-15	Chargement fiche client	Fonctionnel	Prénom, événements chargés
TF-16	URL Google Maps générée	Fonctionnel	Contient google.com/maps
TF-17	Gestion erreur API	Fonctionnel	loading=false, client=null

3.7 Tests End-to-End — Cypress

Fichier : cypress/e2e/app-complete.cy.ts

#	Test	Type	Résultat attendu
E2E-01	Page de login affichée	E2E	Titre et mention "administrateurs" visibles
E2E-02	Erreur champs vides	E2E	Message d'erreur affiché
E2E-03	Connexion admin réussie	E2E	Redirection vers /tabs/events
E2E-04	Liste événements affichée	E2E	Titre "Événements à venir" visible
E2E-05	Navigation tabs présente	E2E	Onglets CGU, CGV visibles
E2E-06	Navigation vers CGU	E2E	Contenu CGU affiché
E2E-07	Navigation vers CGV	E2E	Contenu CGV affiché
E2E-08	Ouverture détail événement	E2E	"Voir fiche client" visible
E2E-09	Ajout note sur événement	E2E	Note visible après ajout
E2E-10	Navigation vers fiche client	E2E	"Actions rapides" visible
E2E-11	Déconnexion	E2E	Retour à /login

4. Synthèse des résultats

Type de test	Nombre	Framework	Résultat
Tests unitaires	19	Jasmine / Karma	19/19 SUCCESS ✓
Tests fonctionnels	17	Jasmine / Karma (HttpTestingController)	17/17 SUCCESS ✓
Tests end-to-end	11	Cypress 13.17	11/11 SUCCESS ✓
TOTAL	47		47/47 SUCCESS ✓

4.1 Couverture de code (Coverage)

La couverture de code est générée automatiquement par karma-coverage (Istanbul) lors de l'exécution de npm test. Résultats obtenus :

Métrique	Couverture	Détail
Statements	85.32%	93 / 109 instructions couvertes
Branches	54.54%	24 / 44 branches couvertes
Functions	73.17%	30 / 41 fonctions couvertes
Lines	84.05%	58 / 69 lignes couvertes

Le rapport HTML détaillé est disponible dans mobile/coverage/innovevents-mobile/index.html.

5. Sécurité et évolutions technologiques

Le plan de tests intègre les évolutions technologiques et les problèmes de sécurité conformément aux critères d'évaluation de la compétence AT3.

- **Contrôle d'accès par rôle** — Les tests TF-02 et TF-03 vérifient que les rôles employée et client sont rejetés à la connexion. Seul le rôle admin est autorisé, conformément au principe du moindre privilège.
- **Protection CORS** — L'utilisation d'un proxy Angular (proxy.conf.json) contourne les restrictions Cross-Origin sans affaiblir la sécurité du backend. Les APIs restent protégées.
- **Gestion sécurisée de la session** — Le test TF-06 vérifie le stockage du token d'authentification. Le test TF-05 vérifie la suppression complète des données de session à la déconnexion.
- **Validation des entrées côté client** — Les tests TU-06 à TU-08 vérifient que le formulaire rejette les soumissions incomplètes avant tout appel réseau, réduisant la surface d'attaque.
- **Résilience aux erreurs serveur** — Les tests TF-04 et TF-11 vérifient que l'application gère gracieusement les erreurs HTTP (401, 500) sans crash ni fuite d'information.
- **Isolation des tests** — L'utilisation de HttpTestingController (mock HTTP) garantit que les tests sont isolés, reproductibles et indépendants de l'état du serveur.
- **Secrets externalisés** — Les identifiants de test E2E sont stockés dans cypress.env.json (exclu du versioning Git) pour éviter toute fuite de credentials dans le code source.