

Faculté des Sciences de Sfax

Département Informatique

# Rapport de Projet de Test Logiciel

## Automatisation Avancée

### Sujet :

Automatisation et Audit Qualité du site

**OpenSky Network (<https://opensky-network.org/>)**

Focus : Responsive overflow, performances (LCP/chargement), erreurs console, intégrité liens, reporting Jira

**Réalisé par :** Amine Harrabi

**Niveau :** 1<sup>re</sup> Année Ingénieur

**Enseignant :** M. Taher Labidi

**Année Universitaire :** 2025–2026

# Table des matières

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Introduction et Contexte du Projet</b>                        | <b>1</b>  |
| 1.1      | Présentation du Projet . . . . .                                 | 1         |
| 1.2      | Cahier des Charges et Objectifs . . . . .                        | 1         |
| 1.3      | Architecture Technique . . . . .                                 | 1         |
| <b>2</b> | <b>Innovations : Bug Hunting, Artifacts et Jira</b>              | <b>3</b>  |
| 2.1      | Collecte Automatique d'Artifacts (preuve de bug) . . . . .       | 3         |
| 2.2      | Mode Audit Strict (transformer les findings en échecs) . . . . . | 3         |
| 2.3      | Intégration Jira (optionnelle) . . . . .                         | 3         |
| <b>3</b> | <b>Analyse Détailée des Suites de Tests</b>                      | <b>5</b>  |
| 3.1      | Suite 1 : Fonctionnel (Pages publiques) . . . . .                | 5         |
| 3.1.1    | Objectif . . . . .   | 5         |
| 3.1.2    | Exemples de scénarios . . . . .                                  | 5         |
| 3.2      | Suite 2 : Performance et Charge (Best-effort) . . . . .          | 5         |
| 3.2.1    | Objectif . . . . .   | 5         |
| 3.2.2    | Scénarios . . . . .  | 5         |
| 3.3      | Suite 3 : Compatibilité Cross-Browser . . . . .                  | 6         |
| 3.3.1    | Objectif . . . . .   | 6         |
| 3.3.2    | Remarque . . . . .   | 6         |
| 3.4      | Suite 4 : Responsive Design . . . . .                            | 6         |
| 3.4.1    | Objectif . . . . .   | 6         |
| 3.5      | Suite 5 : Sécurité (Non-intrusif) . . . . .                      | 6         |
| 3.6      | Suite 6 : Intégrité des Liens . . . . .                          | 6         |
| 3.7      | Suite 7 : Erreurs Console / Erreurs JS runtime . . . . .         | 6         |
| 3.8      | Suite 8 : Fuzzing léger des Inputs . . . . .                     | 7         |
| <b>4</b> | <b>Exécution, Résultats et Exploitation Jira</b>                 | <b>8</b>  |
| 4.1      | Commandes d'exécution . . . . .                                  | 8         |
| 4.2      | Dossier des preuves . . . . .                                    | 8         |
| 4.3      | Modèle de ticket Jira (copiable) . . . . .                       | 9         |
| <b>5</b> | <b>Récapitulatif Exhaustif des Suites et Résultats</b>           | <b>10</b> |
| 5.1      | Légende . . . . .  | 10        |

|          |  |           |
|----------|--|-----------|
| 5.2      | Détail par Suite de Tests . . . . .              | 10        |
| 5.2.1    | Suite 1 : About . . . . .                        | 10        |
| 5.2.2    | Suite 1 : Data . . . . .                         | 10        |
| 5.2.3    | Suite 1 : Feed . . . . .                         | 10        |
| 5.2.4    | Suite 1 : Home . . . . .                         | 11        |
| 5.2.5    | Suite 1 : NonFunctional . . . . .                | 11        |
| 5.2.6    | Suite 1 Functional (détailé) . . . . .           | 11        |
| 5.2.7    | Suite 1 Functional HomePage . . . . .            | 12        |
| 5.2.8    | Suites Inline . . . . .                          | 12        |
| 5.2.9    | Suite 2 : Performance (détailé) . . . . .        | 12        |
| 5.2.10   | Suite 3 : Cross-Browser . . . . .                | 13        |
| 5.2.11   | Suite 4 : Responsive (détailé) . . . . .         | 14        |
| 5.2.12   | Suite 5 : Sécurité (détailé) . . . . .           | 14        |
| 5.2.13   | Suite 6 : Intégrité des Liens . . . . .          | 15        |
| 5.2.14   | Suite 7 : Erreurs Console / JS runtime . . . . . | 15        |
| 5.2.15   | Suite 8 : Fuzzing / Inputs . . . . .             | 15        |
| 5.3      | Résumé et Priorisation (Quick wins) . . . . .    | 15        |
| <b>6</b> | <b>Conclusion et Perspectives</b>                | <b>16</b> |
| 6.1      | Points Forts du Projet . . . . .                 | 16        |
| 6.2      | Perspectives . . . . .                           | 16        |

## Chapitre 1

# Introduction et Contexte du Projet

### 1.1 Présentation du Projet

Ce projet vise à auditer la qualité et la robustesse de la plateforme **OpenSky Network** via une stratégie de tests **boîte noire** (Black Box Testing), orientée **bug-hunting**.

L'objectif n'est pas de « faire passer des tests », mais d'obtenir des **signaux de bugs** exploitables : anomalies UI/UX (responsive overflow), régressions de performance, erreurs JavaScript/console, liens cassés, et défauts de configuration sécurité (headers).

### 1.2 Cahier des Charges et Objectifs

- Mettre en place des suites de tests automatisées (Selenium + Pytest) sur les pages publiques et la carte `map.opensky-network.org`.
- Déetecter des anomalies : scroll horizontal, overflow de blocs de code, erreurs console, erreurs JS runtime.
- Mettre en place des audits non-intrusifs (HTTP headers, intégrité de liens) et les rendre **stricts** via un mode `-audit-strict`.
- Générer automatiquement des **preuves** (captures, HTML, logs console) et faciliter la création de tickets Jira.

### 1.3 Architecture Technique

#### Information

**Stack :** `pytest` (orchestration), `selenium` (UI tests), `requests` (audits HTTP), Page Object Model (POM).

**Objectif :** séparation claire `pages` / `tests` / `utils` + collecte d'artifacts.

- **Pages (POM)** : `pages/base_page.py`, `pages/explorer_page.py`, `pages/login_page.py`
- **Tests** : suites fonctionnelles, performance, cross-browser, responsive, audits (sécurité, liens, console errors)

- **Utils** : utils/web\_audit.py (HEAD/GET + UA), utils/jira\_client.py, utils/selenium\_acti
- **Artifacts** : reports/artifacts/... /screenshot.png, page.html, console.json, js\_errors.json

## Chapitre 2

# Innovations : Bug Hunting, Artifacts et Jira

### 2.1 Collecte Automatique d'Artifacts (preuve de bug)

Lorsqu'un test échoue, le framework sauvegarde automatiquement :

- **Capture d'écran** : screenshot.png
- **Snapshot HTML** : page.html
- **Logs console navigateur** : console.json (si disponible)
- **Erreurs runtime JS** : js\_errors.json (collecteur injecté côté client si support Chrome/CDP)

#### Finding (signal de bug)

Ces fichiers rendent le bug **reproductible** et **vérifiable** (preuve visuelle + contexte DOM + logs).

### 2.2 Mode Audit Strict (transformer les findings en échecs)

Le projet différencie :

- **Mode normal** : log des [FINDING] (utile pour un site externe instable)
- **Mode strict** : -audit-strict force l'échec dès qu'un défaut est détecté

### 2.3 Intégration Jira (optionnelle)

Le mode -jira-create-on-fail peut créer automatiquement un ticket Jira en cas d'échec, et joindre les artifacts.

#### Format Ticket Jira (proposé)

Champs recommandés pour un ticket :

- **Summary** : [AUTOTEST] Nom du test + symptôme (ex : overflow responsive)
- **Environment** : OS, Chrome version, headless on/off

- **Steps to Reproduce** : URL + viewport (si responsive) + actions
- **Expected vs Actual**
- **Attachments** : screenshot + page.html + console.json + js\_errors.json

## Chapitre 3

# Analyse Détailée des Suites de Tests

### 3.1 Suite 1 : Fonctionnel (Pages publiques)

#### 3.1.1 Objectif

Valider que les pages publiques essentielles sont accessibles et cohérentes (Home, About, Feed, Data), et que les liens critiques pointent vers les pages attendues.

#### 3.1.2 Exemples de scénarios

- Vérification de sections principales sur la home (contenu, liens).
- Navigation « Sign in » vers l'auth (robustifiée par click sécurisé).
- Vérification de pages About/FAQ/Terms/Privacy.

### 3.2 Suite 2 : Performance et Charge (Best-effort)

#### 3.2.1 Objectif

Mesurer des indicateurs simples (LCP approximée, load time proxy, 404 latency) et détecter des régressions potentielles.

##### Information

Les seuils sont **stricts uniquement** avec `-audit-strict`, sinon le framework enregistre un [FINDING].

#### 3.2.2 Scénarios

- Cold load (throttling réseau si CDP disponible)
- Warm load (cache)
- API docs sous réseau lent (LCP si disponible)
- 404 probe : latence et taille payload

### 3.3 Suite 3 : Compatibilité Cross-Browser

#### 3.3.1 Objectif

Smoke tests orientés compatibilité (chargement, interaction simple, capture screenshot, vérification logs console).

#### 3.3.2 Remarque

Le site peut ne pas utiliser de balise `<header>`. Le test vérifie désormais `<body>` et journalise les erreurs console en mode normal.

### 3.4 Suite 4 : Responsive Design

#### 3.4.1 Objectif

Déetecter des anomalies UI sur plusieurs viewports (mobile, tablette, desktop) :

- Scroll horizontal (overflow global)
- Overflow de blocs `<pre>/<code>`
- Heuristique de tap targets
- Screenshots pour revue visuelle

#### Finding (signal de bug)

**Candidats Jira typiques** : scroll horizontal sur mobile, code block overflow sur docs, éléments non adaptés au viewport.

### 3.5 Suite 5 : Sécurité (Non-intrusif)

Audit headers HTTP : HSTS, `nosniff`, politiques (Referrer/Permissions), protections de framing (XFO/CSP).

### 3.6 Suite 6 : Intégrité des Liens

Extraction des URLs depuis le DOM (liens/scripts/css/images) puis contrôle HTTP. Le client HTTP utilise un **User-Agent navigateur** pour réduire les faux 403.

### 3.7 Suite 7 : Erreurs Console / Erreurs JS runtime

Collecte des erreurs console (Chrome) + collecteur JS runtime (best-effort). Les anomalies sont des **signaux forts** de bug front-end.

### 3.8 Suite 8 : Fuzzing léger des Inputs

Tests négatifs (payloads XSS usuels) sur la recherche de la carte : absence d'alert et heuristiques d'injection.

## Chapitre 4

# Exécution, Résultats et Exploitation Jira

## 4.1 Commandes d'exécution

Listing 4.1 – Commandes principales (Windows PowerShell)

```
# Activer l'environnement
.\venv\Scripts\activate

# Run complet
python -m pytest

# Bug hunting strict (transformer findings en FAIL)
python -m pytest -m "responsive\uor\ujs\uor\uperformance\uor\ulinks" --audit-
strict

# Audit + cr ation Jira en cas d' chec
python -m pytest -m "responsive\uor\ujs\uor\uperformance\uor\ulinks" --audit-
strict ^
--jira-create-on-fail --jira-project=QA --jira-label=opensky
```

## 4.2 Dossier des preuves

Les preuves sont générées automatiquement dans :

- reports/artifacts/<test\_id>/screenshot.png
- reports/artifacts/<test\_id>/page.html
- reports/artifacts/<test\_id>/console.json
- reports/artifacts/<test\_id>/js\_errors.json

## 4.3 Modèle de ticket Jira (copiable)

### Format Ticket Jira (proposé)

**Summary :** [AUTOTEST] Responsive overflow sur /data/api-docs (Viewport 2560x1440)

**Steps :**

1. Ouvrir <https://opensky-network.org/data/api-docs>
2. Définir viewport : 2560x1440
3. Observer le rendu des blocs de code

**Expected :** Aucun scroll horizontal / pas d'overflow

**Actual :** Overflow visible + scroll horizontal

**Attachments :** screenshot.png + page.html + console.json

## Chapitre 5

# Récapitulatif Exhaustif des Suites et Résultats

## 5.1 Légende

- **PASSED** : Test réussi
- **FAILED** : Test échoué (détaillé ci-dessous)
- **SKIPPED** : Test ignoré (pré-requis manquant ou non applicable)

## 5.2 Détail par Suite de Tests

### 5.2.1 Suite 1 : About

- test-ABOUT-01-faq-searchable : **PASSED**
- test-ABOUT-02-terms-and-privacy : **PASSED**
- test-ABOUT-03-publications-links : **SKIPPED**
- test-ABOUT-04-cross-navigation : **PASSED**

### 5.2.2 Suite 1 : Data

- test-DATA-01-main-data-page-methods : **PASSED**
- test-DATA-02-aircraft<sub>alerts</sub> : **PASSED**
- test-DATA-03-api-docs-navigation : **PASSED**
- test-DATA<sub>04</sub> – tools – page – links : **PASSED**
- test-DATA-05-scientific-datasets : **PASSED**

### 5.2.3 Suite 1 : Feed

- test-FEED-01-main-feed-overview : **PASSED**
- test-FEED-02-raspberry-pi-guide : **PASSED**
- test<sub>FEED</sub><sub>03</sub><sub>debian\_docker\_steps\_present</sub> : **PASSED**
- test<sub>FEED</sub><sub>04</sub><sub>specialized\_feed\_pages</sub> : **PASSED**

- test-FEED-05-downloads-and-removal-documented : **SKIPPED**
- test-FEED-03-debian-docker-steps-present : **PASSED**
- test-FEED-04-specialized-feed-pages : **PASSED**
- test-FEED-05-downloads-and-removal-documented : **SKIPPED**

#### 5.2.4 Suite 1 : Home

- test-HOME-01-homepage-loads-and-sections-visible : **PASSED**
- test-HOME-02-navigation-links : **PASSED**
- test-HOME-03-signin-cta : **PASSED**
- test-HOME-04-news-updates-links : **PASSED**

#### 5.2.5 Suite 1 : NonFunctional

- test-NF-01-browser-compatibility-smoke : **PASSED**
- test-NF-02-page-load-performance : **PASSED**
- test-NF-03-https-and-mixed-content : **PASSED**
- test-NF-05-sitemap-validation-quick : **PASSED**

#### 5.2.6 Suite 1 Functional (détaillé)

- test-01-map-loads-successfully-and-performance : **FAILED**
- test-02-search-input-and-table-presence : **PASSED**
- test-03-map-controls-present : **FAILED**
- test-04-home-page-has-flight-map-link : **PASSED**
- test-05-login-link-points-to-auth : **PASSED**

Échecs détaillés :

- **test-01-map-loads-successfully-and-performance**
  - Objectif : Vérifier le chargement rapide de la carte et la visibilité du canvas.
  - Scénario : Naviguer vers la carte, vérifier la visibilité, mesurer le temps.
  - Résultat attendu : Carte visible, temps sous seuil.
  - Code :

```
assert explorer_page.is_map_visible(), "Map canvas did not become visible within timeout"
assert load_time <= Config.MAP_LOAD_THRESHOLD, f"Map loaded too slowly: {load_time}s"
```

- **test-03-map-controls-present**
  - Objectif : Vérifier la présence des contrôles Home, Follow, Random, Zoom.
  - Scénario : Accéder à la carte, vérifier chaque contrôle par ID.

- Résultat attendu : Tous les contrôles sont visibles.
- Code :

```
assert visible, f"{desc}({ctrl_id}) not visible"
assert explorer_page.is_element_visible((By.ID, 'zoom-in')) or
    explorer_page.is_element_visible((By.ID, 'zoom'))
```

### 5.2.7 Suite 1 Functional HomePages

- test-HOME-01-homepage-loads-and-sections-visible : **FAILED**
- test-HOME-02-navigation-links : **PASSED**
- test-HOME-03-signin-cta : **PASSED**
- test-HOME-04-news-updates-links : **PASSED**

Échec détaillé :

- **test-HOME-01-homepage-loads-and-sections-visible**
  - Objectif : Vérifier la présence des sections principales sur la home.
  - Scénario : Charger la home, vérifier titre, news, liens.
  - Résultat attendu : Toutes les sections présentes.
  - Code :

```
assert "OpenSky Network" in driver.title, f"Unexpected
homepage title: {driver.title}(strict)"
assert news, "Latest News & Updates section not found"
```

### 5.2.8 Suites Inline

- Tests inline (About / Feed / Data inline variants) : la plupart **PASSED**, quelques cas **SKIPPED** (publications, téléchargements) selon disponibilité du contenu.

### 5.2.9 Suite 2 : Performance (détailé)

- test-perf-01-homepage-cold-load-tti : **FAILED**
- test-perf-02-homepage-warm-load-cache-enabled : **FAILED**
- test-perf-06-api-docs-on-slow-3g : **SKIPPED**
- test-perf-11-404-page-speed : **FAILED**
- test-perf-k6-run-smoke : **SKIPPED (k6 absent)**
- TestPerformanceSuite : :test-08-map-load-time : **PASSED**
- TestPerformanceSuite : :test-09-search-response-time : **PASSED**
- TestPerformanceSuite : :test-10-flight-details-panel-response-time : **SKIPPED**
- TestPerformanceSuite : :test-11-map-interaction-stress : **FAILED**
- TestPerformanceSuite : :test-12-concurrent-sessions-load-test : **PASSED**
- TestPerformanceSuite : :test-13-page-refresh-stress-test : **PASSED**

extbfÉchecs performance (extraits) :

- **test-perf-01-homepage-cold-load-tti**

- Objectif : Mesurer LCP/TBT/CLS et proxy load time sous réseau 4G.
- Résultat attendu : LCP  $\leq$  1.5s, load time  $\leq$  2.0s, TBT  $\leq$  0.05s, CLS  $\leq$  0.05.
- Assertion ratée (extrait) :

```
assert lcp <= 1.5, f"LCPUtooohigh:{lcp}s"
assert load-time <= 2.0, f"Loadtime(proxyTTI)tooohigh:{load-time}s"
```

- **test-perf-02-homepage-warm-load-cache-enabled**

- Objectif : Vérifier temps de la visite répétée (cache habilité).
- Résultat attendu : LCP  $\leq$  1.0s, load time  $\leq$  1.2s.
- Assertion ratée (extrait) :

```
if settings.audit_strict:
    assert lcp <= 1.0, f"WarmLCPUtooohigh:{lcp}s"
    assert load-time <= 1.2, f"WarmloadTTIproxytooohigh:{load-time}s"
```

- **test-perf-11-404-page-speed**

- Objectif : Vérifier la latence et la taille de la page 404.
- Résultat attendu :  $\leq$  500 ms et  $\leq$  50 KB.
- Assertion ratée (extrait) :

```
assert elapsed <= 500, f"404page response too slow:{elapsed}ms"
assert len(r.content) <= 50 * 1024, f"404payload too large:{len(r.content)}bytes"
```

- **TestPerformanceSuite : :test-11-map-interaction-stress**

- Objectif : Stress interactions (double-click, drag) et vérifier durée totale.
- Résultat attendu : total-time  $<$  25s.
- Assertion ratée (extrait) :

```
assert total-time < 25, f"Mapinteraction stress test took too long:{total-time:.2f}s."
```

### 5.2.10 Suite 3 : Cross-Browser

- Majorité des cas sur Chrome : **PASSED**
- Plusieurs cas sur Firefox, Edge, Safari : **SKIPPED** (environnements non disponibles)
- test-cb-matrix[CB-09-chrome-old-any] : **PASSED**

### 5.2.11 Suite 4 : Responsive (détailé)

- RWD-01 (320x568 iPhone SE) : **FAILED**
- RWD-02 (375x667 iPhone 8) : **FAILED**
- RWD-03 (390x844 iPhone 14) : **FAILED**
- RWD-04 (414x896 iPhone Plus) : **FAILED**
- RWD-05 (360x800 Budget Android) : **FAILED**
- RWD-06 (768x1024 iPad) : **PASSED**
- RWD-07 (1024x1366 iPad Pro landscape) : **FAILED**
- RWD-08 (1440x900 Laptop) : **FAILED**
- RWD-09 (1920x1080 Desktop) : **FAILED**
- RWD-10 (2560x1440 2K) : **FAILED**
- RWD-11 (1366x768 Zoom125) : **PASSED**
- RWD-12 (1366x768 Dark Mode forced) : **PASSED**

extbfPoints d'échec responsive (générique) :

- Symptômes : scroll horizontal, overflow de blocs <pre>/<code>, tap targets insuffisants.
- Extraits d'assertions ratées :

```
assert not has_scroll, f"Horizontal scroll detected for {case_id} at {w}x{h} on {p}"
assert ok_tap, f"No tap targets >= 44px detected for {case_id} at {w}x{h} on {p}"
assert not overflow, f"Code block overflow detected for {case_id} at {w}x{h} on {p}"
```

- Recommandation rapide : prioriser corrections CSS (overflow, max-width), vérifier styles pour code blocks et responsive meta viewport.

### 5.2.12 Suite 5 : Sécurité (détailé)

- test-security-01-https-redirect : **PASSED**
- test-security-02-security-headers-public-pages[/] : **FAILED**
- test-security-02-security-headers-public-pages[/about] : **FAILED**
- test-security-02-security-headers-public-pages[/data] : **FAILED**
- test-security-02-security-headers-public-pages[/feed] : **FAILED**

extbfErreur typique (headers manquants) :

- Principe : vérifier HSTS, X-Content-Type-Options=nosniff, X-Frame-Options ou CSP frame-ancestors, Referrer-Policy, Permissions-Policy.
- Extrait d'assertion :

```
assert not missing, f"Missing/weak security headers on url:{url}: {missing}"
```

- Recommandation : ajouter HSTS, X-Content-Type-Options : nosniff, et CSP/frame-ancestors ; documenter l'impact des proxy/CDN.

### 5.2.13 Suite 6 : Intégrité des Liens

- Principaux parcours internes testés : Home, About, FAQ, Data, Feed (raspberry) — la majorité **PASSED**.
- Aucun lien critique cassé identifié dans l'index fourni.

### 5.2.14 Suite 7 : Erreurs Console / JS runtime

- test-js-01-no-console-errors-on-key-pages : toutes les variantes citées **PASSED** (Home, About, API, Feed).
- Conclusion : pas d'erreurs console critiques détectées lors des runs fournis.

### 5.2.15 Suite 8 : Fuzzing / Inputs

- test-input-01-map-search-rejects-dom-xss : payloads fournis **PASSED** — aucune exécution d'alerte détectée.

## 5.3 Résumé et Priorisation (Quick wins)

- Priorité Haute : corriger headers de sécurité (pages /, /about, /data, /feed) ; responsive overflows critiques (mobile large desktop viewports).
- Priorité Moyenne : perf cold/warm LCP et stress map (investigation instrumentation LCP/CDP).
- Priorité Basse : étendre cross-browser (environnements manquants) et ajouter historisation métriques.

## Chapitre 6

# Conclusion et Perspectives

### 6.1 Points Forts du Projet

- **Approche orientée bug-hunting** : Framework conçu pour produire des signaux exploitables plutôt que de simplement passer des tests.
- **Preuves automatisées** : Captures, snapshots HTML et logs console attachables automatiquement aux tickets Jira.
- **Mode audit strict** : Permet de transformer des findings en critères bloquants pour des runs de qualification.
- **Couverture large** : Suites fonctionnelles, performance, responsive, sécurité, intégrité liens, console et fuzzing.
- **Extensibilité** : Architecture POM et modules utilitaires facilitant l'ajout de tests et d'artefacts supplémentaires.

### 6.2 Perspectives

- Prioriser corrections sécurité (headers) et responsive overflows identifiés.
- Ajouter historisation des métriques performance pour détection de tendances.
- Étendre les runs cross-browser automatisés (Selenium Grid / CI) pour réduire les SKIP-PED liés aux environnements.
- Intégrer un module d'accessibilité (axe-core) et un dashboard centralisé des findings.