

15ème
édition de la

**Journée
Française
des Tests
Logiciels**



12 juin 2023



Beffroi de
Montrouge

50 Nuances d'Agile

Tutoriel automatisation

Initiation à Playwright



Mahdi, GHARSALLAH
Generali France



Sommaire

1. Présentation du groupe Generali
2. La démarche d'automatisation chez Generali
3. Présentation de la librairie Playwright
4. Installation de Playwright
5. Création d'un test
6. Débogage
7. Ajout d'un test visuel
8. Intégration du test dans un CI Jenkins
9. Conclusion



Durée: 90 min

Le Groupe Generali



Créé en 1831: un des principaux groupes mondiaux d'assurance



+ 60 millions d'assurés dans plus de 50 pays



73 000 collaborateurs dans le monde dont 5 000 en France



Centre d'Ingénierie Test Qualité créé en 2020 - 45 collaborateurs

Le Centre d'Ingénierie Test Qualité Generali



110 projets de recette en 2022



3 expertises: Test Management, Delivery Center, Automation Factory



15 offres de services: cadrage et suivi de stratégie de recette, Conception et exécutions de tests, Automatisation de tests de non-régression etc.



L'Automation Factory Generali



Entité transverse



Créée en juin 2020



Composée d'un Automation Lead, un Automaticien Senior, un Automaticien junior

MISSIONS

- Mettre à disposition des tests automatisés accessibles à toutes les équipes DEV, QA, MOA, OPS
- Garantir un ROI de l'automatisation
- Permettre aux équipes projet de tester plus tôt plus vite et plus souvent



AUTOMATION FACTORY

LIVRER EN TOUTE CONFIANCE

L'Automation Factory en chiffres

- 15 projets d'automatisation
- Type d'applications
 - Web
 - API REST / Webservices
 - Batches
 - Mainframe
 - Client lourd
- Outils: Cypress, Playwright,

RobotFramework, Postman, Uiopath

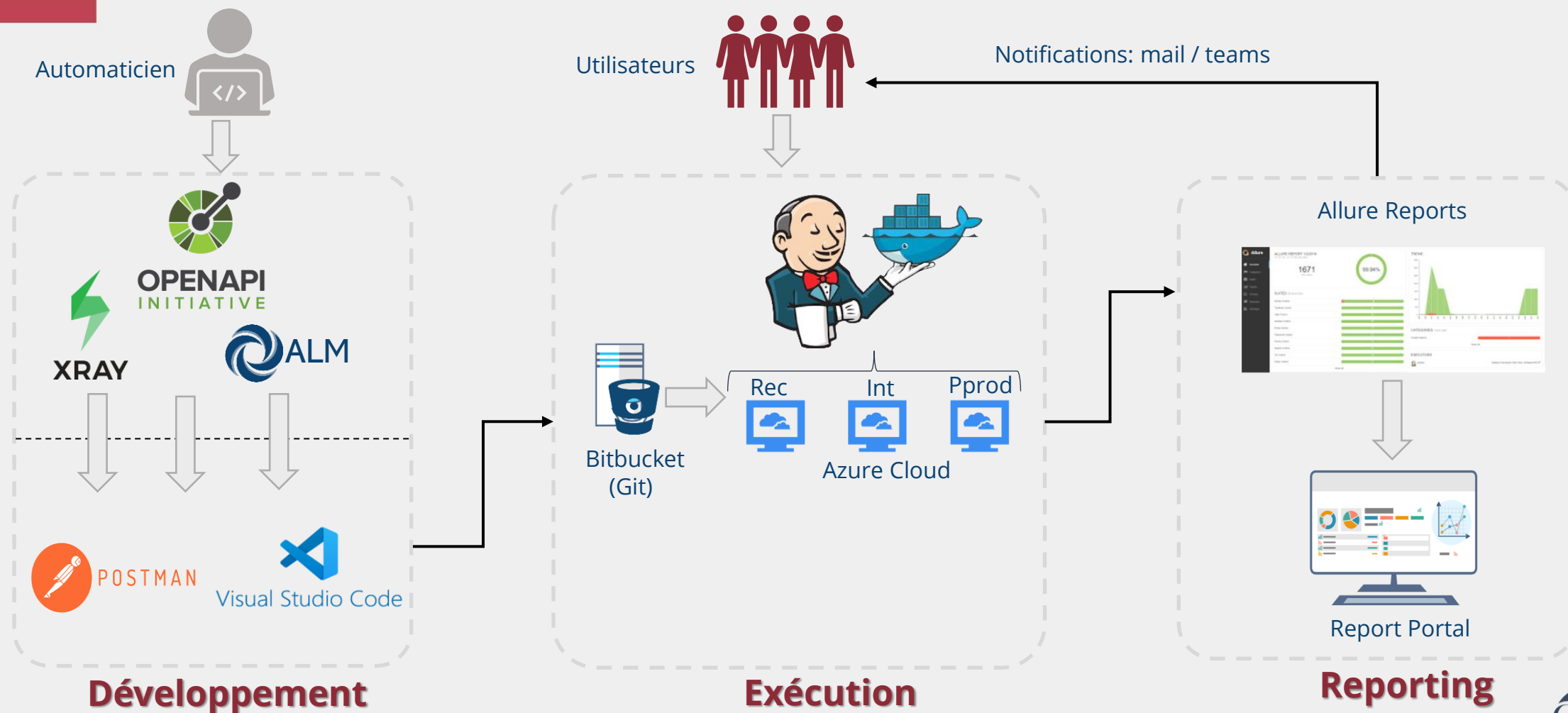
1280 tests automatisés

~300 tests exécutés chaque jour

55% Couverture moyenne TNR

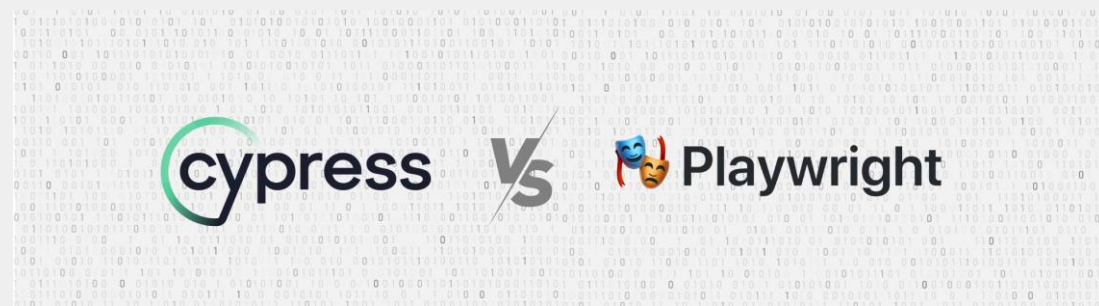
80% de taux de détection d'anomalies par les tests automatisés

Notre démarche d'automatisation



Playwright chez Generali

- Depuis Janvier 2023
- Refonte de certains projets Cypress sur Playwright
- Environ **75** tests PW actuellement
- Objectif 2023: **200** tests



15ème
édition de la

**Journée
Française
des Tests
Logiciels**



12 juin 2023



Beffroi de
Montrouge

50 Nuances d'Agile

Tutoriel automatisation

Initiation à Playwright



Mahdi, GHARSALLAH
Generali France



Sommaire

1. Présentation du groupe Generali
2. La démarche d'automatisation chez Generali
3. Présentation de la librairie Playwright
4. Installation de Playwright
5. Création d'un test
6. Débogage
7. Ajout d'un test visuel
8. Intégration du test dans un CI Jenkins
9. Conclusion



Durée: 90 min

L'outil Playwright

- Librairie **Open Source** développé par la communauté **Microsoft**
- Permet d'automatiser des interfaces Web et des APIs
- Supporte les navigateurs basés sur **Chromium**, **Firefox** et **Webkit**
- Multiplateformes: **Windows**, **Linux**, **macOS**
- Peut être utilisé avec différents langages de programmation tels que: **Java**, **Python**, **C#**, **Node JS** et **Javascript**
- Très configurable, rapide et facile d'utilisation



Installation de Playwright

Prérequis:

- NodeJS
- VS Code



Installation:

- Créer un répertoire « **JFTL_PLAYWRIGHT_TUTO** »
- L'ouvrir dans VS Code
- Dans le terminal, taper la commande: **npm init playwright@latest**
- Valider les choix par défaut
- **Done !!!**



Configuration du projet



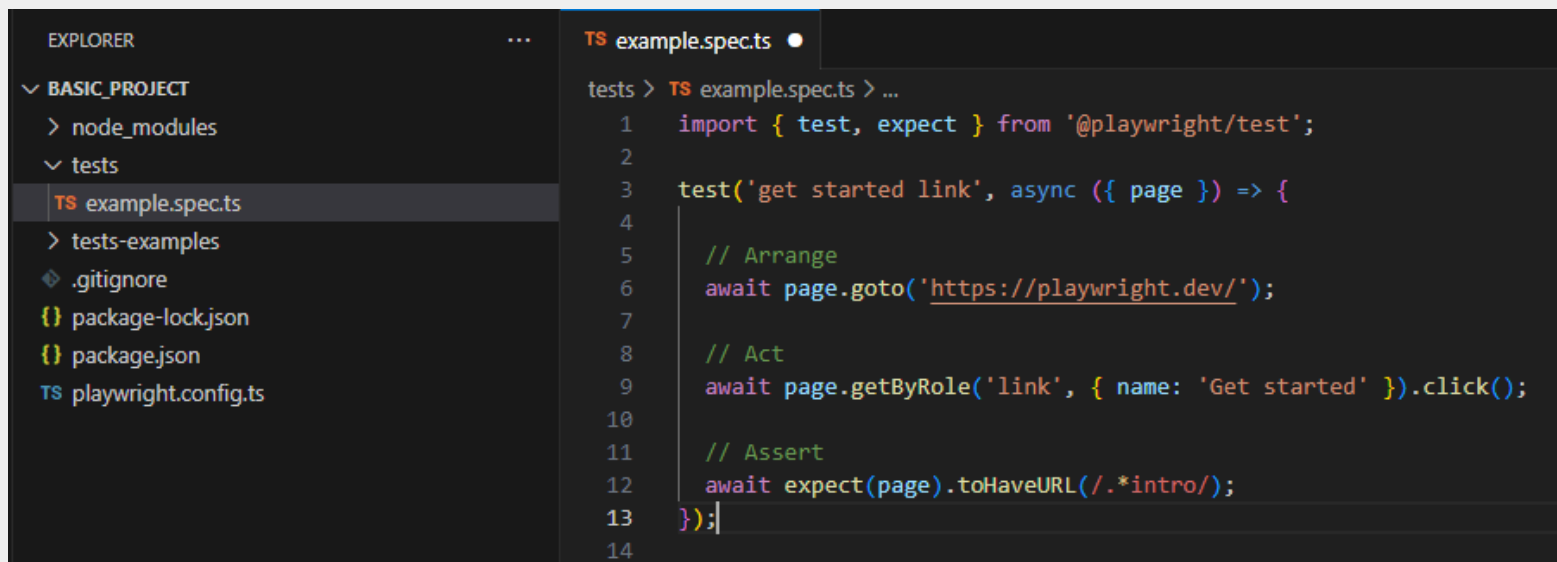
<https://playwright.dev/docs/test-configuration>

- Le fichier **playwright.config.ts**
- Saisir la base URL
- Commenter les sections non nécessaires
- Ajout d'attributs: trace, video,

```
TS playwright.config.ts > [?] default
9  /**
10  * See https://playwright.dev/docs/test-configuration.
11  */
12  export default defineConfig({
13    testDir: './tests',
14    /* Run tests in files in parallel */
15    snapshotDir: './snapshots',
16    fullyParallel: true,
17    /* Fail the build on CI if you accidentally left test.only in the source code. */
18    forbidOnly: !!process.env.CI,
19    /* Retry on CI only */
20    retries: process.env.CI ? 2 : 0,
21    /* Opt out of parallel tests on CI. */
22    workers: process.env.CI ? 1 : undefined,
23    /* Reporter to use. See https://playwright.dev/docs/test-reporters */
24    reporter: 'html',
25    /* Shared settings for all the projects below. See https://playwright.dev/docs/api/class-testoptions. */
26    use: {
27      /* Base URL to use in actions like `await page.goto('/')`. */
28      // baseURL: 'http://127.0.0.1:3000',
29
30      /* Collect trace when retrying the failed test. See https://playwright.dev/docs/trace-viewer */
31      trace: 'on-first-retry',
32    },
33  },
```

Anatomie d'un test Playwright

- Import de librairies
- Déclaration d'un test: nom du test, async, page fixture
- Body du test: await, goto, locator, expect
- La règle du AAA: **A**rrange, **A**ct, **A**ssert
- Exécuter le test: `npx playwright test --headed`



The screenshot shows the Visual Studio Code interface. On the left, the Explorer sidebar displays a project structure with folders 'BASIC_PROJECT' and 'tests', and files like 'example.spec.ts', 'tests-examples', '.gitignore', 'package-lock.json', 'package.json', and 'playwright.config.ts'. The main editor area shows the content of 'example.spec.ts', which is a TypeScript file containing a Playwright test. The code is as follows:

```
tests > TS example.spec.ts > ...
1  import { test, expect } from '@playwright/test';
2
3  test('get started link', async ({ page }) => {
4
5      // Arrange
6      await page.goto('https://playwright.dev/');
7
8      // Act
9      await page.getByRole('link', { name: 'Get started' }).click();
10
11     // Assert
12     await expect(page).toHaveURL(/.*intro/);
13 };
```

Création d'un test



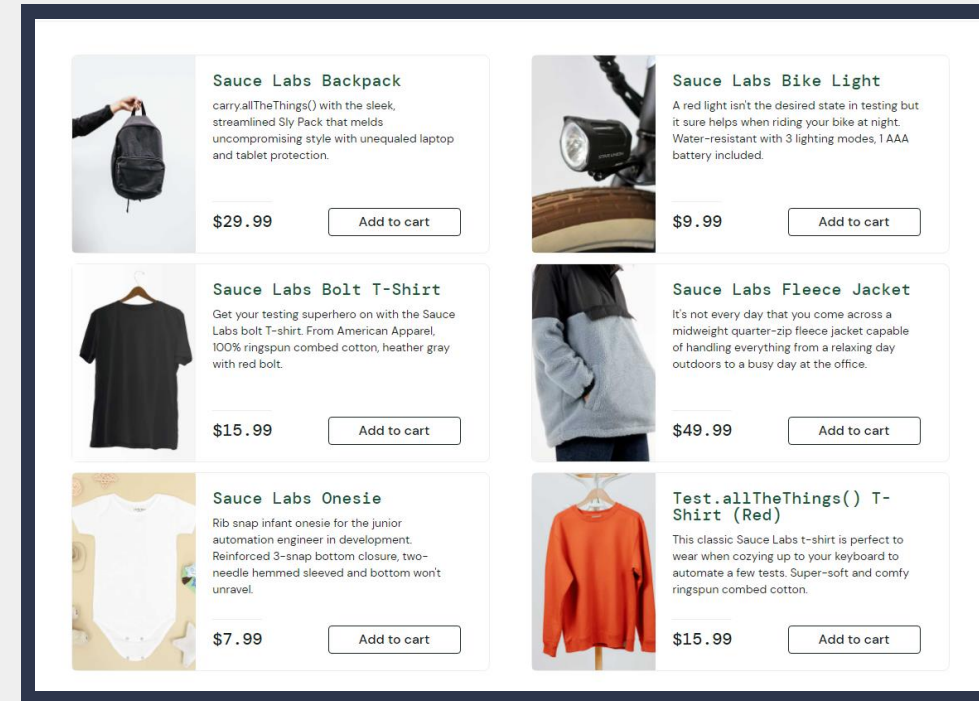
<https://playwright.dev/docs/writing-tests>

Scénario:

- Aller sur un site marchand: <https://www.saucedemo.com/>
- Se connecter
- Ajouter des articles au panier
- Ouvrir le panier
- Payer
- Valider le message de confirmation
- Se déconnecter

Approche de Scripting:

- Utiliser le générateur de code intégré de Playwright
- Améliorer le code généré



Amélioration du script



<https://playwright.dev/docs/pom>

<https://playwright.dev/docs/best-practices>

- Vérifier les « Locators »
- Créer des Page Objects
- Ajouter un beforeEach(), afterEach()
- Externaliser les données de test dans un fichier json
- Ajouter les assertions

```
test.describe('adding articles to cart', () => {
  let page: any;
  let loginPage: LoginPage;
  let shoppingPage: ShoppingPage;
  let checkoutPage: CheckoutPage;
  let orderConfirmationPage: OrderConfirmationPage;

  test.beforeEach(async ({ browser }) => {
    page = await browser.newPage();
    loginPage = new LoginPage(page);
    shoppingPage = new ShoppingPage(page);
    checkoutPage = new CheckoutPage(page);
    orderConfirmationPage = new OrderConfirmationPage(page);
  });

  test.afterEach(async () => {
    await page.close();
  });

  test('should add items to cart and complete checkout', async () => {
    await loginPage.navigateTo();
    // await loginPage.blockPageImages();
    await loginPage.login(testData.user.username, testData.user.password);

    await shoppingPage.checkPageDisplay();
    await shoppingPage.addToCart(testData.productNames.backpack);
    await shoppingPage.addToCart(testData.productNames.tshirt);
    await shoppingPage.openCart();

    await checkoutPage.checkout();
    await checkoutPage.fillForm(testData.checkoutInfo.firstName, testData.checkoutInfo.lastName, testData.checkoutInfo.postalCode);
    await expect(orderConfirmationPage.isConfirmationMessageDisplayed()).toBeTruthy();

    await orderConfirmationPage.backToProducts();

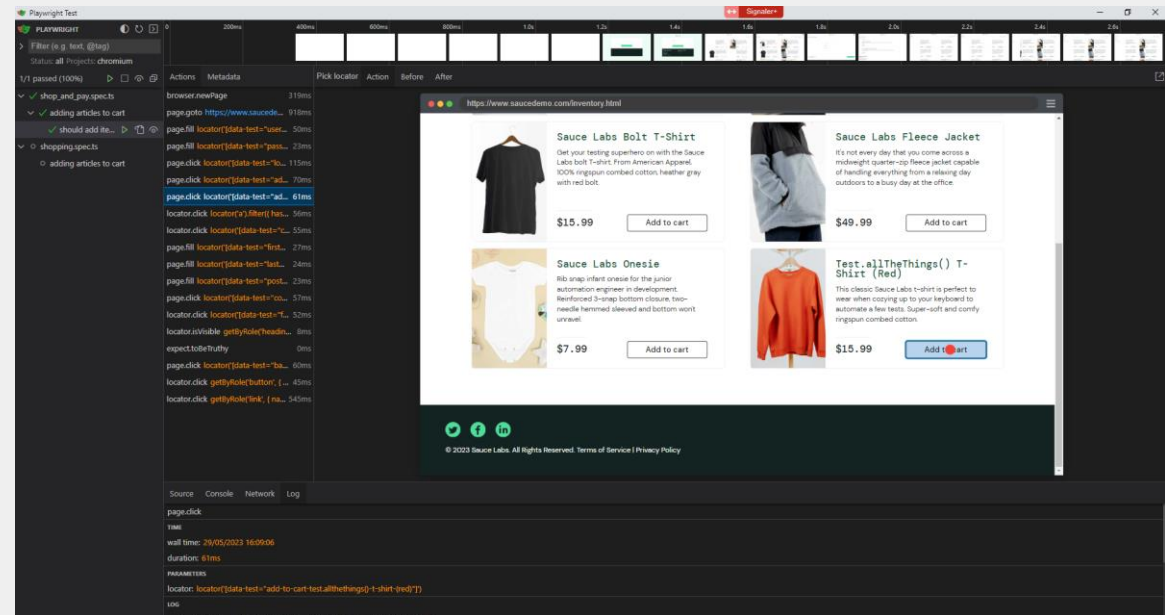
    await shoppingPage.openMenu();
    await shoppingPage.logout();
  });
});
```


Débogage



<https://playwright.dev/docs/debug>

- Debug mode et Break points
- Playwright UI
- Trace viewer



Ajout d'un test visuel



<https://playwright.dev/docs/test-snapshots>

- Qu'est-ce que le visual testing ?
- Rajouter un visual test dans Playwright
- Comparer les snapshots
- Mettre à jour des snapshots suite à des changements d'IHM



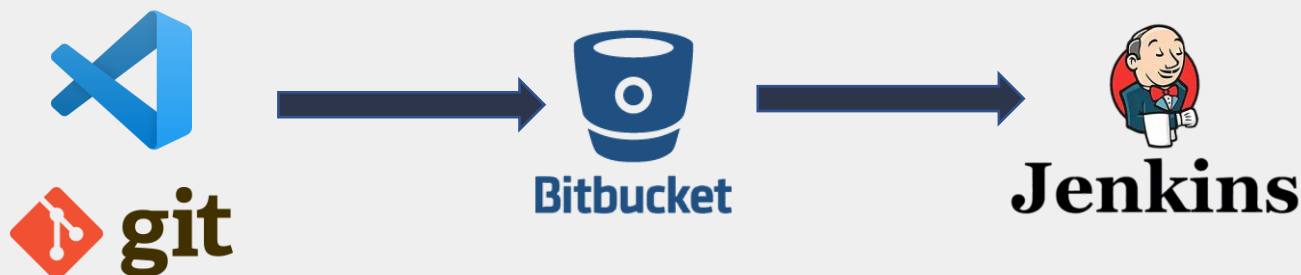
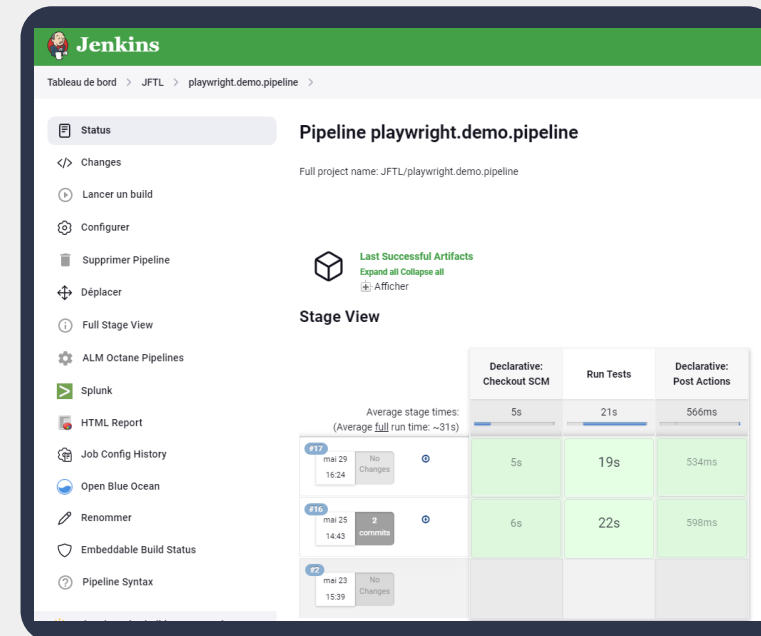
Intégration du test dans une pipeline CI



<https://playwright.dev/docs/ci>

Live Demo:

- Versionner le code source avec git / Bitbucket
- Créer un jenkinsfile
- Lancer les tests depuis un serveur jenkins
- Consulter et partager le rapport HTML



Conclusion

- Vos feedbacks
- Allez-vous considérer Playwright pour vos futurs projets d'automatisation ?
- Questions / Réponses



15ème
édition de la
**Journée
Française
des Tests
Logiciels**



12 juin 2023



Beffroi de
Montrouge

Merci de votre écoute !



Comité Français
des Tests Logiciels

Annexes

- Source code du tuto: <https://github.com/mgharsallah79/tuto-playwright-jftl>

- Commandes Playwright utiles:



Playwright_cmds.txt

- Mon contact:

