End-to-End Automation Workflow for user-journey-explorer Container

This document provides the full sequence of commands for building, running, testing, and generating Allure reports using the user-journey-explorer Docker image. It also includes a diagnostic checklist and optional PowerShell automation script reference.

1️⃣ Build (or rebuild) the Docker image

-
  ```
  cd C:\Projects\XmlPlay\my_pytest_playwright_project
  docker build -t user-journey-explorer:fast .
  ```

2️⃣ Verify the image was built correctly

- `docker images user-journey-explorer:fast`

3️⃣ Confirm entrypoint

- `docker inspect user-journey-explorer:fast --format '{{json .Config.Entrypoint}}'`

4️⃣ Generate XML / HTML / PDF test artifacts

-
  ```
  docker run --rm `
    -v "${pwd}/data:/app/data" `
    user-journey-explorer:fast -- UJX PUBLIC USER
  ```

5️⃣ Run pytest (Allure-enabled)

-
  ```
  docker run --rm `
    -e ROLE_KEY=UJX_PUBLIC_USER `
    -v "${pwd}:/app" `
    -v "${pwd}/allure-results:/app/allure-results" `
    user-journey-explorer:fast -- pytest --alluredir=/app/allure-results -v
  ```

6️⃣ Verify pytest results exist

- `Get-ChildItem .\allure-results`

7️⃣ Generate Allure report (HTML)

-
  ```
  docker run --rm `
    -v "${pwd}/allure-results:/app/allure-results" `
    -v "${pwd}/allure-report:/app/allure-report" `
  ```

```
 user-journey-explorer:fast -- allure generate /app/allure-results --clean -o
/app/allure-report
```

8️⃣ View the Allure report

-

```
cd allure-report
python -m http.server 8080
# Visit http://localhost:8080
```

9️⃣ Diagnostics

-

```
docker images user-journey-explorer:fast
docker ps -a --latest
docker run -it --rm --entrypoint bash user-journey-explorer:fast
docker run --rm user-journey-explorer:fast -- allure --version
Get-ChildItem .\data\output\latest_*.json
```

🔟 Optional — run multiple roles in sequence

-

```
$roles = @("UJX_PUBLIC_USER", "UJX_PUBLIC_ADMIN", "UJX_PRIVATE_USER")
foreach ($role in $roles) {
  docker run --rm `
    -e ROLE_KEY=$role `
    -v "${pwd}:/app" `
    -v "${pwd}/allure-results:/app/allure-results" `
    user-journey-explorer:fast -- pytest --alluredir=/app/allure-results -v
}
```

PowerShell Automation Script (Run-FullCycle.ps1)

This script automates the full build → run → test → report cycle with error handling and timestamps.

```
param(
    [string]$OrgId = "UJX",
    [string]$Project = "PUBLIC",
    [string]$Role = "USER"
)

$ErrorActionPreference = "Stop"
$timestamp = Get-Date -Format "yyyyMMdd_HHmmss"
Write-Host "[$timestamp] 🚀 Starting Full Test Cycle" -ForegroundColor Cyan

# Build image
Write-Host "`n🧱 Building Docker image..." -ForegroundColor Yellow
docker build -t user-journey-explorer:fast .

# Generate XML/HTML/PDF
Write-Host "`n▶ Running Excel converter..." -ForegroundColor Yellow
docker run --rm -v "${pwd}/data:/app/data" user-journey-explorer:fast -- $OrgId $Project $Role

# Run pytest with Allure
Write-Host "`n🧪 Running pytest..." -ForegroundColor Yellow
docker run --rm -e ROLE_KEY="${OrgId}_${Project}_${Role}" -v "${pwd}:/app" -v "${pwd}/allure-results:/app/allure-results" user-journey-explorer:fast -- pytest --alluredir=/app/allure-results -v

# Generate Allure report
Write-Host "`n📊 Generating Allure report..." -ForegroundColor Yellow
docker run --rm -v "${pwd}/allure-results:/app/allure-results" -v "${pwd}/allure-report:/app/allure-report" user-journey-explorer:fast -- allure generate /app/allure-results --clean -o /app/allure-report

Write-Host "`n✅ Cycle complete! View report at ./allure-report/index.html" -ForegroundColor Green
```