Ismail
KTAMI

JOURNÉE FRANÇAISE DES TESTS LOGICIELS

Méta-Modélisation et tests pilotés par les données : l'alliance parfaite pour une automatisation ***industrialisée***

17 JUIN 2025

BEFFROI DE MONTROUGE

CFTL
Comité Français des Tests Logiciels

Au programme

JOURNÉE FRANÇAISE DES TESTS LOGICIELS

- Tests pilotés par les données
- REX Automatisation 150 tests en 2 jours
- Méta-modélisation & Tests dynamiques
- Bibliothèque de tests

Aller vite, bien concevoir

Robert C. Martin

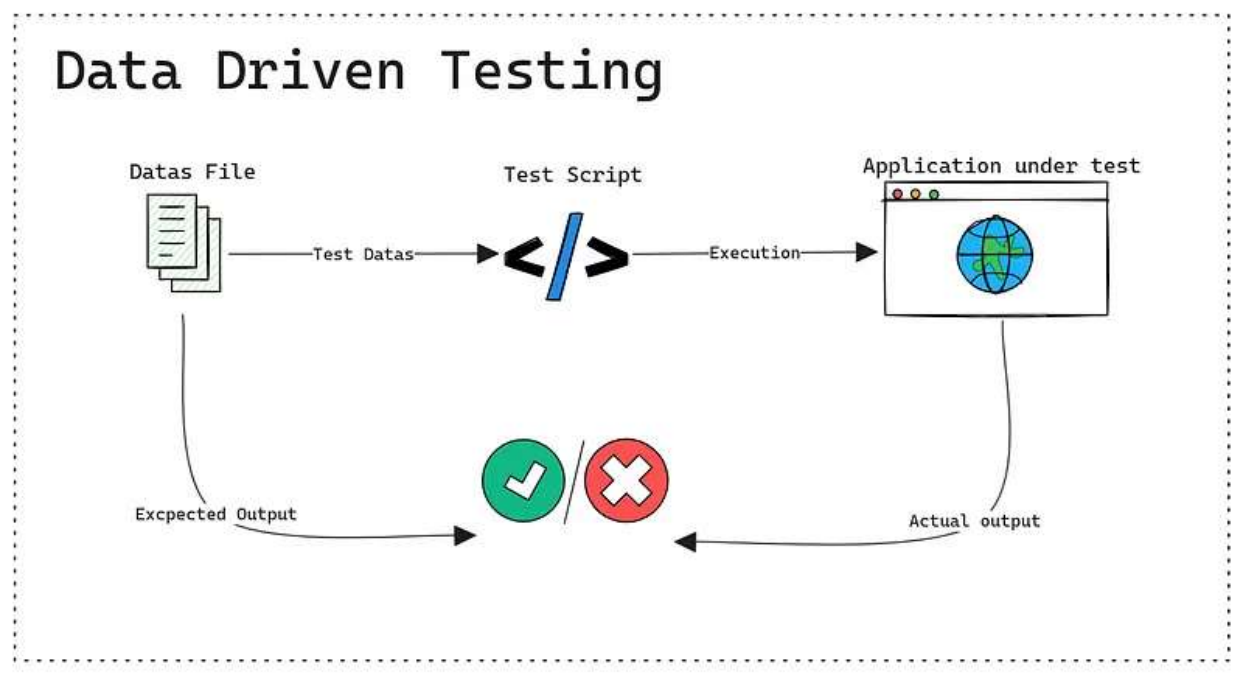
« La seule manière d'aller vite, c'est de bien faire les choses. »

https://www.dailydot.com/unclick/economy-will-collapse-if-you-binge-watch-house-of-cards/

Tests pilotés par les données

Définition :

Un seul scénario, et on le fait tourner avec différentes valeurs extraites d'un fichier ou d'une source externe.



https://medium.com/@thananjayan1988/dynamic-data-driven-tests-playwright-c7426eb877c0

- Modularité
- Réutilisabilité
- Scalabilité

Tests pilotés par les données

JOURNÉE FRANÇAISE DES TESTS LOGICIELS

Filtres produits

Catégorie : Électronique, Meubles, Mode,…
Statut : Promo, Nouveauté,…

```
Feature:
Scenario: Filtrer les produits par le statut disponible et la catégorie Électronique
Given Je suis sur la page liste produits
When  Je sélectionne disponible dans la liste Statut
And   Je sélectionne électronique dans la liste Catégorie
Then  je vérifie que le filtre est appliqué
And   La table affiche les lignes avec statut disponible et la catégorie électronique
```

Tests pilotés par les données

Filtres produits

Catégorie : Électronique, Meubles, Mode
Statut : Promo, Nouveauté

```
Scenario Outline: Filtrer des produits par Catégorie : <categorie> et Statut : <statut>
  Given Je suis sur la page produits
  When Je sélectionne "<categorie>" dans la liste Catégorie
  And Je sélectionne "<statut>" dans la liste Statut
  And Je clique sur rechercher
  Then La table affiche uniquement les lignes correspondant à la catégorie "<categorie>" et au statut "<statut>"

Examples:
| categorie   | statut      |
| Électronique | Disponible  |
| Électronique | En rupture  |
| Meubles     | Disponible  |
| Meubles     | En rupture  |
| ...         | ...         |
```

Scenario Outline

Tests pilotés par les données

5^5=3125 combinaisons possibles à tester :

	Statique (3 125)		**Piloté par les données (1)**
	~ 32 JH (5 mins / scénario)	VS	~ 15 mins ~ 2 heures pour les données
	~ 32 JH (5 mins / scénario)		~ 1 JH
	~ 64 JH (10 mins / scénario)		~ 10 mins

 **≈ 97% sur la rédaction, l'automatisation et l'exécution des tests.**

Automatisation de 150 tests en 2 jours

REX Cdiscount Une étape vers l'industrialisation :

JOURNÉE FRANÇAISE DES TESTS LOGICIELS

Scénario : Visibilité du bouton « ajouter un produit » pour un utilisateur ADMIN.

```
Scenario: Vérifier la visibilité du bouton ajouter un produit pour le rôle ADMIN
Given un utilisateur admin est connecté
When il navigue sur la page produits
Then le bouton ajouter un produit doit être affiché
```

Comment adapter ce scénario pour tester d'autres ?

- **Rôles**
- **Etats de composants**
- **Types de composants**
- **Pages**

Un scénario abstrait

Automatisation de 150 tests en 2 jours

REX Cdiscount Une étape vers l'industrialisation :

JOURNÉE FRANÇAISE DES TESTS LOGICIELS

Étape 1 : Extension du scénario à d'autres états des composants :

```
Scenario Outline: [Page Produits] - Visibilité du bouton <button_name> pour le rôle <user_role>
Given un <user_role> est connecté
When il navigue sur la page produits
Then le bouton <button_name> doit être <expected_state>

Examples:
| user_role | button_name          | expected_state |
| ADMIN     | ajouter un produit | visible        |
```

visible | invisible | enabled | disabled | ● ● ●

Automatisation de 150 tests en 2 jours

REX Cdiscount Une étape vers l'industrialisation :

JOURNÉE FRANÇAISE DES TESTS LOGICIELS

Étape 2 : Extension du scénario à d'autres types de composants :

```
Scenario Outline: [Page Produits] - Visibilité du composant <component_name> pour le rôle <user_role>
Given un <user_role> est connecté
When il navigue sur la page produits
Then le composant <component_name> doit être <expected_state>

Examples:
| user_role | component_name | component_selector | expected_state |
| ADMIN | ajouter un produit | [data-testid='add-product'] | visible
```

button | input | form | div | • • •

Automatisation de 150 tests en 2 jours

REX Cdiscount Une étape vers l'industrialisation :

JOURNÉE FRANÇAISE DES TESTS LOGICIELS

Étape 3 : Extension du scénario à d'autres pages :

```
Scenario Outline: [<page_name>] Vérifier que le composant <component_name> est <expected_state> pour le rôle <user_role>
Given un <user_role> est connecté
When il navigue sur la page <page_name>
And Execute setup <setup>
Then le composant <component_selector> doit être <expected_state>

@DeliveryModes
Examples: Page Modes de livraison
| page_name      | user_role | setup | component_name                | component_selector                              | expected_state |
| Delivery Modes | ADMIN     |       | Ajouter un mode de livraison | [data-testid='add-delivery-mode-button'] | enabled        |
| Delivery Modes | RESTREINT |       | Ajouter un mode de livraison | [data-testid='add-delivery-mode-button'] | not.exist      |
```

[<page_name>] Vérifier que le composant *<component_name>* est *<expected_state>* pour le rôle **<user_role>**

Automatisation de 150 tests en 2 jours

REX Cdiscount Une étape vers l'industrialisation :

JOURNÉE FRANÇAISE DES TESTS LOGICIELS

```
Given('un {string} est connecté', (userRole) => {
  // Commande personnalisée pour se connecter en fonction du rôle utilisateur
  cy.loginByRole(userRole);
});

When("il navigue sur la page {string}", (pageName) => {
  cy.get(`a:contains("${pageName}")`).should('be.visible').click();
});

When("Exécute setup {string}", (setup) => {
 // Certaines pages nécessitent une action préalable pour rendre les composants accessibles
  if (setup)
    cy.get(`${setup}`)();
});

Then("le composant avec {string} doit être {string}", (componentSelector, expectedState) => {
  cy.get(componentSelector).scrollIntoView().should(expectedState);
})
```

 # Keyword Driven Testing

4

Keywords

Methods

Functions

Steps

Conception Driven Automation

Automatisation de 150 tests en 2 jours

REX Cdiscount Une étape vers l'industrialisation :

JOURNÉE FRANÇAISE DES TESTS LOGICIELS

99%
PASS RATE

- **Tests automatisés** — **150** Tests FRONT
- **Anomalies** — 4 Bugs critiques
- **Temps d'exécution** — **5** Minutes

Exemples :

I18N

E2E API

Composants Front

KEYWORD vs DATA driven testing :

JOURNÉE FRANÇAISE DES TESTS LOGICIELS

KDT
Keyword driven testing

Nos keywords sont lisibles
Nous avons 200 keywords

DDT
Data driven testing

On couvre tous les cas métier possibles
Nos données sont complètes

❌
- ConnexionAdmin()
- ConnexionClient()
- ConnexionManager()
- ConnexionInvite()

✅
- Connexion(profil, droits,...)

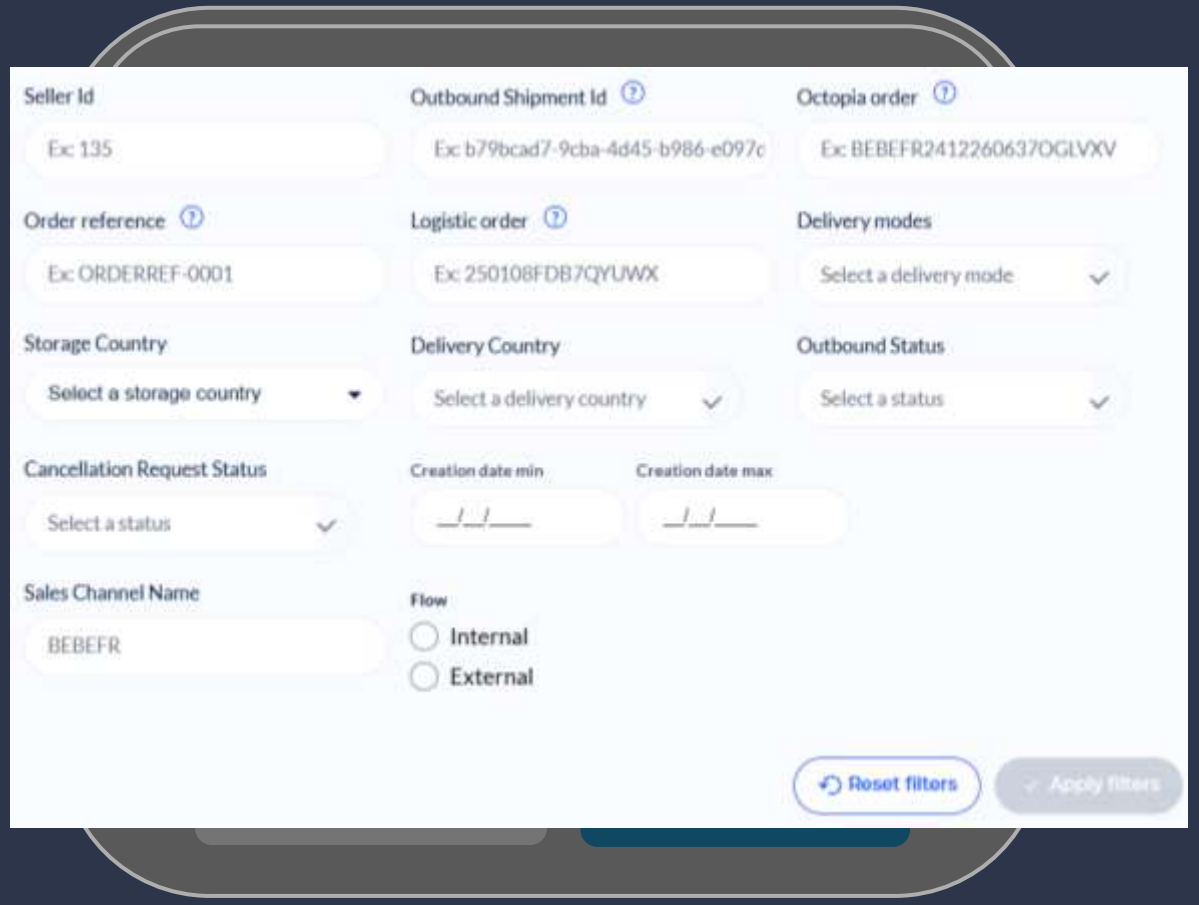
+

Profil	Droits	Page
admin	full	dashboard
client	readonly	catalogue
manager	readonly	dashboard

KDT donne l'architecture, DDT donne la vie. Ensemble, ils se multiplient.

Et si on arrêtait de réinventer la roue ?

JOURNÉE FRANÇAISE DES TESTS LOGICIELS

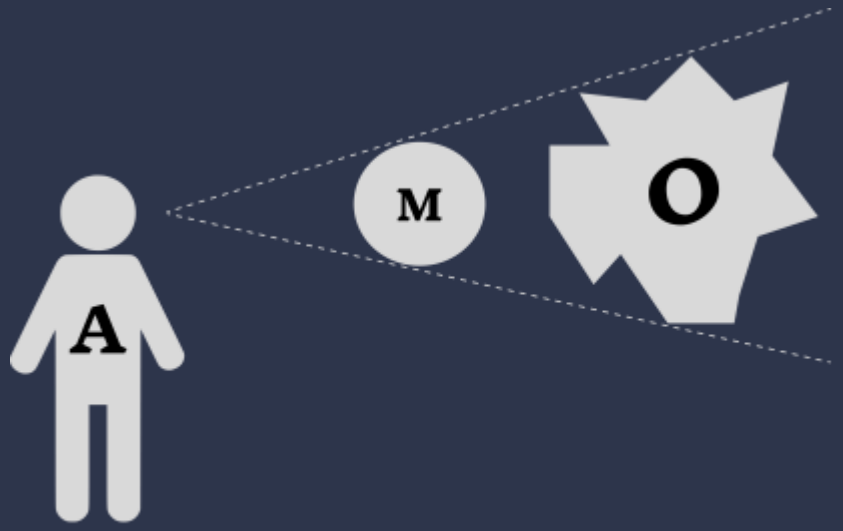


Critères d'acceptances :

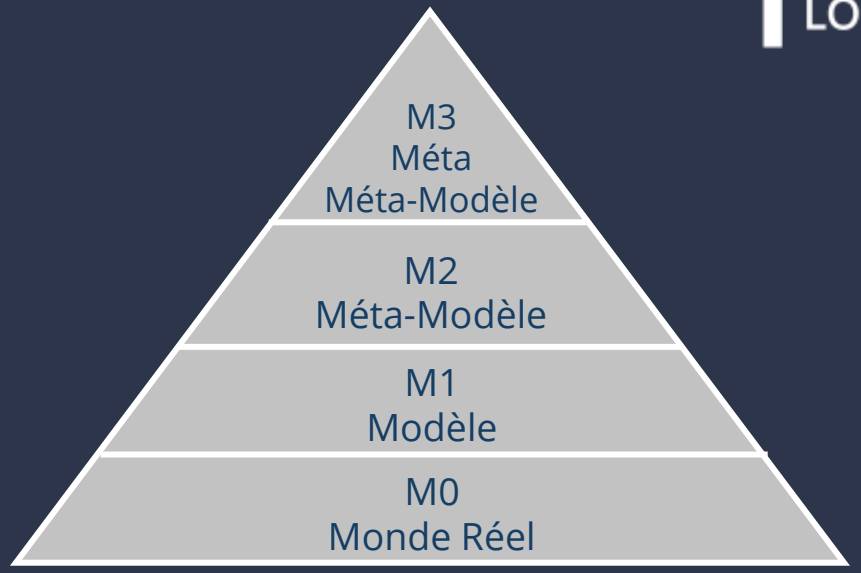
- Champs obligatoires
- Champs optionnels
- Bouton "Valider"
- Bouton "Réinitialiser"
- Format de champ
- Erreur format

Tests
Comportement si un champ requis est vide
Comportement si un champ optionnel est vide
Comportement si un champ est invalid
Réinitialisation du formulaire
Réinitialisation après soumission
Redirection après soumission
...

Modélisation & Méta-Modélisation



M. Minsky [Minsky 1968] : Vision abstraite du système
https://core.ac.uk/download/pdf/42969097.pdf



Architecture à 4 couches de méta-modélisation

Niveau	Rôle	Exemple
M2 Méta-modèle (MOF)	Décrit la grammaire UML	Class, Property, multiplicité…
M1 Modèle (UML)	Diagrammes	Diagramme de classes
M0 Instance concrète	Objet réeel (Code)	Class User { … } en Java

Modélisation avec Gherkin

JOURNÉE FRANÇAISE DES TESTS LOGICIELS

M2	Gherkin AST
M1	Scenario Outline
M0	Scenario

Méta-modéliser en freestyle avec JSON

JOURNÉE FRANÇAISE DES TESTS LOGICIELS

Critères d'acceptances :

- Champs obligatoires
- Mot de passe ≥ 8 caractères
- E-mail valide
- Âge > 16
- Bouton "Créer un compte" actif si champs valides
- Réinitialiser = vider les champs

Imaginez d'autres formulaires avec :

- Composants
- Règles d'interactivité
- Formats de données

Formulaire de création d'un compte utilisateur

Méta-modéliser en freestyle avec JSON

JOURNÉE FRANÇAISE DES TESTS LOGICIELS

```
{
  "formName": "Account Creation",
  "formLink": "FormLink",
  "fields": [
    { "id": "firstName", "label": "Nom", "type": "input", "required": true },
    { "id": "lastName", "label": "Prénom", "type": "input", "required": true },
    { "id": "gender", "label": "Genre", "type": "select", "required": true, "options": ["Male", "Female", "Other"] },
    {"...other fields..."}
  ],
  "actions": [
    { "data-testid": "create-account-btn", "label": "Créer un compte", "defaultState": "disabled" },
    { "data-testid": "reset-btn", "label": "Réinitialiser", "defaultState": "disabled" }
  ],
}
```

Champs

Actions

Tests

Méta-modéliser en freestyle avec JSON

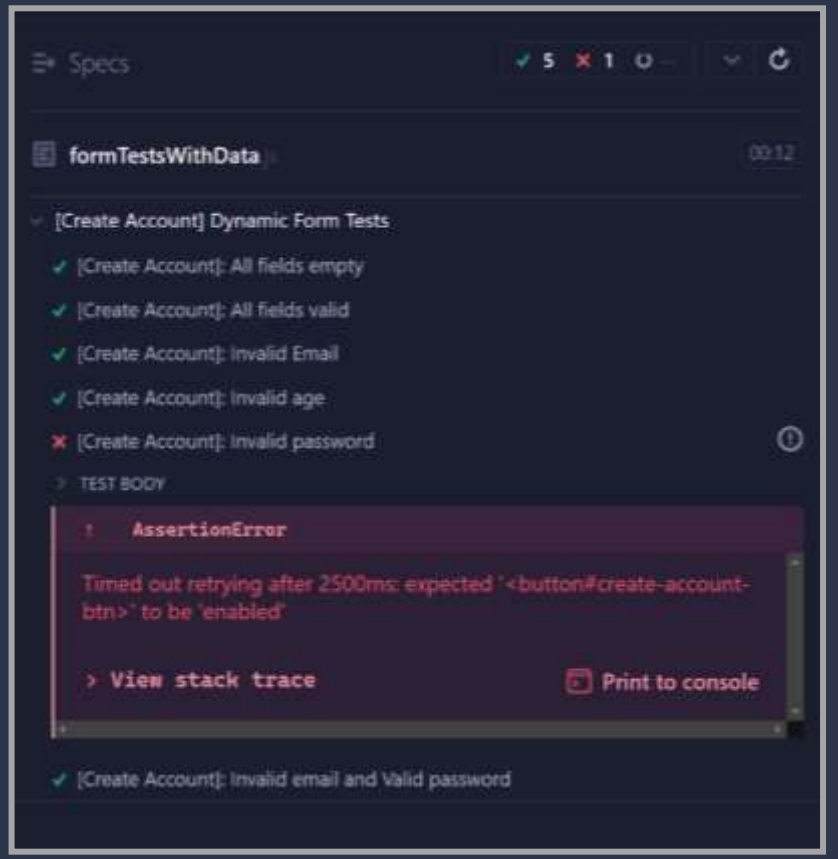
Tests dynamiques Cypress :

JOURNÉE FRANÇAISE DES TESTS LOGICIELS

```
// On importe la configuration JSON qui décrit le formulaire et les testCases
import accountForm from "../../fixtures/accountForm.json";

describe("[Create Account] Dynamic Form Tests", () => {
  // On parcourt chaque objet testCase défini dans le JSON
  accountForm.testCases.forEach((testCase) => {

    // Instancier un test
    it.only(`[Create Account]: ${testCase.testCaseName}`, () => {
      cy.visit(accountForm.pageLink);
      // Pour chaque champ à renseigner (clé = fieldId, valeur = value)
      Object.entries(testCase.inputs).forEach(([fieldId, value]) => {
        cy.get(`#${fieldId}`).then(($el) => {
          const elType = $el.prop("tagName").toLowerCase();
          if (elType === "select") {
            cy.get(`#${fieldId}`).select(value.toString());
          } else {
            cy.get(`#${fieldId}`).clear().type(value.toString());
          }
        });
      });
      // On vérifie les attentes (état attendu pour chacun des boutons)
      testCase.expectations.forEach((expectation) => {
        Object.entries(expectation).forEach(([actionTestId, expectedState]) => {
          cy.get(`[data-testid="${actionTestId}"]`).should("be." + expectedState);
        });
      });
    });
  });
```



Résultats d'exécution des tests dynamiques générés

Méta-modéliser en freestyle V2.0

Données de tests

```
{
  "fields": [
    { "id": "firstName", "type": "input", "required": true, "regex": "^[A-Z][a-z]{2,10}$" },
    { "id": "lastName", "type": "input", "required": true, "regex": "^[A-Z][a-z]{2,15}$" },
    { "id": "gender", "type": "select", "required": true, "options": ["Male", "Female", "Other"] },
    { "id": "age", "type": "number", "required": true, "min": 18, "max": 100 },
    { "id": "email", "type": "email", "required": true, "regex": "^[a-z0-9._%+-]+@[a-z0-9-]+\\.[a-z]{2,}$"
}
  ]
}
```

randexp

Create random strings that match a given regular expression.
Latest version: 0.5.3
last published: 7 years ago.

www.npmjs.com

Weekly Downloads
3750467

Version
0.5.3

License
MIT

Méta-modéliser en freestyle V2.0

Règles de gestion

KEYWORDS		
	ALL_FIELDS_VALID	Un test avec tous les champs valides.
	ANY_FIELD_INVALID	Un ou plusieurs tests par champ invalide en entrée.
	ANY_FIELD_EMPTY	Un ou plusieurs tests par champ vide en entrée.

Méta-modéliser en freestyle V2.0

Stratégie de génération de tests :

KEYWORDS

ANY_FIELD_INVALID

```
{
      "conditionType": "ANY_FIELD_INVALID",
      "fields": ["lastName", "age", "email", "password"],
      "actionStates": [
        { "alias": "submit", "state": "disabled" },
        { "alias": "reset", "state": "enabled" }
      ]
}
```

Méta-modéliser en freestyle V2.0

Stratégie de génération de tests :

KEYWORDS

ALL_FIELDS_VALID

```
{
      "conditionType": "ALL_FIELDS_VALID",
      "actionStates": [
        { "alias": "submit", "state": "enabled" },
        { "alias": "reset", "state": "enabled" }
      ]
}
```

Méta-modéliser en freestyle V2.0

Stratégie de génération de tests :

JOURNÉE FRANÇAISE DES TESTS LOGICIELS

KEYWORDS

ANY_FIELD_EMPTY

```
{
   "conditionType": "ANY_FIELD_EMPTY",
   "fields": ["lastName","age","email", "password","username"],
   "actionStates": [
      {"alias": "submit", "state": "disabled"},
      {"alias": "reset",  "state": "enabled"} ]
}
```

Démonstration :

JOURNÉE FRANÇAISE DES TESTS LOGICIELS

cypress

Playwright

robotframework

Selenium

https://github.com/ismailktami/metamodelisation-jftl2025

Les symptômes qu'il ne faut pas ignorer

- Vous passez plus de temps à maintenir qu'à créer de nouveaux tests.
- Vos scripts sont dupliqués d'un cas à l'autre, avec peu de réutilisation.
- Vous factorisez dans le code mais pas dans la conception.
- Chaque évolution produit impose de modifier plusieurs scripts.
- Vos tests sont liés au code plutôt qu'aux règles fonctionnelles.

Et demain, si vous voulez vous lancer ?

JOURNÉE FRANÇAISE DES TESTS LOGICIELS

Etape1	Etape 2	Etape 3	Etape4
Identifier les cas récurrents	Modélisez vos cas récurrents	Passez à la méta-modélisation	Générez vos tests dynamiquement
➢ Formulaires, API, pagination,traductions… ➢ Commence par les flux répétitifs, faciles à modéliser et à factoriser.	➢ Comportements, états attendus et règles d'interaction. ➢ un modèle clair, structuré, testable.	➢ Un seul méta-modèle pour couvrir plusieurs cas similaires. ➢ Roder votre méta-model.	➢ Utilisez des données valides/invalides de tests. ➢ Amélioration continue des modèles.

L'industrialisation : pas un luxe, une urgence.

Accompagner le changement :
formation & adoption

JOURNÉE FRANÇAISE DES TESTS LOGICIELS

Un effort initial nécessaire

- Déploiement des bonnes pratiques (Conception / Automatisation)
- S'approprier un nouveau vocabulaire (keywords, modèles, méta-modèles…).
- Changer certaines habitudes d'automatisation.

Roi très clair

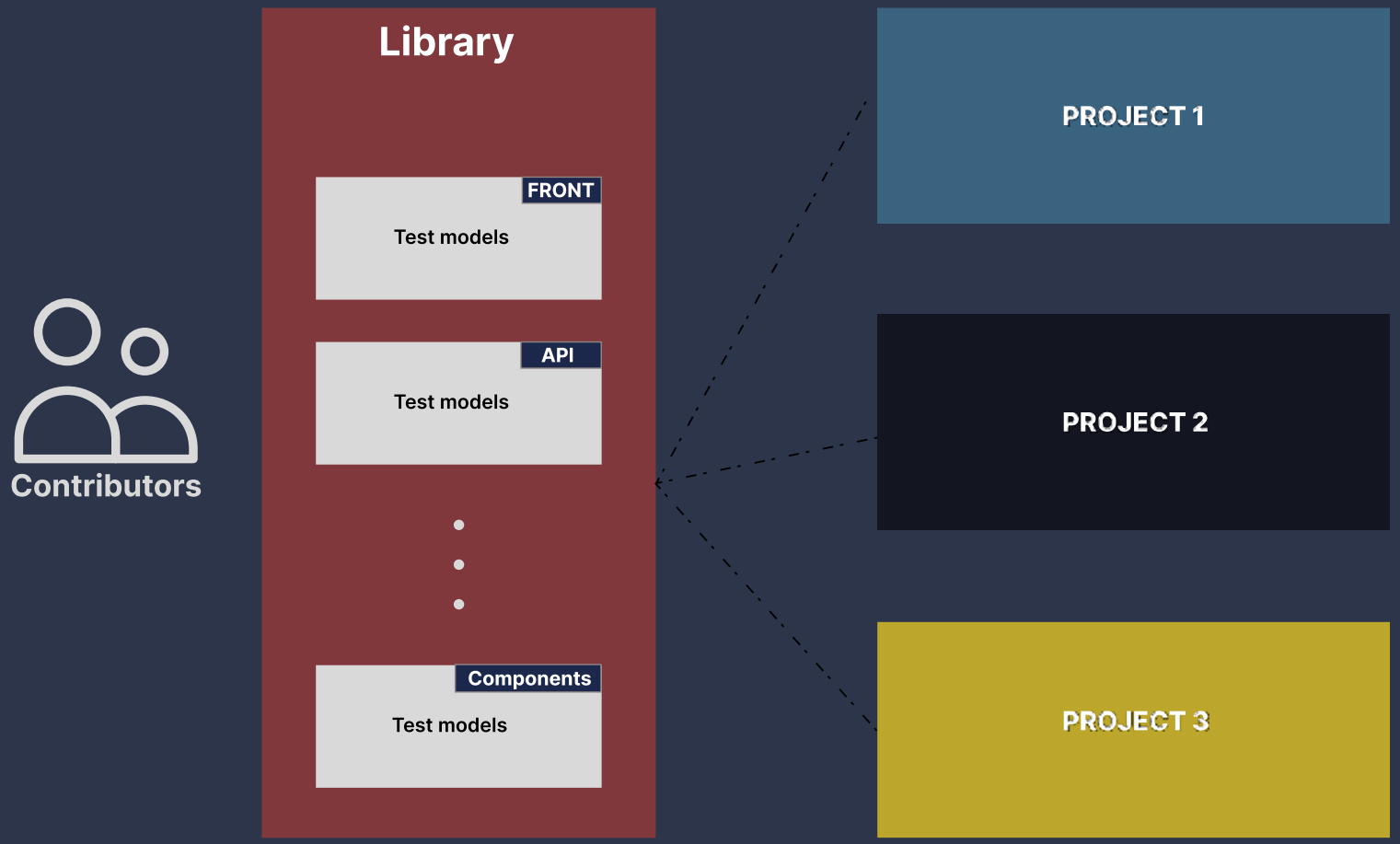
- Moins de duplication, moins de maintenance.
- Des tests plus robustes, générés, plus rapides à créer.
- Industrialiser l'automatisation des flux récurrents
- Une automatisation guidée par la conception et les données.

'' Une approche pilotée par les données, sans changer d'outil, sans formation lourde, simplement en redressant la conception pour rendre les tests plus fiables, plus réutilisables, et bien plus faciles à maintenir. ''

Challenges & Défis

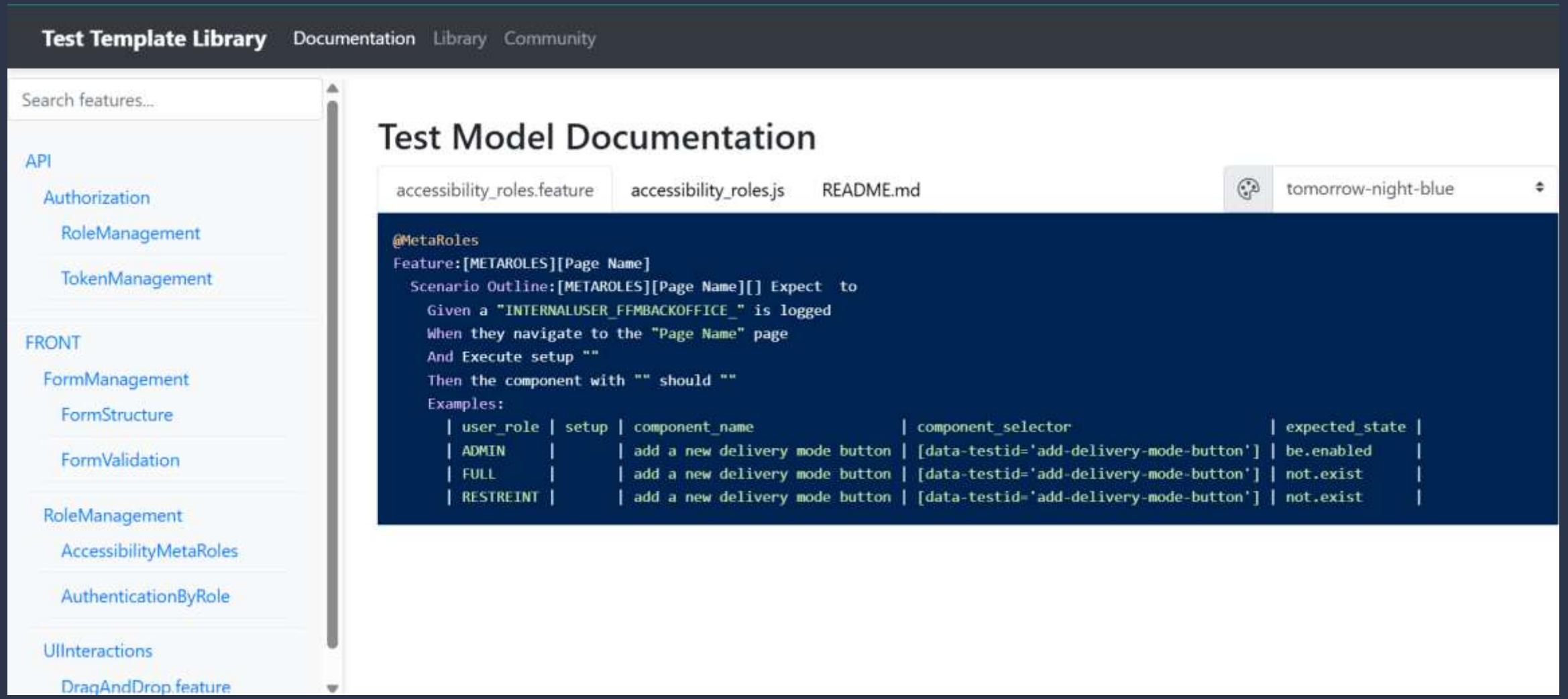
- **Changer de mindset**
- Structurer un socle
- Initialiser votre premier modèle
- Industrialiser sans rigidité
- Attention l'over-engineering

Bibliothèque de tests :



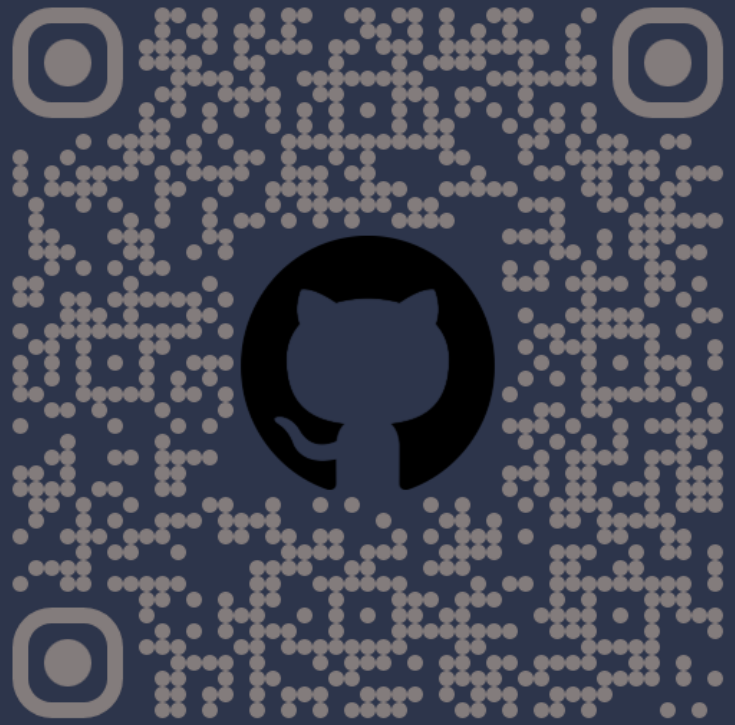
Bibliothèque de tests :

JOURNÉE FRANÇAISE DES TESTS LOGICIELS



Démonstration :

JOURNÉE FRANÇAISE DES TESTS LOGICIELS

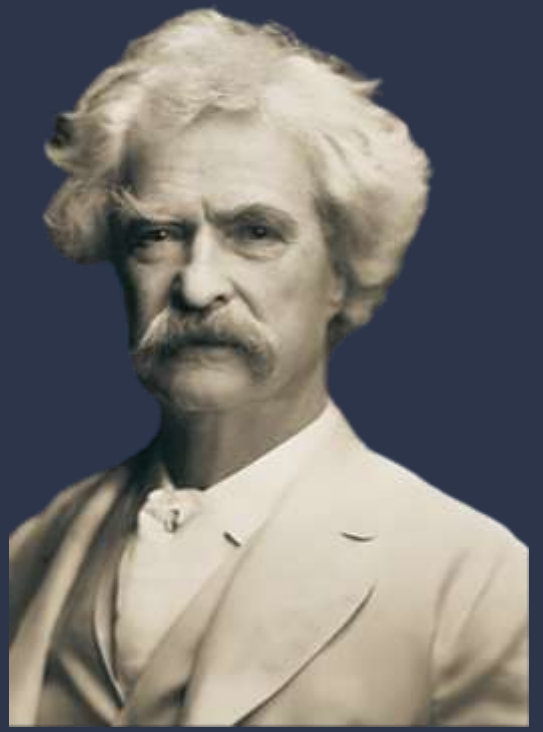


https://github.com/ismailktami/test-templates-library-jftl2025

" L'amélioration continue

vaut mieux que

la perfection retardée"



Mark Twain

Merci de votre écoute !

JOURNÉE FRANÇAISE DES TESTS LOGICIELS

Vote pour la meilleure conférence !

Annexes