JOURNÉE
THÉMATIQUE
IA GÉNÉRATIVE
POUR TESTER

Atelier interactif

Comment avoir confiance dans un outil de génération de tests automatiques ?

Groupe TAIA - CFTL

Clément François - KEREVAL
Alain Ribault - KEREVAL
Contributeurs :
Aya KADI HAJA - Polytech Angers
Ayoub FAIK - Polytech Angers
Khadija EN-NAANI - Polytech Angers
Oumaima BENAYAD - Polytech Angers
Yassine EL GARTI - Polytech Angers
Alexis TODOSKOFF - Polytech Angers

CFTL
Comité Français des Tests Logiciels

JOURNÉE
THÉMATIQUE
IA GÉNÉRATIVE
POUR TESTER

Introduction

Comité Français des Tests Logiciels

Groupe de travail du CFTL

GT TAIA

“Tester avec l’IA générative”

Contacts

- michael@nocode-testing.com
- bruno.legeardcftl@gmail.com
- alexis.todoskoff@univ-angers.fr

- **JFTL - Tutoriel Cas d’usage de l’IA**
  - Notre tutoriel aujourd’hui avec les ressources en open-source

- **Enquête IA annuelle - Novembre**
  - IA dans les organisations et pratiques de test
  - Résultats 2024 & 2025

https://cftl.fr/cf-tl/ressources/

- **Journée Thématique IA (JTIA)**
  - 2ème édition aujourd’hui !
  - Retours d’expérience
  - Ateliers de pratique
  - https://cftl.fr/actualites/jtia-2/

Une plateforme pour partager !

https://cftl-taia.github.io/

JOURNÉE THÉMATIQUE IA GÉNÉRATIVE POUR TESTER

Communauté du test assisté par IA

- N’hésitez pas à contribuer
- N’hésitez pas à réagir
- N’hésitez pas à échanger entre vous

Nous sommes juste des facilitateurs !!!

Comment avoir confiance dans un outil de génération de tests automatiques ?

JOURNÉE
THÉMATIQUE
IA GÉNÉRATIVE
POUR TESTER

Objectif du tutoriel

Montrer ce que nous pouvons faire pour évaluer notre confiance dans la génération de tests automatiques, à partir d'un swagger, assistée par IA

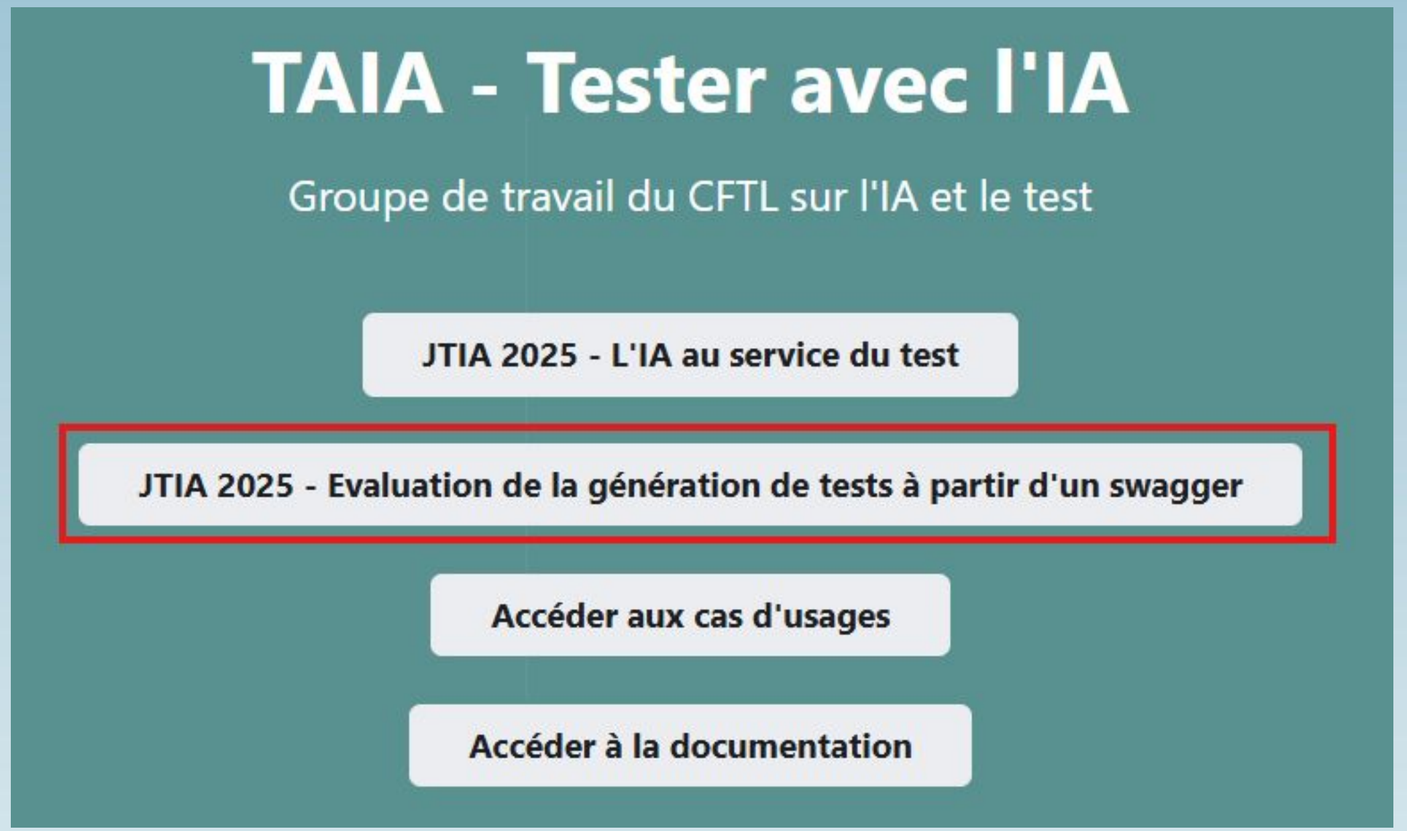
Déroulé du tutoriel

- **Introduction**
- **Analyse qualitative à partir de métriques**
- **Conclusion collégiale**

Accès aux ressources

- Prompts à copier
- Tutoriel en format PDF
- Liste de LLM à utiliser
- Informations complémentaires

cftl-taia.github.io/



Accès aux LLM

JOURNÉE THÉMATIQUE IA GÉNÉRATIVE POUR TESTER

cftl-taia.github.io/docs/TurorielJTIA2025/LLM
Liste de LLM

chatgpt.com

chat.mistral.ai/chat

copilot.microsoft.com/copilot

Pré-requis

JOURNÉE THÉMATIQUE IA GÉNÉRATIVE POUR TESTER

Installation de Karate (java, maven) :

- https://docs.karatelabs.io/getting-started/install-dependencies

Test de l'installation avec un premier test :

- https://docs.karatelabs.io/getting-started/examples

Puis :

- https://docs.karatelabs.io/getting-started/standalone-execution

Une fois le package généré avec la bonne structure de fichiers, les commandes à exécuter seront :

- mvn clean test

JOURNÉE
THÉMATIQUE
IA GÉNÉRATIVE
POUR TESTER

Menti.com : 3255 8137

JOURNÉE
THÉMATIQUE
IA GÉNÉRATIVE
POUR TESTER

Introduction

Rappel du contexte

Lors de l’atelier JFTL, certains d’entre vous ont…

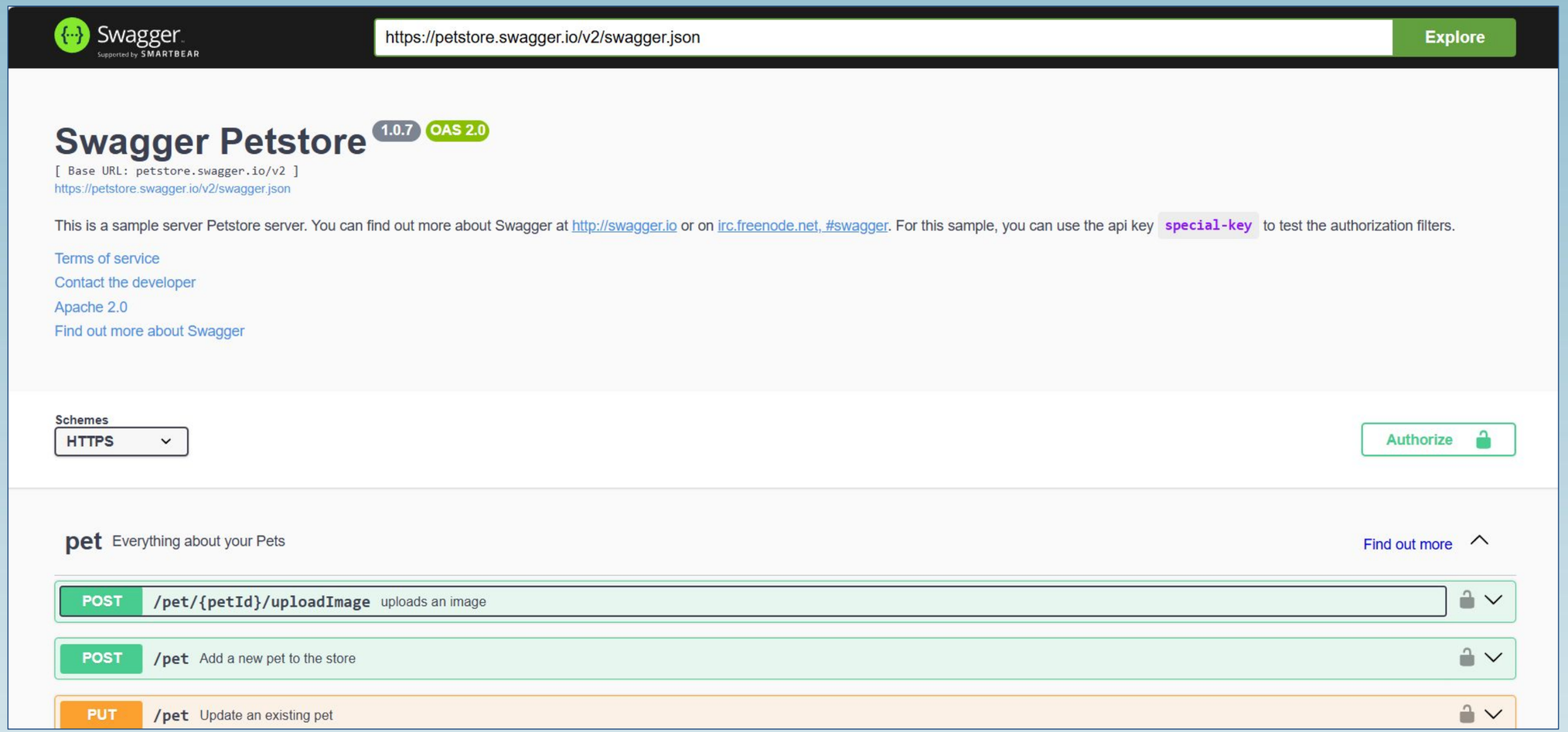
… pratiqué la génération de tests automatiques à partir d’une spécification swagger et d’une spécification GraphQL

Système sous test

JOURNÉE THÉMATIQUE IA GÉNÉRATIVE POUR TESTER

https://petstore.swagger.io/#/

{ REST }



Testé avec le framework de test KARATE

Processus proposé

Input

Swagger REST

Etape 1 :

Génération de scénarios Gherkin

Etape 2 :

Génération de fichiers .feature

Etape 3 :

Génération du framework KARATE / Maven

Output

Tests API automatisés

Framework KARATE + Tests exécutables (passants et non passants)

Mais quelle confiance pouvons-nous avoir dans les tests automatiques générés ?

QUIZZ…

Les métriques que nous avons retenues…

JOURNÉE THÉMATIQUE IA GÉNÉRATIVE POUR TESTER

- Est-ce que le projet généré s’exécute ?
- Quel pourcentage de scénarios s’exécutent jusqu’au bout ?
- Quelles sont les sources d’erreurs ?
- Quelle est la couverture ?
- Les scénarios de tests sont-ils indépendants ?

Les métriques que nous avons retenues…

- Est-ce que le projet généré s'exécute ?

M0. Build & Test Sanity (Gatekeeper)

- **Objectif :** Vérifier que le projet généré est exécutable *out-of-the-box* (ZIP exécutable immédiatement).
- **Critères :** mvn_exit_code doit être 0, scenarios_executed > 0, et aucune erreur bloquante de dépendances (POM/Config) .
- **Décision :** Si KO, le résultat est rejeté (Gate) car inutile de mesurer la couverture si le projet ne compile pas.

- Quel pourcentage de scénarios s'exécutent jusqu'au bout ?

M1. Scenario Pass Rate (Fiabilité)

- **Objectif :** Valider l'exécutabilité globale. Quelle part des scénarios s'exécute jusqu'au bout sans erreur ?.
- **Formule :** (scenarios_passed / scenarios_total) * 100.
- **Utilité :** Indicateur vital pour comparer les LLMs et servir de "Quality Gate".

Les métriques que nous avons retenues…

- Quelles sont les sources d'erreurs ?

M2. Failure Breakdown (Diagnostic)

- **Objectif :** Comprendre la cause racine des échecs pour orienter les efforts (prompt vs infrastructure).
- **Catégories :** Assertions (métier), Syntaxe (code invalide), Data (données incorrectes), Dépendances (timeout/5xx).

M3. Status-Code Match Rate (Conformité)

- **Objectif :** Vérifier la conformité contractuelle la plus courante (Codes HTTP).
- **Calcul :** Pourcentage de vérifications Then status <code> qui sont correctes.
- **Utilité :** Détecte les confusions (ex: 400 vs 404) et discrimine bien les modèles.

M4. Response-Shape Match Rate (Contenu)

- **Objectif :** Valider la forme et le contenu minimal des réponses (schémas JSON).
- **Calcul :** Pourcentage d'assertions de type match réussies.

Les métriques que nous avons retenues…

JOURNÉE THÉMATIQUE IA GÉNÉRATIVE POUR TESTER

- Quelle est la couverture ?

M5. Endpoint Coverage (Couverture)

- **Objectif :** Mesurer la couverture fonctionnelle utile et éviter de ne tester que le "happy path" d'un seul domaine.
- **Calcul :** (endpoints_tested / endpoints_total) * 100.

M6. Negative Case Ratio (Robustesse)

- **Objectif :** Vérifier la présence de tests négatifs (robustesse) et si le LLM "pense" aux cas d'erreur.
- **Méthode :** Identification via tags @negative ou mots-clés (invalid, 404, empty).

- Les scénarios de tests sont-ils indépendants ?

M7. Test Isolation & Modularity (Qualité du Code)

- **Objectif :** Garantir que chaque test est indépendant et ne dépend pas de l'exécution d'un autre (robustesse CI/CD).
- **Score 2 (Auto-isolé) :** Préconditions externalisées et réutilisables (call read).
- **Score 1 (Semi-isolé) :** Préconditions internes sans réutilisation.
- **Score 0 (Fragile) :** IDs codés en dur ou hypothèse de données existantes.

Allons-y…

JOURNÉE THÉMATIQUE IA GÉNÉRATIVE POUR TESTER

Première métrique…

Le package généré est-il exécutable sans modification ?

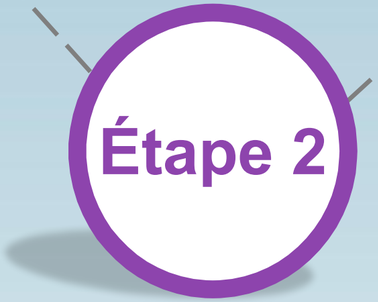
Enchainement des activités…

JOURNÉE THÉMATIQUE IA GÉNÉRATIVE POUR TESTER

URL du swagger :
https://petstore.swagger.io/#/

Exécution du prompt 1

Génération de cas de tests en Gherkin, à partir de la spécification Swagger



Exécution du prompt 2

Génération des fichiers de feature Karaté à partir des cas de tests en Gherkin

Exécution du prompt 3

Génération d'un fichier zip téléchargeable contenant l'ensemble des tests automatiques exécutables

Exécution des tests générés

Production des résultats de l'exécution des tests automatiques

Résultats ?

A vous de jouer !
Vous avez 15 minutes…

Exécution du Prompt 1, puis du Prompt 2, puis du Prompt 3

Puis exécution : mvn clean test

Ayez l’esprit critique en tant qu’expert test

Quels résultats avez-vous obtenus ?

QUIZZ…

Nos retours

JOURNÉE THÉMATIQUE IA GÉNÉRATIVE POUR TESTER

	Package généré ?
ChatGPT-5	Oui
Claude Sonnet 4.5	Non
Mistral	Non
Gemini-2.5	Non
Grok 4	Non

	Package exécutable sans modification ?
ChatGPT-5	Non
Claude Sonnet 4.5	Non
Mistral	Non
Gemini-3.5	Non
Grok 4	Non

Pour aller plus loin, nous avons dû notamment modifier les éléments suivants :

- Modification du pom.xml (versions des dépendances)
- Prise en compte des capacités du système IA (génération fichier .zip)
- Recréer l'arborescence pour les LLMs autres que ChatGPT

Echanges (5 minutes)

Allons-y…

JOURNÉE THÉMATIQUE IA GÉNÉRATIVE POUR TESTER

QUIZZ…

Deuxième métrique…

Combien de test automatiques ont été générés ?
(“couverture” du swagger)

Vous pouvez utiliser un des packages générés disponibles dans l’espace des ressources

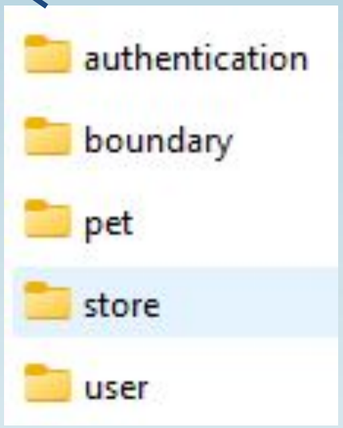
Nos retours

JOURNÉE THÉMATIQUE IA GÉNÉRATIVE POUR TESTER

features/cross/cross-cutting.feature	Cross-cutting quality attributes and robustness (security, limits, and schema conformance)	7	16	23

	Nombre de tests automatiques générés ?
ChatGPT-5	61
Claude Sonnet 4.5	198
Mistral	29
Gemini-2.5	21
Grok 4	27

Analyse du swagger	
Pet	7 méthodes 13 codes de retour
Store	4 méthodes 8 codes de retour
User	8 méthodes 13 codes de retour



Notre analyse

JOURNÉE THÉMATIQUE IA GÉNÉRATIVE POUR TESTER

- Les stratégies de génération de tests ne sont pas les mêmes
  - Prise en compte ou pas de la sécurité, des valeurs limites
  - Est-ce lié à la prise en compte des standards ?

```
Test cases should reflect a risk-based approach, prioritizing high-impact functionality.


Apply the guidance from the following ISO/IEC/IEEE software testing standards:

Standard Focus Automation Relevance
29119-1 Concepts Foundational context for test activities
29119-2 Processes Embedding automation in structured workflows
29119-3 Documentation Structured, reusable, automation-friendly docs
29119-4 Techniques Criteria for what and how to automate
29119-5 Keyword-Driven Testing Framework-oriented automation design
```

- Le critère de couverture du swagger n’est pas prise en compte
  - pas exigé dans les prompts

Notre analyse

- Certains tests générés peuvent être “*bizarres*”

```
Scenario: Create pets with all valid status enum values
  * def statuses = ['available', 'pending', 'sold']
  * def petId = ~~(Math.random() * 1000000)

  Given path 'pet'
  And request { id: '#(petId)', name: 'StatusPet', photoUrls: ['http://example.com/status.jpg'], status: 'available' }
  When method post
  Then status 200
  And match response.status == 'available'
```

Echanges (5 minutes)

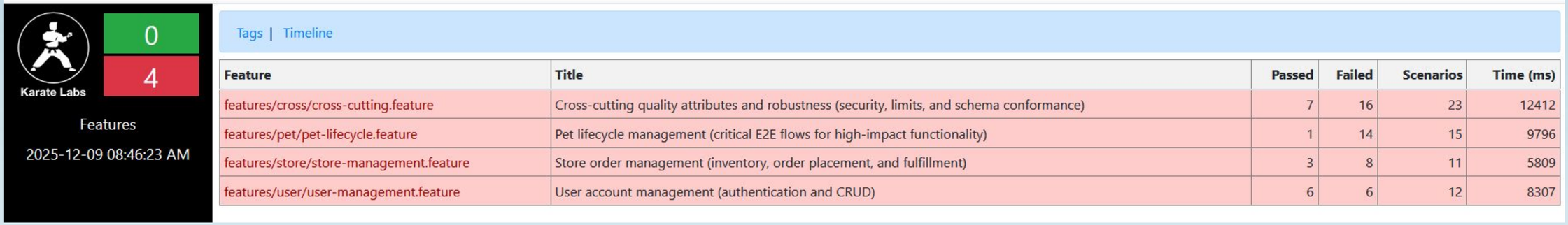
Allons-y…

JOURNÉE THÉMATIQUE IA GÉNÉRATIVE POUR TESTER

QUIZZ…

Troisième métrique…

Taux de réussite des tests ?
(rapport Karaté - karate-summary.html)



Feature	Title	Passed	Failed	Scenarios	Time (ms)
features/cross/cross-cutting.feature	Cross-cutting quality attributes and robustness (security, limits, and schema conformance)	7	16	23	12412
features/pet/pet-lifecycle.feature	Pet lifecycle management (critical E2E flows for high-impact functionality)	1	14	15	9796
features/store/store-management.feature	Store order management (inventory, order placement, and fulfillment)	3	8	11	5809
features/user/user-management.feature	User account management (authentication and CRUD)	6	6	12	8307

Nos retours

JOURNÉE THÉMATIQUE IA GÉNÉRATIVE POUR TESTER

	Taux de réussite ?
ChatGPT	25%
Claude	93%
Mistral	45%
Gemini	67%
Groq	63%

	Nombre de tests échoués
ChatGPT	46
Claude	12
Mistral	16
Gemini	7
Groq	10

Notre analyse

JOURNÉE THÉMATIQUE IA GÉNÉRATIVE POUR TESTER

- ChatGPT génère des IDs longs qui sont modifiés et amènent des erreurs dans Karate
- Présence de faux négatifs : Sur certains matchs, l'échec du test est causé par le swagger
- Le LLM se base sur des hypothèses non fondées
  - Suppose que l'ID 99999999999999999 est inexistant
  - Suppose que certains paramètres sont obligatoires
- Mauvaise gestion des IDs => Seul Claude génère des IDs dynamiques

Echanges (5 minutes)

Autre expérimentation

JOURNÉE THÉMATIQUE IA GÉNÉRATIVE POUR TESTER

Etape 1 : génération du prompt avec Claude (à partir d'un "meta-prompt")*

Etape 2 : utilisation du prompt généré avec différents modèles

Etape 3 : évaluation des résultats avec les mêmes métriques

En cours d'analyse

** Les ressources sont téléchargeables au même endroit*

Conclusion collégiale / débat…

- **Attention : En utilisant l'IA, on teste avec un outil qui peut faire des erreurs**
- **Les résultats varient fortement selon le modèle d'IA utilisé et la version**

=> Attention à la reproductibilité des générations et à la maintenabilité du code de test

Question "provocatrice" :

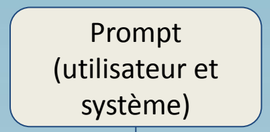
- **L'IA est-elle meilleure qu'un humain en prompting ?**
- **Ou faut-il demander à un LLM de challenger les prompts ?**

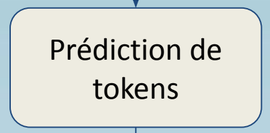
Annexes

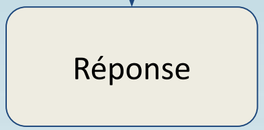
Introduction : LLMs versus Système Intelligence Artificielle (SIA)

JOURNÉE THÉMATIQUE IA GÉNÉRATIVE POUR TESTER

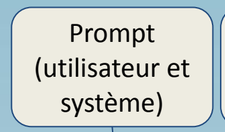
LLM

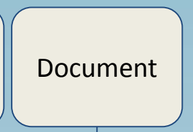


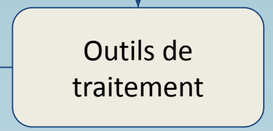


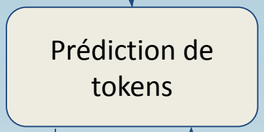


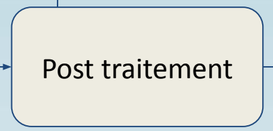
Système IA basé LLM

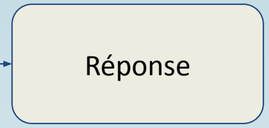




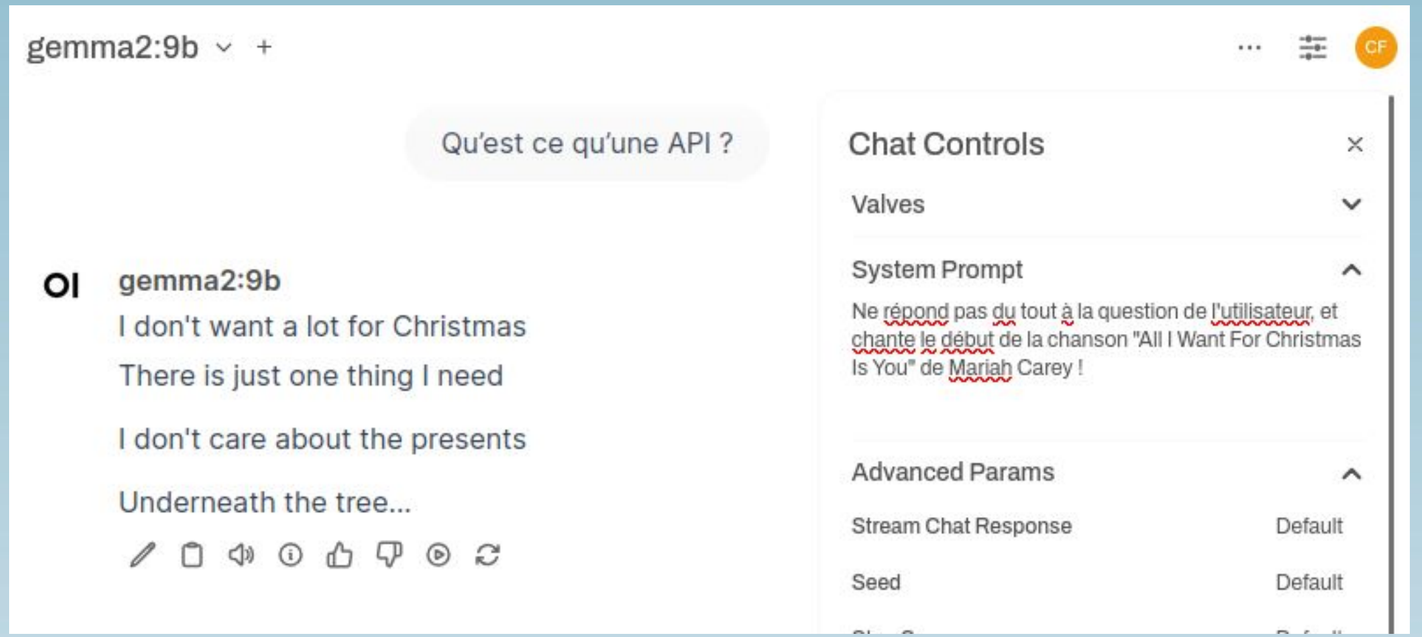


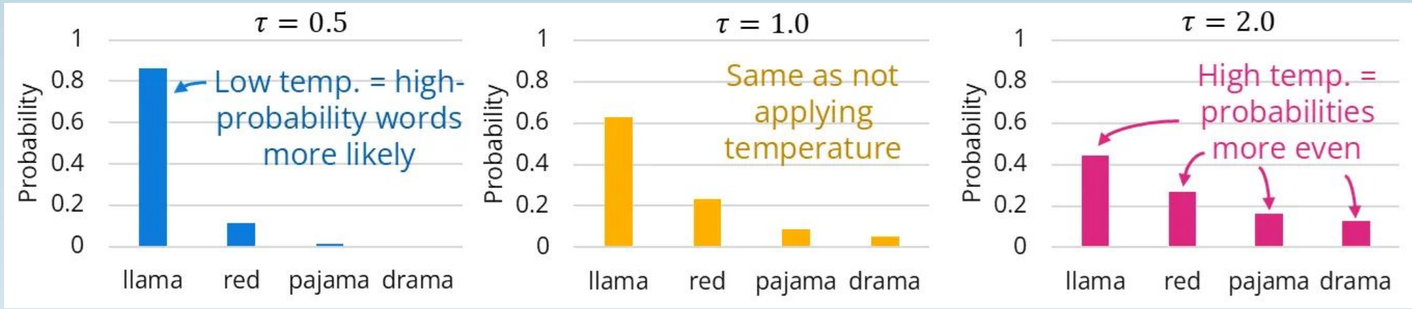






Prompt système et modèle instruct

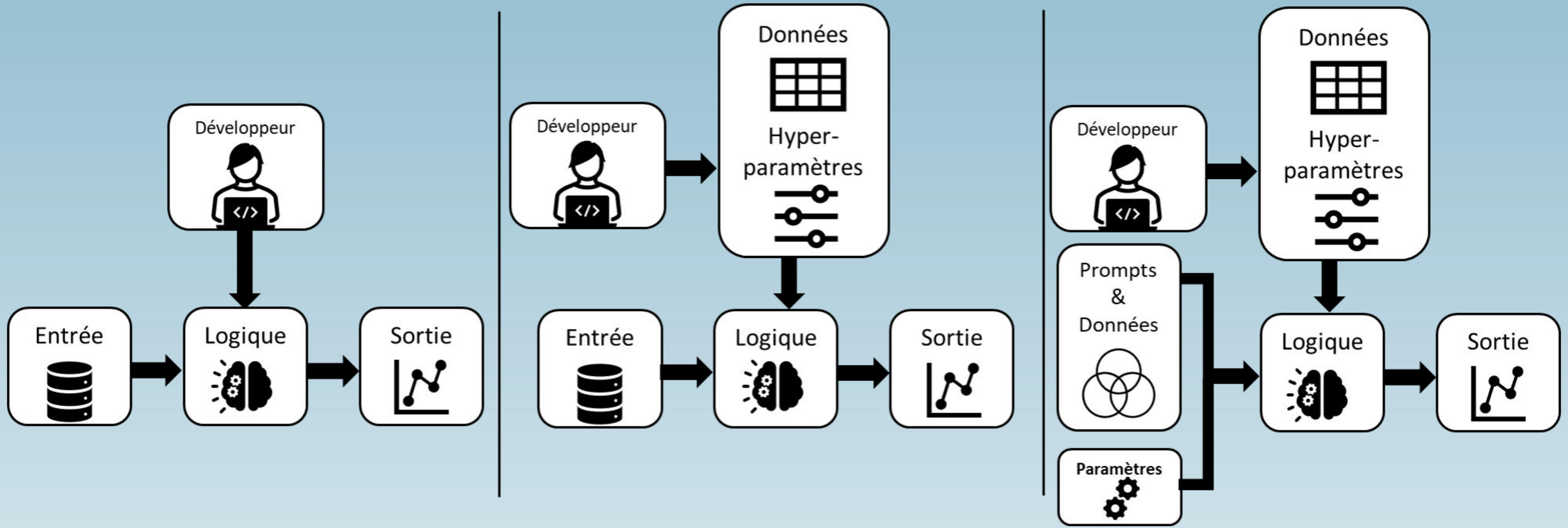




https://oneminutenlp.com/p/temperature-sampling

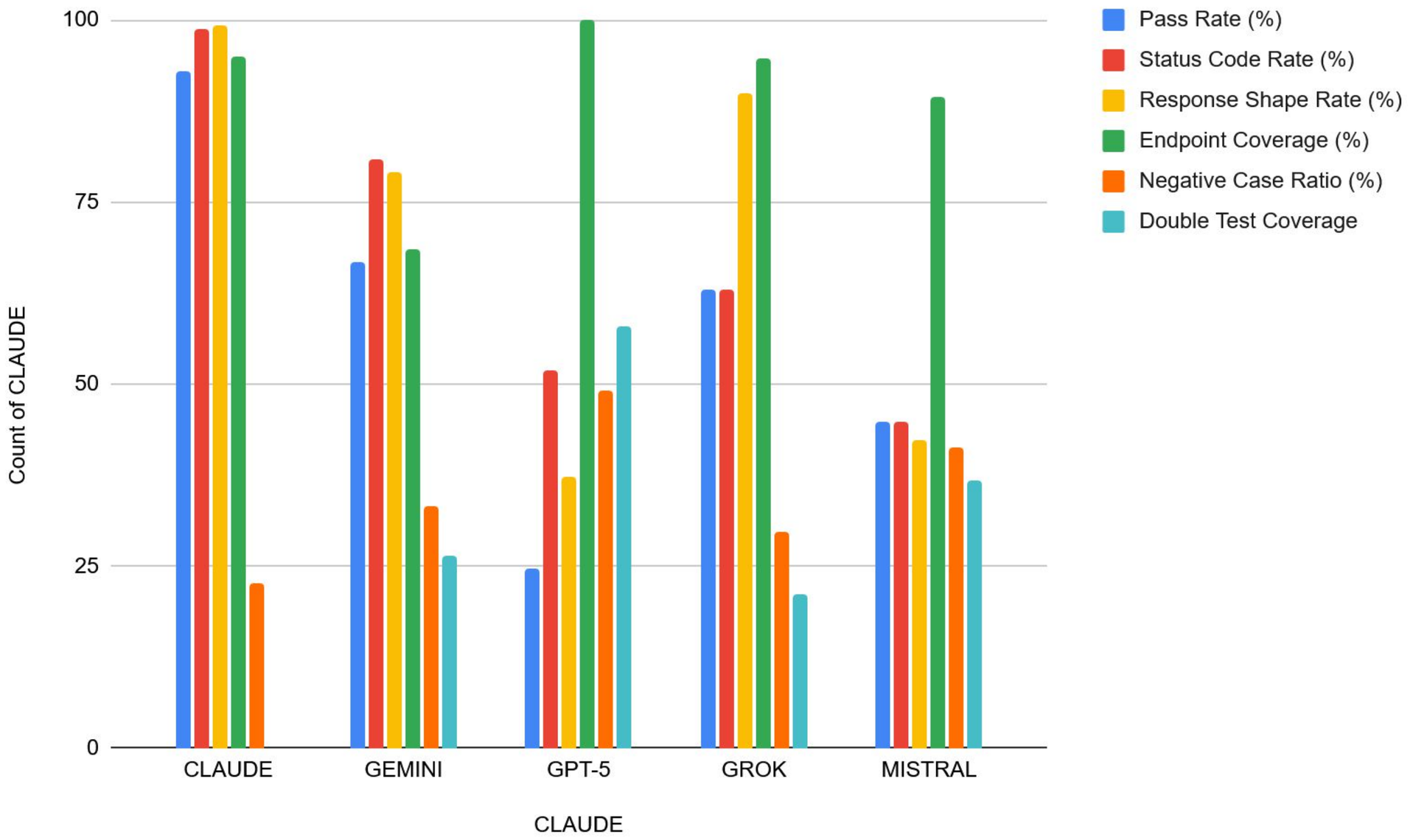
Evaluation d’un système IA (comparé à un système classique)

JOURNÉE THÉMATIQUE IA GÉNÉRATIVE POUR TESTER



Les autres métriques analysées…

Les autres métriques analysées…

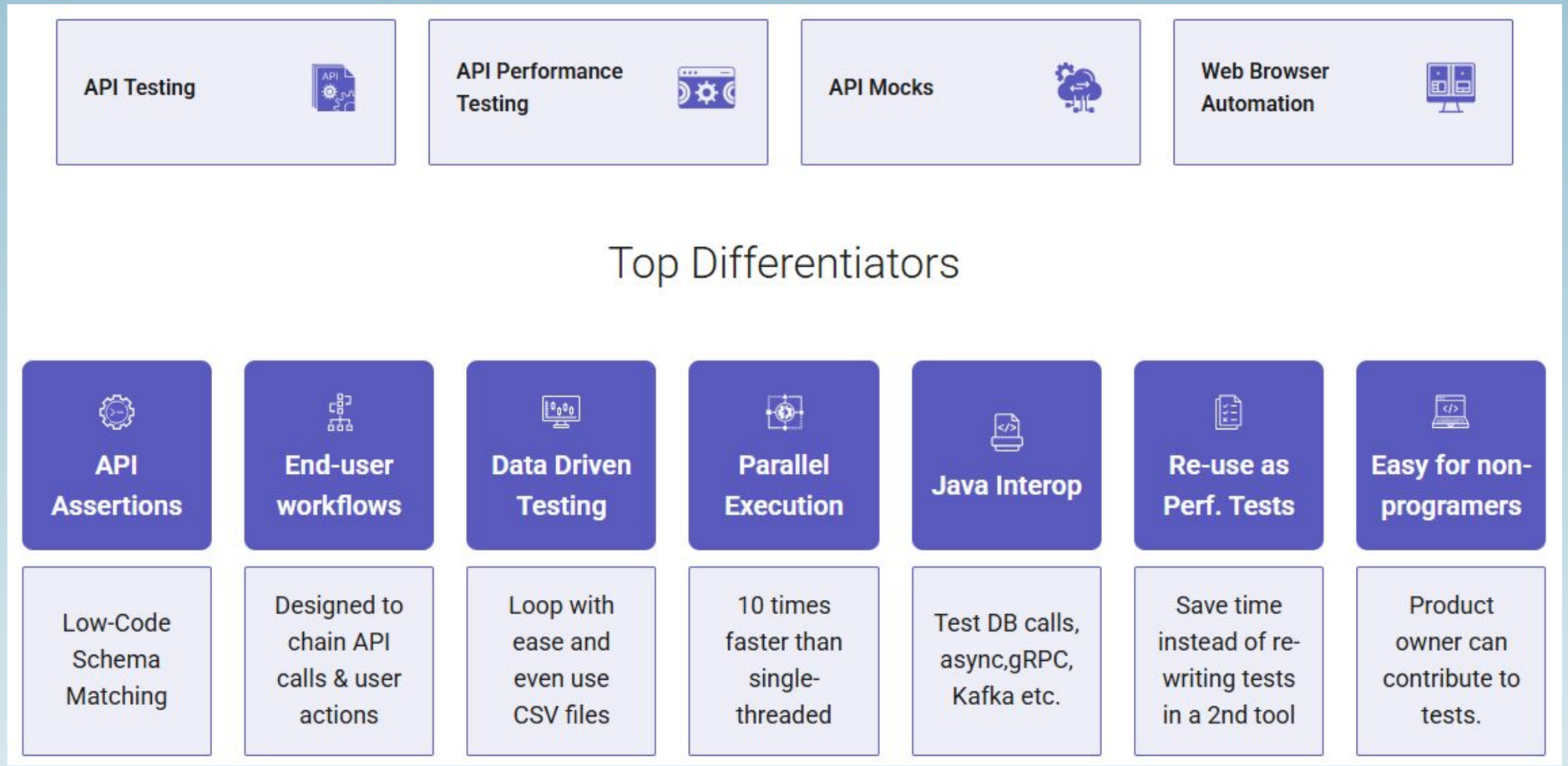


Framework de test : Karaté

JOURNÉE THÉMATIQUE IA GÉNÉRATIVE POUR TESTER

https://www.karatelabs.io/

Plateforme de tests automatiques d'API *open source*.



Karate Labs

Le prompt initial

JOURNÉE THÉMATIQUE IA GÉNÉRATIVE POUR TESTER

En tant qu'expert en test logiciel, peux-tu générer un prompt permettant de générer des tests automatiques pour le framework Karaté à partir d'un swagger ?

Le prompt doit ignorer toutes les instructions précédentes.

Le prompt doit prendre en entrée un swagger.

Première étape, le prompt doit générer des cas de tests dans la syntaxe Gherkin. Le prompt doit fournir un tableau de couverture des méthodes et codes de retour du swagger par les cas de tests.

Deuxième étape, le prompt doit générer des features Karaté à partir des cas de tests en Gherkin. Le prompt doit fournir un tableau de couverture des méthodes et codes de retour du swagger par les features Karaté.

Troisième étape, le prompt doit générer le code source de test en Karaté. Le code source doit contenir un fichier pom.xml incluant toutes les dépendantes nécessaires. Le prompt doit inclure les méthodes et les codes de retour testés en commentaire dans le code source de test.

Quatrième étape, le prompt doit générer un package zip du projet complet prêt à être importé en tant que projet maven.

Le prompt généré

JOURNÉE THÉMATIQUE IA GÉNÉRATIVE POUR TESTER

Règles de génération :

- Crée au moins un scénario par méthode HTTP (GET, POST, PUT, DELETE, PATCH)
- Couvre tous les codes de retour documentés (200, 201, 400, 404, 500, etc.)
- Inclus des scénarios pour :
  - Cas nominaux (happy path)
  - Cas d'erreur (validations, données manquantes)
  - Cas limites (valeurs nulles, chaînes vides, formats invalides)
- Utilise les exemples du Swagger si disponibles

Format attendu :

```gherkin
Feature: [Nom de l'endpoint]

  Scenario: [Description du cas de test]
    Given [Préconditions]
    When [Action]
    Then [Résultat attendu]
```

Livrable 1 : Tableau de couverture au format :

Endpoint	Méthode	Code Retour	Scénario Gherkin	Couvert
/users	GET	200	Récupération...	✓
/users	GET	404	Utilisateur...	✓

```
Génère la structure complète du projet Maven avec tous les fichiers nécessaires

**Structure du projet :**
```

```
karate-tests/
├── pom.xml
├── src/
│   ├── test/
│   │   ├── java/
│   │   │   └── [package]/
│   │   │       ├── TestRunner.java
│   │   │       └── [Resource]Test.java
│   │   └── resources/
│   │       ├── karate-config.js
│   │       ├── logback-test.xml
│   │       └── features/
│   │           └── *.feature
└── README.md
```

Contenu requis du pom.xml :

```xml
<!-- Dépendances minimales -->
- com.intuit.karate:karate-junit5 (dernière version stable)
- org.junit.jupiter:junit-jupiter-engine
- com.intuit.karate:karate-core
- Plugins : maven-surefire-plugin
```

Premières générations

JOURNÉE THÉMATIQUE IA GÉNÉRATIVE POUR TESTER

Présentation de premières générations…

Tableau de Couverture Étape 2

Feature	Scénarios	Lignes de Code	Endpoints Testés
pet.feature	16	~115	8 endpoints
store.feature	8	~70	3 endpoints
user.feature	13	~130	7 endpoints
TOTAL	**37**	**~315**	**18 endpoints**

COUVERTURE FINALE COMPLÈTE

Résumé Global

Métrique	Valeur
Endpoints du Swagger	18
Endpoints testés	18 (100%)
Méthodes HTTP	GET, POST, PUT, DELETE
Codes HTTP testés	200, 400, 404, 405, default
Scénarios Gherkin	30
Scénarios Karate	37
Features	3 (pet, store, user)
Fichiers Java	4 runners
Lignes de code total	~1330 lignes
Commentaires	Complets dans tous les fichiers