

Article

[Irène Mykhailova](#) · Jan 12, 2023 3m de lecture

[Open Exchange](#)

## Tests unitaires pour les transformations de données

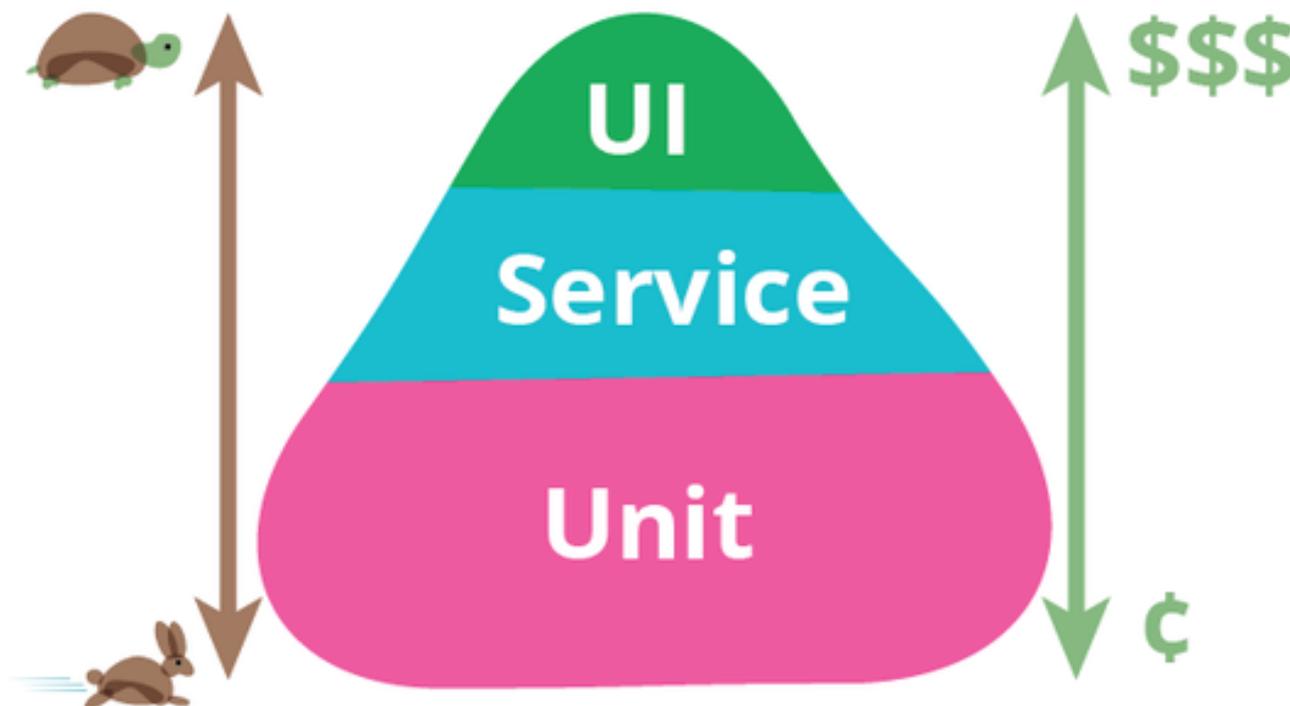
Voulez-vous être sûr que vos transformations de données fonctionnent comme prévu avec une seule commande ? Et que diriez-vous d'écrire des tests unitaires pour vos transformations de données de manière simple et rapide ?

Les transformations de données sont généralement très présentes lorsqu'on parle d'interopérabilité. Ces transformations de données sont utilisées pour convertir les données entre différents systèmes ou applications dans votre code, et elles ont donc un rôle très important à jouer.

### Les stratégies de test

Après avoir examiné le concept de la [Pyramide de tests] (<https://martinfowler.com/bliki/TestPyramid.html>) et certains [articles] (<https://medium.com/@timothy.cochran/test-pyramid-the-key-to-good-automat...>) à ce sujet, on peut se faire une idée rapide du fait qu'il est préférable de disposer d'une base solide de tests automatisés de bas niveau moins coûteux que de tester uniquement l'interface utilisateur.

Dans un contexte d'interopérabilité, j'ai constaté dans plusieurs projets qu'il vaut vraiment la peine d'investir un peu d'effort dans l'écriture de tests unitaires de transformation de données, surtout lorsque nous traitons différents scénarios (par exemple HL7, messages personnalisés, etc.). Cela nous permettra d'être sûrs que notre logique de transformation de données fonctionne comme prévu après l'introduction de nouveaux changements. Même après la résolution d'un problème avec une transformation de données, nous pouvons facilement créer un nouveau test avec le message qui a causé le problème, de sorte que nous sommes sûrs de ne pas obtenir la même erreur à l'avenir.



Un petit assistant qui utilise le cadre %UnitTest

Pour vous aider à écrire et à exécuter des tests de transformation de données, je partage un exemple qui utilise le cadre [IRIS %UnitTest framework] (<https://docs.intersystems.com/irislatest/csp/docbook/DocBook.UI.Page.cls...>). L'objectif est de vous permettre de définir des ensembles différents de messages d'entrée et de messages de sortie correspondants prévus pour chaque transformation de données à tester, en tant que fichiers texte externes.

Supposons que vous vouliez créer des tests unitaires pour une transformation de données appelée `Sample.Health.a31ToPatient`, alors vous devez procéder comme suit :

1. Créez une nouvelle classe étendue `DataTransformTestCase`. Définissez le paramètre `<TestDirectory>` comme le répertoire dans lequel vous allez stocker vos tests de transformation de données.
2. Dans votre `<TestDirectory>`, créez un sous-répertoire appelé `Sample.Health.a31ToPatient`. Ce sous-répertoire stockera les ensembles d'entrées et de sorties attendues que vous souhaitez tester dans la transformation de données.
3. Dans le sous-répertoire `Sample.Health.a31ToPatient` is faut ajouter les messages d'entrée et la sortie attendue comme `.in.txt` and `.out.txt`.
4. Lancez vos tests ! Vous pouvez voir les résultats dans le portail %UnitTest Framework dans IRIS. Si vous obtenez une erreur, le fichier `*.gen.txt` sera généré, afin que vous puissiez comparer la sortie réelle à la sortie attendue et voir ce qui ne va pas.

Lancez l'exemple vous-même !

Vous pouvez télécharger, lancer l'exemple et lire plus de détails dans [Open Exchange](#).

[#DTL](#) [#Interopérabilité](#) [#InterSystems IRIS](#) [#InterSystems IRIS for Health](#)  
[Voir l'application sur InterSystems Open Exchange](#)

---

URL de la source: <https://fr.community.intersystems.com/post/tests-unitaires-pour-les-transformations-de-donn%C3%A9es>