

ソースコードと GUI で編集可能なシナリオテスト作成ツールの開発

— GUI ベースでシナリオテストを作成する —

1. 背景

ソフトウェアの品質保証は昨今のプロダクト開発における重要な課題であり、致命的なバグはビジネスリスクや信用の損失につながる。そこで私が着目しているのがシナリオテストである。シナリオテストは、実際にユーザーがサービスを利用する流れに沿ってシステムを操作し、期待する挙動をするかを確認するテストである。バックエンドの API サーバに着目すると、複数のエンドポイントを跨いでリクエスト送り、期待するレスポンスが得られるかを確認する。

このようなテストでは、テストシナリオをどのように管理するのが重要となる。しかし、プロダクトによっては 100 を超えるテストシナリオが存在し、これらを俯瞰的に把握することは困難になっている。この状態では、テストケースの抜け・漏れがないか確認するだけでも大変な作業である。

このような状況下で、複数のテストシナリオを効率的かつ簡単に閲覧および編集できるソリューションが求められている。

上記に加え、多くの場合テストファイルはエンジニアや QA 担当者のみによって管理されている。しかしプロダクトマネージャーやデザイナー、カスタマーサポートなどより多くの人々がテストケースを把握することで、品質向上に寄与できると考えている。

2. 目的

本プロジェクトの目的は、GUI で直感的にシナリオを閲覧・編集できるシナリオテスト作成ツールを構築することである。YAML 形式と双方向同期を可能にすることで、ソースコードと GUI エディタの両方で操作可能であり、また、グラフ構造を採用し、複数のシナリオを単一画面で把握できる仕組みを実現する。

3. 開発の内容

開発したアプリケーションのスクリーンショットを図 1 に示す。中央に表示されているのが GUI エディタである。ユーザーはここで自由にブロックを配置し、ブロック同士を連結させることでシナリオを作成することができる。

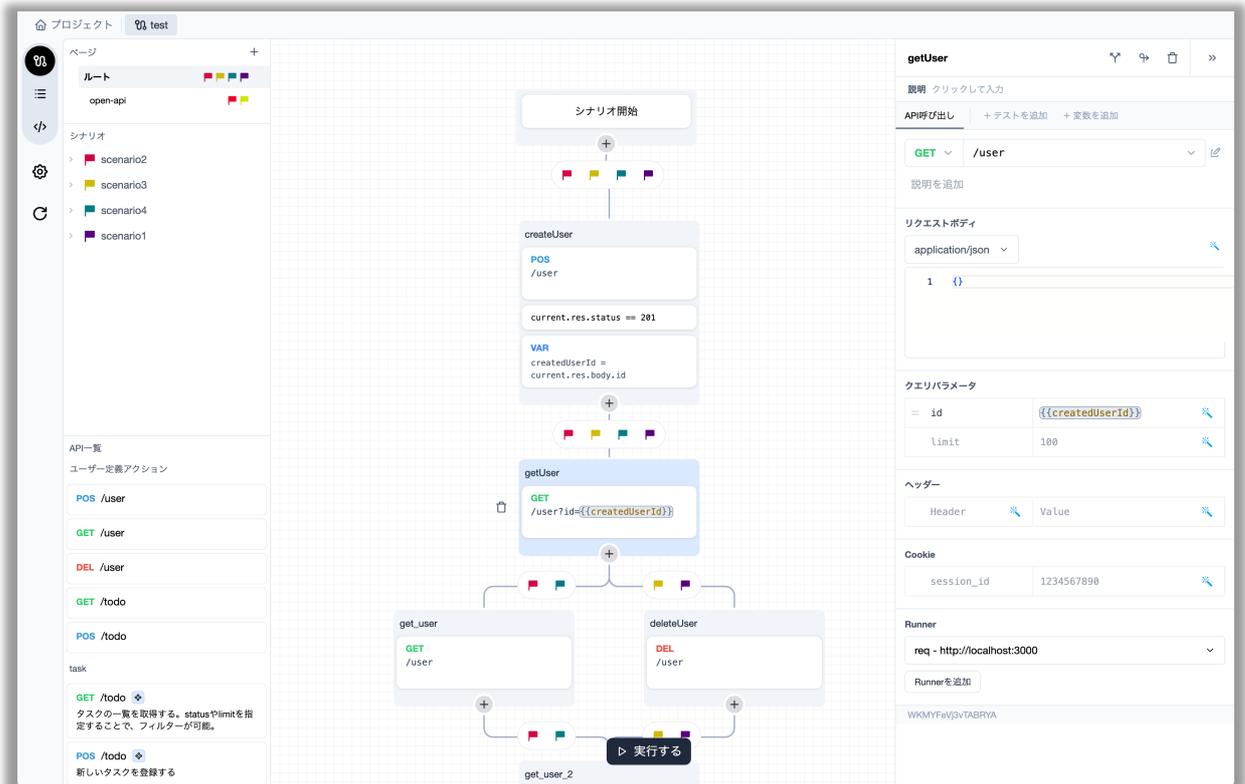


図 1：テストシナリオの編集画面

各ブロックは API 呼び出しやデータベースへのクエリなどのアクションを表す。また、レスポンスに対してテストを追加したり、後続のブロックで利用するための変数を定義できたりする機能も実装している。

作成したテストシナリオは、自動でローカルの YAML ファイルに同期される。この YAML ファイルを git 管理含めてリモート環境にプッシュすることで、CI 上でレビューをしたり、テストを実行したりすることも可能である。

本ツールでは、単にシナリオを GUI で作成できるだけでなく、グラフ構造を利用した定義が可能である。例えば図 2 を見ると、1つのグラフ構造で3つのシナリオが定義されていることが分かる。一般的なアプリケーションにおいてテストシナリオは枝分かれが多いため、このような表現が適していると考えている。

本ツールは、既存の OSS テストツールである runn をサポートしている。本ツールが出力する YAML ファイルは、そのまま runn が解釈可能なフォーマットである。runn と互換性のあるフォーマットをサポートすることで、runn が持つエコシステムを利用できることや、既に runn を利用しているプロジェクトに対しての導入ハードルを下げることができるというメリットがある。今後、runn 以外のテストツールに関しても対応していきたい。

なお、ツール上で「実行する」のボタンを押すことで、runn を直接実行することなくテストを実行できる機能を備えている。これは、ツールが runn を代わりに呼び出し、その結果をパースして UI 上に反映させることで実現している (図 3)。

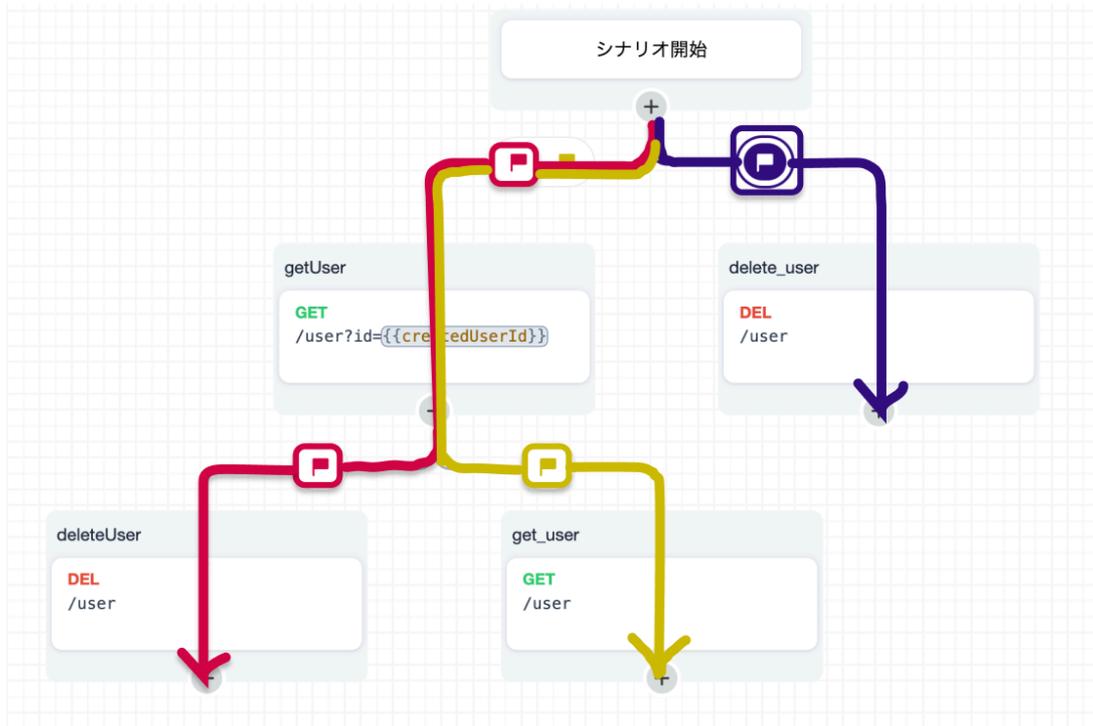


図 2：単一のグラフで複数のシナリオが表現されている

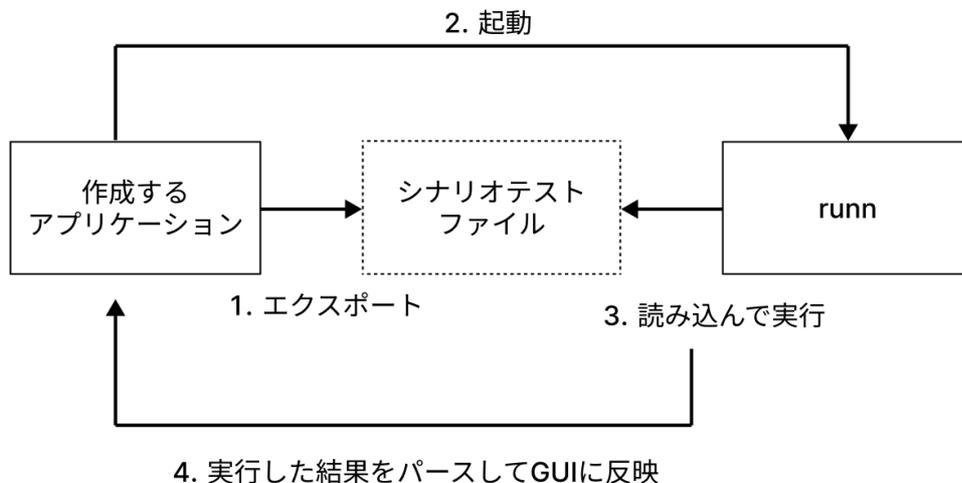


図 3：テスト実行のアーキテクチャ

4. 従来の技術（または機能）との相違

現在、シナリオテストを記述する方法は、1. プログラミング言語を利用して記述する、2. YAML 形式で記述する、3. GUI ツールを利用する、の主に 3 通りの方法に分類される。

しかし、複数のシナリオを単一の画面で表示するグラフ構造を用いたアプローチは多くはない。GUI ツールを利用する場合は同等のことが実現可能ではあるものの、YAML ファイルへの書き出しが難しく、CI 環境での実行には向いていない場合が多い。

本ツールは、GUIでの操作と、既存のライブラリが読み込み可能な形式のYAMLを出力することによって、これらの問題を解決している。

5. 期待される効果

シナリオテストを実施している開発現場において、テストシナリオ管理の効率の向上に寄与できると考えている。特にテストケースに抜け・漏れがないか、仕様と相違がないかなどの確認が容易になると考えられる。また新規参画メンバーにとっても、テストの全体を把握する上で有用なツールになる。

また、GUIでシナリオを構築できるため、コーディングに抵抗があるQA担当者でも積極的にテストケースの策定に関わることができるようになるかもしれない。

6. 普及の見通し

まずは、細かいバグ等を潰した上でリリースを目指す。その後、ユーザーフィードバックやインタビュー等を通じて、より使いやすいツールへと改善を続ける予定である。

7. クリエータ名（所属）

池奥 裕太（エムスリー株式会社）