

デザインの継続的インテグレーション支援ソフトウェア

— 圧倒的に効率の良いWebデザイン検証サービス —

1. 背景

多くのWebサービスにはユーザーが操作するためのUIが備わっている。それらは通常HTMLで文書構造を、スタイルシートで見た目を定義しているが、一般的なプログラミング言語とは異なりそれらに対してテストの実行を行うことはほとんどない。それはUIの検証のためには開発者による目視確認が必要不可欠であることに起因する。1行1行に明確な意味を定義できるプログラミング言語とは異なり、UIの定義に用いられるスタイルシートでは明確な入力と出力を定義することができない。そのため、ユニットテストなどの概念をそのまま適用することができず、開発者自身によってブラウザ上、もしくはスクリーンショットとして生成された画像を目視確認するという極めて効率の悪い作業が常態化していた。

2. 目的

これまでテストの実施による恩恵を受けられなかったWebUIの分野に対して、効率の良いCI手法の開発と運用が行えるサービスの開発を目指した。小さく見つけづらいデザインのバグであっても迅速に見つけ出し、見た目におけるリグレッションテストを実現することを目的としている。

3. 開発の内容

本プロジェクトでは、Webサービスの開発者がこれまで手をつけづらかったUIの検証を支援するためのCIツール「Eyecatch」を開発した。

Eyecatchは一般的なCIサービスと同様に、Githubなどのリポジトリ管理サービスを開発者が操作した際に発生するフックがトリガーとなり起動する。Eyecatchのバックエンドにて開発者がもたらしたコードの変更による見た目の変化を、全てのページを対象に検出し、その結果を提示する。全てのビルドは隔離された環境下で実行されるため、他のビルドの影響を受けることなく検証が行われる。これにより、例えばデータベースアプリケーションのインストールが求められるWordpressのようなWebサイトであっても、一般的な静的サイトと同じように見た目の検証が行われるほか、事前にコンパイルが必要になる言語であってもEyecatchを使用することができる。大規模なWebサービスであれば、異なるページが300や400存在することは決して珍しいものではない。Eyecatchは変化したページのみをより抜いて表示するため、圧倒的に効率の良いUI検証を提供する。

また、ユーザーがブラウザ上で行える動作の多くをEyecatchに代行させることができる。具体的には、テキストフォームへの文字入力や、ボタンやリンクのクリックなどを定義できるため、ユーザー認証が必要となるようなページであっても検証を行える。

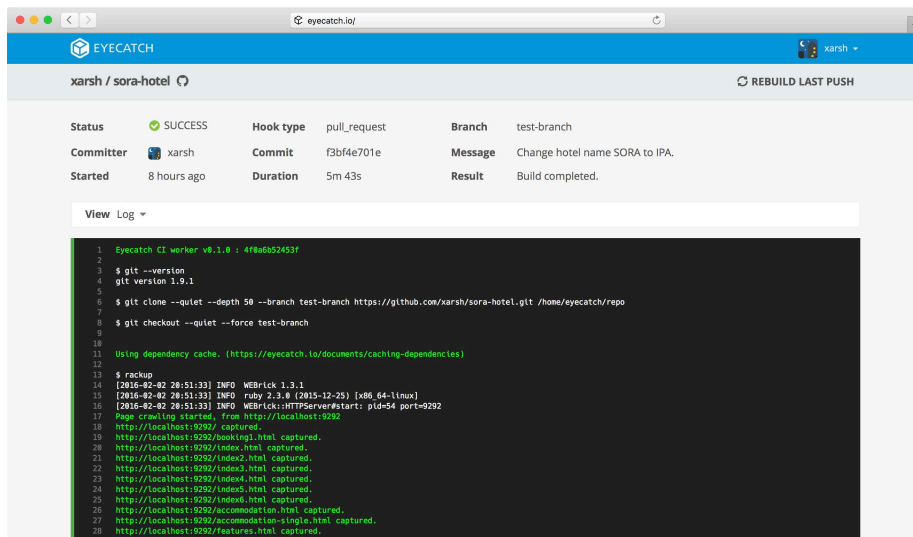


図1 ビルド結果と動作ログ

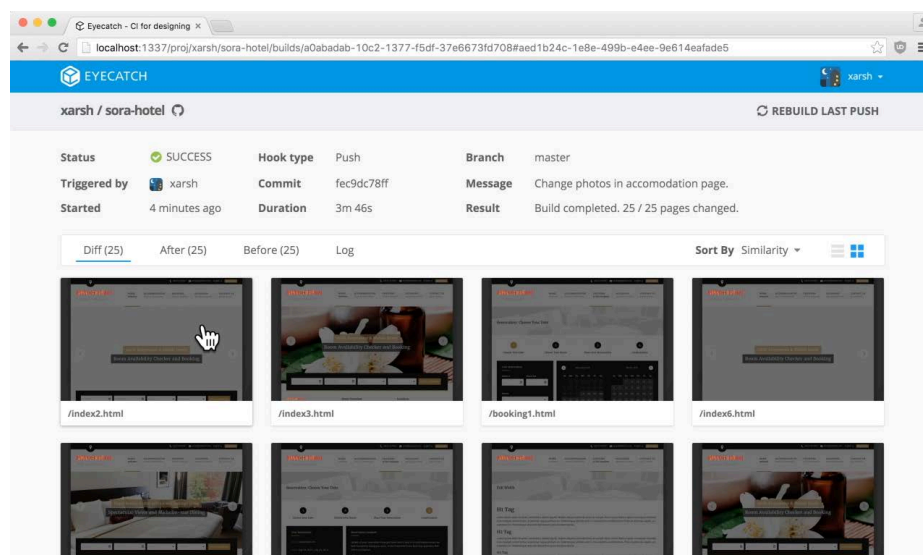


図2 解析結果の一覧

開発者による1箇所の変更が複数のページにまたがって存在する場合、具体的にはヘッダーやフッターといった共通要素に対する変更がもたらされた場合、Eyecatchはその全てを検出してしまう。動作としては正常なものではあるが、もし1000ページに渡って同様の変化が生じた場合、1000ページ分の目視確認が必要になってしまうことは好ましくない。そのため、Eyecatchでは画像解析により変化の内容自体を確認し、同様の変化だと推測されたものに関してはそれらを一つにまとめあげる機能を備えている。これにより、たとえばヘッダーに含まれるロゴマークを変更した場合であっても一度に複数ページを確認できるため、非常に効率の良い検証作業が行える。

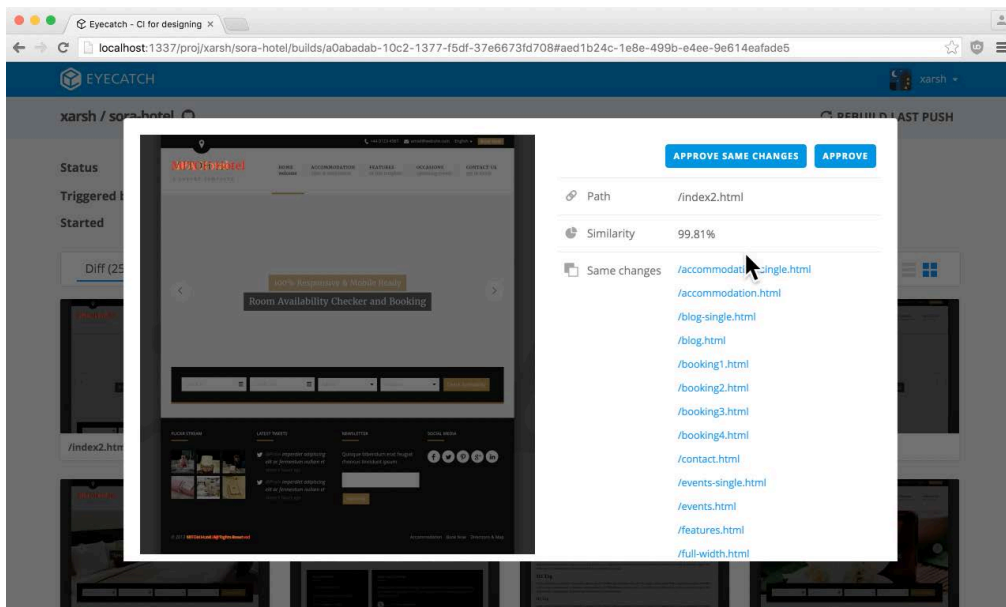


図3 同一変化のまとめ

4. 従来の技術（または機能）との相違

既存のCIツールとして広く用いられているJenkinsやCircleCI、TravisCIなどは、単純に開発者が事前に作成したテストコードをクラウド上で実行するというだけに他ならない。したがって、そもそもテストコードを記述することが困難であるUIの検証には活かすことが出来なかった。UI検証のために開発されたCIツールの一つとしてSauceLabsやBrowserstackなどが挙げられるが、これらも同様の問題を抱える。さらにCIツールの1サイクルでページの数だけスクリーンショットが生成され、見た目の変更の有無にかかわらず開発者が確認しなければならなかった。

Eyecatchでは自動で全てのページを確認し、変化が生じているかどうかをEyecatch自身が判断してくれるため、不要な確認作業を省略することができる。

また、これまで検証に手間が掛かっていたインタラクションを求めるサイト、例えばモーダルウィンドウの表示やユーザー認証後に表示されるページなどであっても検証できるように設計してあるため、動的にコンテンツが生成されるような複雑なサイトなども含め、非常に幅広い開発現場で用いることができる。

CIツールではなく、単純にWebUIの差分を見つけるためのサービスやライブラリも存在する。InstadiffやWraithなどが挙げられるが、これらは前述した問題点（大量の画像の目視確認、ユーザー認証など）を解決している訳ではなく、これらをそのまま開発のフローに導入することはできない。

5. 期待される効果

UIのテスト作業は、プログラムのテスト作業に比べて遅れていると評価されてきている。Eyecatchはこれまでのテスト作業に一般的に求められたテストコードの記述という作業をなくし、開発者による時系列的な変化の流れを元に検証する。そのためEyecatchは、その導入へのハードルは非常に低くなっており、エンジニアだけではなくプログラミングの経験のないデザイナーであっても使いこなせるサービスを実現した。

また、Eyecatchの仕組みはWebサイトの検証だけにとどまらず、他の分野、例えばモバイルアプリケーションのUI検証にも活かすような応用が考えられる。現状の仕組みでは、対象がCordovaのようなHTML+CSSでUIが作られていることが条件になってしまうが、画像の解析やサービスのアーキテクチャ自体は他の対象にも使えるため、UIの検証サービスとして多様な場面で使えるようにしたい。

6. 普及の見通し

プロジェクト期間中にKDDIやリクルートなど複数のIT系企業を対象にヒアリングした結果、確実な需要があると確信が持てた。Web上でサイトを持つ企業は無数にあり、またWebサービスを提供する企業も同様に世界中に数限りなく存在する。これらの企業でごく当たり前に使ってもらえるようにサービスの質を改善し続けたい。

7. クリエータ名 (所属)

内藤 剛生

(Tallinn University of Technology, Computer and Systems Engineering)

(参考) 関連URL

Eyecatch

<https://eyecatch.io/>