Visual UI Testing with Percy

Maroš Kutschy

PNSQC 2021
Let me introduce myself

Maroš Kutschy
Senior QA Automation Engineer at Ness
Košice, Slovakia
Linkedin:
https://sk.linkedin.com/in/maros-kutschy-83100878

creator of Jasecu framework:
https://github.com/maroskutschy/Jasecu
http://jasecu.mablog.eu/blog/
Quotes

“Success is the ability to go from failure to failure without losing enthusiasm”
- Winston Churchill

“I have not failed, I've just found 10,000 ways that won't work”
- Thomas A. Edison
Agenda

1. Visual testing vs functional testing
2. Percy overview
3. How to integrate Percy into test framework
4. Percy UI
5. Percy CSS
6. Advantages
7. Disadvantages
8. Business Impact
9. Supported SDKs
10. Percy Alternatives
11. Demo
12. Questions
1. Visual testing vs functional testing

• **TASK**: Validate hundreds of elements on the page

• *Traditional* (*functional testing*) *approach* – code many separate validations for each element on the page

• *Modern* (*visual testing*) *approach* – do one validation of the whole page
2. Percy Overview

PERCY is a visual review platform

External tool for visual testing - automated process of detecting and reviewing visual UI changes

https://percy.io/
2. Percy Overview

Percy automatically tests UI across browsers and screens through pixel-by-pixel comparisons.
3. Percy integration into framework

1. Add percy-java-selenium to project dependencies → pom.xml

```xml
<dependency>
  <groupId>io.percy</groupId>
  <artifactId>percy-java-selenium</artifactId>
  <version>0.1.4</version>
</dependency>
```

2. Install @percy/agent npm package (node.js is required)

3. Set percy token

When you sign into Percy and you create a new organization and project, then you'll be shown a token environment variable in ‘Project Settings’ tab.

```
$ set PERCY_TOKEN=<your token here>
```
3. Percy integration into framework

4. Create new Percy object with webdriver instance as a parameter, call percy object and make screenshot / screenshots

```
Percy percy = new Percy(webDriver);
percy.snapshot(SNAPSHOT_NAME);
```

```java
public void makeScreenshot(String name) {
    Percy percy = new Percy(getDriver());
    percy.snapshot(name);
}
```
3. Percy integration into framework

5. Run command - tests with result in Percy can ONLY be run via terminal

```
npx percy exec -- mvn install -Dtest=SetOfTestsRunnerPercy.java -Denv=int2 -Daccountld=255264 -Dfluentlenium.webDriver=chrome -Dfluentlenium.capabilities=chrome -DfailIfNoTests=true -Dmaven.test.failure.ignore=true -Dmaven.test.haltafterfailure=true
```
4. Percy UI – runs and statuses
4. Percy UI – differences
4. Percy UI – 1st run + resolution
4. Percy UI – new baselines

When is a new baseline created?

- If you run Percy test from a new branch
- If you change the name of the screenshot
5. Percy CSS

Features to make visual testing more useful

- Hide or change UI elements to help minimize false positives
- Freeze CSS animations and GIFs
5. Percy CSS

1. Hide title in green box

2. Hide title between arrows

3. Hide circle from calendar and change color of font to grey
5. Percy CSS

Visual Testing with Percy
5. Percy CSS

```java
public void makeScreenshot(String name) throws InterruptedException {
    Thread.sleep(2000);
    Percy percy = new Percy(getDriver());
    List<Integer> widths = Arrays.asList(1280, 375, 1080);
    String percyCSS = "div[class='v-date-picker-title'] {visibility:hidden} ";
    percy.snapshot(name, widths, minHeight: 1024, enableJavaScript: false, percyCSS);
}
```
5. Percy CSS
5. Percy CSS
5. Percy CSS

Visual Testing with Percy
5. Percy CSS

```java
public void makeScreenshot(String name) throws InterruptedException {
    Thread.sleep(2000);
    Percy percy = new Percy(getDriver());
    List<Integer> widths = Arrays.asList(1280, 375, 1080);
    String percyCSS = "div[class='v-date-picker-title']{visibility:hidden}"

    div[class='v-date-picker-header__value'] {visibility:hidden}

    button[class='v-btn v-date-picker-table__current v-btn--active v-btn--text v-btn--rounded theme-light accent'] {visibility:hidden;}
    div[class='v-btn__content'] {visibility:visible;}
    div[class='v-btn__content'] {color:#7a8296;"

    percy.snapshot(name, widths, 1024, false, percyCSS);
}
```
5. Percy CSS

The mde-date-picker is a stand-alone component that can be utilized in many existing legacy components. It offers the user a visual representation for selecting dates/months.

Usage:

- **Month picker**:
  - Landscape
  - Portrait
  - Full width
  - Close current date
  - Month picker
  - Multiple

- **Week picker**:
  - Week display
  - Week events

changes from 2023-02-05

Changes from Percy-test 5 minutes ago

The mde-date-picker is a stand-alone component that can be utilized in many existing legacy components. It offers the user a visual representation for selecting dates/months.

Usage:

- **Month picker**:
  - Landscape
  - Portrait
  - Full width
  - Close current date
  - Month picker
  - Multiple

- **Week picker**:
  - Week display
  - Week events

Baseline from Percy-test 8 minutes ago

Navigate snapshots with → A
5. Percy CSS - examples

"div[class=v-calendar-weekly__head] {color: black}"
"div[data-preview=Introduction] { visibility: hidden;}"
"div[class='v-btn__content'] {visibility:visible;}
"div[class=rsg--tab-39] { display : none;}"
6. Percy advantages

- 5000 screenshots monthly for free
- snapshot rendering is consistent
- easy snapshot approval process
- test on different resolutions possible
- not depending on screen size or resolution of the computer/server because it is rendered for validation on Percy cloud
7. Percy disadvantages

- In Cucumber you cannot see the result, you have to go to Percy to confirm differences
- Only supports Chrome/Firefox
- Capturing parts of a screen can be a big pain. You’re better off just snapshotting the whole page but you can use Percy css to hide/show things
- Not easy to debug on Percy side, as it is an external tool
8. Business impact

- Visual Testing was the only reasonable option, otherwise there wouldn’t be any testing
- Automatic feedback after each deploy
- Developers are more involved in testing
- Motivational aspect – everything is possible
- Increased automation skills
- 400 Percy validations during the run
- Fully cover the application testing via Percy
9. Supported SDK-s

- PercyScript
- Ember
- Rails
- Capybara
- Selenium
- Other
- Cypress
- Puppeteer
- WebdriverIO
- NightmareJS
- Nightwatch
- Protractor
10. Percy Alternatives

Applitools

Screener.io
11. Demo
12. Questions
Thanks for your attention