

#### **PREMIER SPONSORS**

**Keep Looking Good:** An Introduction to **Visual Regression** Testing

> Loren Klingman **Big Nerd Ranch**

Sample Repo: https://github.com/loren138/testing-demo

Slides: http://files.klingman.us/keep-looking-good.pdf



Loren Klingman - Big Nerd Ranch

Sample Repo: https://github.com/loren138/testing-demo



#### About Loren

I've been developing websites for 15 years. Working in PHP, Laravel, Angular, React, Spring, and currently Vue.js and Node.js

My Websites: https://klingman.us https://github.com/loren138





# Where we're going

- Types of Testing
- Why You Want Visual Regression Tests
- Tools for Visual Regression Tests
- Implement/Demo Visual Regression Tests
- When to Use Visual Regression Tests



🔴 🔵 🌒 💙 Testing Demo	× +		
← → ♂ ☆	localhost:8080/#/contact		⊪\ © ● ₩ ▼ © ≡
		Home   Contact	
		Contact Loren	
		Your name:	
		Your email:	
		Message:	
		Submit form	

•••	V Testing Demo	× +					
	ŵ	③ localhost:8080/#/contact		lii\ C	0	Ħ	
			Home   Contact				
			Contact Loren				
			Your name:				
			Name is Required!				
			Your email:				
			Email is Required!				
			Message:				
			Message is Required!				
			Submit form				

	V Testing Demo	× +					
$\leftrightarrow$ $\rightarrow$ G	ŵ	Iccalhost 8080/#/contact	⊠ ☆	lii\ 🗉	0	# 🗸 📽	
		Home   Contact					
		Contact Loren					
		Thanks for contacting us someone will follow up with you within 2 business days.					

# Types of Tests



});

#### Unit Testing - Units of Code A validator works as expected

import Contact from "@/views/Contact.vue";

```
describe("Contact.vue", () => {
    it("validator works", () => {
        const localThis = { formData: {} };
```

expect(Contact.methods.validate.call(localThis)).toBe(false);
expect(localThis.nameError).toBe("Name is Required!");
expect(localThis.emailError).toBe("Email is Required!");
expect(localThis.messageError).toBe("Message is Required!");
});



}); });

#### Unit Testing - Render Closer to Integration Testing

import { shallowMount } from "@vue/test-utils"; import Contact from "@/views/Contact.vue";

```
describe("Contact.vue", () => {
    it("shows required errors", () => {
        const wrapper = shallowMount(Contact);
    }
}
```

wrapper.find("form").trigger("submit.prevent");

expect(wrapper.find("#nameError").text()).toBe("Name is Required!"); expect(wrapper.find("#emailError").text()).toBe("Email is Required!"); expect(wrapper.find("#messageError").text()).toBe("Message is Required!");



});

#### Unit Testing - Snapshot Tests that it renders the same as it did before

import { shallowMount } from "@vue/test-utils"; import Contact from "@/views/Contact.vue";

```
describe("Contact.vue", () => {
    it("shows required errors", () => {
        const wrapper = shallowMount(Contact);
    }
}
```

wrapper.find("form").trigger("submit.prevent");

```
expect(wrapper).toMatchSnapshot();
});
```



#### Unit Testing - Snapshot Not a Visual Snapshot

exports[`Contact.vue shows required errors 1`] = ` <div class="content-box contact"> <div> <h2 class="contact title"> Contact Loren </h2> <!---> <form name="contact" method="post"><input type="hidden" name="form-name" value="contact"> <div class="sender-info"> <div><label for="name" class="error">Your name:</label> <input id="name" type="text" name="name"> <div id="nameError" class="error"> Name is Required! </div> </div> <div><label for="email" class="error">Your email:</label> <input id="email" type="email" name="email"> <div id="emailError" class="error"> Email is Required! </div> </div> </div> <div class="message-wrapper"><label for="message" class="error">Message:</label> <textarea id="message" name="message"></textarea> <div id="messageError" class="error"> Message is Required! </div> </div> <button id="sendBtn" type="submit"> Submit form </button> </form> </div> </div>



#### End-to-End Testing Tests the fully rendered page

describe("The Contact Page", function() {
 it("shows errors", function() {
 cy.visit("/#contact");
 cy.contains("h2", "Contact Loren");
 cy.get("#sendBtn").click();
 cy.contains("#nameError", "Required");
 cy.contains("#messageError", "Required");
 cy.contains("#messageError", "Required");
 });
});



#### Visual Regression Testing Tests the looks of the fully rendered page

Scenario("Empty Form Errors", (I) => {
 const imageName = subfolder + "contact-form-invalid.png";
 I.amOnPage("#/contact");
 // Make sure the page has loaded
 I.waitForElement("#sendBtn");
 I.click("#sendBtn");
 // Make sure error message has loaded
 I.waitForElement("#nameError");
 I.waitForText("Name is Required!");
 I.saveScreenshot(outputFolder + imageName);
 I.seeVisualDiff(imageName, { tolerance: 2, prepareBaseImage: false });
});



#### Visual Regression Testing Tests the looks of the fully rendered page

Home	Contact

#### **Contact Loren**

Your name:
Name is Required!
Your email:
Email is Required!
Message:
Message is Required!
Submit form

# Why Do You Want Visual <u>Regression Tests?</u>

New Feature: Thank You for Visiting





#### What went wrong?

•••	V Testing Demo	× +	
	ŵ	I localhost:8080/#/contact	⊪\ © ● ₩ ¥ \$   =
		Home   Contact	
		Contact Loren	
		Thanks for contacting us someone will follow up with you within 2 business days.	



#### Ad Panels

•••	V Testing Demo	× +				
$\leftrightarrow$ $\Rightarrow$ G	ŵ	localhost:8080/#/contact				II\ ⊡ ● ## ¥ © ≡
			Home   Con	ntact		
			Contact I	.oren		
			Your name: Ad Panel Your email: Message:	Ad Panel		
			Submit form			



#### How can we stop it?

- Unit Tests
  - Would have checked the form validation methods
- Unit Integration Tests
  - $\circ \quad \mbox{Would still pass with fully covered fields}$
- Snapshot Tests
  - $\circ$  HTML only
- End-to-End Tests
  - $\circ \quad \ \ {\rm Still \, filled \, and \, submitted \, a \, form}$
  - Could still see the thank you message
  - Could see the error messages show for invalid fields
- Visual Regression Tests

### Tools for Visual Regression Tests





#### CodeceptJS





#### \* 🔹 🖯 🖬 🗸 🔕 🕂 🔻 🔿 👘 🗄

	R
	Kesemple.J5
bu	Id passing 🔿 code quality 🔥 🔿 coverage 🔤 License MIT downloads 46k/m
Analyse and compare ima	iges with Javascript and HTML5. More info & Resemble.js Demo. Compatible with Node.js >8.
Get it	
npm install resemblejs	



## Test Case for Initial Form

Feature("Contact Form");

```
const subfolder = "contact/";
const outputFolder = "output/";
```

```
Scenario("Initial Form", (I) => {
    const imageName = subfolder + "contact-form.png";
    I.amOnPage("#/contact");
    // Make sure the page has loaded
    I.waitForElement("#sendBtn");
```

```
I.saveScreenshot(outputFolder + imageName);
I.seeVisualDiff(imageName, { tolerance: 2, prepareBaseImage: false });
});
```



# Test Case for Form Validation

Scenario("Empty Form Errors", (I) => {
 const imageName = subfolder + "contact-form-invalid.png";
 I.amOnPage("#/contact");
 // Make sure the page has loaded
 I.waitForElement("#sendBtn");
 I.click("#sendBtn");
 // Make sure error message has loaded
 I.waitForElement("#nameError");
 I.waitForText("Name is Required!");
 I.saveScreenshot(outputFolder + imageName);
 I.seeVisualDiff(imageName, { tolerance: 2, prepareBaseImage: false });
});

# Test Case for Submitting Form

```
Scenario("Thank You", (I) => {
 const imageName = subfolder + "contact-form-thank-you.png";
 I.amOnPage("#/contact");
 I.waitForElement("#sendBtn");
 // Fill Form
 I.fillField("#name", "Loren");
 I.fillField("#email", "email@email.com");
 I.fillField("#message", "Hello From Codecept");
 // Send
 I.click("#sendBtn");
 // Make sure at least one error message has loaded
 I.waitForElement("#thanks");
 I.waitForText("Thanks for contacting us", 1, "#thanks");
```

I.saveScreenshot(outputFolder + imageName);
I.seeVisualDiff(imageName, { tolerance: 2, prepareBaseImage: false });
});



## Storing Images

Each test in a folder with it's images



### Getting the Reference Images

- You can temporarily set prepareBaseImage to be true
- The screenshots will be placed in the output folder so you can also just copy them over from there after the first run of the tests.



## Running the Tests

- You will need JDK installed for selenium to work (brew cask install java)
- Runnpm install
- Run npm run selenium (Leave this running while running tests)
- Run npm run serve (to have a server running)
- Runnpm run test:codecept
- After you're done testing, kill the background processes



# Running the Tests



### Demo Visual Regression Testing

#### New Feature: Thank You For Visiting





## Output

Home Contact

**Contact Loren** 



#### -- FAILURES:

1) Contact Form Thank You:

> MissMatch Percentage 3.53 + expected - actual

-false +true

# When to Use Visual Regression Tests

- Add when mid/final-UI is implemented
- Validate screenshot changes with UX
- Limit 1 screenshot per page scenario
  - One for Base Page
  - One for Error
  - One for Success
- Great libraries for React Storybook

# Tips for Stable Visual Testing

- Disable Animations
- Mock Dates (Timecop/Sinon.js)
- Reproducible Data (No random placeholder data)
- Use local images as much as possible/avoid rate limited remote ones

Pat O'Callaghan - Visual Regression Testing Made Easy (Percy)

#### Tips for Stable E2E Tests

- Use (short, 1s) waits anywhere you can over immediate checks
- Use longer waits when waiting for an API call to return
- Don't use long waits everywhere as it causes failing test runs to take an extremely long time
- Be sure your backend is stable/fast
- Setup clean test data before each test
- Rerun failing tests once or twice before failing the build
- Rerun new/changed tests twice and require them to pass every time

# Speeding up E2E Tests

- Test multiple scenarios in one block (ie full flows) to avoid browser reboots
- Be sure scenarios and/or files can be run in parallel (no dependencies on previous files/scenarios)
- Run the tests in parallel
- Speed up your backend, tests can't run any faster than they can fetch the data

#### Tomorrow: Testing Your Front-End: How to Divide Your Time and Resources



We design and build exceptional apps and teach others to do the same.

We're hiring web, iOS, and Android.

https://www.bignerdranch.com/

#### Questions?

lklingman@bignerdranch.com https://klingman.us https://www.bignerdranch.com

Sample Repo: https://github.com/loren138/testing-demo

Slides: http://files.klingman.us/keep-looking-good.pdf