



PNSQC

OCTOBER 10-12 2022

TestZeus:
Automate the
Lightning and
Thunder of
Salesforce
Testing

Celebrating 40 Years
Of Software Quality

About Me



Robin Gupta

Associate Vice President - Innovation at Provar

Twitter: @smilinrobin

Linkedin: <https://www.linkedin.com/in/polymorphicrobin/>

Email: smilin.robin@gmail.com



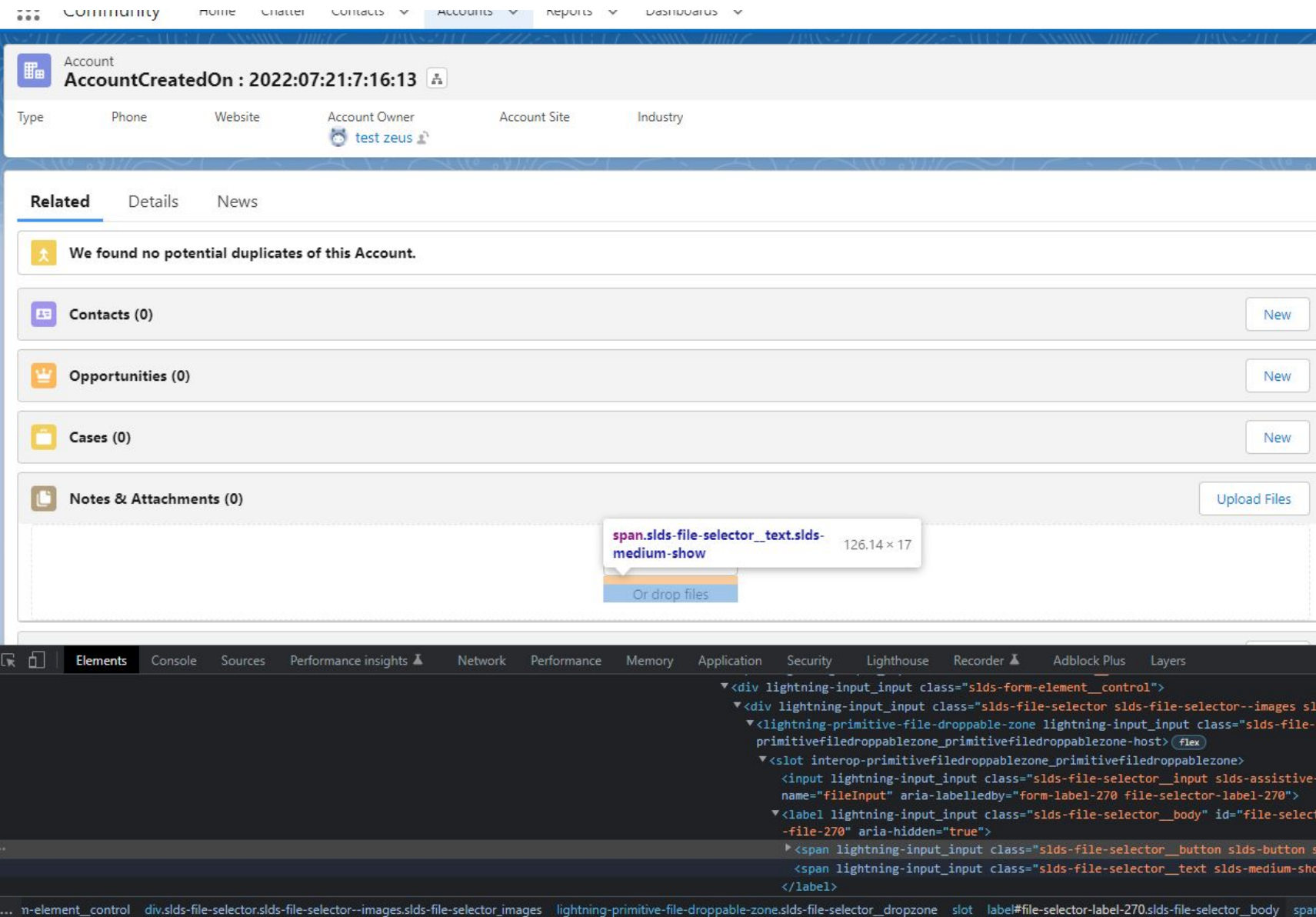
Show of Hands

How many of you have
worked on a SAAS
application?

Then you know the pain



It's always calm on the surface

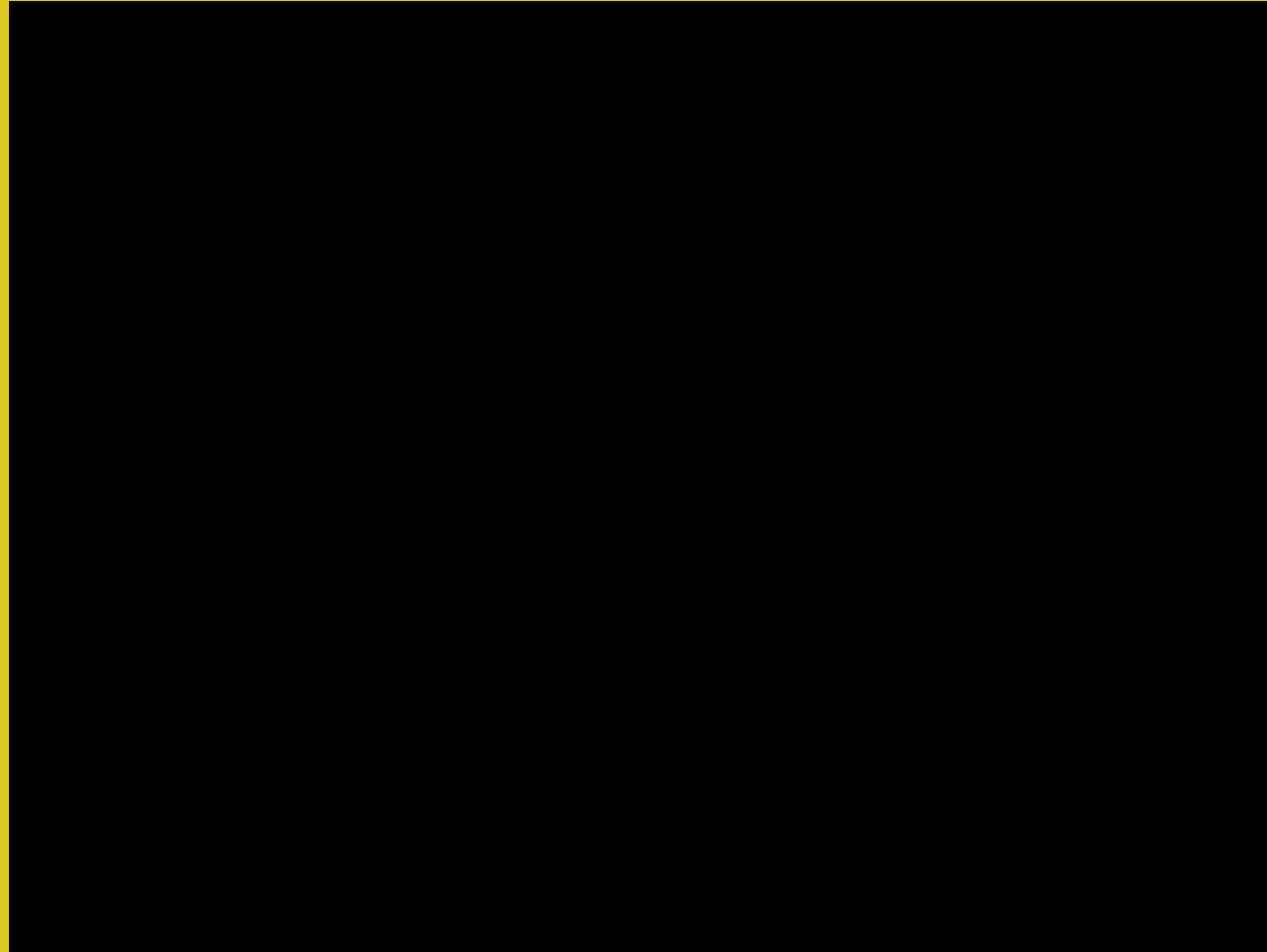


The screenshot shows a Salesforce Account record page. The account is named "test zeus" and was created on 2022-07-21 7:16:13. The page displays related records for Contacts, Opportunities, Cases, and Notes & Attachments, all of which are currently empty. A tooltip is visible over a file selector button, showing the selector `span.slds-file-selector__text.slds-medium-show` and its dimensions `126.14 x 17`. At the bottom, the Chrome DevTools Elements panel is open, displaying the DOM tree for the file selector button. The DOM structure is as follows:

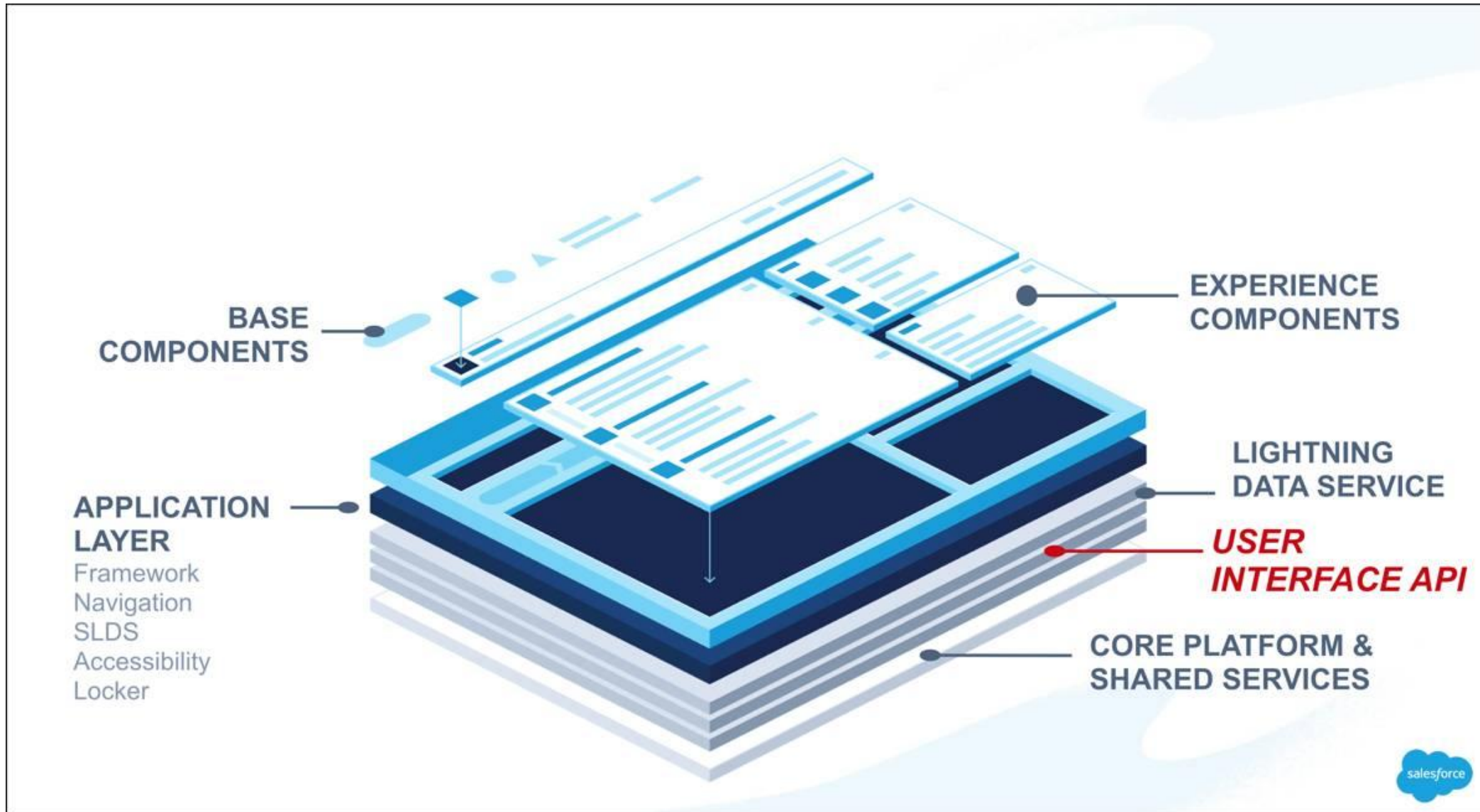
```
<div lightning-input_input class="slds-form-element__control">
  <div lightning-input_input class="slds-file-selector slds-file-selector--images slds-file-selector--medium">
    <lightning-primitive-file-droppable-zone lightning-input_input class="slds-file-selector__file-droppable-zone">
      <slot interop-primitivefiledroppablezone_primitivefiledroppablezone-host>
        <input lightning-input_input class="slds-file-selector__input slds-assistive-text" type="text" value="file" />
        <label lightning-input_input class="slds-file-selector__body" id="file-selector-label-270" aria-hidden="true">
          <span lightning-input_input class="slds-file-selector__button slds-button slds-button--ghost" type="button" value="file" />
          <span lightning-input_input class="slds-file-selector__text slds-medium-show" type="text" value="file" />
        </label>
      </slot>
    </lightning-primitive-file-droppable-zone>
  </div>
</div>
```

1. **Frequent system updates:** Salesforce regularly updates their platform to enhance user experiences or fix underlying issues. For QA teams, this means a lot of maintenance.
2. **Shadow DOMs:** Salesforce uses Shadow DOMs to isolate components. This makes it difficult to identify elements in UI test automation.
3. **Heavy DOM structure:** Salesforce's DOM structure is heavy with a complex tree structure. This means that automation tools will need more time to access them.
















And that's where TestZeus comes in.



How is the UI rendered for Salesforce?



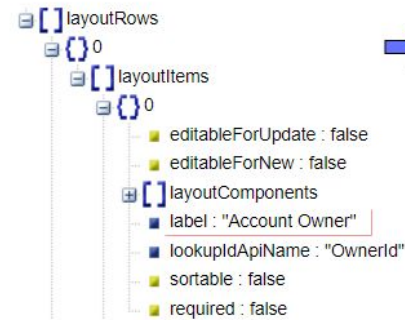
AutoLocator Strategy

Related	Details	News
Account Owner	 test zeus	
Account Name	AccountCreatedOn : 2022:07:21:7:16:13	
Parent Account		
Account Number		
Account Site		
Type		
Industry		
Annual Revenue		
Billing Address		
Customer Priority		
SLA Expiration Date		
Number of Locations		
Active		
Created By	 test zeus, 7/21/2022, 6:46 AM	
Description		
Billing		

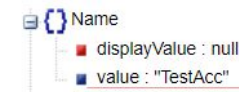
UI API

GET /ui-api/record-ui/001R0000003GeJ1IAK

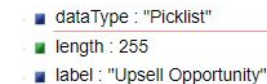
Automate user interfaces that let users work with records, list views, actions, favorites, and more. Not only do you get data and metadata in a single response, but the response matches metadata changes made to the org by Salesforce admins. You don't have to worry about layouts, picklists, field-level security, or sharing—all you have to do is build super reliable automation scripts.



TestZeus gets the Sections, Labels and Datatypes from the UI API to create stable locators and based on the datatypes, contextually provides actions to interact with the UI



You also get the field values for the record (so that you don't have to driver.findElement(By.xpath("crazyXPathHere")).getText();



And you get the DataType for the record, so that we can build contextually aware actions for the fields.

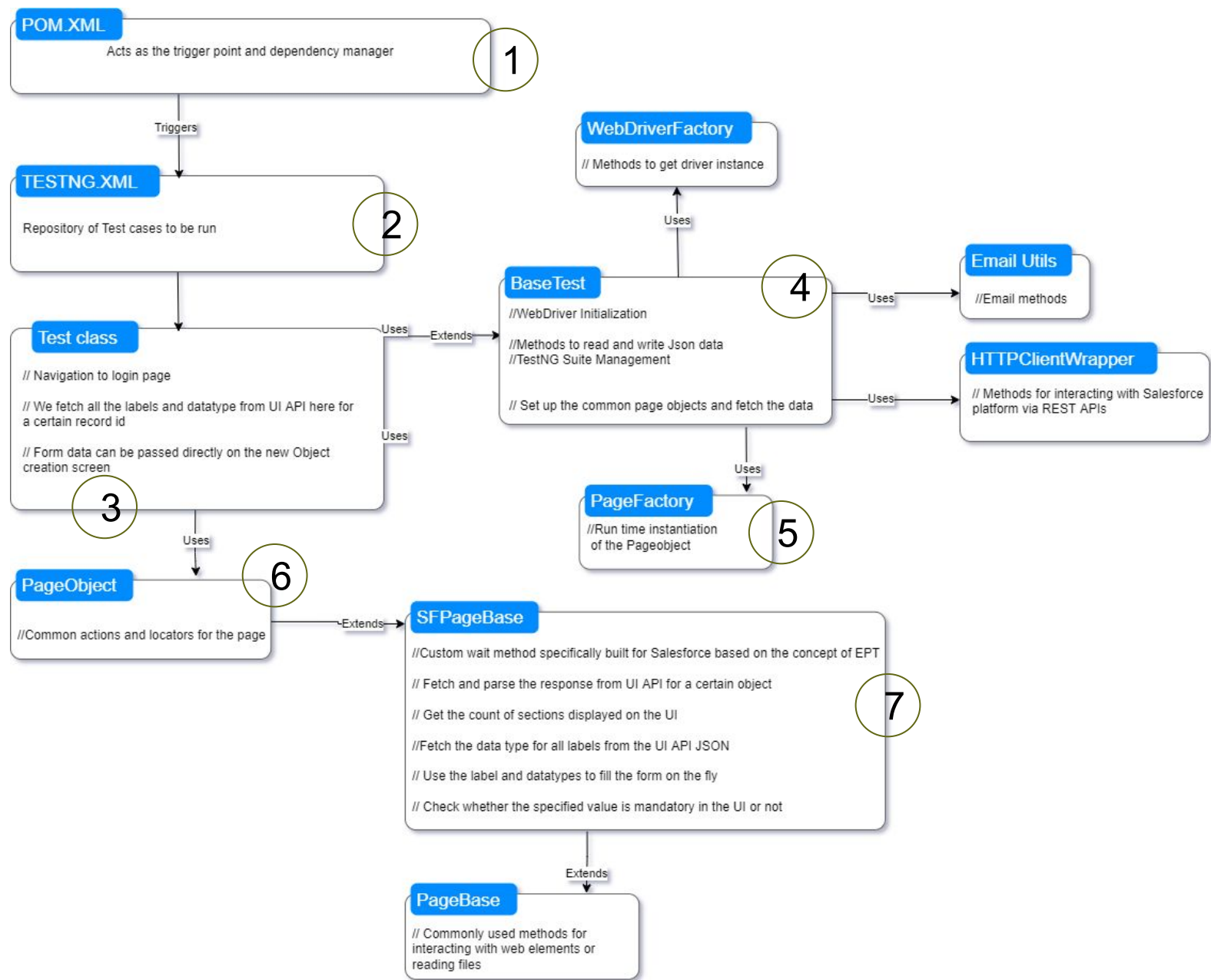
AutoWaits based on EPT

What Is EPT?

Experienced Page Time (EPT) is a performance metric Salesforce uses in Lightning to measure page load time. EPT measures how long it takes for a page to load into a state that a user can meaningfully interact with.



High Level Framework diagram 🏠:





Demo

Personal Experience and other footnotes

TestZeus helps us via:

- Accelerating the Test automation effort by removing the need for writing explicit locators
- Adding stability via automatically waiting for the page to load
- Reducing the maintenance effort for feature and platform level changes

TestZeus is an open source test automation framework created as a Proof of Work for utilizing Salesforce provided APIs to accelerate the test automation process.

Target users for TestZeus are Technically strong engineering teams, willing to experiment with open source selenium based test automation frameworks.

We are a GPL V3 Open source project and open for contributions 🙌👼

The road ahead

Life does not come with a compass,
no map to guide the way.
No torch to light the road ahead.
no answer when we pray

We all must choose the path we tread,
living from day to day.
and trust whatever route we take
will not lead us astray.

- [Shacknofsky](#)

Resources and support:

1. Salesforce community for Test Automation :
[Test Automation Trailblazers](#)
2. Free Training modules on Salesforce:
[Trailhead for learning](#)
3. [Selenium community on Slack](#)
4. [TestZeus Github Repository](#)

Thanks



www.testzeus.com