

Lightning Performance and Debugging Tips for Developers



Sara Morgan Nettles
Lead Technical Curriculum Engineer
at Salesforce
@SaraHasNoLimits
<http://saramorgan.net>

Round of applause for our Platinum Sponsors!



shellblack



zennify



Lightning Champions



Round of applause for our Gold Sponsors!



Thanawalla Digital



I Am Also a Pluralsight Author



- [Getting Started Building SPAs with Lightning Components](#)
- [Customizing Salesforce with Lightning Components](#)
- [Lightning Component Development Best Practices](#)



Agenda

- Top 5 Performance Tips
- Top 5 Debugging Tips
- Demo of Using Lightning Inspector
- Questions



Top 5 Performance Tips



1 - Check Component Caching Setting

To Enable/Disable Component Caching:

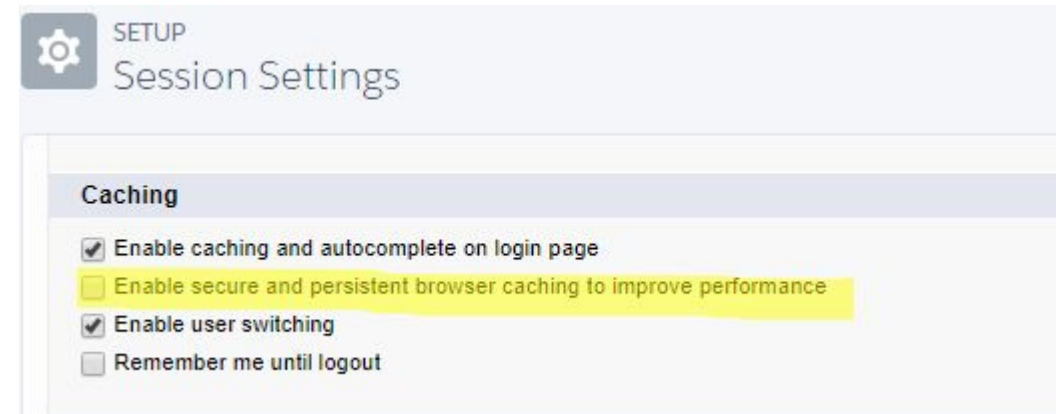
1. From Setup, go to **Session Settings**.
2. Select/Deselect the **Enable secure and persistent browser caching to improve performance** checkbox.
3. Click **Save**.



Definitely Enable for Production



May Want to Disable for Development



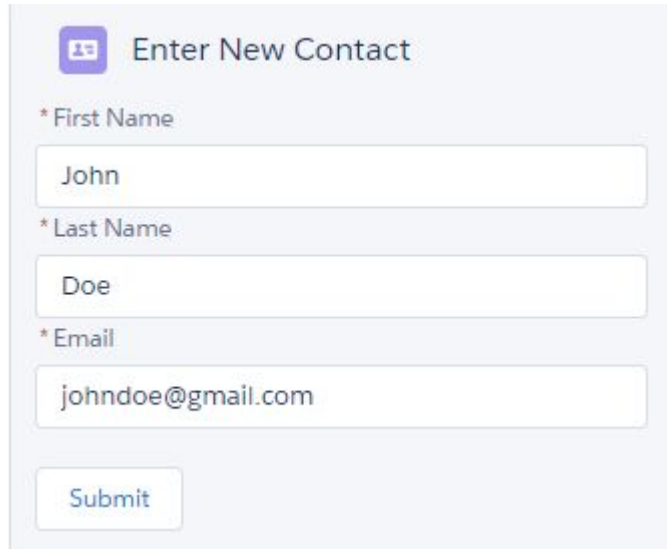
1 - Check Component Caching Setting

```
{ } project-scratch-def.json ●
1  {
2      "orgName": "sara.morgan Company",
3      "edition": "Developer",
4      "orgPreferences" : {
5          "enabled": ["S1DesktopEnabled"],
6          "disabled": ["S1EncryptedStoragePref2"]
7      }
8  }
9  |
```

For SalesforceDX developers, you can disable the setting automatically for scratch orgs with the following line in your config file



2 - <aura:if> for Conditional Rendering



Enter New Contact

* First Name
John

* Last Name
Doe

* Email
johndoe@gmail.com

Submit

```
<div aura:id="submitBtn" class="slds-hide">  
  <lightning:button label="Submit"  
    onclick="{!c.newRecord}" />  
</div>
```

```
cmp.find("submitBtn").toggleClass(!isVisible, "slds-hide");
```



Don't Use this



2 - <aura:if> for Conditional Rendering

```
<aura:if isTrue="{!v.isVisible}">
```

```
    <lightning:button label="Submit"
```

```
        onclick="{!c.newRecord}" />
```

```
</aura:if>
```

← Use this instead

<https://bit.ly/2Lf3YRp> for more info



3 - Limit Data Returned

- Make sure SELECT statement only returns data you absolutely need
- Use the LIMIT clause to limit # of rows returned
- Consider adding pagination to lists of data
- <https://bit.ly/2J89QiP> for more info

10 Races • page 1 of 2



4 - Use the Lightning Data Service

AKA: The Standard Controller for Lightning

Benefits include:

- No APEX or SOQL needed

- FLS and CRUD security built-in

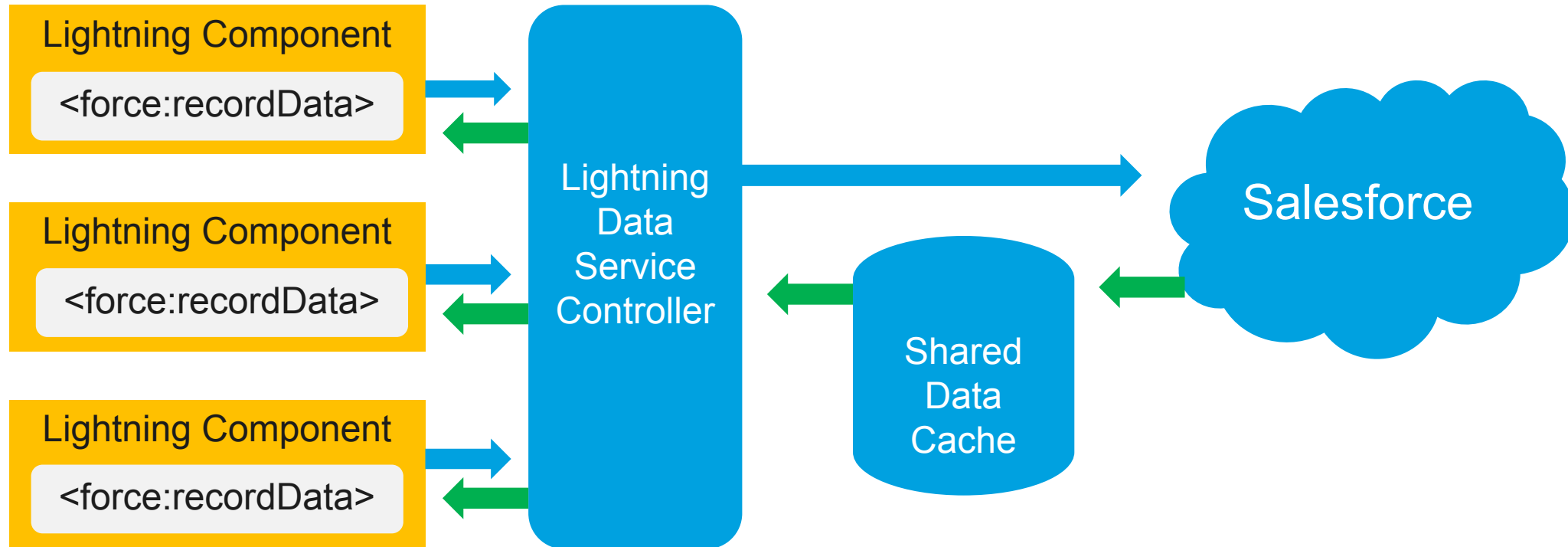
- Auto-notify on record changes

- Offline access for Salesforce Mobile

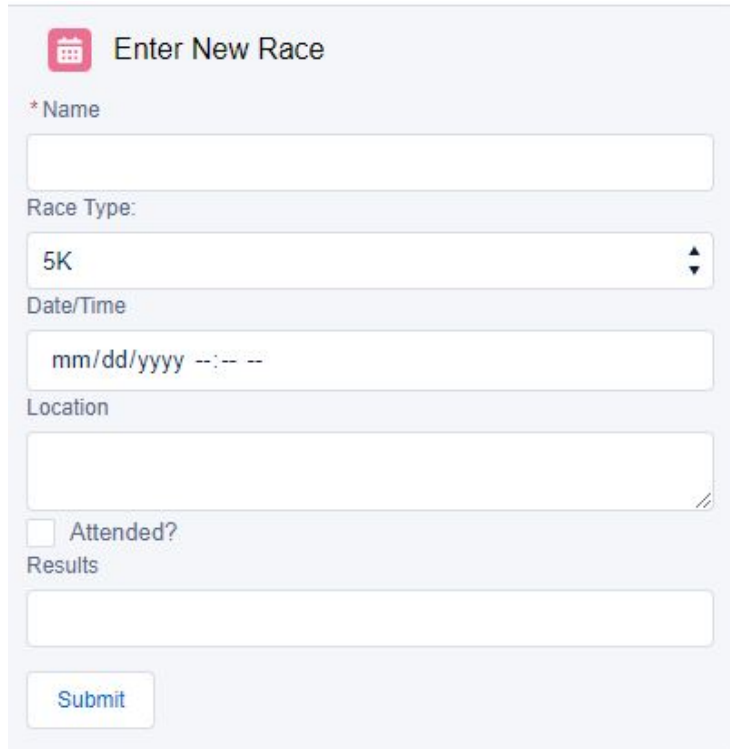
- Single Request and Cached Response**



4 - Use the Lightning Data Service



4 - Use the Lightning Data Service



Enter New Race

* Name

Race Type:

5K

Date/Time

mm/dd/yyyy --:--

Location

☐ Attended?

Results

Submit

```
<force:recordData aura:id="raceRecordCreator"
    layoutType="FULL"
    targetRecord="{!v.newRaceRec}"
    targetFields="{!v.newRace}"
    targetError="{!v.errorMsg}" />
```

<https://bit.ly/2xGxKfS> for more info



5 - Use Storable Actions When Possible

ACCOUNT NAME	ACCOUNT NUMBER	ANNUAL REVENUE
Burlington Textiles Corp of America	CD656092	\$350,000,000
Dickenson plc	CC634267	\$50,000,000
Edge Communications	CD451796	\$139,000,000
Express Logistics and Transport	CC947211	\$950,000,000
GenePoint	CC978213	\$30,000,000

13 Accounts • page 1 of 3

top

Filter

Default levels

Page 1 loaded in 545.1550000000002ms	LightningDataGrid.js:38
Page 2 loaded in 297.7900000000009ms	LightningDataGrid.js:38
Page 3 loaded in 271.0150000000067ms	LightningDataGrid.js:38
Page 2 loaded in 0.6000000000058208ms	LightningDataGrid.js:38
Page 1 loaded in 0.46999999998661224ms	LightningDataGrid.js:38

Notice the Pagination



5 - Use Storable Actions When Possible

```
var action = component.get('c.getRacesDB');
```

```
action.setStorable();
```



Caching in a
Single Line of
Code

```
action.setCallback(this, function(response) {
```

```
    // Code here to handle the response
```

```
}
```

```
$A.enqueueAction(action);
```

<https://bit.ly/2wdRHFR> for more info



Top 5 Debugging Tips



1 - Check Debug Mode Setting

To Enable/Disable Debug Mode:

1. From Setup, go to **Lightning Components**.
2. Select/Deselect the **Enable Debug Mode** checkbox.
3. Click **Save**.



ONLY Enable for
Development or Sandboxes



Make sure you Disable
for Production

The screenshot shows the Salesforce Setup interface for Lightning Components. At the top, there's a 'SETUP' header with a gear icon and the text 'Lightning Components'. Below this, the title 'Lightning Components' is displayed. A subtitle reads: 'View Lightning components and enable debug mode to make it easier to debug JavaScript code.' Underneath, there's a section titled 'Lightning Components' containing a checkbox labeled 'Enable Debug Mode' which is currently checked. At the bottom right of the form, there are two buttons: 'Save' and 'Cancel'.



2 - Use the Salesforce CLI Linting Tool

Salesforce DX includes a linting tool you can use during development to avoid Locker Service conflicts and anti-patterns

<https://sforce.co/2LW7mSd> for more about the Salesforce CLI



2 - Use the Lightning Linter Service?

Lightning Linter Service

```
2  getRaces : function(component) {
3      var action = component.get('c.getRacesDB');
4      action.setCallback(this, function(response) {
5          var state = response.getState();
6          if (component.isValid() && state === "SUCCESS") {
7              component.set("v.races", response.getReturnValue());
8          } else {
9              //Handle errors
10             var errors = response.getError();
11             console.log('error:' + errors[0].message)
12         }
13     });
14     $A.enqueueAction(action);
15 }
16 },
17 addToRaces: function(component, event) {
18     var race = event.getParam('race');
```

<https://lightning-linter.herokuapp.com/>

Parse



11¹⁷

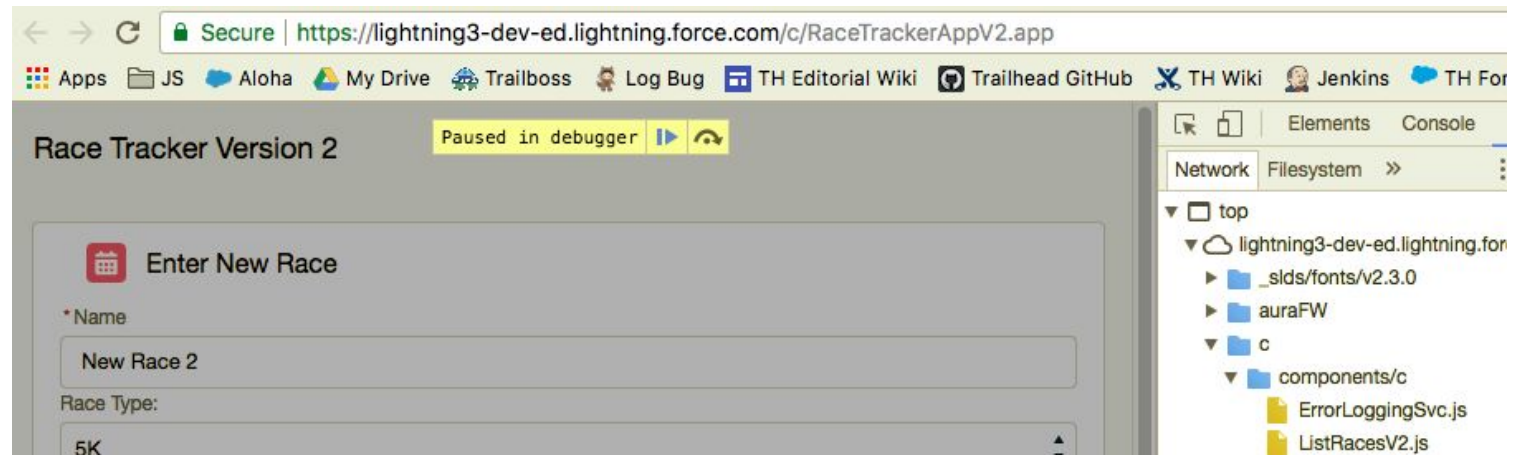
Unexpected console statement.
console.log('error:' + errors[0].message)



3 - Use debugger and Dev Tools

```
18  
19 ▼  
20   createNewRace : function(component) {  
21     var log = component.find("logService");  
22     debugger; ← Pause Execution with debugger statement  
23     component.find("raceRecordCreator").saveRecord(function(saveResult) {  
24       if(saveResult.state === "SUCCESS" || saveResult.state === "DRAFT") {
```

<https://bit.ly/2k498IH>
for more info



4 - Use Lightning Testing Service

Set of tools used to create test suites in Jasmine or Mocha

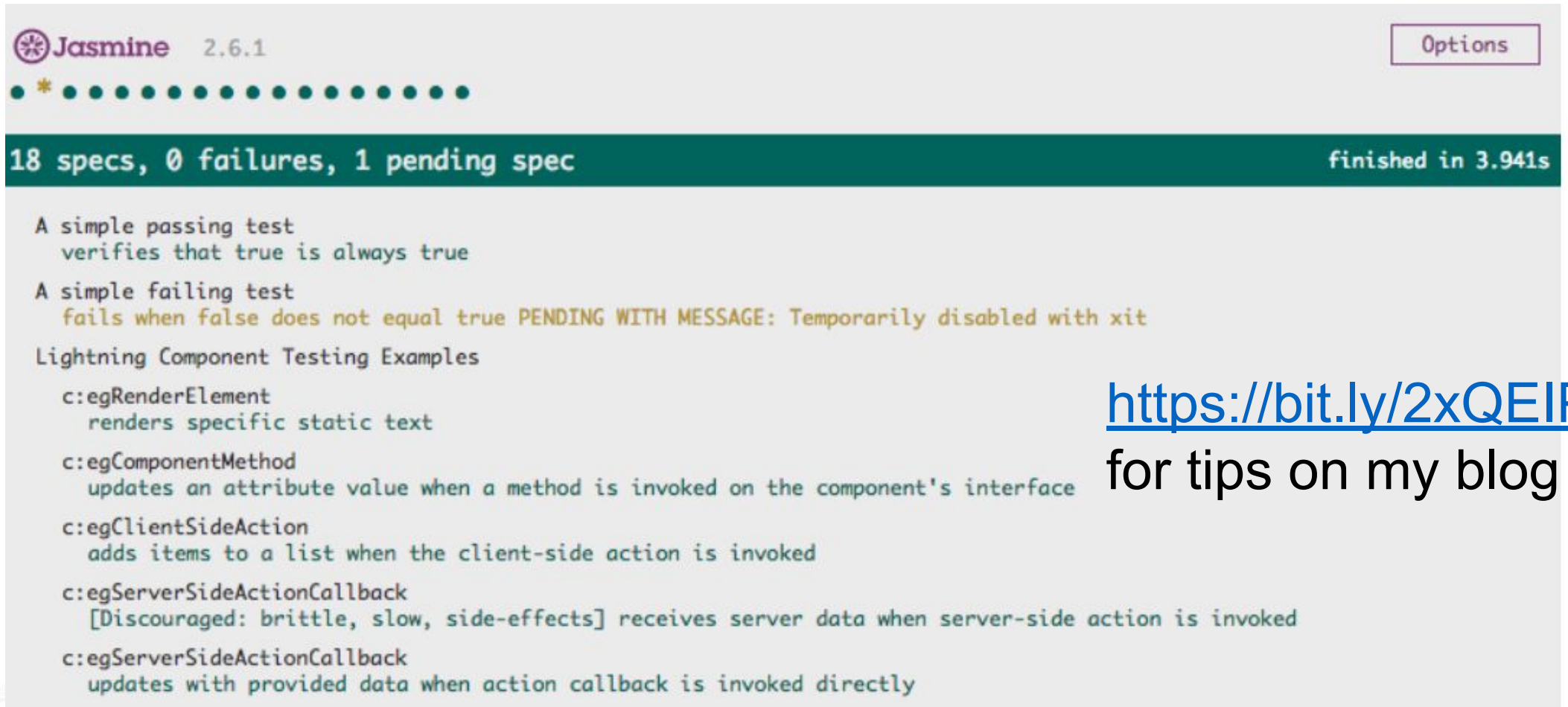
```
/**
 * This is a 'hello world' Jasmine test spec
 */
describe("A simple passing test", function() {
  it("checks that true is always true", function() {
    expect(true).toBe(true);
  });
});
```

<https://sforce.co/2kUXegv>
for more info

Currently in Pilot



4 - Use Lightning Testing Service



The image shows the Jasmine 2.6.1 test runner interface. At the top left is the Jasmine logo and version number. At the top right is an 'Options' button. Below the header is a progress bar consisting of 18 dots, with the first one highlighted. A green status bar displays '18 specs, 0 failures, 1 pending spec' and 'finished in 3.941s'. The main area contains a list of test specifications:

```
Jasmine 2.6.1 Options
● * ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●
18 specs, 0 failures, 1 pending spec finished in 3.941s

A simple passing test
  verifies that true is always true

A simple failing test
  fails when false does not equal true PENDING WITH MESSAGE: Temporarily disabled with xit

Lightning Component Testing Examples

c:egRenderElement
  renders specific static text

c:egComponentMethod
  updates an attribute value when a method is invoked on the component's interface

c:egClientSideAction
  adds items to a list when the client-side action is invoked

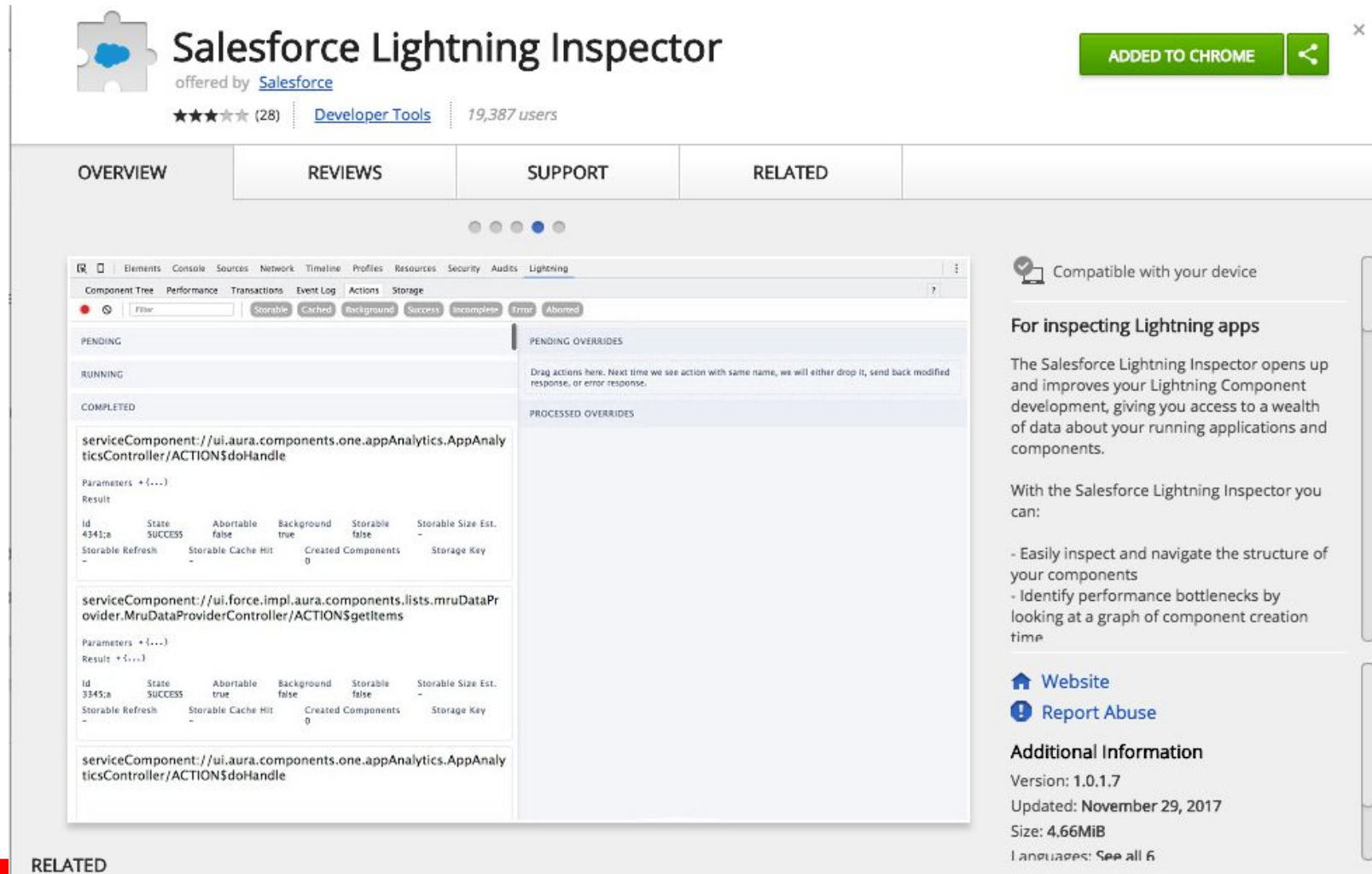
c:egServerSideActionCallback
  [Discouraged: brittle, slow, side-effects] receives server data when server-side action is invoked

c:egServerSideActionCallback
  updates with provided data when action callback is invoked directly
```

<https://bit.ly/2xQEIPs>
for tips on my blog



5 - Use the Lightning Inspector Plugin



The screenshot displays the Salesforce Lightning Inspector Chrome extension interface. At the top, the extension is identified as "Salesforce Lightning Inspector" offered by Salesforce, with a 4.5-star rating from 28 reviews and 19,387 users. A green button indicates it has been "ADDED TO CHROME". Below this is a navigation bar with tabs for OVERVIEW, REVIEWS, SUPPORT, and RELATED. The main content area is divided into two panels. The left panel shows a list of components under the "Lightning" tab, with sub-tabs for Component Tree, Performance, Transactions, Event Log, Actions, and Storage. The right panel displays details for a selected component, including its serviceComponent, parameters, and a table of results. The results table lists components with their IDs, states, abortable status, background status, storable status, and storage size. The right sidebar contains a "Compatible with your device" message, a description of the inspector's purpose, a list of capabilities, and links to the website and report abuse page. Additional information at the bottom right includes the version (1.0.1.7), update date (November 29, 2017), size (4.66MiB), and languages (See all 6).

Salesforce Lightning Inspector
offered by [Salesforce](#)
★★★★★ (28) | [Developer Tools](#) | 19,387 users

ADDED TO CHROME

OVERVIEW | REVIEWS | SUPPORT | RELATED

Lightning

Component Tree | Performance | Transactions | Event Log | Actions | Storage

PENDING

PENDING OVERRIDES

Drag actions here. Next time we see action with same name, we will either drop it, send back modified response, or error response.

PROCESSED OVERRIDES

serviceComponent://ui.aura.components.one.appAnalytics.AppAnalyticsController/ACTION\$doHandle

Parameters + (...)

Result

Id	State	Abortable	Background	Storable	Storable Size Est.
4341:a	SUCCESS	false	true	false	-
Storable Refresh	-	Storable Cache Hit	Created Components	Storage Key	

serviceComponent://ui.force.impl.aura.components.lists.mruDataProvider.MruDataProviderController/ACTION\$getItem

Parameters + (...)

Result + (...)

Id	State	Abortable	Background	Storable	Storable Size Est.
3345:a	SUCCESS	true	false	false	-
Storable Refresh	-	Storable Cache Hit	Created Components	Storage Key	

serviceComponent://ui.aura.components.one.appAnalytics.AppAnalyticsController/ACTION\$doHandle

Compatible with your device

For inspecting Lightning apps

The Salesforce Lightning Inspector opens up and improves your Lightning Component development, giving you access to a wealth of data about your running applications and components.

With the Salesforce Lightning Inspector you can:

- Easily inspect and navigate the structure of your components
- Identify performance bottlenecks by looking at a graph of component creation time

[Website](#)
[Report Abuse](#)

Additional Information

Version: 1.0.1.7
Updated: November 29, 2017
Size: 4.66MiB
Languages: [See all 6](#)



5 - Use the Lightning Inspector Plugin

<https://sforce.co/2jNXJr2>
for more info

Race Tracker Version 2

Enter New Race

* Name

Race Type: 5K

Date/Time

Location

Attended?

Results

Elements Console Sources Network Performance Memory Application Security Audits **Lightning**

Component Tree Performance Transactions Event Log Actions Storage

Expand All Show Global IDs Update On Select

```
<aura:if isTrue="function(cmp, fn) { return (cmp.get("v.page")<cmp.get("v.pages")); }" else=[] template=[1]>
  <lightning:buttonIcon onfocus="null" onblur="null" iconName="utility:right" variant="border" size="medium" disabled="false" onclick="{!c.nextPage}" type="button" privateComputedButtonClass="slds-button slds-button--icon-border" privateIconClass="slds-button__icon">
    <button aura:id="privateButton" class="{!v.privateComputedButtonClass}" onblur="{!c.handleBlur}" value="{!v.value}" accesskey="{!v.accesskey}" type="{!v.type}" name="{!v.name}" onfocus="{!c.handleFocus}" disabled="{!v.disabled}" tabindex="{!v.tabindex}" onclick="{!c.handleClick}" title="{!v.privateTitle}">
      <lightning:primitiveIcon iconName="{!v.iconName}" variant="bare" svgClass="{!v.privateIconClass}" privateSvgComputedClass="slds-button__icon">
        <span>
          <aura:if isTrue="function(cmp, fn) { return !(fn.empty(cmp.get("v.alternativeText"))); }" else=[1] template=[1]>
            <aura:if isTrue="function(cmp, fn) { return !(fn.empty(cmp.get("v.footer"))); }" else=[] template=[1]>
```

Descriptor: markup://aura:html
Global ID: 268:99;a
Is Rendered: true
Is Valid: true
HTML Elements: 4
Rerendered Count: 0
aura:id: privateButton
Attribute & Facet Value Provider
 <lightning:buttonIcon globalId="266:99;a"
 onfocus="null" onblur="null"
 iconName="utility:right" variant="border"
 size="medium" disabled="false"
 onclick="{!c.Paginator\$controller\$nextPage}"
 type="button" privateComputedButtonClass="slds-button slds-button--icon-border"
 privateIconClass="slds-button__icon">
Attributes
body: [2]



Demo Time



Check Out These Amazing Trails:

- [Lightning Component Tips and Gotchas](#)
- [Lightning Data Service Basics](#)
- [Lightning Component Basics](#)



Thank You!



Sara Morgan Nettles
<http://saramorgan.net>
@SaraHasNoLimits

Happy *trails* 'til next
year, y'all!

