

# BlazeMeter<sup>®</sup>

## Scriptless Testing for Salesforce Lightning Applications

*Easily create and automate tests for your Salesforce Lightning applications.*

### Key Benefits

- Accurately locate elements located in your application—even if they are encapsulated by the shadow DOM.
- Generate GUI functional tests using the BlazeMeter Chrome Extension recorder.

### Get Started

You are one step closer to running frustration-free automated tests for your Salesforce Lightning applications. Use the following resources to get started:

- Sign up for a free trial of BlazeMeter and start testing at [blazemeter.com](https://blazemeter.com).
- Read the how-to blogs:
  - [Web Components and Shadow DOM: Why and How to Use Them](#)
  - [How to Create Automated Tests for Salesforce Lightning Applications with BlazeMeter](#)
- Explore our online courses and even get certified. Visit the [BlazeMeter University](#).

### Overview

BlazeMeter Scriptless Testing is one of the few automated testing tools on the market that enables you to accurately and reliably test applications built on Salesforce Lightning and other applications using web components.

### Business Challenges

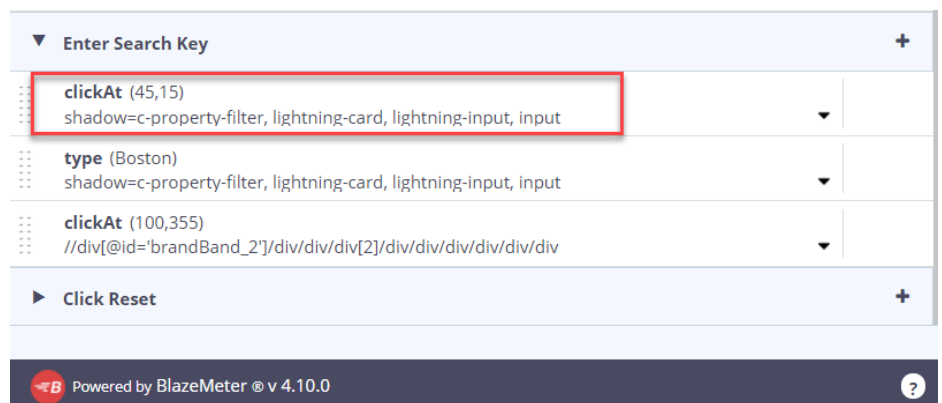
Salesforce Lightning applications are becoming increasingly popular because of their ease of development and flexibility. The lightning framework uses web components and the shadow DOM extensively.

Shadow DOM encapsulates web elements, and keeps markup structure, style, and behavior hidden and separate from the rest of the page for portability and reusability reasons. Since Selenium is the standard framework for GUI functional testing and does not easily support the shadow DOM, it cannot locate any of the hidden elements that need to be tested. Elements that cannot be properly identified and located cannot be tested, let alone tested in an automated fashion.

### Solution Overview

Blazemeter Scriptless Testing automatically identifies locators of elements in the shadow DOM. Tests can be recorded and further automated so you can continuously test for functional correctness at scale.

#### An Example of a Shadow Locator Identified by Blazemeter



▼ Enter Search Key		+
⋮	<b>clickAt</b> (45,15) shadow=c-property-filter, lightning-card, lightning-input, input	▼
⋮	<b>type</b> (Boston) shadow=c-property-filter, lightning-card, lightning-input, input	▼
⋮	<b>clickAt</b> (100,355) //div[@id='brandBand_2']/div/div/div[2]/div/div/div/div/div/div	▼
▶ Click Reset		+

Powered by BlazeMeter @ v 4.10.0