

Feature List:

Features:

**Pre-built spec/concept files:** This gives a strong running start for testers to work from. These spec files can be modified and implemented for the needs of an application under test. This feature is only available on the full version of BOLT Test.

**Custom Specs:** Organize test cases by page or by feature and create a custom spec for each. Every application under test will need customized spec files as no two applications are the same. These files do not need any modifications when upgrading to the full version.

Example:

**Install Test**

=====

**Tags:** install, regression

**Install Test Case**

-----

**Tags:** installTest, smokeTest

\* **Go to** "http://www.google.com"

**Custom Concepts:** Concepts bring multiple steps together and allow data to be sent to them for those steps. This helps reduce the repetition of steps in multiple test cases, making it easier to create and read test cases. These files do not need any modifications when upgrading to the full version of Bolt Test.

Example:

**# Enter Form Data** <field1> <field2> <field3>

\* Enter text <field1> into field with default value "field1"

\* Enter text <field1> into field with default value "field2"

\* Enter text <field1> into field with default value "field3"

\* Click "submit" button

**Smart Elements**

Need info on this

**Custom Steps:** While BOLT Test comes with many pre-generated steps to build tests from, every software is custom. It is simple to create your own custom step that can do any action, set of actions, or even process data while completing action(s). This feature is only available on the full version of BOLT Test.

**Custom Actions:** BOLTTest is able to complete most webpage interactions right out of the box, but there is always more that can be done. Custom actions can be created to do anything, even adding integration with other software packages. This feature is only available on the full version of BOLT Test.

**Appium:** BOLT Test is built to include support for mobile testing by utilizing Appium. To utilize this feature, a user simply needs to have an appium server up and running with a connected emulated

device. As this feature is in its infancy, it is included with BOLT Test, but as it is advanced, this feature will likely be restricted.

**Environment Property Files:** As testing is conducted against an application, it often moves through different environments (dev, QA, pre-prod, prod). These property files can store the information needed to run tests on each environment and can be specified when running via command line.

**Negative Testing:** BOLT Test is designed with negative tests in mind. Simply by marking a spec as negative ("-NEGATIVE"), some of the default validations are not run and the testcase will need to conduct the validations.

**Custom Element Definitions:** BOLT Test smart elements are designed to handle the interaction with most elements, but sometimes it just isn't good enough. To handle this situation, elements can be defined in a csv file and BOLT Test will utilize the given definitions.

**Database Connections:** Databases are often the source of truth and need to be used with validation. BOLT Test makes this process simple with integration for database calls. This feature is only available on the full version of BOLT Test.

**Performance Testing:** BOLT Test is able to both run JMeter test scripts and pass variables into a JMeter script. This allows performance and API tests to be run by any tester. This feature is only available on the full version of BOLT Test.

**SoapUI Testing:** SoapUI is the most common API testing tool, so we just had to integrate it with BOLT Test. Just like JMeter, SoapUI tests can be run from BOLT Test and variables can be passed in. This feature is only available on the full version of BOLT Test.

**Reporting:** HTML, XML, Spectacle, and Flash reports can all be created from BOLT Test. This are simply added by running a terminal command similar to "gauge install html-report".

**Browser Support:** BOLT Test can support most common browsers including Chrome, Firefox, IE, and Safari. BOLT Test Lite is restricted to only Chrome support unless utilizing Appium.

**Parallel Testing:** As applications grow, the number of test cases grow very quickly. To cover all the test cases quickly, BOLT Test is able to be run in parallel utilizing Selenium Grid or an Appium server. This feature is only available on the full version of BOLT Test.

**Tags:** Often only smoke tests or tests covering a specific feature are required. To easily manage this, each spec or testcase can be given one or multiple tags. If one or more tags are specified (when running via command line), then only the tests with the given tag are run.

**Jira Bug Creation:** Upon a test failure, BOLT Test is able to automatically generate a bug within Jira if there isn't an open bug already. This helps speed up the process of reporting bugs and fixing them. This feature is only available on the full version of BOLT Test.

**Remote Testing:** It often makes more sense to run automated tests remotely utilizing a service like Sauce Labs, Browserstack, or a custom server/pipeline. This feature is only available on the full version of BOLT Test.

## Bolt Build Features

**AWS:** Template to deploy Bolt Build Components on EC2 Instances.

**Azure:** Template to deploy Bolt Build Components on Azure Instances.

**Docker Cloud:** Bolt Build Template for running on Docker Cloud.

**Openshift** – Use Openshift to manage the Bolt Build components and the CI/CD pipeline.

**Jenkins CI/CD:** Fully configured Pipeline for running tests in any environment and utilizing any agent including 3<sup>rd</sup> party agents such as Sauce Labs and BrowserStack.

**Jenkins Bolt Slaves:** Custom Slaves for executing Bolt tests.

**Selenium Grid** – Selenium Grid Container to manage all browsers

**Selenium Browsers** – Spin as many containers as needed to support distributed testing.

**Jmeter Load Agents** – Spin as many agents as needed to support load tests.

**Docker:** Use Docker to deploy all components in Containers

**Docker Compose** – Templates describing all containers to spin up.

**Sonar:** Add a Step to perform Unit tests and Code Quality checks

**Nexus:** Add local repository for project artifacts

**ZAP OWASP:** Add s Security tests to the pipeline.

**Portainer:** Monitors containers and overall Node health of Bolt Build system.

**HQ Dashboard:** Central location Hub with access to all Bolt Build components as well as managing run data, reports, issues, instrumentation.

**Custom Integrations:** The ability to add additional services.

## Requirement List:

BOLT Test:

Java 8

Gauge

IDE - IntelliJ (recommended) or Visual Studio

Gauge IDE plugin

