

Title: Integration testing of multi-war application in Spring Boot

Post Body:

I have an application composed of multiple maven war projects.

I have another maven project that runs JUnit integration tests against the manually-started tomcat-deployed multi-war application using `org.springframework.web.client.RestTemplate` calls.

However, I'd like my integration test project to actually start my multi-war application (once for the duration of the entire suite) before it runs my tests...in spring-boot!

From my integration test project, I'd like to be able to run all the war projects together as a spring-boot application, each with their own contextPaths (e.g. `localhost:8080/a` for project 'a', `localhost:8080/b` for project 'b', etc.), and without changing the original war projects (that are not (yet) spring-boot aware). If I can't make these projects run from my integration test project in spring-boot without changing them then I'd at least like to minimize the use of spring-boot dependencies and configuration in packaged war files...as much as possible.

I was able to get my integration test project to depend on a single war project, start it up and run tests against it...but I was unsuccessful getting two war projects running together in spring-boot under separate contextPaths.

Any suggestions would be welcome!

Here are some of the resources I've been using to put this together:

(Spring-boot documentation) <http://docs.spring.io/spring-boot/docs/current/reference/html/boot-features-testing.html>

(Blog post touching on starting spring app once for test suite) <http://www.nurkiewicz.com/2010/12/speeding-up-spring-integration-tests.html>

(Suggestions for including war files as dependencies in a integration test project pom)

<http://eureka.ykyuen.info/2009/10/30/maven-dependency-on-jarwar-package/>

Accepted Answer:

As per Andy's suggestion, I went with the Tomcat7 Maven Plugin and it worked just fine. The Jetty Maven Plugin was another option (and better documented IMO) although I couldn't find a way to avoid having to provide a 'path' to my WAR files. The Tomcat7 Maven Plugin, let me load up my WARs from my local .m2 repository. I should also say that the following links were helpful as well...

- <http://cupofjava.de/blog/2013/02/05/integration-tests-with-maven-and-tomcat/>
- <https://stackoverflow.com/a/16936585/1098564>

Here's part of my integration test project pom...

```
<build>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-s
```

Highest Rated Answer: None