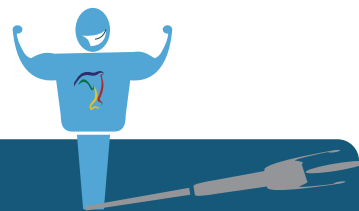


¿Qué ofrece Autentia Real Business Solutions S.L?

Somos su empresa de **Soporte a Desarrollo Informático**.
 Ese apoyo que siempre quiso tener...

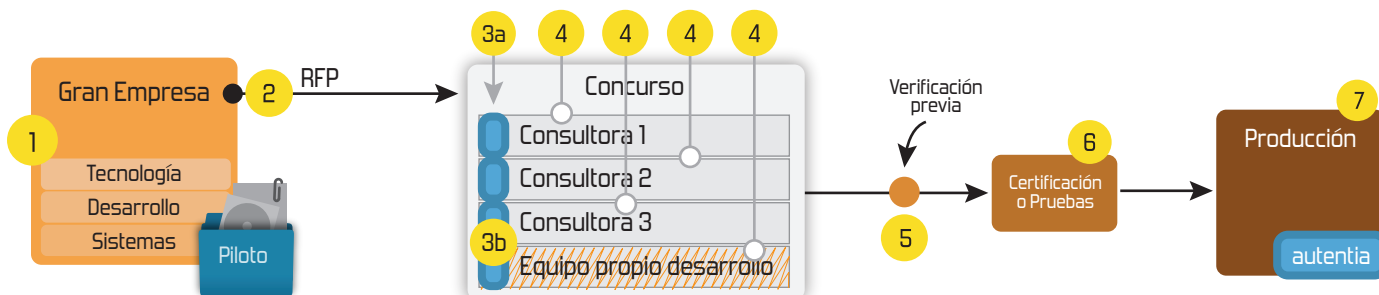
1. Desarrollo de componentes y proyectos a medida



2. Auditoría de código y recomendaciones de mejora

3. Arranque de proyectos basados en nuevas tecnologías

1. Definición de frameworks corporativos.
2. Transferencia de conocimiento de nuevas arquitecturas.
3. Soporte al arranque de proyectos.
4. Auditoría preventiva periódica de calidad.
5. Revisión previa a la certificación de proyectos.
6. Extensión de capacidad de equipos de calidad.
7. Identificación de problemas en producción.



4. Cursos de formación (impartidos por desarrolladores en activo)

Spring MVC, JSF-PrimeFaces /RichFaces,
 HTML5, CSS3, JavaScript-jQuery

Gestor portales (Liferay)
 Gestor de contenidos (Alfresco)
 Aplicaciones híbridas

Tareas programadas (Quartz)
 Gestor documental (Alfresco)
 Inversión de control (Spring)

Control de autenticación y
 acceso (Spring Security)
 UDDI
 Web Services
 Rest Services
 Social SSO
 SSO (Cas)

JPA-Hibernate, MyBatis
 Motor de búsqueda empresarial (Solr)
 ETL (Talend)

Dirección de Proyectos Informáticos.
 Metodologías ágiles
 Patrones de diseño
 TDD

BPM (jBPM o Bonita)
 Generación de informes (JasperReport)
 ESB (Open ESB)

AdictosAlTrabajo

Terrakas 1x05
¡¡Ya está en la web!! :-)
terrakas.com



Entra en Adictos a través de  

E-mail

Contraseña

Entrar

[Deseo registrarme](#)
[Olvidé mi contraseña](#)

[Inicio](#) [Quiénes somos](#) [Formación](#) [Comparador de salarios](#) [Nuestro libro](#) [Más](#)

» Estás en: [Inicio](#) [Tutoriales](#) Lanzando nuestros tests de jasmine-node con IntelliJ IDEA



Alejandro Pérez García

Alejandro es socio fundador de Autentia y nuestro experto en J2EE, Linux y optimización de aplicaciones empresariales.

Ingeniero en Informática y Certified ScrumMaster

[Seguir a @alejandropgarcia](#) 777 seguidores

Si te gusta lo que ves, puedes contratarle para darte ayuda con soporte experto, impartir **cursos presenciales** en tu empresa o para que **realicemos tus proyectos como factoría** (Madrid). Puedes encontrarme en [Autentia](#): Ofrecemos servicios de soporte a desarrollo, factoría y formación.

[Ver todos los tutoriales del autor](#)



Fecha de publicación del tutorial: 2012-12-26

Tutorial visitado 1 veces [Descargar en PDF](#)

Lanzando nuestros tests de jasmine-node con IntelliJ IDEA

Creación: 23-12-2012

Índice de contenidos

1. Introducción
2. Entorno
3. Instalando los plugins
4. Creando un proyecto web con IntelliJ
5. Ejecutando los tests de Jasmine con jasmine-node desde IntelliJ IDEA 12
- 5.1. Ejecutando los tests bajo demanda
- 5.2. Ejecutando los tests automáticamente
6. Conclusiones
7. Sobre el autor

1. Introducción

En el tutorial "[Hello Jasmine! Primeros pasos para hacer BDD/TDD con JavaScript](#)" vimos distintas maneras de lanzar los tests de Jasmine. En esta ocasión vamos a ver como integrar dentro de **IntelliJ IDEA 12** la ejecución de los tests de Jasmine con jasmine-node.

2. Entorno

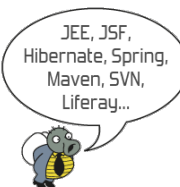
El tutorial está escrito usando el siguiente entorno:

- Hardware: Portátil MacBook Pro 15' (2.5 GHz Intel i7, 8GB 1333 Mhz DDR3, 256GB Solid State Drive).
- AMD Radeon HD 6770M 1024 MB
- Sistema Operativo: Mac OS X Lion 10.8.2
- IntelliJ IDEA 12
- Node.js v0.8.15

3. Instalando los plugins

IntelliJ es un IDE que podemos extender a base de plugins en función de nuestras necesidades. En esta ocasión vamos a instalar el **plugin de Node.js** y el plugin de jsTestDrive.

Catálogo de servicios Autentia



Síguenos a través de:



Últimas Noticias

» AdictosAlTrabajo os desea
¡¡¡FELICES FIESTAS!!!

» **ESTRENO** de Terrakas
1x05: "Un lugar en el mundo"

» Mi retrospectiva sobre la
CAS 2012

» Autentia estuvo en el
Duatlón de Torrejón de
Ardoz

» Participamos en la Carrera
de las Empresas 2012

[Histórico de noticias](#)

Últimos Tutoriales

» Maquetación de un libro
con Adobe InDesign

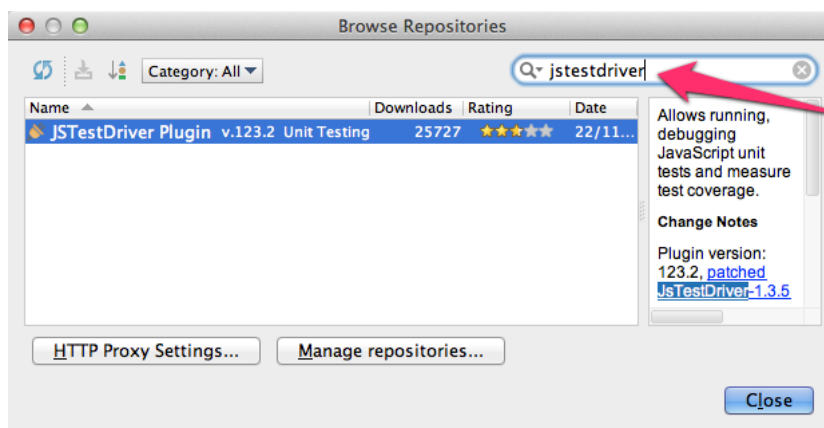
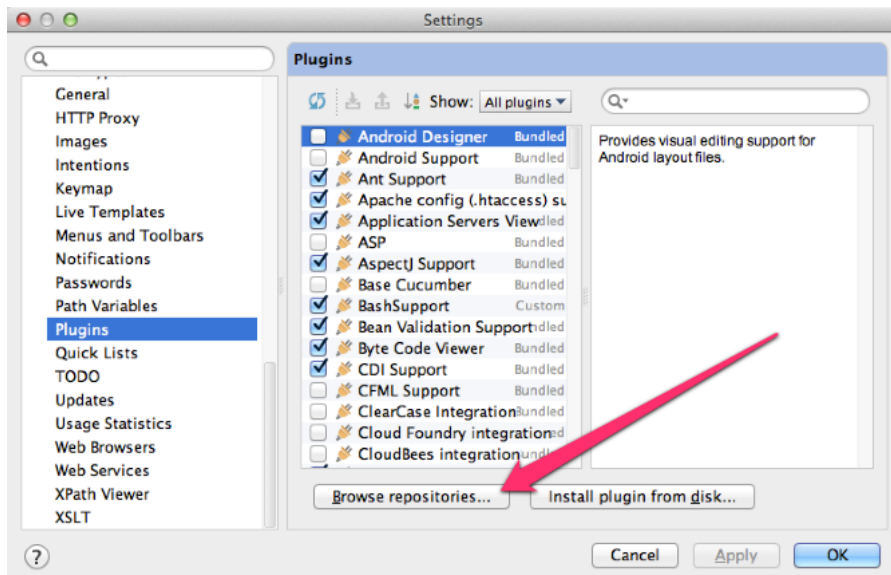
» Maquetación de un
documento con Adobe
InDesign

» Sonar y Total Quality:
midiendo la calidad total de
nuestros proyectos

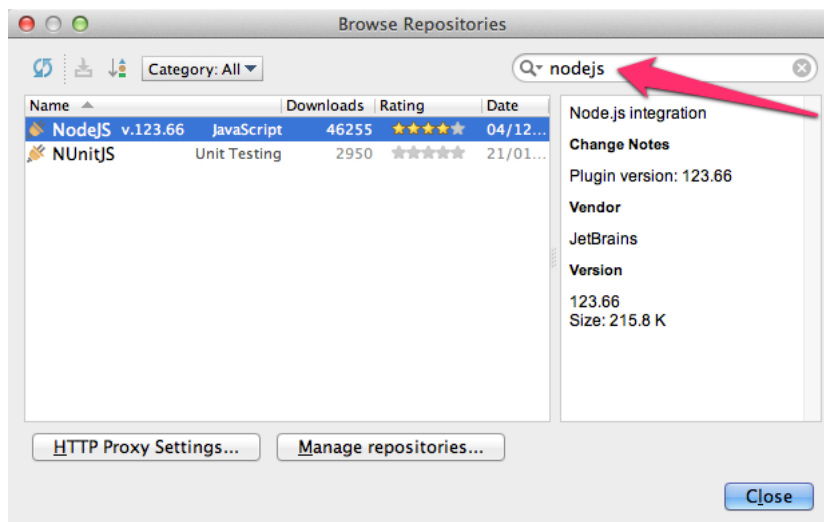
» Tutorial de Responsive
Design

» Hello Jasmine! Primeros
pasos para hacer BDD/TDD
con JavaScript

Para ello vamos a *IntelliJ IDEA* --> *Preferences...* --> *Plugins*



El plugin de **JSTestDriver** nos va a permitir lanzar test con varios frameworks de JavaScript, incluido Jasmine, en distintos navegadores; y también nos va a permitir el auto completado cuando estemos escribiendo los tests.



Gracias al plugin de **Node.js** vamos a poder lanzar ejecuciones de node desde dentro de nuestro IntelliJ, haciendo más fácil el trabajo con los tests, ya que no tendremos que estar cambiando constantemente a la consola (como nos pasaba en el tutorial que hemos mencionado en la introducción).

4. Creando un proyecto web con IntelliJ

IntelliJ es un IDE especialmente pensado para Java, pero también nos permite trabajar de forma impecable con HTML, CSS, y JavaScript. Para crear un proyecto web sencillo hacemos *File* --> *New Project...*

Últimos Tutoriales del Autor

» Hello Jasmine! Primeros pasos para hacer BDD/TDD con JavaScript

» Cómo instalar Java7 en Mac OS X

» Cómo usar el GPS en nuestras aplicaciones iOS 5 (iPhone, iPad, ...)

» Como internacionalizar nuestras aplicaciones iOS 5 (iPhone, iPad, ...)

» TDD con PHP gracias a PHPUnit

Últimas ofertas de empleo

2011-09-08
Comercial - Ventas - MADRID.

2011-09-03
Comercial - Ventas - VALENCIA.

2011-08-19
Comercial - Compras - ALICANTE.

2011-07-12
Otras Sin catalogar - MADRID.

2011-07-06
Otras Sin catalogar - LUGO.



Alejandro Pérez
alejandropgarci

alejandropgarci @Ananada7
estáis riquísimos!!! Feliz Navidad
y pasarlo muy bien jovencitos!!!
:D
yesterday · reply · retweet · favorite

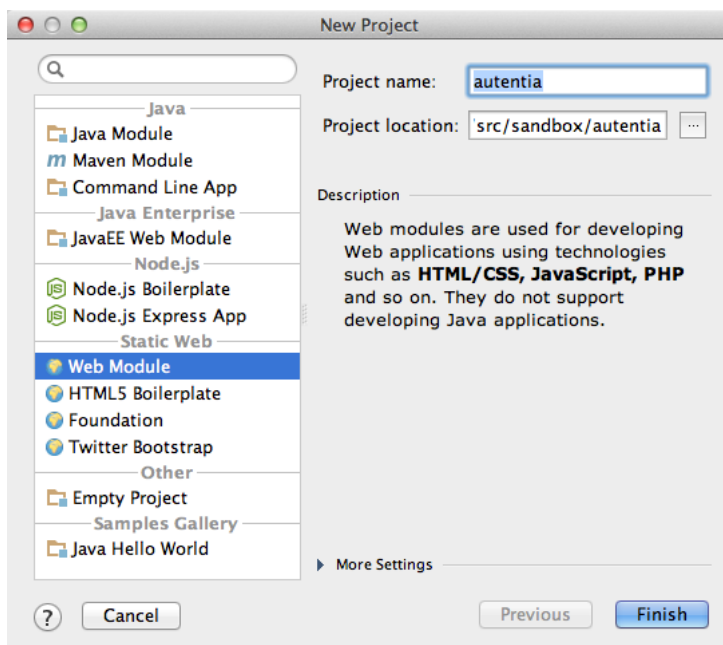
alejandropgarci @albertovilches
juas ya te digo. Hay puertas que
es mejor no abrir!!! :P Feliz
Navidad !!!!!!!!!!!
yesterday · reply · retweet · favorite

AutomatedTester Code review
pic.twitter.com/5m79lkj7
yesterday · reply · retweet · favorite

pilonway - Me estás siendo infiel,
a mi no me engañas - Te
contradices.
9 days ago · reply · retweet · favorite



Join the conversation

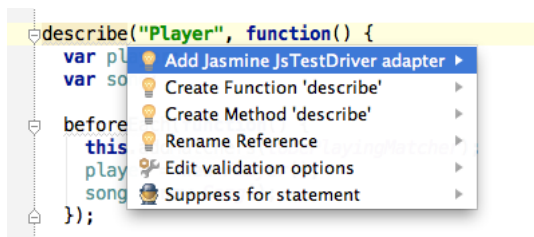


Ahora nos copiamos los ficheros del ejemplo con **jasmine-node** del tutorial [Hello Jasmine! Primeros pasos para hacer BDD/TDD con JavaScript](#). Además también crearemos un directorio **lib** para guardar otras librerías de JavaScript.

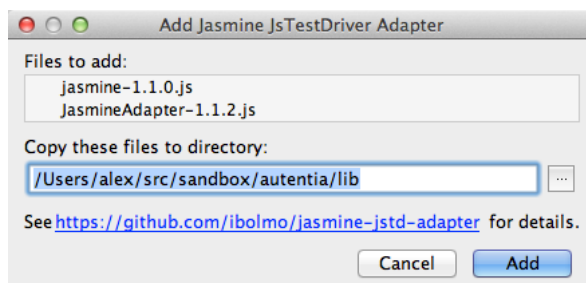
Si, por ejemplo, abrimos el fichero **player.spec.js**, podemos ver que no estamos consiguiendo auto completado de la librería de Jasmine



Basta con ponernos encima y hacer **Alt + Intro** y nos aparecerá un menú donde podemos indicar que queremos añadir el adaptador de Jasmine para lanzar los tests, conseguir auto completado, ...



Nos preguntará que dónde queremos guardar las librerías para el adaptador. Le indicamos el directorio **lib** que habíamos creado anteriormente (ojo porque estas librerías no deberían llegar nunca a producción, son simplemente para lanzar los tests y facilitarnos la vida en el entorno de desarrollo).



Si ahora nos volvemos a fijar en el código fuente deberíamos ver que si nos está reconociendo las funciones de Jasmine, y que nos funciona perfectamente el auto completado.



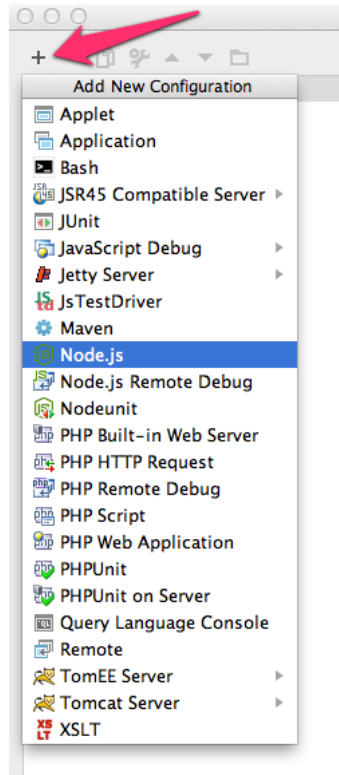
```
1 var Player = require('../src/Player');
2 var Song = require('../src/Song');
3 var toBePlayingMatcher = require('./toBePlayingMatcher');
4
5 describe("Player", function() {
6     var player;
7     var song;
8
9     beforeEach(function() {
10         this.addMatchers(toBePlayingMatcher);
11         player = new Player();
12         song = new Song();
13     });
14 });
```

5. Ejecutando los tests de Jasmine con jasmine-node desde IntelliJ IDEA 12

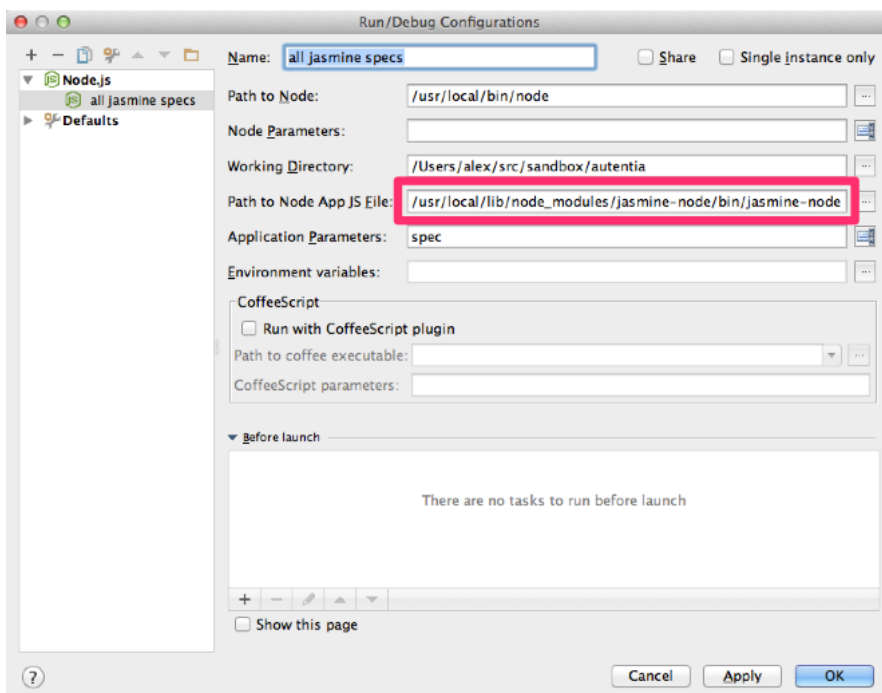
5.1. Ejecutando los tests bajo demanda

En lugar de usar el JSTestDrive, que nos levantaría un navegador, vamos a lanzar los test usando directamente jasmine-node. Además esto tiene la ventaja que podemos usar el modo *autotest* para que los tests se lancen continuamente de forma automática en cuanto hagamos cualquier cambio en cualquier fichero (lo veremos en el punto siguiente).

Para ello hacemos *Run --> Edit Configurations...* y damos sobre el + para ver las opciones que tenemos disponibles, seleccionaremos Node.js

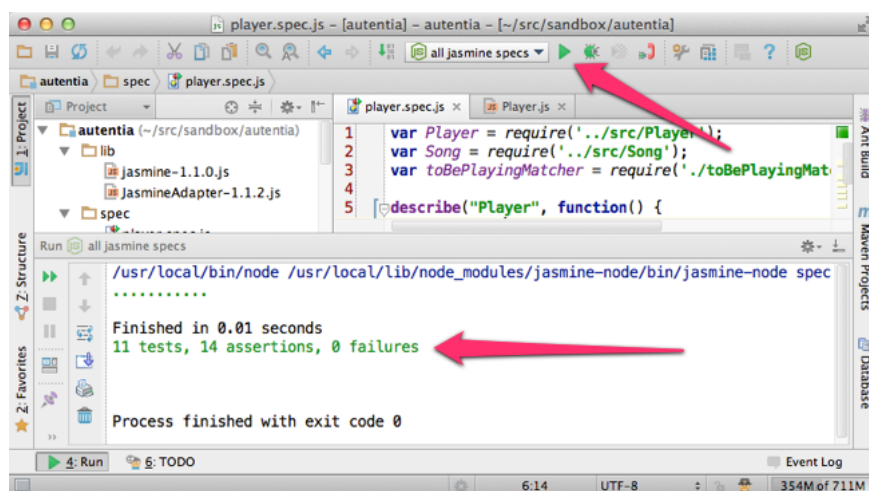


Ahora rellenamos las opciones tal como se ve en la imagen.



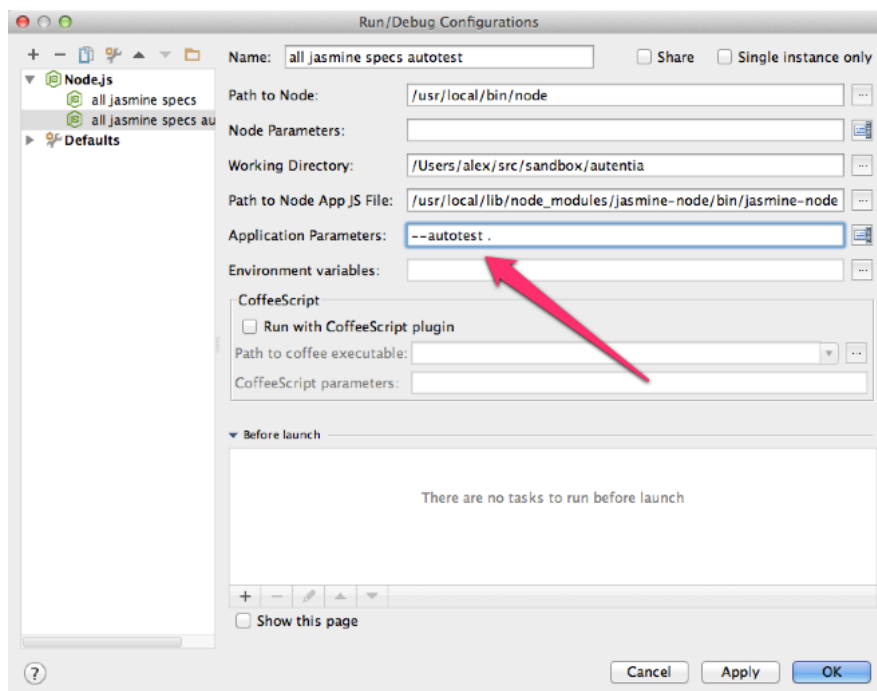
Se puede ver como hemos indicado donde se encuentra el ejecutable de node, así como el script de jasmine-node que vamos a ejecutar. También podemos ver como indicamos como parámetro de esta aplicación el directorio **spec**, que será donde se buscarán los tests que hay que ejecutar (esto nos permite jugar con la estructura de directorios para crearnos varias configuraciones y así lanzar sólo los tests que nos interesa en cada momento).

Ahora basta con pulsar sobre el botón de *play* y veremos como se ejecutan los tests.



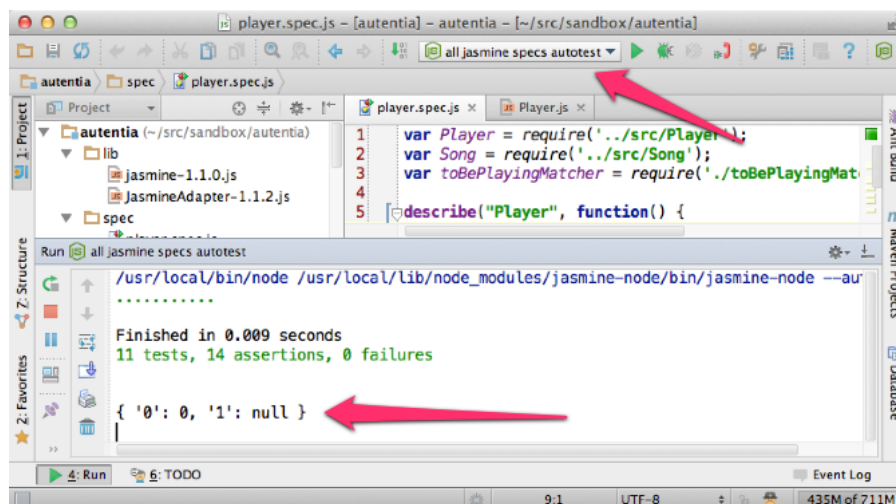
5.2. Ejecutando los tests automáticamente

Vamos a crear un nuevo **runner** para ejecutar los test automáticamente. Tan sólo tenemos que cambiar la configuración tal como se muestra en la imagen.

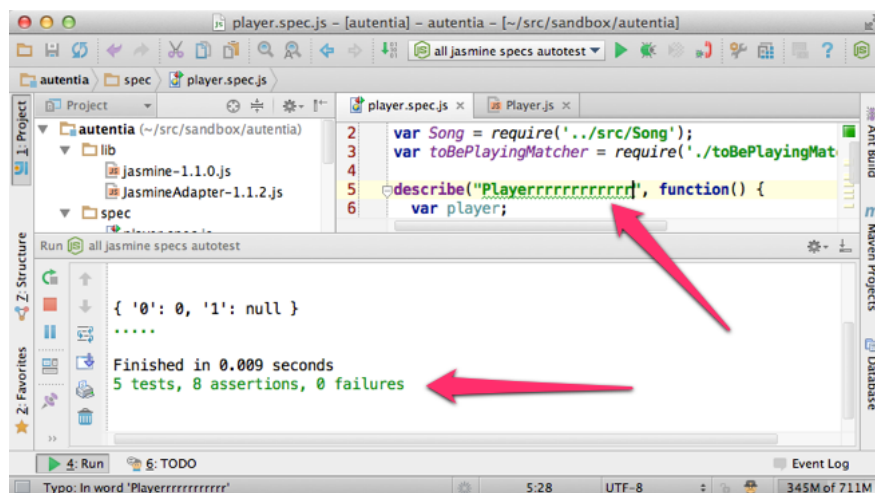


Se ve como hemos cambiado los parámetros que le pasamos a jasmine-node para indicar que queremos que esté continuamente comprobado si hay cambios en los ficheros. Al indicar . como el directorio a partir del cual hay que comprobar si cambia algo, conseguimos que se lancen los tests tanto si cambiamos los ficheros con los specs, como si cambiamos el código de producción.

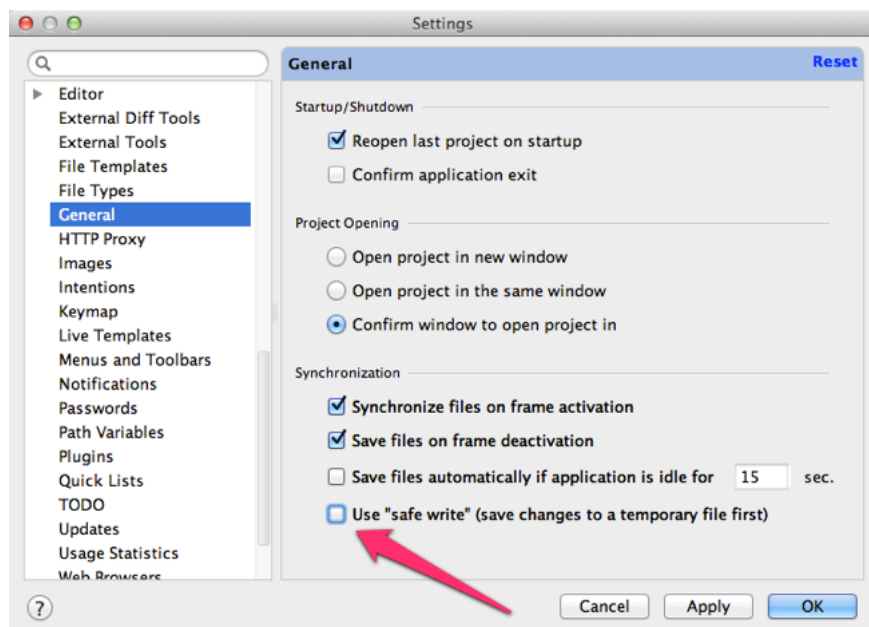
Ahora si damos al play para este runner, veremos algo como:



Se ve como se han lanzado los tests, pero el proceso no ha terminado, está esperando. Si ahora cambiamos un fichero y guardamos los cambios, veremos como se de forma inmediata se vuelven a ejecutar los tests.



Si vemos que esto no está funcionando correctamente y que parece que cuando guardamos los cambios no se lanzan los tests de forma automática, conviene que revisemos la configuración IntelliJ IDEA --> Preferences... --> General --> Use "safe write", para ver que está deshabilitada.



6. Conclusiones

Ya sabéis que es fundamental conocer bien las herramientas con las que trabajamos a diario para sacarles el mayor partido. En esta ocasión hemos visto como configurar IntelliJ IDEA 12 para poder lanzar cómodamente los tests de Jasmine, sin necesidad de cambiar a la consola de comandos.

7. Sobre el autor

Alejandro Pérez García, Ingeniero en Informática (especialidad de Ingeniería del Software) y Certified ScrumMaster
Socio fundador de Autentia (Desarrollo de software, Consultoría, Formación)

<mailto:alejandropg@autentia.com>

Autentia Real Business Solutions S.L. - "Soporte a Desarrollo"

<http://www.autentia.com>

A continuación puedes evaluarlo:

[Regístrate para evaluarlo](#)

Por favor, vota +1 o compártelo si te pareció interesante

[Share](#) |

[0](#)

Anímate y coméntanos lo que pienses sobre este **TUTORIAL**:

» [Regístrate](#) y accede a esta y otras ventajas «



Esta obra está licenciada bajo licencia [Creative Commons de Reconocimiento-No comercial-Sin obras derivadas 2.5](#)

IMPULSA

Impulsores

Comunidad

¿Ayuda?

sin clicks

0 personas

han traído clicks a esta página

+

+

+

+

+

+

+

+

powered by

karmacracy