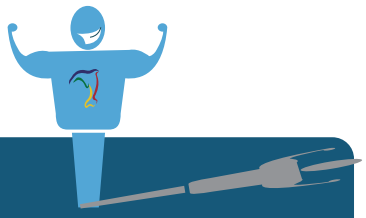
autentia

Avenida de Castilla,1 - Edificio Best Point - Oficina 21B
28830 San Fernando de Henares (Madrid)
tel./fax: +34 91 675 33 06
info@autentia.com - **www.autentia.com**

¿**Qué ofrece** Autentia Real Business Solutions S.L?

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

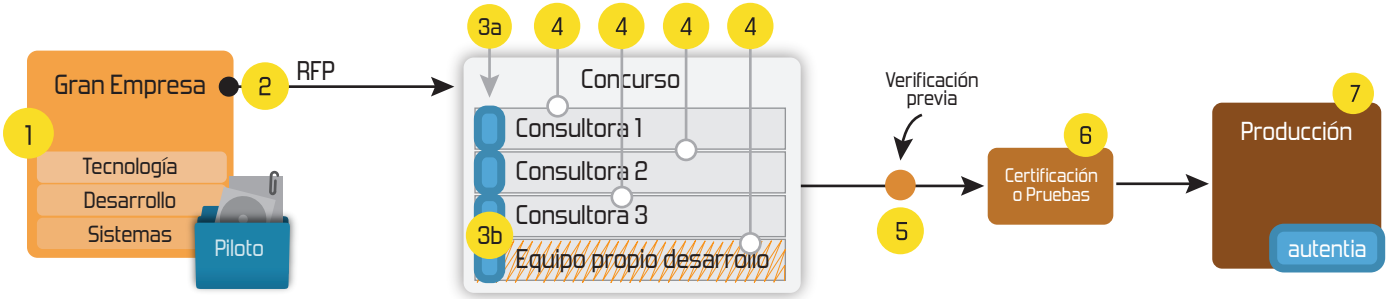
1. Desarrollo de componentes y proyectos a medida



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

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

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

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

Compartimos nuestro conocimiento en:
www.adictosaltrabajo.com

Para más información visítenos en:
www.autentia.com

DESARROLLADO POR:
Jose Manuel Sánchez Suárez

Consultor tecnológico de desarrollo de proyectos informáticos.

Puedes encontrarme en Autentia: Ofrecemos servicios de soporte a desarrollo, factoría y formación

Somos expertos en Java/J2EE

Fecha de publicación del tutorial: 2011-02-07

19

Share | Regístrate para votar

Eclipse custom templates.

0. Índice de contenidos.

1. Introducción

Las plantilas en Eclipse son pequeñas piezas de código con marcadores predefinidos que nos permiten no repetir la escritura de código. La idea es no tener que escribir un código relativamente repetitivo (como puede ser la signatura de un bucle) o no tener que recordar la sintaxis de una pieza de código más o menos compleja.

No se trata de no aprender u olvidar cómo se escribe un sentencia foreach, se trata de que:

- me la se de memoria porque llevo años escribiéndola y quiero que, con una simple combinación de teclas, el IDE (hoy toca Eclipse) la escriba para mi,
- estoy aprendiendo y quiero tener algo que me facilite su escritura, a base de verla repetidas veces, terminaré memorizando su sintaxis.

El objetivo de las plantillas es ser más productivo y, con menos esfuerzo, escribir más código.

Cada plantilla tiene un nombre, basta con escribir ese nombre en el código y pulsar CTRL + espacio y el código de la plantilla se expanderá.

2. Entorno.

El tutorial está escrito usando el siguiente entorno:

- Hardware: Portátil MacBook Pro 17' (2.93 GHz Intel Core 2 Duo, 4GB DDR3 SDRAM).
- Sistema Operativo: Mac OS X Snow Leopard 10.6.1
- Eclipse 3.6 (Helios)

3. Plantilla predefinidas.

Eclipse viene provisto de una serie de plantillas predefinidas que podemos consultar y modificar accediendo, dentro de la opción preferencias, al lenguaje con el que trabajemos, dentro de su opción Editor > templates, así:

- Preferences > XML > XML Files > Editor > Templates
- Preferences > Java > Editor > Templates

CatÃ¡logo de servicios Autentia

Últimas Noticias

- XIII Charla Autentia - AOS y TDD - Vídeos y Material
- Las metodologías ágiles como el catalizador del cambio
- XIV Charla Autentia - ZK
- Informática profesional: Las reglas no escritas para triunfar en la empresa. 2ª EDICIÓN ACTUALIZADA.
- Pequeño coding dojo con Carlos Ble en las oficinas de Autentia.

Histórico de NOTICIAS

Últimos Tutoriales

- Haaala! para Android: Facebook y Email con Voz
- Librería de acceso a datos con Spring y JPA
- Como editar XML o HTML con el plugin xmledit de Vim
- Notificación de eventos en Nut
- Cosas que no funcionan en JSF2 como uno podría esperar (es decir, bugs)

Últimos Tutoriales del Autor

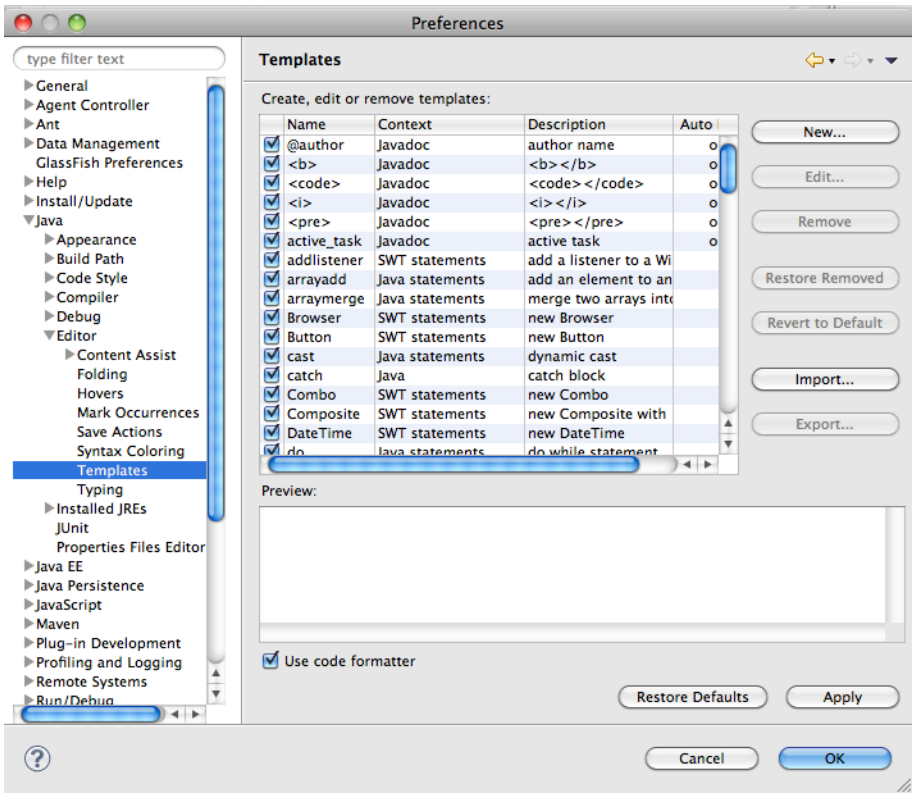
- Zen-coding: una nueva forma de escribir código HTML
- IAQ (Interesting Asked Questions), implementado una interfaz SPI con jQuery
- Google Chrome Developer Toolbar.
- Introducción a HTML5.
- IAQ (Interesting Asked Questions), recordando la posición del scroll con el soporte de jQuery.

Síguenos a través de:

Últimas ofertas de empleo

2010-10-11
Comercial - Ventas - SEVILLA.

2010-08-30
Otras - Electricidad -

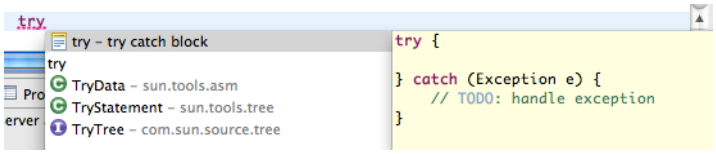


Una pantilla se compone de:

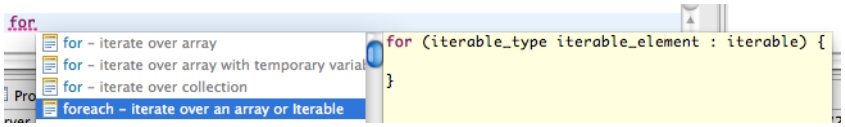
- un nombre,
- una descripción,
- un contexto en función del lenguaje (en java, si estamos en el código, en el javadoc, ...) y
- un pattern, que es el código de la plantilla. Dentro del código de la plantilla podemos usar texto fijo o una serie de variables predefinidas, por ejemplo:
  - ${cursor}: posición en la que se establecerá el cursor de texto tras desplegar el código de la plantilla,
  - ${enclosing_type}: tipo de la clase en la que nos encontramos, ${enclosing_method}: nombre del método en el que nos encontramos,
  - ${todo}: TODO: task code,
  - ${year}: año en curso, ${time}: hora en curso,
  - ...

Como hemos comentado, esas plantillas se mostrarán como sugerencias en el código tras comenzar a escribir su nombre y pulsar CTRL + espacio.

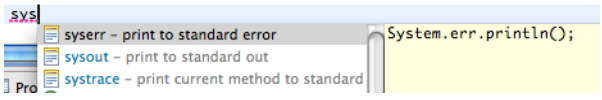
Ejemplo con try



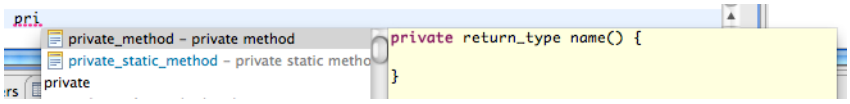
Ejemplo con for



Ejemplo con sys



Ejemplo con private

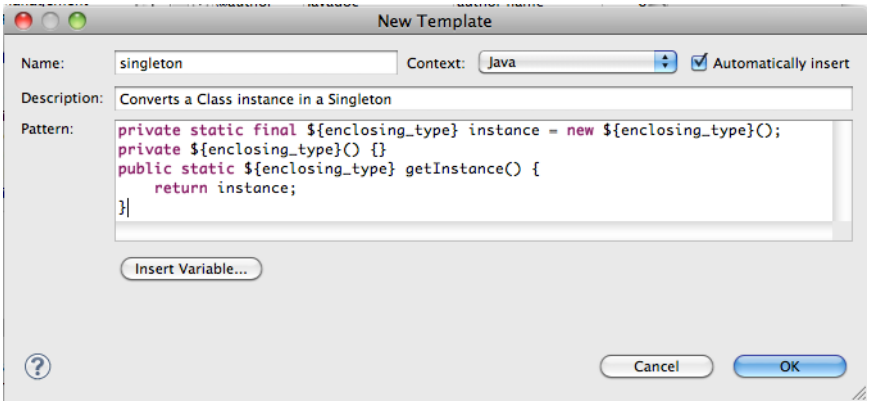


Basta con seleccionar la plantilla y pulsar INTRO para que el código de la plantilla se añada a nuestro código y si requiere de la introducción de algún tipo de texto se posicionará en aquella parte del código que lo requiera.

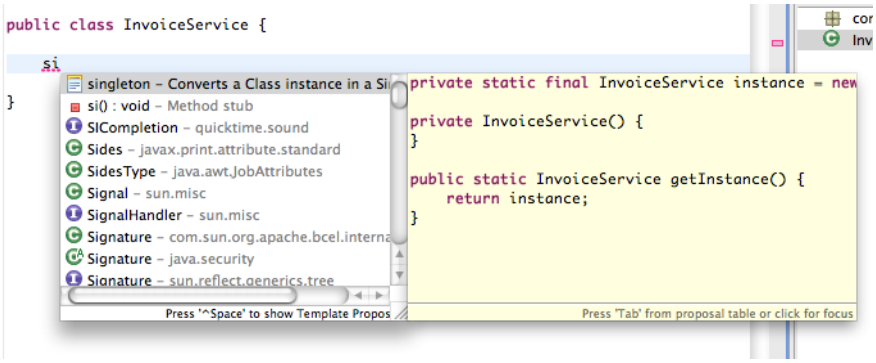
4. Creación de una plantilla personalizada.

Lo más interesante es que nosotros podemos crearnos nuestras propias plantillas, además de modificar las existentes. Para ello no tenemos más que añadir una nueva desde la opción de Templates, asignarle un nombre, descripción y elegir el código que queremos que se muestre al seleccionar la misma.

El siguiente es un ejemplo de un singleton:



Desde código no tenemos más que introducir las primeras letras del nombre de la plantilla y pulsar CTRL + espacio:



Las plantillas, al igual que el formateador del estilo de código se pueden importar de un fichero externo de modo que cada empresa puede tener las suyas propias, en el siguiente enlace podéis encontrar un ejemplo: http://sproutee.com/mytemplates.xml

Del mismo modo la configuración de las plantillas se almacena en el workspace, con lo que si cambias o creas un workspace nuevo se perderán. De ahí la necesidad de disponer de esos archivos xml para importarlos.

5. Un par de plantillas muy últiles.

A continuación un par plantillas que pueden sernos de gran utilidad:

singleton - Converts a Class instance in a singleton

```
private static final ${enclosing_type} instance = new ${enclosing_type}();

private ${enclosing_type}() {}

public static ${enclosing_type} getInstance() {
  return instance;
}
```

La que hemos visto en el ejemplo de creación de una nueva plantilla.

logger - create new Logger

```
${:import(org.apache.log4j.Logger)}
private static final Logger logger = Logger.getLogger(${enclosing_type}.class.getName());
```

La creación de una instancia del logger que usemos es origen de un bug conocido, tiene incluso una regla de PMD, porque lo más común es no escribir el código del mismo sino copiar y pegar el código de otra clase **sin cambiar el parámetro del método getLogger()**, con lo que logamos para la clase de la que copiamos el código no la nuestra. Con esta plantilla evitamos ese error.

6. Referencias.

- http://eclipse.dzone.com/news/effective-eclipse-custom-templ
- http://snippets.dzone.com/posts/show/3618
- http://stackoverflow.com/questions/1028858/useful-eclipse-java-code-templates
- http://www.ibm.com/developerworks/opensource/tutorials/os-eclipse-code-templates/index.html

7. Conclusiones.

Cualquiera que desarrolle software necesita un medio probado para producir mejor, más rápido y más barato (The Productive Programmer). Conocer las características y facilidades de las herramientas que usamos día a día es nuestra responsabildiad.

Un saludo.

Jose

jmsanchez@autentia.com

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

Puedes opinar o comentar cualquier sugerencia que quieras comunicarnos sobre este tutorial; con tu ayuda, podemos ofrecerte un mejor servicio.

Enviar comentario

(Sólo para usuarios registrados)

» **Registrate** y accede a esta y otras ventajas «

COMENTARIOS