

¿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)



Hosting patrocinado por **enredados**

- [Inicio](#)
- [Quienes somos](#)
- [Tutoriales](#)
- [Formación](#)
- [Empleo](#)
- [Colabora](#)
- [Comunidad](#)
- [Libro de Visitas](#)
- [Comic](#)

NUEVO ¿Quieres saber cuánto ganas en relación al mercado? pincha **aquí...**

[Ver cursos que ofrece Autentia](#)

[Descargar comics en PDF y alta resolución](#)



Estamos escribiendo un libro sobre la profesión informática y estas viñetas formarán parte de él. Puedes opinar en la seccion [comic](#).

Catálogo de servicios Autentia (PDF 6,2MB)

[En formato comic...](#)

Google

Web
 www.adictosaltrabajo.com

Últimos tutoriales

- 2008-06-14 [Hibernate Validator, y como definir las validaciones sobre los objetos de negocio](#)
- 2008-06-13 [Toda la potencia de un buscador como Google en tu base de datos, gracias a Hibernate Search](#)
- 2008-06-09 [Arquetipos de maven: cómo crear, distribuir y generar proyectos con JSF e ICEfaces, JBoss y EJB3](#)
- 2008-06-04 [Desarrollos Web en PHP con AppServ 2.5.6 y Eclipse PDT](#)
- 2008-06-02 [Ficheros de mapeo de Hibernate desde las clases](#)
- 2008-06-02 [Un vistazo a Gantt Project](#)
- 2008-05-29 [Manejar presentaciones con UNO](#)
- 2008-05-25 [Composición de música con TUXGUITAR](#)
- 2008-05-14 [Spring + Hibernate + Anotaciones = Desarrollo Rápido en Java](#)
- 2008-05-06 [J2ME. Internacionalización de aplicaciones para móviles](#)

Tutorial desarrollado por



Alejandro Pérez García

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

Ingeniero en Informática

Si te gusta lo que ves, puedes contratarle para impartir **cursos presenciales** en tu empresa o para **ayudarte en proyectos** (Madrid). Puedes encontrarme en [Autentia](#)

Catálogo de servicios de Autentia

[Descargar \(6,2 MB\)](#)

[Descargar en versión comic \(17 MB\)](#)

[AdictosAlTrabajo.com](#) es el Web de difusión de conocimiento de [Autentia](#).



[Catálogo de cursos](#)

Descargar este documento en formato PDF: [hibernateValidator.pdf](#)

Fecha de creación del tutorial: **2008-06-14**

Hibernate Validator, y como definir las validaciones sobre los objetos de negocio

Creación: 14-06-2008

Índice de contenidos

1. Introducción
2. Entorno
3. Configuración del entorno de desarrollo
4. Validaciones predefinidas
5. Añadiendo validaciones a nuestro POJO
6. Ejemplos de como se valida nuestro POJO
7. Recursos
8. Conclusiones
9. Sobre el autor

1. Introducción

Cuando construimos aplicaciones, lo normal es que tengamos que hacer validaciones de las entradas del usuario. Frameworks como JSF o Struts nos suelen proporcionar ayudas para que podamos definir más o menos de manera fácil estas validaciones en la vista.

Los problemas que tiene esto son:

- Nos podemos encontrar repitiendo las mismas validaciones en diferentes formularios de nuestra aplicación que tratan con los mismos objetos de negocio.
- Si luego queremos llamar a negocio desde otro sitio que no sea nuestra aplicación web (por ejemplo si montamos un servicio web para acceder a la misma lógica), tendremos que repetir estas validaciones en el servicio web.

Es decir, estamos rompiendo el principio DRY (Don't repeat yourself - no te repitas), ya que nos vemos escribiendo las mismas validaciones una y otra vez.

Hibernate Validator (<http://www.hibernate.org/412.html>) pretende resolver este problema, de forma que definiremos una única vez las validaciones en nuestros objetos de negocio (en nuestros POJOS), e invocaremos esas validaciones desde el punto que nos interese.

- [Anuncios Google](#)
 [Tutoriales 3D](#)
 [Manual Project](#)
 [Hibernate Libraries](#)
 [Java Protect Code](#)
 [Hibernate ORM](#)

Últimas ofertas de empleo

Principales características de Hibernate Validator:

- Definimos las las validaciones muy fácilmente con anotaciones (es posible anotar tanto los atributos como los getters).
- Trae un conjunto predefinido de validaciones típicas, conjunto que podemos extender fácilmente con nuestras propias validaciones.
- El sistema soporta internacionalización: Ya trae mensajes de error traducidos a diez idiomas. Estos mensajes los podemos cambiar fácilmente, simplemente escribiendo nuestro propio fichero de propiedades y sobrescribiendo los mensajes que nos interese.
- Se integra directamente con Hibernate, y en general con cualquier OR Mapping, de forma que antes de hacer una inserción o actualización se validarán los objetos.
- Si usamos Hibernate, las validaciones que indiquemos se tendrán en cuenta a la hora de generar el DDL (los scripts de creación de la base de datos).

2008-06-13
Comercial - Ventas -
VALENCIA.

2008-06-12
T. Información - Analista
Funcional (otros) - MADRID.

2008-06-11
T. Información - Técnico de
Soporte (Help Desk) -
MALAGA.

2008-06-09
Otras Sin catalogar -
ASTURIAS.

2008-06-07
T. Información - Otros no
catalogados - MADRID.

Anuncios Google

2. Entorno

El tutorial está escrito usando el siguiente entorno:

- Hardware: Portátil Asus G1 (Core 2 Duo a 2.1 GHz, 2048 MB RAM, 120 GB HD).
- Nvidia GEFORCE GO 7700
- Sistema Operativo: GNU / Linux, Debian (unstable), Kernel 2.6.24, KDE 3.5
- Java Sun 1.6.0_06
- Hibernate 3.2.6.ga
- Hibernate Validator 3.0.0 GA
- JUnit 4.4

3. Configuración del entorno de desarrollo

La configuración del entorno, como siempre, la vamos a hacer con Maven. De esta forma resulta tan sencillo como añadir a nuestro pom.xml las siguientes líneas:

```

01. view plain print ?
02. ...
03. <repositories>
04.     ...
05.     <repository>
06.         <id>repository.jboss.org</id>
07.         <name>JBoss Maven Repository</name>
08.         <url>http://repository.jboss.org/maven2</url>
09.         <layout>default</layout>
10.     </repository>
11.     ...
12. </repositories>
13. <dependencies>
14.     ...
15.     <dependency>
16.         <groupId>org.hibernate</groupId>
17.         <artifactId>hibernate-validator</artifactId>
18.         <version>3.0.0.GA</version>
19.     </dependency>
20.     ...
21. </dependencies>
22. ...

```

4. Validaciones predefinidas

Para ver las validaciones que Hibernate Validator trae "out of the box", lo mejor es ir directamente a la documentación: http://www.hibernate.org/hib_docs/validator/reference/en/html/validator-defineconstraints.html#validator-defineconstraints-builtin

5. Añadiendo validaciones a nuestro POJO

Ahora vamos a crear un POJO y le pondremos un par de validaciones:

```

view plain print ?
01. @Entity
02. public class Book {
03.
04.     @Id
05.     @GeneratedValue
06.     private Integer id;
07.
08.     @NotEmpty
09.     private String title;
10.
11.     private String summary;
12.
13.     @Past
14.     @Temporal(TemporalType.TIMESTAMP)
15.     private Date publicationDate;
16.
17.     ...
18.     // Resto de constructores, métodos, getter y setters, etc
19.     ...

```

Vamos a comentar exclusivamente las anotaciones de Hiberante Validator:

- Línea 8, `@NotEmpty`: Esta anotación indica que el título no puede ser `null` ni una cadena vacía.
- Línea 13, `@Past`: Indica que la fecha tiene que estar en el pasado con respecto al momento actual.

6. Ejemplos de como se valida nuestro POJO

Vamos a ver un fragmento de código de un test donde se hacen varias comprobaciones sobre nuestro POJO:

```

view plain print ?
01. @Test
02. public void testHibernateConstraints() {
03.     final Calendar calendar = Calendar.getInstance();
04.     calendar.set(3008, Calendar.FEBRUARY, 28);
05.     final Book book = new Book("", "lo mejor del ocultismo", calendar.getTime());
06.     try {
07.         dao.persist(book);
08.         Assert.fail();
09.     } catch (InvalidStateException e) {
10.         assertEquals(2, e.getInvalidValues().length);
11.     }
12.
13.     ClassValidator<Book> validator = new ClassValidator<Book>(Book.class);
14.
15.     InvalidValue[] invalidValues = validator.getInvalidValues(book);
16.     assertEquals(2, invalidValues.length);
17.
18.     invalidValues = validator.getInvalidValues(book, "publicationDate");
19.     assertEquals(1, invalidValues.length);
20.
21.     invalidValues = validator.getPotentialInvalidValues("publicationDate", calendar.getTime());
22.     assertEquals(1, invalidValues.length);
23.
24.     invalidValues = validator.getPotentialInvalidValues("title", "");
25.     assertEquals(1, invalidValues.length);
26.
27. }

```

- Línea 5: Se ve como creamos un libro con un título vacío y con fecha en el futuro.
- Línea 6 a 11: Persistimos el libro con Hibernate. Esto debería hacer saltar la validación, por lo que nunca deberíamos alcanzar la línea 8. Cuando fallan las validaciones se lanzará una excepción de tipo `org.hibernate.validator.InvalidStateException` (capturada en la línea 9). Esta excepción contiene toda la información sobre que es lo que ha fallado. Finalmente en la línea 10 se ve como se comprueba que hay dos valores inválidos: el título no puede ser vacío, y la fecha tiene que estar en el pasado.
- Línea 13: Se ve como se crea un `org.hibernate.validator.ClassValidator`. Esta clase es la que utilizaremos para llamar al sistema de validaciones desde cualquier sitio que nos interese (no sería necesario si sólo se hacen las validaciones por Hibernate al insertar o actualizar un objeto en la base de datos).
- Línea 15 en adelante: Se hacen varias comprobaciones. Cabe destacar especialmente el uso del método `getPotentialInvalidValues` en las líneas 20 y 23. Este método lo que hace es validar si a un campo determinado se le puede aplicar un valor dado. Es decir, la validación es antes de que se aplique el valor (si os fijáis no se hace la validación con una instancia concreta de libro, simplemente se está preguntando si un determinado valor puede ser potencialmente peligroso).

JBoss Seam (<http://www.jboss.com/products/seam>) ya incluye la integración de este tipo de validaciones en nuestras páginas JSF (<http://docs.jboss.com/seam/2.1.0.A1/reference/en/html/validation.html>). Hacer lo mismo en cualquier framework de JSF sería bastante sencillo. Simplemente habría que hacernos un pequeño componente que invocara a las validaciones del estilo mostrado en las líneas 20 y 23.

7. Recursos

Si queréis probar los ejemplos, en el tutorial sobre Hibernate Search (<http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=hibernateSearch>) encontraréis un proyecto de ejemplo donde podréis integrar muy fácilmente el código que es han mostrado aquí.

8. Conclusiones

Siempre tenemos que intentar no repetir cosas, y mucho más si ya están inventadas. En este caso, Hibernate Validator puede ser una muy buena opción para gestionar las validaciones sobre nuestros POJOs.

Ya sabéis, en Autentia (www.autentia.com) siempre os recomendamos el uso de librerías y estándares. Siempre es mucho más barato (en coste y esfuerzo) integrar que desarrollar. Por eso siempre es muy recomendable a la hora de empezar un proyecto, buscar a ver que podemos encontrar que nos haga la vida un poco más fácil.

9. Sobre el autor

Alejandro Pérez García, Ingeniero en Informática (especialidad de Ingeniería del Software)
Socio fundador de Autentia (Formación, Consultoría, Desarrollo de sistemas transaccionales)

<mailto:alejandropg@autentia.com>

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

<http://www.autentia.com>

- Puedes opinar sobre este tutorial [haciendo clic aquí](#).
- Puedes firmar en nuestro libro de visitas [haciendo clic aquí](#).
- Puedes asociarte al grupo AdictosAlTrabajo en XING [haciendo clic aquí](#).
- Añadir a favoritos Technorati. 



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

Recuerda

Autentia te regala la mayoría del conocimiento aquí compartido ([Ver todos los tutoriales](#)). Somos expertos en: J2EE, Struts, JSF, C++, OOP, UML, UP, Patrones de diseño ... y muchas otras cosas.

¿Nos vas a tener en cuenta cuando necesites consultoría o formación en tu empresa?, ¿Vas a ser tan generoso con nosotros como lo tratamos de ser con vosotros?

Somos pocos, somos buenos, estamos motivados y nos gusta lo que hacemos ...

Autentia = Soporte a Desarrollo & Formación.

info@autentia.com



Servicio de notificaciones:

Si deseas que te enviemos un correo electrónico cuando introduzcamos nuevos tutoriales.

Formulario de suscripción a novedades:

E-mail

Tutoriales recomendados

Nombre	Resumen	Fecha	Visitas	pdf
Como implementar el método equals(Object) en objetos persistentes de Hibernate, y otras consideraciones.	En este tutorial Alejandro Pérez nos muestra cómo implementar el método equals(Object) en objetos persistentes de Hibernate, y otras consideraciones.	2008-02-05	1145	pdf
Hibernate 3 y los tipos de datos para cadenas largas	En este tutorial se contará la experiencia que hemos tenido a la hora de manejar los diferentes tipos de datos existentes para grandes cadenas de texto, tales como el tipo Clob de Oracle, o los tipos TEXT de MySQL y SQLServer, utilizando la última versión	2007-03-23	7764	pdf
Hibernate y el mapeo de la herencia	En este tercer tutorial de la saga vamos a ver en este tutorial como implementar las relaciones de herencia con las anotaciones de JPA	2007-06-27	4043	pdf
Comparativa entre Hibernate y EJB3 en la Capa de Persistencia	El presente documento pretende dar algunas luces a la comparativa entre la opción de usar Hibernate y/o EJB3 para la capa de persistencia	2007-08-16	5591	pdf
Introducción a Hibernate	Cesar Crespo nos enseña como utilizar unos de los sistemas más extendidos de mapeo de objetos a estructuras relacionales (tablas de base de datos)	2004-08-14	54569	pdf
Hibernate 3.1, Colecciones, Fetch y Lazy	En este tutorial vamos a ver cómo se comportan ciertas relaciones, y cómo podemos optimizar las consultas a la base de datos con Hibernate	2006-04-17	15967	pdf
Ficheros de mapeo de Hibernate desde las clases	Vamos a ver cómo las HibernateTools nos permiten generar de manera automática los ficheros de mapeo de Hibernate a partir de anotaciones en las clases	2008-06-02	549	pdf
Spring + Hibernate + Anotaciones = Desarrollo Rápido en Java	Alejandro Pérez nos enseña lo fácil y rápido que es desarrollar en Java usando Spring e Hibernate, y usando anotaciones	2008-05-14	2147	pdf
Toda la potencia de un buscador como Google en tu base de datos, gracias a Hibernate Search	En este tutorial vamos a ver una introducción de como montar un sistema para poder hacer búsquedas textuales tipo Google sobre la información que tenemos almacenada en nuestra base de datos	2008-06-13	53	pdf
Manejar dos bases de datos distintas con Hibernate	Alejandro Pérez nos enseña como manejar dos bases de datos distintas con Hibernate	2005-07-13	15687	pdf

Nota:

Los tutoriales mostrados en este Web tienen como objetivo la difusión del conocimiento. Los contenidos y comentarios de los tutoriales son responsabilidad de sus respectivos autores. En algún caso se puede hacer referencia a marcas o nombres cuya propiedad y derechos es de sus respectivos dueños. Si algún afectado desea que incorporemos alguna reseña específica, no tiene más que solicitarlo. Si alguien encuentra algún problema con la información publicada en este Web, rogamos que informe al administrador rcanales@adictosaltrabajo.com para su resolución.