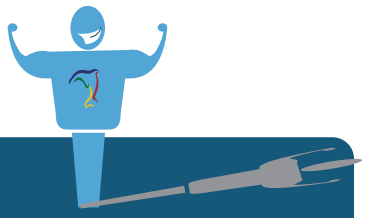
autentia

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...

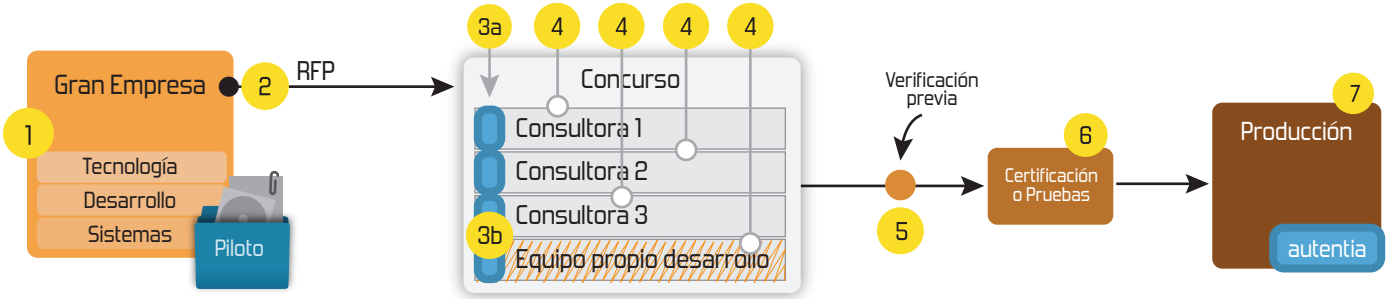
1. Desarrollo de componentes y proyectos a medida



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:
Juan Alonso Ramos

Consultor tecnológico de desarrollo de proyectos informáticos.

Ingeniero Técnico en Informática de Gestión e Ingeniero en Informática, especialidad en Ingeniería del Software

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

Somos expertos en Java/J2EE

Fecha de publicación del tutorial: 2009-02-26

DataTable con paginación en base de datos con Primefaces

Índice de contenidos

1. Introducción

En este tutorial veremos un ejemplo de uso del dataTable de Primefaces que permite la carga bajo demanda de los datos a mostrar en la tabla. Esta forma de sacar la información es la más recomendable ya que para grandes volúmenes de datos la memoria se vería sobrecargada en exceso y con ello degradaríamos la escalabilidad de nuestra aplicación. La solución a este problema está en paginar el listado en base de datos mostrando más datos a medida que el usuario los solicita. Este planteamiento tiene el inconveniente de que se necesitan más consultas a base de datos pero conseguimos traernos únicamente los datos que se requieren y como decía anteriormente liberando mucha más memoria al servidor.

El código fuente del ejemplo lo puedes descargar de aquí.

2. Entorno

- MacBook Pro 15' (2.8 GHz Intel Core 2 Duo, 4GB DDR3 SDRAM).
- Sistema Operativo: Mac OS X Snow Leopard 10.6.4
- JDK 1.6.0_20
- Primefaces 2.1
- JSF 2.0.2
- Spring 3.0.4
- Hibernate Core 3.3.1

3. Uso del Componente

Dentro de los atributos disponibles del dataTable los que son importantes para nuestro ejemplo son el lazy y dynamic. Ambos hay que ponerlos a true. En el ejemplo sacaremos un listado de futbolistas.

```
<H:FORM>
    <P:DATATABLE loadingMessage="#{msg['entity.loading']}" dynamic="true" lazy="true" rows="5" paginatorPosition="bottom" paginator="true" value="#{datatableCtrl.footballers}" var="footballer">
        <P:COLUMN>
            <F:FACET name="header">
                <H:OUTPUTTEXT value="#{msg['Footballer.number']}"></H:OUTPUTTEXT>
            </F:FACET>
            <H:OUTPUTTEXT value="#{footballer.number}"></H:OUTPUTTEXT>
        </P:COLUMN>

        <P:COLUMN>
            <F:FACET name="header">
                <H:OUTPUTTEXT value="#{msg['Footballer.name']}"></H:OUTPUTTEXT>
            </F:FACET>
            <H:OUTPUTTEXT value="#{footballer.name}"></H:OUTPUTTEXT>
        </P:COLUMN>

        <P:COLUMN>
            <F:FACET name="header">
                <H:OUTPUTTEXT value="#{msg['Footballer.lastname']}"></H:OUTPUTTEXT>
            </F:FACET>
            <H:OUTPUTTEXT value="#{footballer.lastname}"></H:OUTPUTTEXT>
        </P:COLUMN>
    </P:DATATABLE>
</H:FORM>
```

En la parte del ManagedBean debemos implementar el código encargado de construir el dataTable y proporcionarle el medio para que pueda recargar el listado con los datos una vez que el usuario pagine. Esto lo haremos en el método init anotado con @PostConstruct para que Spring lo llame una única vez tras la creación del objeto.

```
@ManagedBean
@ViewScoped
public class DatatableCtrl implements Serializable {

    private static final long serialVersionUID = 4089937794197364974L;

    private LazyDataModel<FOOTBALLER> footballers;

    @ManagedProperty(value = "#{dao}")
    private Dao dao;

    @SuppressWarnings("unused")
    @PostConstruct
    private void init() {
        final Long numEvents = (Long) dao.findByQuery(
                "select count(id) from Footballer", -1, -1, null).get(0);

        footballers = new LazyDataModel<FOOTBALLER>(numEvents.intValue()) {

            private static final long serialVersionUID = 8885722005055879976L;

            @SuppressWarnings("unchecked")
            @Override
            public List<FOOTBALLER> fetchLazyData(int first, int pageSize) {
                return dao.findByQuery("from Footballer", first, pageSize, null);
            }
        };
    }

    public LazyDataModel<FOOTBALLER> getFootballers() {
        return footballers;
    }

    public void setDao(Dao dao) {
        this.dao = dao;
    }
```

ayudas, ejemplo Wink, extensiones y encuesta

Configuración de jCaptcha en Struts

MySQL - Sensibilidad a mayúsculas/minúsculas de los nombres de las tablas

Cómo evitar tener más de dos cabezas en Mercurial

Reunión Madrid Ágil 14-10-2010: Equipos autogestionados, y motivación del individuo y del equipo

Últimos Tutoriales del Autor

Generación de Informes con JasperReports en PHP

Múltiples datasources en JasperReports

JCaptcha - Generación de Captchas en Java

Instalar Puente PHP -Java en Tomcat

Sacar Release de un proyecto con Maven

Síguenos a través de:

Últimas ofertas de empleo

2010-10-11
Comercial - Ventas - SEVILLA.

2010-08-30
Otras - Electricidad - BARCELONA.

2010-08-24
Otras Sin catalogar - LUGO.

2010-06-25
T. Información -

```
37 }
```


En el ManagedBean nos creamos una instancia de LazyDataModel, el tipo de datos encargado de trabajar con el listado de futbolistas. Esta clase tiene un método abstracto, el fetchLazyData, que estamos obligados a implementar y donde meteremos el código encargado de consultar la lista de elementos paginados. El componente, a medida que vayamos paginando, llamará a este método pasándole la página en la que se encuentra el usuario y el tamaño de la lista que viene dado por el atributo rows del dataTable, en nuestro caso 5 elementos.

Un consideración a tener en cuenta a la hora de construir el LazyDataModel es que hay que pasarle el número de elementos que tendrá el dataTable para que el componente pueda pintar el número de páginas totales. Para proporcionárselo hacemos un count y se lo pasamos al constructor del LazyDataModel.

El resultado del ejemplo es el siguiente:

Número	Nombre	Apellidos
Realizando la operación, por favor espere un momento...		
1	Iker	Casillas
3	Pepe	Lima
4	Sergio	Ramos
18	Raúl	Albiol
17	Álvaro	Arbeloa
2	Ricardo	Carvalho
14	Xabi	Alonso
24	Sami	Khedira
8	Kaka	Dos Santos
11	Esteban	Granero

1 2

Número	Nombre	Apellidos
21	Pedro	León
16	Sergio	Canales
23	Mesut	Ozil
22	Angel	Di María
7	Cristiano	Ronaldo
9	Karim	Benzema
20	Gonzalo	Higuaín

1 2

A través del atributo loadingMessage del componente se puede indicar un mensaje de información de espera por si la consulta tarda demasiado tiempo. Es el que aparece en la parte superior de los registros de la tabla.

Conclusiones

Dentro de las múltiples configuraciones disponibles del componente DataTable de Primefaces creo que la que hemos visto en este tutorial es una de las más recomendables si en vuestras aplicaciones tenéis que sacar grandes volúmenes de datos por no mencionar su facilidad de uso y configuración.

Este tutorial va dedicado a Cagatay Civici, creador de Primefaces y declarado seguidor del Barca al cual, como no podía ser de otra forma, agradecemos enormemente su trabajo y aportación de tan grande librería de componentes JSF.

Espero que te haya servido de ayuda.

Un saludo. Juan.

COMENTARIOS