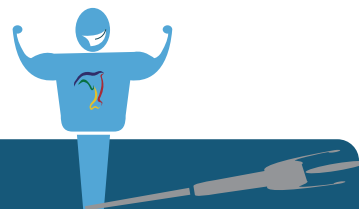


¿Qué ofrece Autentia Real Business Solutions S.L?

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

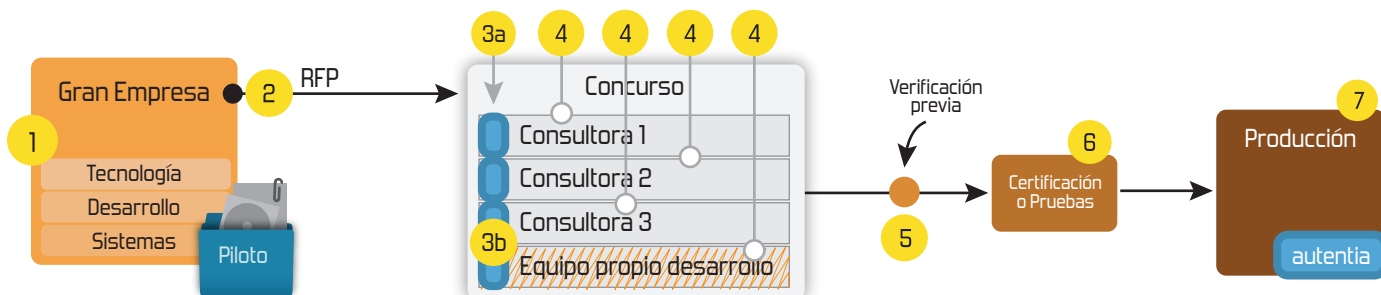
1. Desarrollo de componentes y proyectos a medida



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

autentia
 Soporte a desarrollo informático
 Hosting patrocinado por
enREDados

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](#) ZKPushState: Manejar el API de Historial de navegación en HTML5 directamente...

Francisco Ferri Pérez

Consultor tecnológico de desarrollo de proyectos informáticos.

Desarrollador de proyectos informáticos, Microsoft Certified IT Professional - Enterprise Administrator

[Ver todos los tutoriales del autor](#)
**FRUSTRATED WITH
JAVA REDEPLOYS!?**
JRebel
 FREE TRIAL

Fecha de publicación del tutorial: 2013-01-14

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

Framework ZK



ZKPushState: Manejar el API de Historial de navegación en HTML5 directamente desde Java

18 de diciembre de 2012, artículo original de [Monty Pan](#)

Introducción

La especificación HTML5 introduce un API para manejar el histórico de navegación del browser.

- El objeto history puede usar pushState() para inyectar la información en el histórico de navegación de la sesión y cambiar la url del browser por la que le hemos proporcionado, si es que le proporcionamos una.
- Cuando el historial cambia, el navegador dispara el evento "onpopstate" a en la ventana, "window.onpopstate".

El artículo [Administración del histórico con el API HTML desde ZK](#), explica cómo los desarrolladores de ZK pueden usar ese API con código JavaScript.

Voy a enseñaros cómo puedes hacerlo directamente desde código Java. Lo conseguimos introduciendo un AddOn "ZKPushState", que es un envoltorio que permite a los desarrolladores manejar esta característica, directamente desde Java, lo que hace que manejar el histórico, más y más del estilo que ZK hace las cosas y simplifica la dificultad de tener que codificar y descodificar la información.

¿Cómo lo hacemos?

La única cosa que debes hacer antes de empezar a profundizar en el código es descargar el fichero ZKPS.jar y ubicarlo en el classpath, de forma que puedas llamar al método PushState.push(Map, String, String) desde Java y añadir el atributo onPopUpState en el elemento root de tu ZUL.

Para verlo más claro, sigue el siguiente ejemplo.

Ejemplo

Siguiendo el siguiente ejemplo: [Administración del histórico con el API HTML desde ZK](#), que contiene tres TextBoxes y botones de filtro. Puedes escribir texto en las casillas de texto para buscar, haciendo click en el botón de filtro, el componente listBox te mostrará filtrada la información basándose en el texto que introdujiste en el textbox.Reescribiremos el ejemplo usando **ZKPushState** y **MVVM**, el código ZUL de los botones quedará como sigue:

```
view plain print ?
01. <button label="Filter" width="50px" onClick="@command("filter", f1=filter1.value, f2=filter2.value, f3=filter3.value)"/>
```

En el ViewModel, el método de comando queda como sigue:

```
view plain print ?
01. @Command
02. @NotifyChange("f1")
03. public void filter(@BindingParam("f1") String f1, @BindingParam("f2") String f2, @BindingParam("f3") String f3) {
04.     doFilter(f1, f2, f3);
05. }
```

Catálogo de servicios Autentia



Síguenos a través de:



Últimas Noticias

» ¿Qué es aportar valor como técnico/programador?

» Como arrancar un nuevo proyecto e integrar el agilismo en una organización

» Hoy es el primer día para cambiar tu sector

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

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

[Histórico de noticias](#)

Últimos Tutoriales

» Modela tu mercado en base a la demanda y no a la oferta.

» Mirar el todo: llevar el agilismo a las grandes organizaciones.

» Creación de un gif animado con Adobe Photoshop

» Lanzando nuestros tests de jasmine-node con IntelliJ IDEA

» Maquetación de un libro con Adobe InDesign

```

05. Map<String, String> map = new HashMap<String, String>();
06. map.put("f1", f1);
07. map.put("f2", f2);
08. map.put("f3", f3);
09. PushState.push(map, "Search results", "?q="+f1+f2+f3);
10. }

```

La función de doFilter es para reconstruir el modelo del componente listBox. Al final del método filter(), llamamos a PushState.push(), el cual ejecutará el history.pushState() en la parte del cliente automáticamente. En este punto del tiempo, la URL cambiará y el usuario puede ir a su estado previo haciendo click en el botón de "volver" en la barra de herramientas del navegador, volviendo a la página anterior.

Esto es lo que sucede: después de algunas búsquedas, el navegador mantendrá algunos estados para enlazar con la URL que le proporcionamos. Cuando el usuario clickea el botón de "volver" del navegador, ZKPushState lanza el PopupStateEvent, por lo tanto, necesitamos añadir el atributo onPopupState en el elemento root de nuestro ZUL del siguiente modo:

```

view plain print ?
01. <window onPopupState="@command("popupState", event=event)">

```

En el ViewModel, el comando se queda como sigue:

```

view plain print ?
01. @Command
02. @NotifyChange("")
03. public void filter(@BindingParam("f1") String f1, @BindingParam("f2") String f2, @BindingParam("f3")
04. {
05. doFilter(f1, f2, f3);
06. Map<String, String> map = new HashMap<String, String>();
07. map.put("f1", f1);
08. map.put("f2", f2);
09. map.put("f3", f3);
10. PushState.push(map, "Search results", "?q="+f1+f2+f3);

```

Podemos recibir el estado con event.getState() el cual retorna una instancia del objeto Map>String, ?<. Para hacer esto, llamamos al método doFilter con los argumentos que retorna PopupStateEvent, el listBox mostrará la misma información que antes (antes de que el usuario saliera de la página previamente).

Puedes descargar el código fuente completo del ejemplo y ZKPushState del repositorio de github aquí.

Cualquier feedback o comentario será bienvenido! :)

Recursos interesantes

- Descargar ZK 6.5.1.1
- ZK 6.5.1.1's Release Notes
- Artículo original de Monty Pan

Este documento es un extracto de la documentación oficial del Framework ZK, traducido y ampliado por Francisco Ferri. Colaborador de Potix (creadores del Framework ZK). Si quieres contactar con él puedes hacerlo en franferri@gmail.com, en twitter [@franciscoferri](https://twitter.com/franciscoferri) o en [LinkedIn](#)

A continuación puedes evaluarlo:

[Regístrate para evaluarlo](#)

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

Share |

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

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

Últimos Tutoriales del Autor

» Tutorial de Responsive Design

» ZK 6.5 Consejos para dispositivos móviles

» ZK 6.5 Empezando con el Responsive Design

» ZK 6.5 Responsive design

» ZK Responsive Design, enfoque/filosofía

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.



Esta obra está licenciada bajo licencia Creative Commons de Reconocimiento-No comercial-Sin obras derivadas 2.5

IMPULSA

Impulsores

Comunidad

¿Ayuda?

0 personas han traído clicks a esta página

sin clicks

+

+

+

+

+

+

+

+

powered by [karmacracy](#)

