

# Egy újabb webalkalmazás framework ZK

Bakay Árpád

NETvisor kft

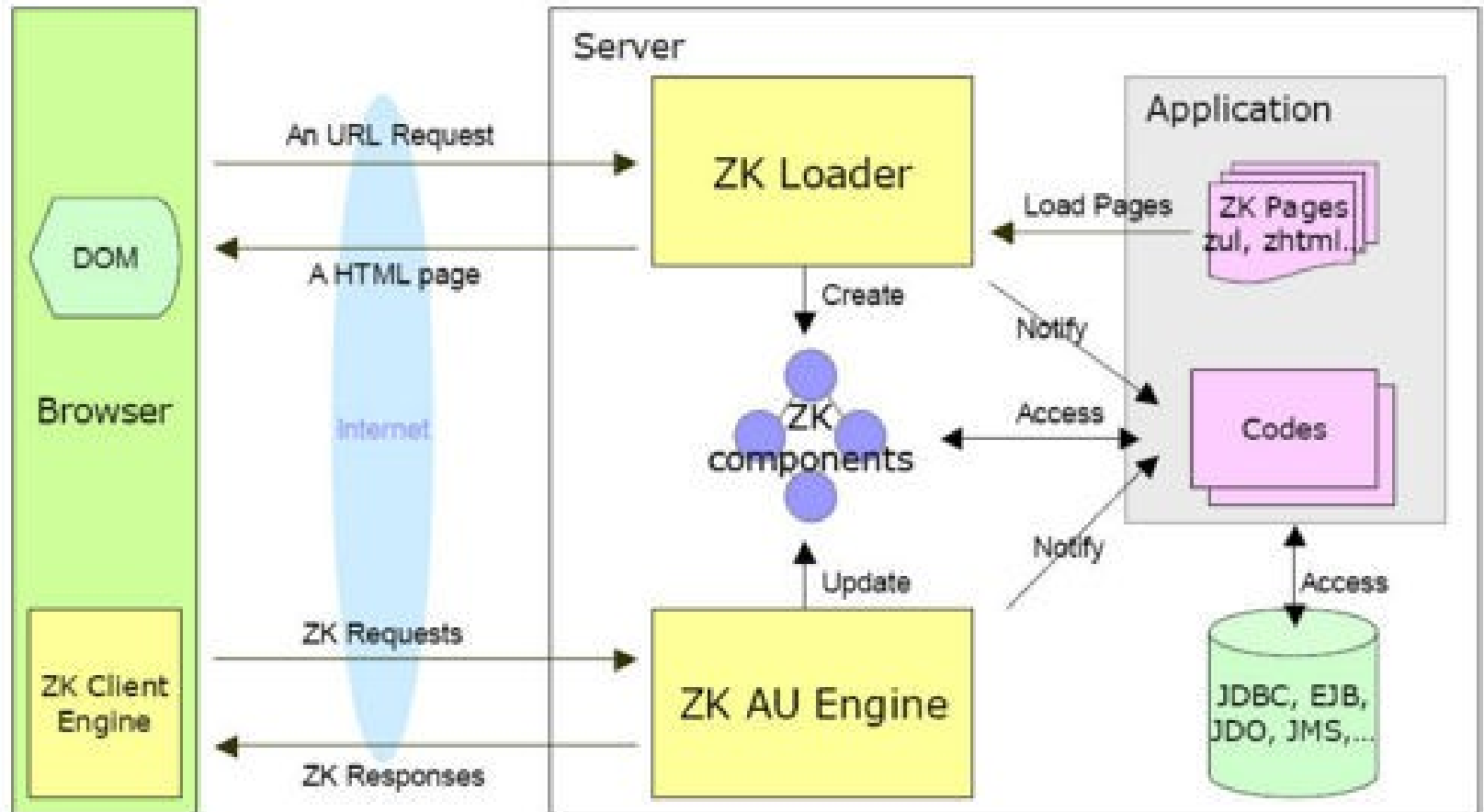
(30) 385 1711

[arpad.bakay@netvisor.hu](mailto:arpad.bakay@netvisor.hu)

# Amit már láttunk...

- Komponens-alapú, eseményvezérelt
  - (egymásba ágyazott widgetek)
- View: egy dokumentum
- Backing bean modul: egy Java osztály
  - Hozzáfér a view objektumaihoz
  - Opcionálisan mapping ezek között:
    - **bean propertyk ↔ html aktív elemek**
- AJAX-ot támogat
  - Kommunikáció XMLHttpRequest Bridge-en keresztül
- Nem kell HTML-t, Javascriptet fejleszteni

# Architektura



AU: Asynchronous Update

# Ami újdonság

- Új, XML alapú nyelv (nem HTML/JSP-szerű)
  - XML formátumú ZUML '.zul' fileok
  - Nem Web-es GUI-hoz hasonló szemlélet
    - Eseményvezérlők, aszinkron update-ek.
- ZSCRIPT: ZUML-ban támogatott az embedded Java kód
  - Részben a backing bean helyett is használható: = nem tiszta MVC
  - Szerver oldalon fut, azaz a formai hasonlóság dacára NEM a JavaScript megfelelője
    - Inkább hasonlít a JSP scriptletekhez ( <% ... %> )
  - A ZSCRIPT-ben más nyelvek szintaxisa is használható (Ruby, Python, Groovy)
- Nincs a faces-config-xml-hez hasonló központi XML
  - Backing bean és oldalak összerendelése magán az oldalon történik
  - Nincs szükség JSF navigációs rule-okra, mert itt ritkán lépünk lapok között, hanem a UI nagy része egyetlen komplex oldalon van megoldva.

# Backing bean vagy ZSCRIPT

- AZ UI logika vezérlőjét kétféleképpen is megírhatjuk:
  - JSF-szerű backing bean
    - Ez a „hivatalos” mód javasolt a „komoly” pl. businesslogic funkciók kezelésére.
  - ZUML-ba írt ZSCRIPT tag-ek
    - Elsősorban prototípusokhoz, ill. egyszerű GUI-közeli funkciókra.
    - A script közvetlenül (az id-t változóként használva) eléri az adott <window>-t és a benne levő komponenseket

# ZUML – és ZSCRIPT

```
<window title="To do list" width="640px" border="normal">
<zscript>
import events.Event;
import events.EventDAO;
import java.util.ArrayList;

//fetch all events from database
EventDAO evtDAO = new EventDAO();
List allEvents = evtDAO.findAll();

void add() { // DAO OPERATIONS NOT DETAILED HERE
    ListItem li = new ListItem();
    li.appendChild(new ListCell(name.value)); // same for priority and date
    box.appendChild(li);
}
void update() { ... } // NOT DETAILED HERE
</zscript>
<listbox id="box" multiple="true" rows="4">
<listhead>
<listheader label="Item" />
<listheader label="Priority" width="50px" />
<listheader label="Opened" width="90px" />
</listhead>
<listitem forEach="{allEvents}" value="{each}">
<listcell label="{each.name}" />
<listcell label="{each.priority}" />
<listcell label="{each.date}" />
</listitem>
</listbox>
<groupbox>
<caption label="New" />
Item: <textbox id="name" cols="50" />
Priority: <intbox id="priority" cols="1" />
Date: <datebox id="date" cols="8">
<button label="Add" width="36px" height="24px, onClick="add()"/>
<button label="Update" width="46px" height="24px onClick="update()"/>
<button label="Delete" width="46px" height="24px"/>
</groupbox>
</window>
```

Az mellékelt kód egy egyszerű to-do lista kezelőt mutat, ahol backing bean helyett ZSCRIPT-ben van megírva UI logika.

A fekete részek írják le az UI felületet: egy táblázatszerű „lisbox”, alatta egy csoportban 3 beviteli mező, ill. 3 gomb..

A kék rész az a ZSCRIPT, ami inicializálja az adatokat tartalmazó allEvents objektumot, ami a piros kódokon keresztül a táblázatba kerül

**FONTOS**, hogy mind ez, mind pedig a gombok eseménykezelői a SZERVER oldalon futnak.

Sőt: a megtévesztő látszat ellenére az onClick () metódusok sem JavaScript-ek, hanem szerver oldalon futó (Ajax bridge-en indított) ZSCRIPT-ek!!!

Ezt a példát innen loptam:

<http://www.zkoss.org/doc/step%20by%20step.dsp>

# Ugyanez Backing bean-nel

```
<window id="win" use=„hu.pgy.en.ToDoController"
title="To do list" width="640px" border="normal">
<listbox id="box" multiple="true" rows="4">
  <listhead>
    <listheader label="Item" />
    <listheader label="Priority" width="50px" />
    <listheader label="Opened" width="90px" />
  </listhead>
  <listitem forEach="{win.allEvents}" value="{each}">
    <listcell label="{each.name}" />
    <listcell label="{each.priority}" />
    <listcell label="{each.date}" />
  </listitem>
</listbox>
<groupbox>
  <caption label="New" />
  Item: <textbox id="name" cols="50" />
  Priority: <intbox id="priority" cols="1" />
  Date: <datebox id="date" cols="8">
  <button label="Add" width="36px" height="24px,, onClick=„win.add()"/>
  <button label="Update" width="46px" height="24px onClick=„win.update()"/>
  <button label="Delete" width="46px" height="24px"/>
</groupbox>
</window>
```

*A backing bean-t az use attributum rendeli az oldalhoz..*

*Mivel ez igazi Java (nem BeanShell, mint a ZSCRIPT-nél), window alatti ui komponenseket a getFellow()-val lehet elérni.*

```
public class ToDoController extends Window {

  List allEvents;

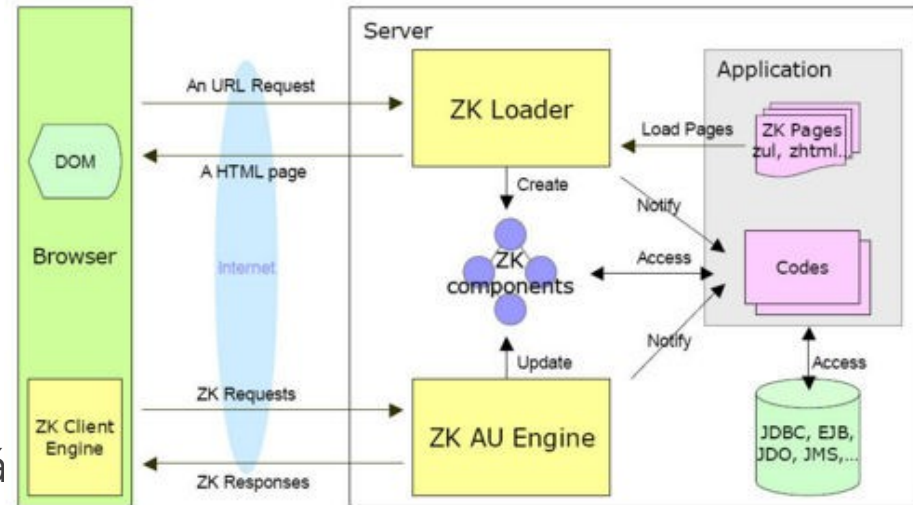
  ToDoController() {
    EventDAO evt dao = new EventDAO();
    allEvents = evt dao.findAll();
  }

  void add() {
    Listitem li = new Listitem();
    li.appendChild(
      new Listcell(getFellow(„name”).value));
    // same for priority and date
    getFellow(„box”).appendChild(li);
  }

  void update() {
  }
}
```

# Standard életciklus

- Oldal kezdeti kérése
  - ZUML -> HTTP rendering
  - Közben:
    - létrejönnek a widgetek tükör objektumai
    - meghívódhatnak az alkalmazás Java eseménykezelői is
- ZK client engine parancsokat küld a szervernek
  - Komponens tükör obj. update a kliens státusz alapján
  - Események mehet(nek) az alkalmazásnak. Ez az komponens objektumokon újabb módosításokat jelent(het).
  - Az AU válasza lényegében DOM változtatási kéréseket tartalmaz.



# Komponensek

- Összesen kb. 200 készen kapott komponens!
- Package org.zkoss.zul : ZUL component set that are used for HTML-based clients. It includes window, button, listbox.
- Package org.zkoss.zkex.zul : Zul Extension library. It includes borderlayout, jasperreport.
- Package org.zkoss.zkmax.zul : Components and UI utilities available in the Enterprise edition.

# Komponens Fa - fogalmak

- Desktop – egy komplett HTML válasz
- Page – ZUL komponensek gyűjteménye
  - Amit egyetlen ZUML generál
  - Több is lehet a Desktopon
    - Tehát 1 HTML Page != ZK Page

The screenshot displays a web browser window with the address bar showing 'www.zkstore.com(Desktop)'. The page content is organized into several sections:

- ProductList.zul (Page):** A table listing products with columns for Name, Price, Quantity, Arrive Date, and operation. The items listed are Cookies (Price: 4.0, Quantity: 30, Arrive Date: 2010/10/31 11:39), Toast (Price: 3.0, Quantity: 2), and Chocolate (Price: 5.1, Quantity: 12, Arrive Date: 2010/10/24 11:42). Each item has an 'add' button.
- Shopping Cart:** A table showing the items in the cart with columns for Name, Price, Amount, and subTotal. The items are Cookies (Price: 4.0, Amount: 4, subTotal: \$ 16.0), Toast (Price: 3.0, Amount: 2, subTotal: \$ 6.0), and Chocolate (Price: 5.1, Amount: 5, subTotal: \$ 25.5). The total amount is \$ 47.5. There are 'submit' and 'clear' buttons.
- Orders.zul (Page):** A table showing order details with columns for No, description, and Sub Total. The orders listed are:

No	description	Sub Total
processing : 2010/10/23 01:59	No additional requests	73.5
processing : 2010/11/03 06:11	Note for this order.	92.5

# Komponens Fa - fogalmak

- Window
  - Egy másik félreérthető ZK fogalom.
  - A window egy container komponens (mint panel, groupbox, stb.)
    - „Alap container”
    - A gyermekeit egymás alá rendezve mutatja
  - Egy Page gyökér-eleme gyakran (v. mindig?) egy window
  - A window egyben ZSCRIPT namespace is
    - Ld. alább

# ZSCRIPT Hierarchikus láthatóság

A ZSCRIPT nem igazi Java, hanem BeanShell  
Multi-context, automatikus változók

- Minden „window” tag egy új hierarchikus namespace-t hoz létre.

```
<window id="A">
  <zscript>var1 = "abc";</zscript>
  <window id="B">
    <zscript>
      var2 = compld1;    // változókét látszanak a child komponensek
      var2.setText(var1); // látszanak a szülő namespace változói
    </zscript>
    <textbox id=„compld” />
  </window>
  // de innen már nem lászik var2 vagy a compld
</window>
```

Backing bean-nél nincs ilyen, ott a getFellow() val érjük el a komponenseket (id szerint)

# ZK – kételyek és korrekciók

- Nincs HTML – de vannak tag attribútumok és igényes alkalmazásokhoz használni kell a CSS-t is.
  - \_ Újabb fejlemények:
    - JSP -vel kombinált ZK ZUML
    - Java-ból összeállított komponens hierarchia
      - \_ Richletek: Java nyelven összerakott ZUML\_ből használható komponens
- Nincs JavaScript – de van ZSCRIPT
  - \_ BeanShell-ben vágrehajtott Java
  - \_ Visszalépés az MVC-től?
- ZSCRIPT szerver oldalon fut (az AU engine-en keresztül)
  - \_ Nem lassú ez?
  - \_ Kombinálható a JQuery frameworkkel, kliens oldali interaktivitás céljából.

# Példák

1. Egyszerű AJAX + Gmaps használat

*<http://www.zkoss.org/doc/tutorial.dsp>*

1. ZSCRIPT Használata

*<http://www.zkoss.org/doc/step%20by%20step.dsp>*

1. ZK + Spring framework

*<http://zkoss.org/smalltalks/contact/contact.dsp>*

# Ellenőrző kérdések

- Vázolja fel egy ZK alkalmazás architektúráját, a kliens és a szerver kapcsolatait.
- Mi a hasonlóság a ZK és a JSF között?
- Mikben különbözik a ZK a JSF-től?
- Mi a ZK UI logika elkészítésének két lehetősége?
- Mi a legfontosabb különbség a JavaScript és a ZSCRIPT között?
- Soroljon fel 3-4 alapvető ZK komponenst
- Soroljon fel 3-4 érdekesebb ZK komponenst