



Lập trình Web với ZK framework

GV. Trần Trung Hiếu
Khoa CNTT - VNUA

Nội dung chính

- 1. Giới thiệu ZK framework**
- 2. Kiến trúc ứng dụng ZK framework**
- 3. Cài đặt và tạo ứng dụng ZK trên Eclipse**
- 4. Một số thành phần giao diện cơ bản**
- 5. Ứng dụng làm quen**
- 6. Ứng dụng web căn bản**

1. Giới thiệu ZK framework



Enterprise



Rapid development



Security



Simply Java



Innovation technology

1. Giới thiệu ZK framework



1. Giới thiệu ZK framework

- ZK framework được phát triển bởi Potix Đài Loan cùng với cộng đồng lập trình viên mã nguồn mở
- Đặc trưng công nghệ của ZK framework là server-centric (các xử lý tập trung ở phía server-side). Khác với GWT, Vaadin và ExtJS là client-centric (từ bản ZK 8 cho phép tích hợp thư viện javascript phía client)
 - Các nhà phát triển tập trung vào phát triển logic kinh doanh của mình ở phía máy chủ và để lại mọi thứ khác như giao tiếp giữa máy chủ, tương thích trình duyệt, mạng, bảo mật...
 - Triết lý của ZK là AJAX without Javascript, cho phép XD các ứng dụng Web phong phú mà không cần bất kỳ kiến thức nào về Ajax và JavaScript
 - Các nhà phát triển có thể tận dụng tất cả các công nghệ Java, như Java EE và Spring, trong khi dữ liệu và dịch vụ back-end có sẵn
 - Tất cả logic nghiệp vụ được lưu trữ tại máy chủ đồng nghĩa với việc tăng cường bảo mật cho các ứng dụng Enterprise. Không có logic kinh doanh nào được đưa ra cho khách hàng, bảo vệ bạn khỏi 10 mối quan tâm bảo mật hàng đầu của XSS, CSRF, DOS và OWASP.

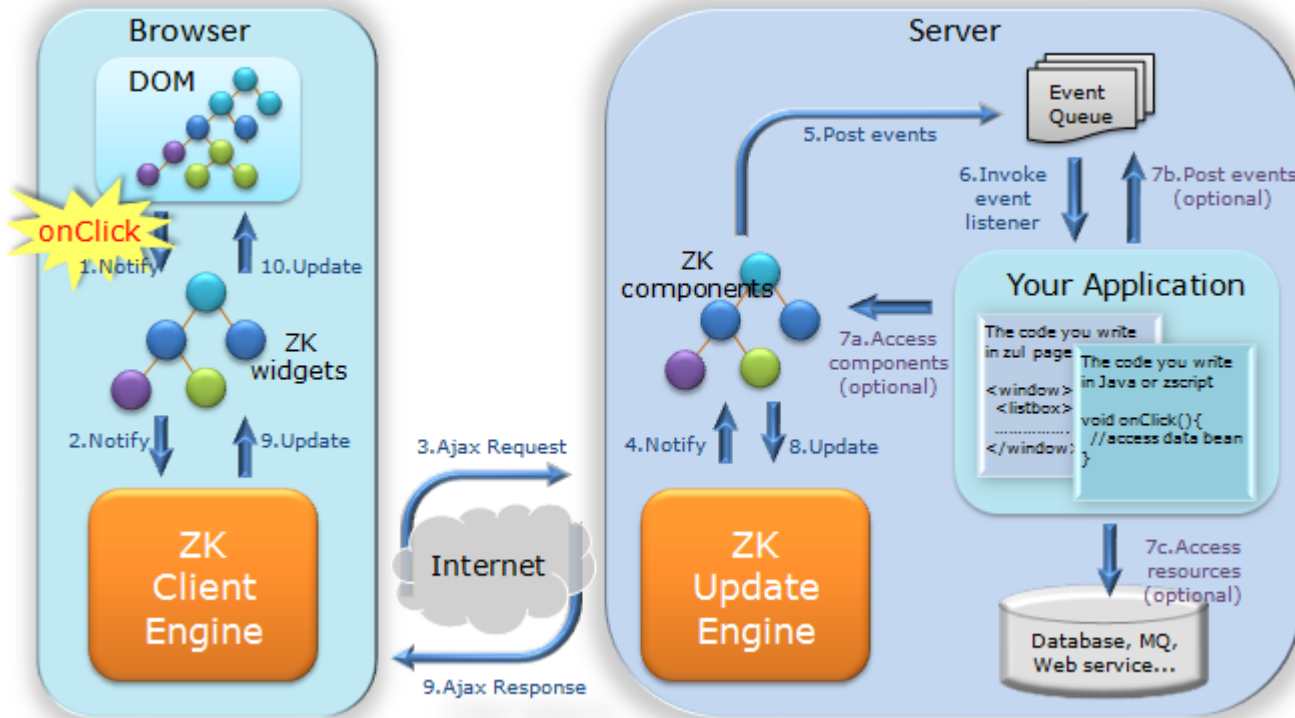
1. Giới thiệu ZK framework

- ZK giúp lập trình viên Java server-side xây dựng ứng dụng doanh nghiệp nhanh dạng Rich Internet Application (RIA)
 - Người dùng ZK báo cáo trung bình tăng gấp 4 - 16 lần năng suất.
- Có nhiều tiện ích hỗ trợ các component tiêu chuẩn như:
 - Các thành phần UI cơ bản với hơn 200 thành phần và chủ đề sẵn sàng sử dụng
 - ZK Spreadsheet: bảng tính
 - ZK Charts: hỗ trợ rất nhiều loại biểu đồ trực quan
 - ZK Calendar: xây dựng bảng chấm công, lịch trình, thời gian biểu
 - ZK Pivottable: hiển thị tập dữ liệu phức tạp, kích thước lớn

1. Giới thiệu ZK framework

- Công nghệ tiên tiến, kể từ lần phát hành đầu tiên vào năm 2005, ZK đã dẫn đầu trong ngành:
 - Là Java framework đầu tiên hỗ trợ mẫu MVVM (Có thể sử dụng MVC)
 - Giải pháp đáp ứng PC + Mobile đầu tiên hỗ trợ ghi một lần chạy ở mọi nơi,
 - Kiến trúc hợp nhất máy khách + máy chủ đầu tiên
 - Thành phần bảng tính web đầu tiên.
- ZK có vài phiên bản miễn phí (ZK CE) và có trả phí (ZK EE, PE). Mã nguồn của ZK là mã nguồn mở
- Tài liệu tra cứu, các ví dụ minh họa trực quan kèm mã nguồn rất phong phú
 - Demo các UI: <https://www.zkoss.org/zkdemo/grid?zkn=1>
 - Chạy thử code online:
 - <https://www.zkoss.org/zksandbox/>
 - <https://zkfiddle.org/>
 - Chi tiết các UI: https://www.zkoss.org/wiki/ZK_Component_Reference
 - Xây dựng UD: https://www.zkoss.org/wiki/ZK_Developer's_Reference
 - Tài liệu khác: <https://www.zkoss.org/documentation#References>

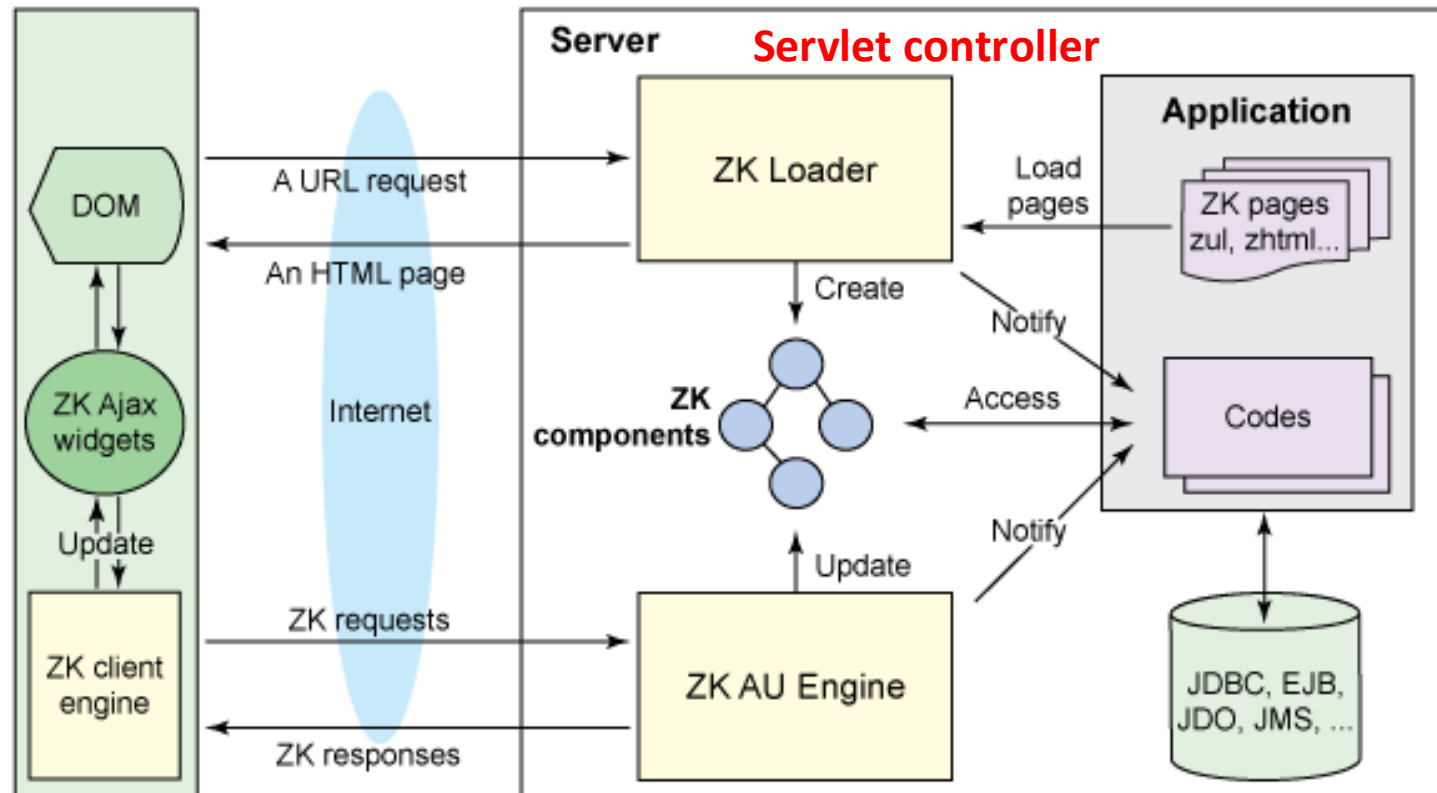
2. Kiến trúc ứng dụng ZK framework



Các thành phần UI trên trang HTML DOM (Document Object Model) được ZK mô hình hóa bởi ZK widgets (phía client, là các đối tượng javascript) và ZK components (phía server, là các đối tượng Java POJO)

- ZK Client Engine: Được viết bởi javascript đóng vai trò quản lý các widget và giao tiếp phía client
- ZK Update Engine: Được viết bởi java, đóng vai trò cập nhật trạng thái các component và giao tiếp phía server

2. Kiến trúc ứng dụng ZK framework



Yêu cầu trang mới qua URL được đáp ứng bởi ZK Loader, sau đó các hoạt động trên trang được thông qua ZK request, response

3. Cài đặt và tạo ứng dụng ZK trên Eclipse

- **Các phần mềm cần chuẩn bị**

- Java JDK (version 8):

<https://www.oracle.com/technetwork/java/javase/downloads/java-archive-javase8-2177648.html>

- Eclipse (version for Enterprise Java Developers):

<https://www.eclipse.org/downloads/packages/>

- Apache Tomcat (version 9):

<https://archive.apache.org/dist/tomcat/tomcat-9/v9.0.24/bin/>

- Thư viện ZK framework (version 8.6, dùng bản miễn phí ZK CE):

<https://www.zkoss.org/download/zk>

- Cơ sở dữ liệu:

- *MySQL community (version 8.0.19, download bộ installer và chỉ cài Server ko cài Full):*

<https://dev.mysql.com/downloads/mysql/>

<https://o7planning.org/vi/10221/huong-dan-cai-dat-va-cau-hinh-mysql-community>

- *Công cụ quản trị CSDL: SQLyog community (version 13.1.5):*

<https://github.com/webyog/sqlzog-community/wiki/Downloads>

- *MySQL JDBC Driver (jar file):*

<https://mvnrepository.com/artifact/mysql/mysql-connector-java/8.0.19>

3. Cài đặt và tạo ứng dụng ZK trên Eclipse

- **Cài đặt Tomcat trên Eclipse**

- Windows > Preferences > Server > Runtime Environments > Add > Apache Tomcat 9 > Browser đường dẫn tới thư mục Tomcat đã giải nén

- **Cài đặt ZK studio trên Eclipse**

- Help > Eclipse Marketplace > Tìm kiếm cụm từ “ZK studio” > Cài đặt

(Lưu ý: Security Warning > OK, Restart Eclipse, Activate ZK studio là tùy chọn, ZK studio sẽ tự động cài với thư viện ZK EE là bản ko miễn phí, cần thay thế bởi ZK CE ở bước sau)

- Thay đổi phiên bản ZK: Window > Preferences > ZK > ZK package > Add File > Chọn file nén ZK CE đã download > Tích chọn phiên bản ZK CE

3. Cài đặt và tạo ứng dụng ZK trên Eclipse

- **Tạo ứng dụng HelloWorld trên Eclipse**
 - File > New > Project > ZK project > Đặt tên cho project > Next.. > Tích chọn sinh file web.xml > ...
- **Chạy trên server:**
 - Chuột phải vào tên project > Run As > Run on server
 - Mặc định file index.zul được chạy trên server

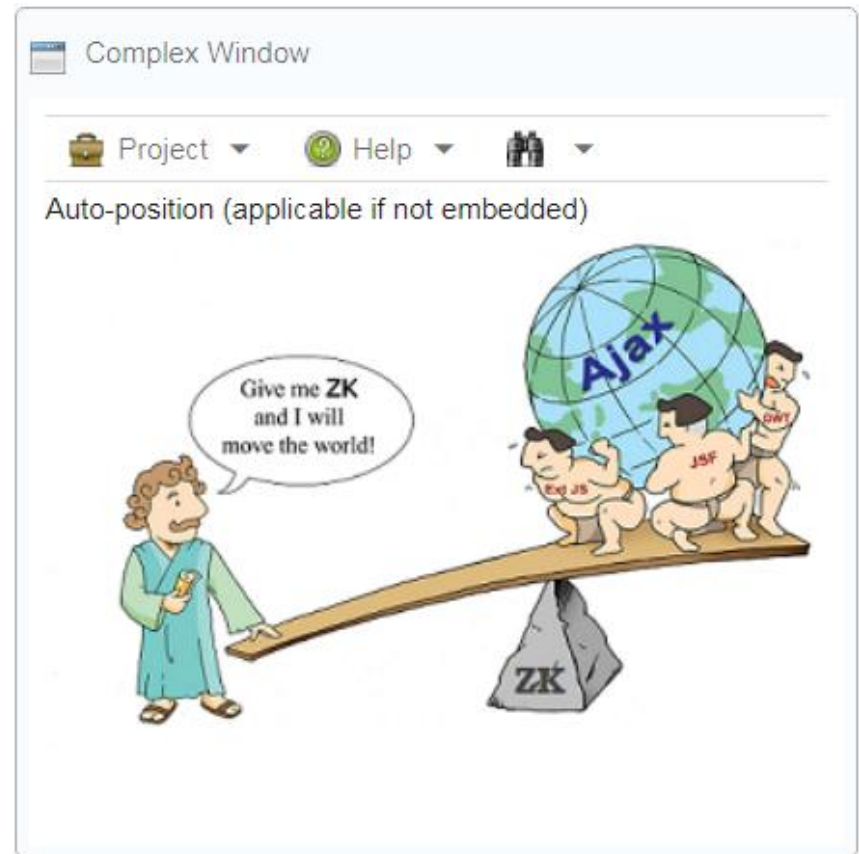
Hello World!!

You are using:
8.6.0.1

```
<?page title="Auto Generated index.zul"?>
<window title="Hello World!!" border="normal" width="200px">
<label value="You are using: ${desktop.webApp.version}"/>
</window>
```

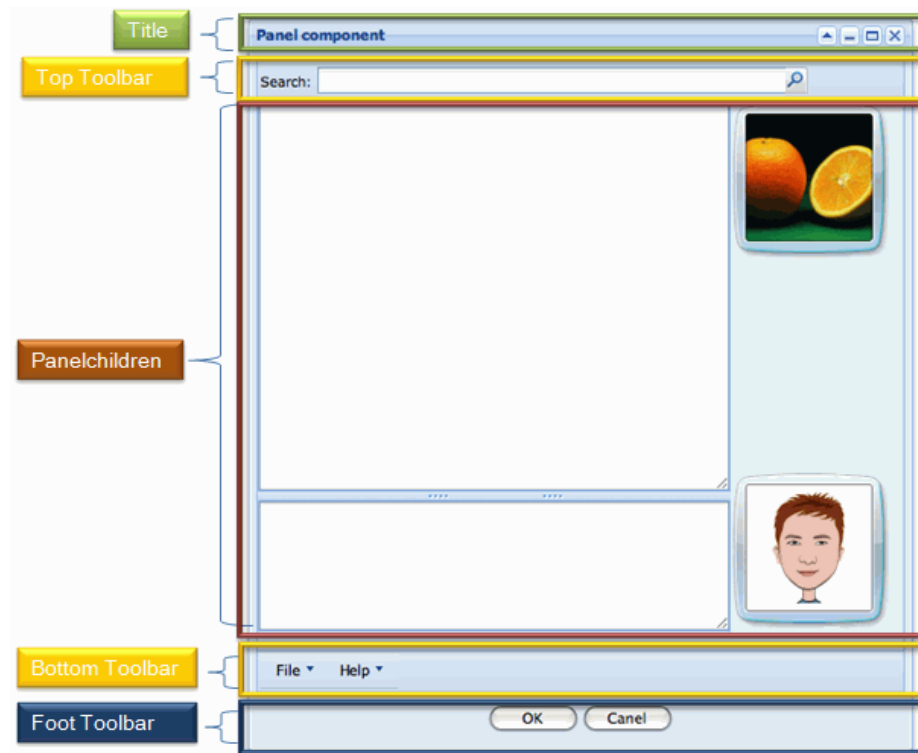
4. Một số thành phần giao diện cơ bản

- **Các thành phần vùng chứa (container)**
 - Div: tương tự thẻ Div trong HTML, là một khối để nhóm các thành phần bên trong nó
 - Window: tương tự thẻ Div của HTML, nó có thể hiển thị ở 5 chế độ: embedded (default), overlapped, popup, modal và highlighted. Window thực thi “ID space” (không gian ID độc lập) nên ID của các thành phần con có thể trùng ID của các thành phần bên ngoài Window.



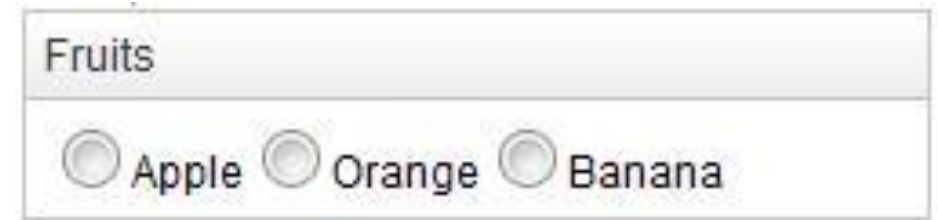
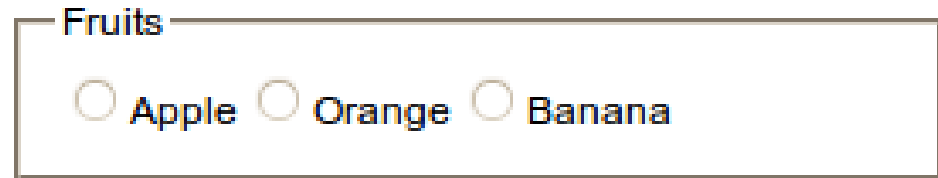
4. Một số thành phần giao diện cơ bản

- **Các thành phần vùng chứa (container)**
 - Panel: tương tự Window, Panel hỗ trợ 3 vị trí toolbar gồm top toolbar, bottom toolbar, foot toolbar. Các thay đổi của Panel chỉ diễn ra trong nội bộ thành phần cha chứa nó. Panel không thực thi “ID space”, các thành phần con của Panel có thể truy cập trực tiếp từ ngoài.



4. Một số thành phần giao diện cơ bản

- **Các thành phần vùng chứa (container)**
 - Groupbox: được sử dụng để nhóm các thành phần vào một khối, có hỗ trợ caption và đường viền



4. Một số thành phần giao diện cơ bản

- **Các thành phần vùng chứa (container)**
 - Tabbox: được sử dụng để nhóm các thành phần vào các tab
 - Span: tương tự thẻ span của HTML, được sử dụng để nhóm một nhóm nhỏ ở chế độ inline thường dùng trong phạm vi một dòng (ví dụ, một nhóm thành phần trong một paragraph)
 - Caption: là header thêm vào cho Groupbox, Window, Panel

4. Một số thành phần giao diện cơ bản

- **Các thành phần trình bày dạng danh sách**
 - Grid: Tương tự bảng trong HTML
 - Có thể bao gồm *auxhead* (khối tiêu đề), dòng tên các cột, các cột, dòng, footer.
 - Có thể nhóm các dòng, phân trang, sắp xếp, chỉnh sửa dữ liệu trực tiếp tại các ô.
 - Grid có thể chứa các thành phần giao diện hoặc dữ liệu

Type	Content
File:	<input type="text"/>
Type:	Java Files (*.java) <input type="button" value="Browse..."/>
Option:	<input type="text"/>

Order #	Item Purchased	Purchased Time	Paid
101	Macchiato	AM 08:16:15	<input checked="" type="checkbox"/> yes
102	Latte	AM 08:26:17	<input type="checkbox"/> no
103	Mocha	AM 08:16:52	<input checked="" type="checkbox"/> yes
104	Macchiato	AM 08:17:32	<input type="checkbox"/> no
105	Latte	AM 08:11:24	<input checked="" type="checkbox"/> yes
106	Lungo	AM 08:17:05	<input checked="" type="checkbox"/> yes

<< < 1 / 5 > >>

[1 - 6 / 25]

4. Một số thành phần giao diện cơ bản

- **Các thành phần trình bày dạng danh sách**
 - Listbox: Tương tự như grid, chỉ nên sử dụng Listbox khi cần lưu trữ các mục có thể tích chọn

Quality Assurance			
<input type="checkbox"/> Product	Quality	Comment	Confirm
<input type="checkbox"/> Product #1	Medium ▼	<input type="text"/>	<input type="button" value="confirm"/>
<input type="checkbox"/> Product #2	Medium ▼	<input type="text"/>	<input type="button" value="confirm"/>
<input type="checkbox"/> Product #3	Medium ▼	<input type="text"/>	<input type="button" value="confirm"/>
<input type="checkbox"/> Product #4	Medium ▼	<input type="text"/>	<input type="button" value="confirm"/>
<input type="checkbox"/> Product #5	Medium ▼	<input type="text"/>	<input type="button" value="confirm"/>

4. Một số thành phần giao diện cơ bản

- **Các thành phần trình bày dạng danh sách**

- Tree: Hiển thị dữ liệu dạng cây, có thể có thêm các thành phần header, footer, có thể cho phép tích chọn các thành phần

Name		Description
<input checked="" type="checkbox"/>	Item 1	Item 1 description
<input checked="" type="checkbox"/> ▲	Item 2	Item 2 description
<input type="checkbox"/> ▲	Item 2.1	
<input type="checkbox"/>	Item 2.1.1	
<input type="checkbox"/>	Item 2.1.2	

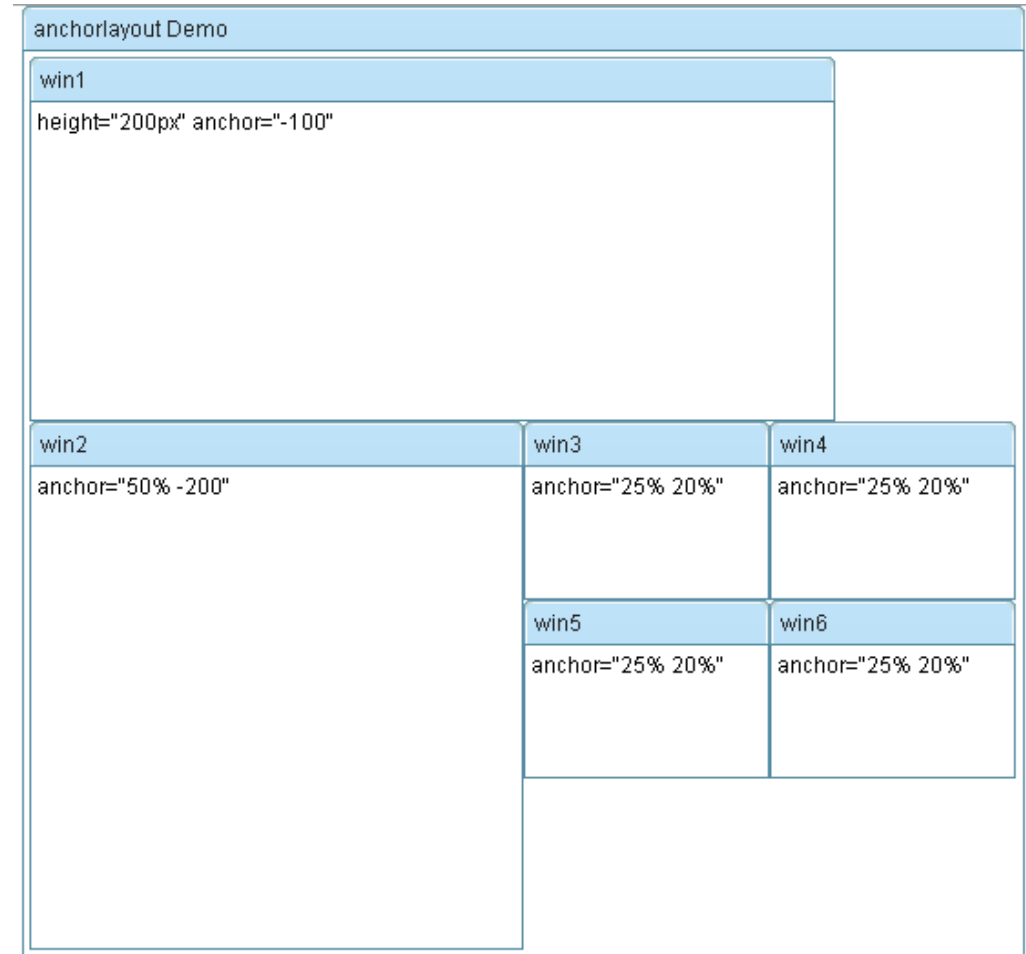
4. Một số thành phần giao diện cơ bản

- **Các thành phần đồ thị (diagrams)**
 - Chart: hỗ trợ nhiều loại đồ thị phong phú
 - Gmaps: hỗ trợ nhúng Gmap vào ứng dụng ZK
 - Timeline
 - Timeplot

4. Một số thành phần giao diện cơ bản

- **Các thành phần layout**

- Absolutelayout:
Được sử dụng để chứa đựng các thành phần có vị trí cố định
- Anchorlayout: Cho phép chứa đựng các thành phần con với kích thước có thể thay đổi được phụ thuộc vào chiều dài, chiều rộng của Anchorlayout



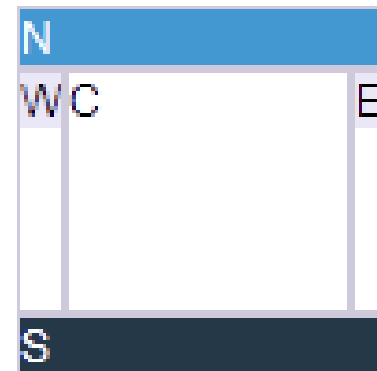
4. Một số thành phần giao diện cơ bản

- Các thành phần layout

- Borderlayout:

- Layout được chia thành 5 vùng chứa bắc, đông, tây, nam, trung tâm. Có thể khuyết một vùng nào đó.
 - Chiều cao của vùng bắc, nam được xác định trước, vùng trung tâm là phần còn lại, vùng đông, tây có kích thước bằng vùng trung tâm

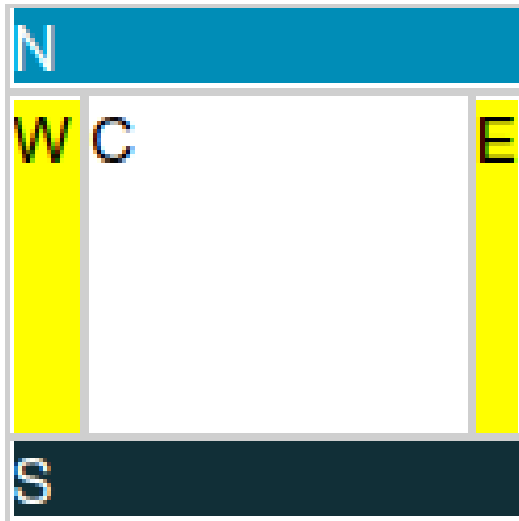
```
<borderlayout width="100px" height="100px">  
  <north>  
    <div style="background:#008db7;color:white;">N</div>  
  </north>  
  <south>  
    <div style="background:#112f37;color:white;">S</div>  
  </south>  
  <center>  
    <div>C</div>  
  </center>  
  <east>  
    <div style="background:#f2f2f2;">E</div>  
  </east>  
  <west>  
    <div style="background:#f2f2f2;">W</div>  
  </west>  
</borderlayout>
```



4. Một số thành phần giao diện cơ bản

- **Các thành phần layout**

- **Borderlayout:** Có thể cấu hình thuộc tính `vflex`, `flex` để các thành phần con vừa với phân vùng



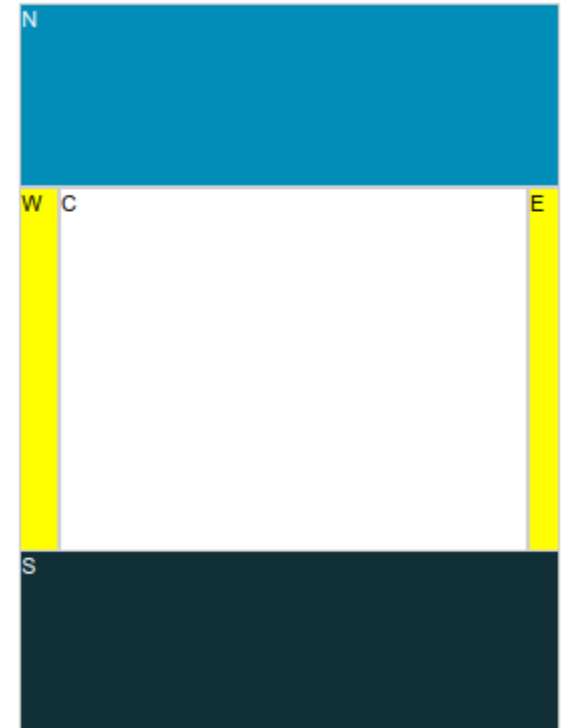
```
<borderlayout width="100px" height="100px">
  <north>
    <div style="background:#008db7;color:white;">N</div>
  </north>
  <south>
    <div style="background:#112f37;color:white;">S</div>
  </south>
  <center>
    <div>C</div>
  </center>
  <east>
    <div vflex="1" style="background:yellow;">E</div>
  </east>
  <west>
    <div vflex="1" style="background:yellow;">W</div>
  </west>
</borderlayout>
```

4. Một số thành phần giao diện cơ bản

- **Các thành phần layout**

- **Borderlayout**
: Có thể cấu hình thuộc tính `vflex`, `flex` để các phân vùng tự điều chỉnh vừa với thành phần con chứa bên trong

```
<borderlayout width="300px" vflex="min">
  <north vflex="min">
    <div height="100px" style="background:#008db7;color:white;">N</div>
  </north>
  <south vflex="min">
    <div height="100px" style="background:#112f37;color:white;">S</div>
  </south>
  <center vflex="min">
    <div height="200px">C</div>
  </center>
  <east flex="true">
    <div width="30px" style="background:#f2f2f2;">E</div>
  </east>
  <west flex="true">
    <div width="20px" style="background:#f2f2f2;">W</div>
  </west>
</borderlayout>
```



4. Một số thành phần giao diện cơ bản

- **Các thành phần layout**

Borderlayout: Nếu borderlayout có chiều cao cố định khi đó chiều cao của vùng trung tâm không co giãn theo thành phần còn chứa bên trong nên cần cấu hình chế độ thanh cuộn cho nó nếu cần

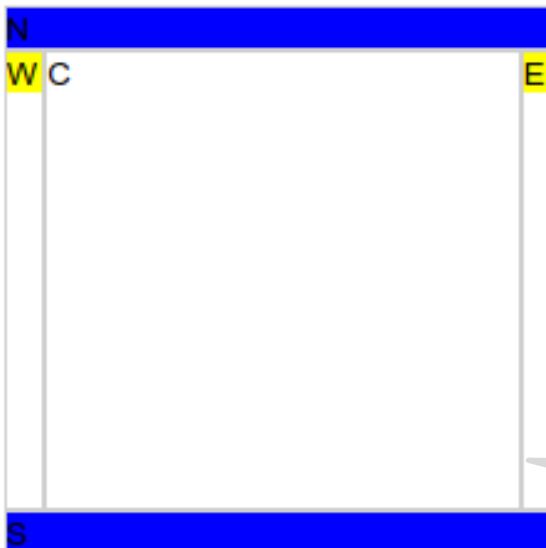


```
<borderlayout width="300px" height="300px">
  <north>
    <div height="100px" style="background:#008db7;color:white;">
      N
    </div>
  </north>
  <south>
    <div height="100px" style="background:#112f37;color:white;">
      S
    </div>
  </south>
  <center autoscroll="true">
    <div height="200px">C</div>
  </center>
  <east flex="true">
    <div width="30px" style="background:#f2f2f2;">E</div>
  </east>
  <west flex="true">
    <div width="20px" style="background:#f2f2f2;">W</div>
  </west>
</borderlayout>
```

4. Một số thành phần giao diện cơ bản

- Các thành phần layout**

Borderlayout: nếu đặt trong một khung chứa (vd window) sẽ không có giãn chiều cao theo kích thước thành phần con chứa bên trong nên phải đặt kích thước chiều cao cố định cho nó hoặc khung chứa phải có chiều cao cố định.



```
<window title="win" border="normal">
  <borderlayout height="200px" width="200px">
    <north>
      <div style="background:blue">N</div>
    </north>
    <south>
      <div style="background:blue">S</div>
    </south>
    <center>
      <div>C</div>
    </center>
    <east>
      <div style="background:yellow">E</div>
    </east>
    <west>
      <div style="background:yellow">W</div>
    </west>
  </borderlayout>
</window>
```

Bị mất khuôn dạng nếu bỏ thuộc tính height

4. Một số thành phần giao diện cơ bản

- **Các thành phần layout**

- Box: Dạng layout đơn giản như một hàng hay cột, các thành phần chứa bên trong có thể theo hướng ngang hoặc dọc

`<zk>`

`<box orient="vertical" spacing="2em">`

`<button label="Button 1"/>`

`<button label="Button 2"/>`

`</box>`

`<box orient="horizontal">`

`<button label="Button 3"/>`

`<button label="Button 4"/>`

`</box>`

`</zk>`

Button 1

Button 2

Button 3

Button 4

4. Một số thành phần giao diện cơ bản

- **Các thành phần layout**

- Box: thuộc tính căn lề sử dụng align (ngang) và pack (dọc)

<zk>

<panel border="normal" height="150px" width="100px">

<panelchildren>

<box align="end" pack="center" height="90%" width="100%">

<button label="Button 1" />

<button label="Button 2" />

</box>

</panelchildren>

</panel>

</zk>



pack / align		stretch	start	center	end
	<div>1</div> <div>2</div> <div>3</div>	<div>1</div> <div>2</div> <div>3</div>	<div>1</div> <div>2</div> <div>3</div>	<div>1</div> <div>2</div> <div>3</div>	<div>1</div> <div>2</div> <div>3</div>
start	<div>1</div> <div>2</div> <div>3</div>	<div>1</div> <div>2</div> <div>3</div>	<div>1</div> <div>2</div> <div>3</div>	<div>1</div> <div>2</div> <div>3</div>	<div>1</div> <div>2</div> <div>3</div>
center	<div>1</div> <div>2</div> <div>3</div>	<div>1</div> <div>2</div> <div>3</div>	<div>1</div> <div>2</div> <div>3</div>	<div>1</div> <div>2</div> <div>3</div>	<div>1</div> <div>2</div> <div>3</div>
end	<div>1</div> <div>2</div> <div>3</div>	<div>1</div> <div>2</div> <div>3</div>	<div>1</div> <div>2</div> <div>3</div>	<div>1</div> <div>2</div> <div>3</div>	<div>1</div> <div>2</div> <div>3</div>

4. Một số thành phần giao diện cơ bản

- **Các thành phần layout**

- Box: Có thể nhúng thành phần Cell vào trong Box để tổ chức layout và định style
- Box được dịch ra Table trong HTML, do đó nếu đặt các thành phần giao diện input như textbox, combobox trong box thì chỉ định chiều cao (rộng) của box sẽ không phản ánh đúng (Ví dụ cho thấy chiều rộng đã bị vượt quá 100px)

```
<box>  
  <cell sclass="years">  
    ...  
  </cell>  
</box>
```

1	1

```
<box orient="horizontal" height="100px"  
width="100px" style="border: 1px solid  
red">  
  <textbox hflex="1" value="1" />  
  <textbox hflex="1" value="1" />  
</box>
```

4. Một số thành phần giao diện cơ bản

- **Các thành phần layout**

- Hbox (Vbox): Kế thừa thuộc tính của Box, sử dụng Hbox (Vbox) giúp ta giảm bớt cấu hình thuộc tính orient như trong Box
- Hlayout (Vlayout): Tương tự như Hbox (Vbox), tuy nhiên Hlayout (Vlayout) không hỗ trợ align, pack, splitter. Hlayout (Vlayout) được dịch ra tương ứng thẻ Div trong HTML, trong trường hợp đơn giản, sử dụng Hlayout (Vlayout) cho hiệu suất cao hơn Hbox (Vbox)

4. Một số thành phần giao diện cơ bản

- **Các thành phần layout**

- Splitter: thanh ngăn cách các phân vùng được dùng trong Box (Hbox, VBox), có thể thu gọn (collapse) các thành phần trước hoặc sau nó tùy việc cấu hình

```
<hbox spacing="0" width="100%">
```

```
<vbox height="200px">
```

Column 1-1

```
<splitter collapse="after" />
```

Column 1-2

```
</vbox>
```

```
<splitter collapse="before" />
```

Column 2

```
</hbox>
```

Column 1-1: The left-top box. To know whether a splitter is collapsed, you can listen to the onOpen event.

Column 1-2: You can enforce to open or collapse programming by calling setOpen method.

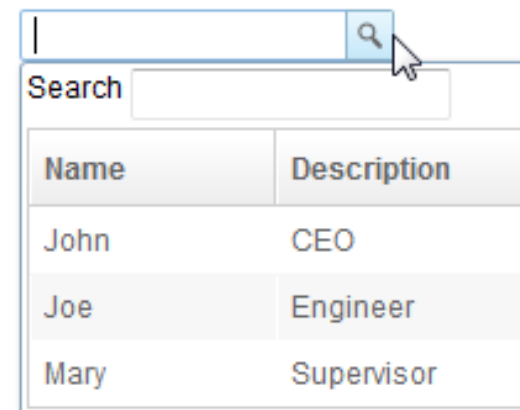
Column 2: Whether a splitter allows users to open or collapse depending on the collapse attribute.

4. Một số thành phần giao diện cơ bản

- **Các thành phần nhập dữ liệu (input)**

- **Bandbox:** một loại textbox đặc biệt, có kèm theo popup được kích hoạt khi bấm nút hoặc nhấn Alt+Down để cung cấp giá trị. Popup có thể chứa bất kỳ thành phần gì, thường là một hộp thoại tìm kiếm, giá trị từ popup có thể được lấy để đưa vào ô text của bandbox. Popup được đóng nếu bấm ESC hoặc Alt+Up

```
<bandbox id="bd">  
  <bandpopup>  
    // nội dung popup  
  </bandpopup>  
</bandbox>
```



4. Một số thành phần giao diện cơ bản

- **Các thành phần nhập dữ liệu (input)**

- Calendar

- `<calendar/>`

- Datebox

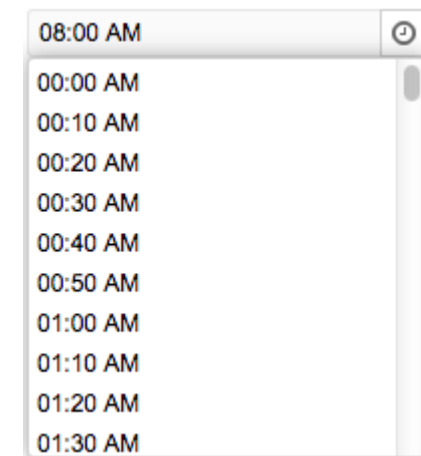
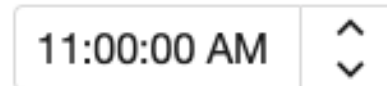
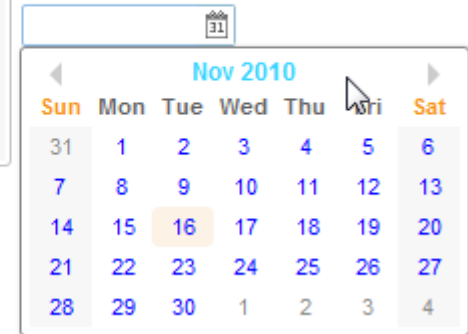
- `<datebox/>`

- Timebox

- `<timebox/>`

- Timepicker

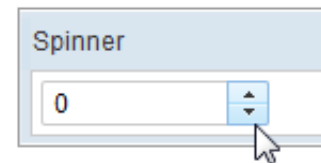
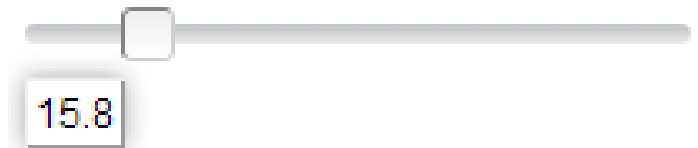
- `<timepicker/>`



4. Một số thành phần giao diện cơ bản

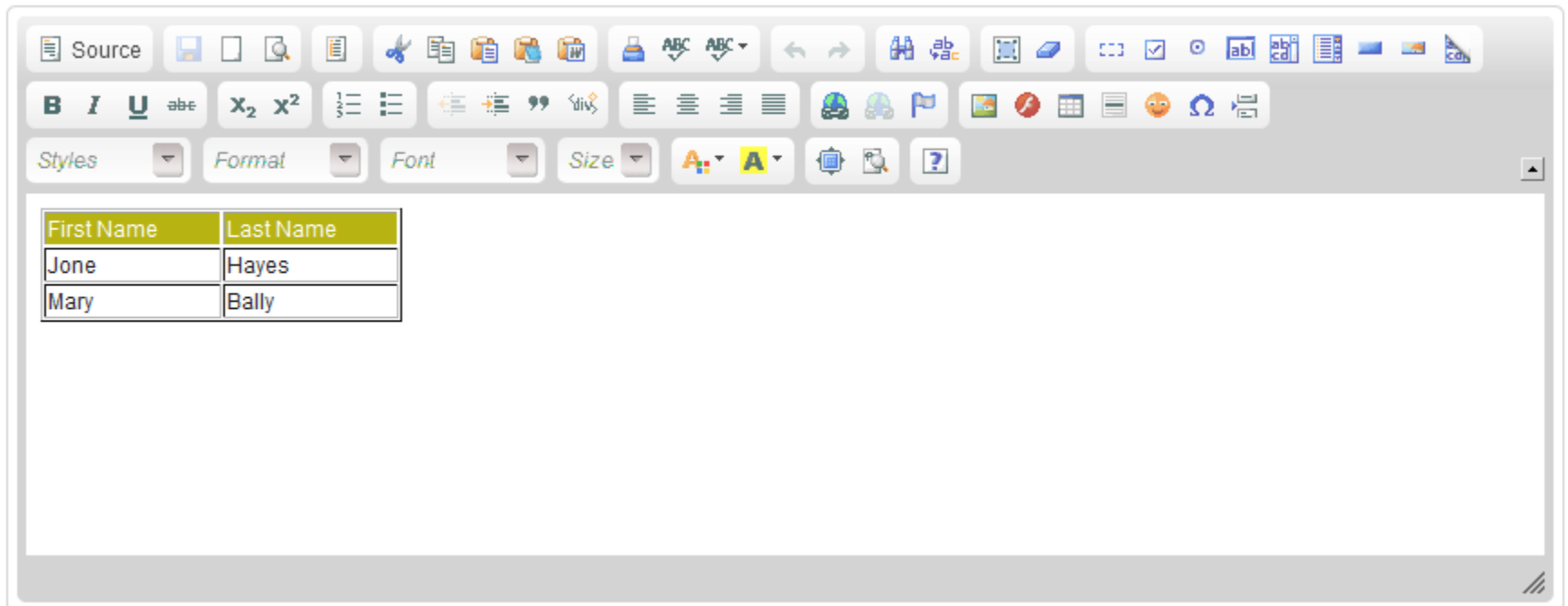
- **Các thành phần nhập dữ liệu (input)**

- Checkbox
- Combobox
- Radio, RadioGroup
- Textbox
- Slider: thanh chọn dạng trượt (ngang/dọc)
- Spinner: ô chọn giá trị nguyên
- Doublespinner



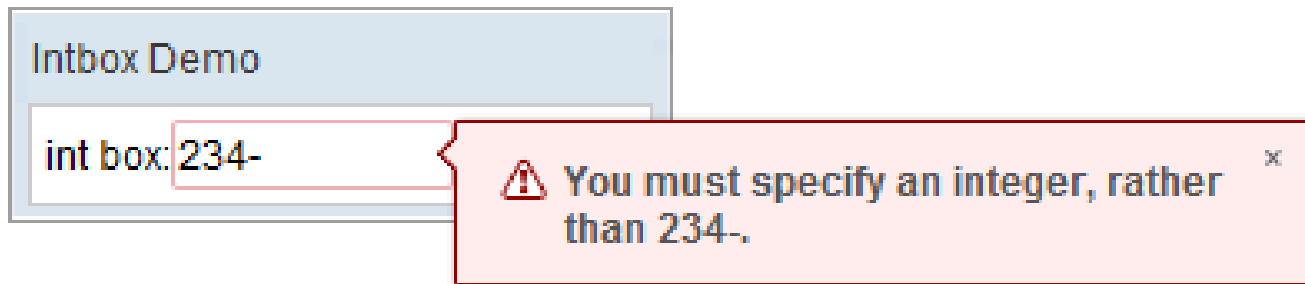
4. Một số thành phần giao diện cơ bản

- **Các thành phần nhập dữ liệu (input)**
 - Tbeditor: vùng soạn thảo đơn giản
 - CKEditor: vùng soạn thảo đầy đủ tính năng



4. Một số thành phần giao diện cơ bản

- **Các thành phần nhập dữ liệu (input)**
 - Decimalbox : ô dành cho nhập giá trị big decimal
 - Doublebox: ô dành cho nhập giá trị float hoặc double
 - Intbox: ô dành cho nhập số nguyên kiểu int
 - Longbox: ô dành cho nhập số nguyên kiểu long



```
1 <window title="Intbox Demo" border="normal" width="200px">
2   int box:<intbox/>
3 </window>
```

4. Một số thành phần giao diện cơ bản

- **Các thành phần cơ bản khác**

- **A**: tương tự thẻ a trong HTML, có thể thiết lập href, label, image, autodisable và thành phần con cho nó nếu cần

- Ví dụ:

```
<a href="http://www.zkoss.org" label="Visit ZK!"  
    image="https://www.zkoss.org/zksandbox/img/ZK-Logo.gif">
```

```
<grid>
```

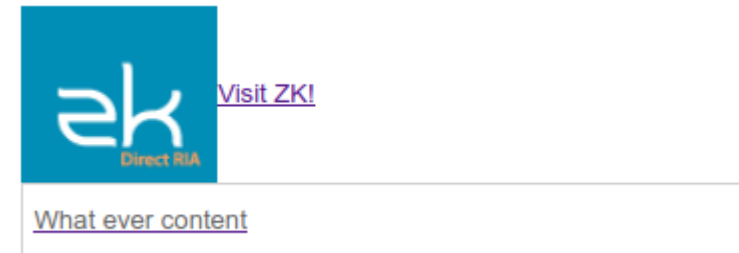
```
<rows>
```

```
<row>What ever content</row>
```

```
</rows>
```

```
</grid>
```

```
</a>
```



- *Autodisable*: tự động disable khi kích hoạt và chờ đến khi yêu cầu được phục vụ tự enable trở lại (tránh user click nhiều lần)
 - Ví dụ: ``

4. Một số thành phần giao diện cơ bản

- **Các thành phần cơ bản khác**

- Button: bao gồm các thuộc tính label, image, href, upload, autodisable

- Ví dụ:

- `<button label="Left" image="/img/network.gif" width="125px"/>`

- `<button label="Right" image="/img/network.gif" dir="reverse" width="125px"/>`

- `<button label="Above" image="/img/network.gif" orient="vertical" width="125px"/>`

- `<button label="Below" image="/img/network.gif" orient="vertical" dir="reverse" width="125px"/>`



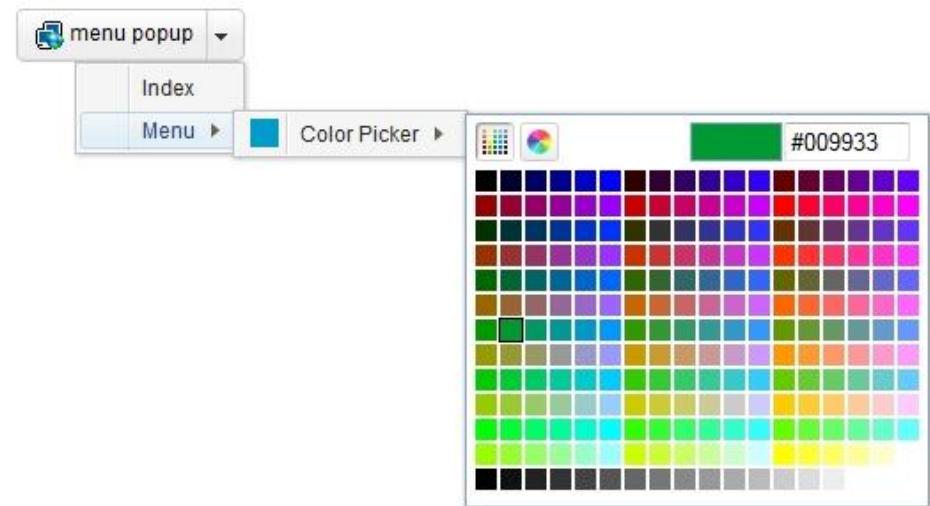
4. Một số thành phần giao diện cơ bản

- Các thành phần cơ bản khác

Combobutton: Một loại button đặc biệt với một popup hoặc menupopup kèm theo, được hiển thị khi user click

```
<combobutton label="popup"
  image="/img/network.gif">
  <popup>
    <vbox>
      .....
    </vbox>
  </popup>
</combobutton>
```

```
<combobutton label="menu popup"
  image="/img/network.gif">
  <menupopup>
    <menuitem label="Index" />
    <menu label="Menu">
      <menupopup><menu label="Color Picker"
content="#color=#029BCB" />
      </menupopup>
    </menu>
  </menupopup>
</combobutton>
```



4. Một số thành phần giao diện cơ bản

- **Các thành phần cơ bản khác**
 - **Fileupload:** có 2 cách upload file bằng cách dùng thẻ uploadfile hoặc thực hiện lệnh Fileupload.get()

- Ví dụ 1:

<zk>

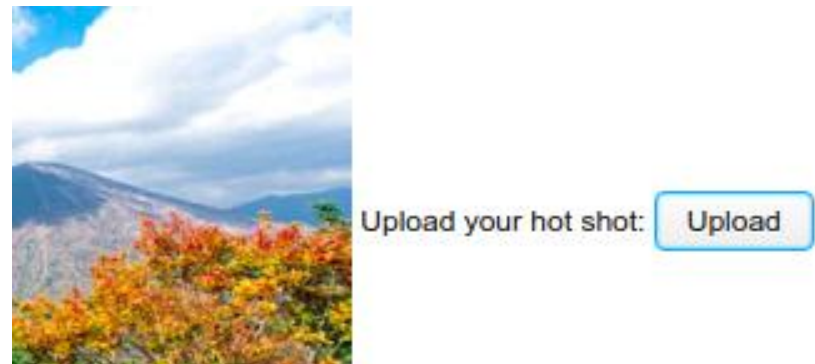
<image id="img" />

Upload your hot shot:

<fileupload label="Upload"

onUpload="img.setContent(event.media)" />

</zk>



4. Một số thành phần giao diện cơ bản

- Các thành phần cơ bản khác

- Fileupload:

- Ví dụ 2: Code xử lý file upload ko phải ảnh

<zk>

<fileupload label="Upload Image">

<attribute name="onUpload"><![CDATA[

org.zkoss.util.media.Media media = event.getMedia();

if(media instanceof org.zkoss.image.Image) {

org.zkoss.zul.Image image = new org.zkoss.zul.Image();

image.setContent(media);

image.setParent(*pics*);

} else {

MessageBox.show("Not an image: "+media, "Error", MessageBox.OK, MessageBox.ERROR);

break;

}

]]></attribute>

</fileupload>

<separator />

<vlayout id="*pics*" height="360px" style="overflow:auto"/>

</zk>



Có thể thay thẻ fileupload bởi thẻ button như sau:

<button label="Upload Image" upload="true, maxsize=300">

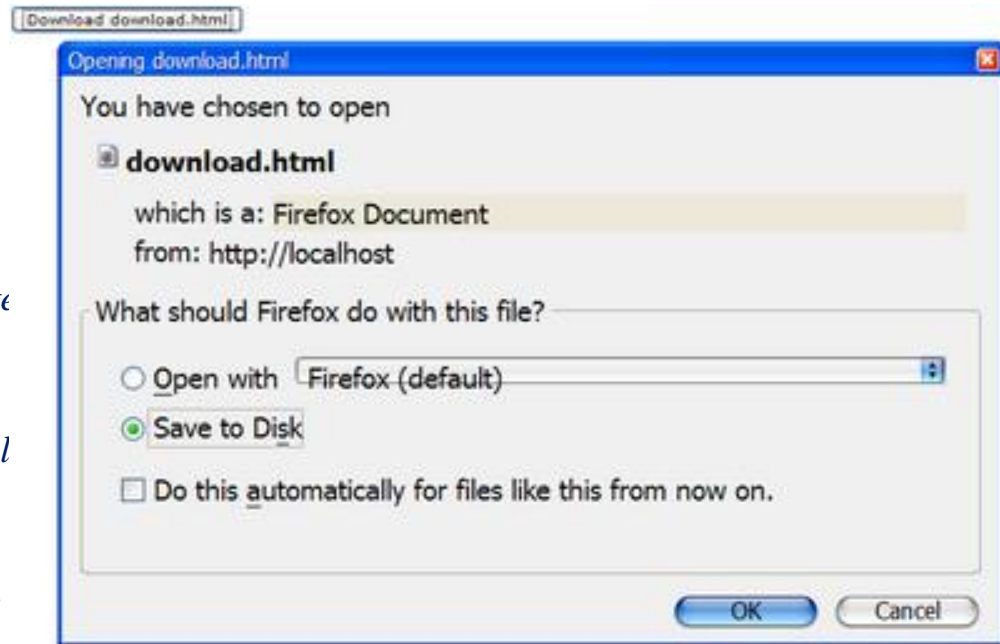
4. Một số thành phần giao diện cơ bản

- **Các thành phần cơ bản khác**

- **Filedownload:** là một dạng tiện ích hỗ trợ download, giúp mở hộp thoại để người dùng chọn vị trí tải xuống.

- Ví dụ:

```
<button label="Download">  
  <attribute name="onClick">{  
    java.io.InputStream is  
desktop.getApp().getResourceAsStream("/test/download.html");  
    if (is != null)  
      Filedownload.save(is, "text/html  
download.html");  
    else  
      alert("/test/download.html not found");  
  }  
</attribute>  
</button>
```



4. Một số thành phần giao diện cơ bản

- **Các thành phần cơ bản khác**

- Html: thẻ do ZK cung cấp cho phép nhúng đoạn code sử dụng thẻ HTML trong ZK. Các thẻ HTML này sẽ ko được coi là thành phần con của trang, nó được đặt trong thẻ span khi biên dịch thẻ html
- Ví dụ:

```
<html> <![CDATA[  
  <h4>Hi, ${win.title}</h4>  
  <p>It is the content of the html  
component.</p>  
]]>  
</html>
```



```
<span id="...">  
  <h4>Hi, Html Demo</h4>  
  <p>It is the content of the html  
component.</p>  
</span>
```

4. Một số thành phần giao diện cơ bản

- **Các thành phần cơ bản khác**

- Iframe: được dịch ra ứng với thẻ iframe trong HTML, iframe là một cửa sổ trình duyệt độc lập được nhúng trong một cửa sổ khác. Có thể thay đổi nội dung load trong iframe từ cửa sổ cha. Iframe có thể được dùng để hiển thị ảnh, file pdf hay một trang web
- Ví dụ:

```
<zk>
  <vbox height="300px">
    <iframe id="iframe" src="http://www.zkoss.org" vflex="1" width="40%"
      style="border: 1px solid gray" />

    <button upload="true" label="show data in iframe">
      <attribute name="onUpload">{
        Object media = event.getMedia();
        if (media != null)
          iframe.setContent(media);
      }</attribute>
    </button>
  </vbox>
</zk>
```



4. Một số thành phần giao diện cơ bản

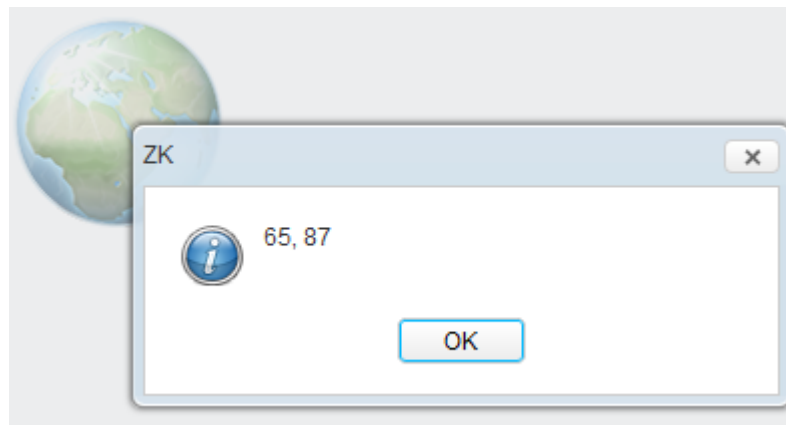
- **Các thành phần cơ bản khác**
 - **Include:** thẻ này được dùng để gắn kết quả thực hiện của một servlet khác vào trang hiện tại, servlet khác có thể là JSF, JSP hay một trang ZUML khác.
 - Nếu nội dung include là một trang ZUML khác, các thành phần của trang này sẽ được coi như một thành phần của trang include nó.
 - Ví dụ: Sử dụng include trong thiết kế layout trang web

```
<zk>
<borderlayout width="100%" height="100%">
  <north border="none" size="10%" class="header">
    <include src="header.zul"/>
  </north>
  <center border="none" flex="true" class="center">
    <hlayout>
      <div hflex="2"></div>
      <div hflex="5" vflex="1" class="center-left">
        <include src="sidebar.zul" class="center-left-
inner"/>
      </div>
      <div hflex="7" vflex="1" class="center-right">
        <include src="content.zul" />
      </div>
      <div hflex="2"></div>
    </hlayout>
  </center>
  <south border="none" class="footer" size="10%">
    <include src="footer.zul"/>
  </south>
</borderlayout>
</zk>
```

4. Một số thành phần giao diện cơ bản

- **Các thành phần cơ bản khác**

- Image: Thẻ hiển thị hình ảnh
 - Ví dụ: `<image src="/my.png">`
- Imagemap: Mang đầy đủ đặc tính của một hình ảnh, tuy nhiên khi người dùng click chuột vào ảnh, tọa độ của vị trí click được gửi về server, gốc tọa độ (0,0) tại vị trí góc cao bên trái của ảnh
 - Ví dụ: `<imagemap src="/img/sun.jpg" onClick="alert(event.x + "; " + event.y)"/>`



4. Một số thành phần giao diện cơ bản

- **Các thành phần cơ bản khác**

- **Label: Hiển thị một nhãn**

- Ví dụ:

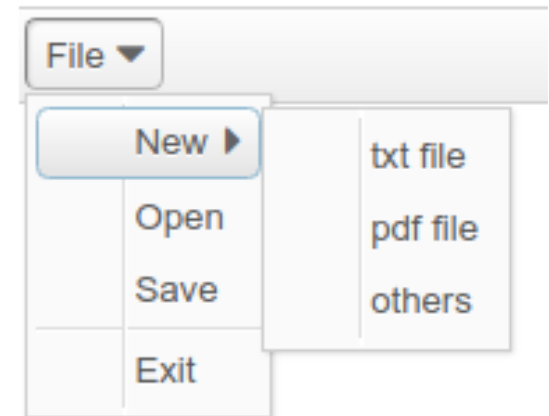
```
<window border="normal">  
    <label style="color: red" value="Hello World" />  
</window>
```



- **Menu: gồm các thành phần menubar, menu, menupopup**

- Ví dụ:

```
<menubar>  
  <menu label="File">  
    <menupopup>  
      <menu label="New">  
        <menupopup>  
          <menuitem label="txt file" onClick="alert(self.label)" />  
          <menuitem label="pdf file" onClick="alert(self.label)" />  
          <menuitem label="others" onClick="alert(self.label)" />  
        </menupopup>  
      </menu>  
      <menuitem label="Open" onClick="alert(self.label)" />  
      <menuitem label="Save" onClick="alert(self.label)" />  
      <menuseparator />  
      <menuitem label="Exit" onClick="alert(self.label)" />  
    </menupopup>  
  </menu>  
</menubar>
```



4. Một số thành phần giao diện cơ bản

- **Các thành phần cơ bản khác**

- Popup: có 3 loại popup, tooltip và context menu. Có thể thiết lập vị trí xuất hiện của popup như after_start, before_start..v.v..

Vd 1:

```
<zk>  
  <label value="Click Me!" popup="status,  
position=after_start"/>
```

```
  <popup id="status" width="300px">  
    <vlayout>  
      This user is online now !  
      <a label="Mail him/her"/>  
    </vlayout>  
  </popup>
```

```
</zk>
```

Click Me!

This user is online now !
[Mail him/her](#)

Vd 2:

```
<zk>  
  <label value="Mouse Over Me!"  
tooltip="status, position=after_start"/>
```

```
  <popup id="status" width="300px">  
    <vlayout>  
      This user is online now !  
      <a label="Mail him/her"/>  
    </vlayout>  
  </popup>
```

```
</zk>
```

Mouse Over Me!

This user is online now !
[Mail him/her](#)

Vd 3:

```
<zk>  
  <label value="Right Click Me!"  
context="status, position=after_start"/>
```

```
  <popup id="status" width="300px">  
    <vlayout>  
      This user is online now !  
      <a label="Mail him/her"/>  
    </vlayout>  
  </popup>
```

```
</zk>
```

Right Click Me!

This user is online now !
[Mail him/her](#)

4. Một số thành phần giao diện cơ bản

- **Các thành phần cơ bản khác**

- **Progressmeter**: thanh trạng thái chỉ tiến độ công việc, giá trị cần nằm trong khoảng từ 0 đến 100

- Ví dụ:

```
<zkc>  
  <zscript><![CDATA[  
    int percent = 0;  
    void go(int step) {  
      percent += step;  
      pm.setValue(percent);  
      progress_label.setValue(percent + "%");  
    }  
    void reset(){  
      pm.setValue(0);  
    }  
  ]]>  
</zscript>  
  
<vbox>  
  <progressmeter id="pm" value="0" width="300px" />  
  <label id="progress_label" value="0%" />  
  <button label="Reset" onClick='reset()' />  
  <button label="increase 10 percent" onClick='go(10)' />  
</vbox>  
</zkc>
```



4. Một số thành phần giao diện cơ bản

- **Các thành phần cơ bản khác**

- Rating: cho phép bình chọn

- Ví dụ: (*iconClass: thư viện tích hợp sẵn*)

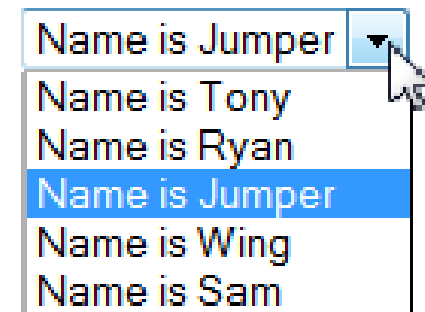
```
<vbox>
  <rating rating="2"/>
  <rating iconClass="z-icon-bolt" rating="3" max="7"/>
</vbox>
```



- Selectbox: một dạng lựa chọn danh sách đơn giản, có hỗ trợ nguồn dữ liệu tạo bởi ListModel, Renderer, Databinding

Ví dụ:

```
<zk>
  <zscript>
    <![CDATA[
      String[] userName = { "Tony", "Ryan", "Jumper", "Wing", "Sam" };
      ListModelList model = new ListModelList(userName);
    ]]></zscript>
    <selectbox model="${model}" onSelect='alert(model.get(event.getData()));'>
      <template name="model">
        Name is ${each}
      </template>
    </selectbox>
  </zk>
```



4. Một số thành phần giao diện cơ bản

- **Các thành phần cơ bản khác**

- **Separator:** được sử dụng để chèn vào một khoảng trống giữa hai thành phần
 - *Mặc định separator chèn vào một dòng trống giữa hai thành phần (hướng ngang). Có thể cấu hình thuộc tính `orient` để chèn theo hướng dọc, khi đó separator chèn vào một phím cách*
 - *Nếu để thuộc tính `bar = "true"`, dòng trống hay phím cách sẽ hiển thị dưới dạng đường kẻ ngang (dọc)*
 - *Sử dụng thuộc tính `spacing` để đặt độ dài separator*
- **Space:** là separator với mặc định hướng ngang
- Ví dụ:

<z<

line 1 by separator

<separator />

line 2 by separator

<separator bar="true" spacing="20px" />

line 3 by separator

<space bar="true" />

another piece

<separator spacing="20px" />

line 4 by separator

<space bar="true" spacing="20px" />

another piece

</z>

line 1 by separator

line 2 by separator

line 3 by separator | another piece

line 4 by separator | another piece

4. Một số thành phần giao diện cơ bản

- **Các thành phần cơ bản khác**

- **Script:** thẻ được sử dụng để nhúng mã script chạy phía trình duyệt client, thường là mã javascript
- Ví dụ:

```
<zk>
<window>
  <textbox id="hai" value="HAI"></textbox>
  <button id="export" label="Click"
      onClick='Clients.evalJavaScript("myFunction()")'>
  </button>

  <script type="text/javascript">
function myFunction()
{
  document.body.style.backgroundColor="lavender";
  alert(zk.Widget.$("$hai").getValue());
}
  </script>

</window>
</zk>
```

4. Một số thành phần giao diện cơ bản

- **Các thành phần cơ bản khác**

Style:

Thêm trực tiếp

```
<textbox style="color: red; font-style: oblique;" />
```

Thêm trực tiếp trong code java

```
Textbox tb = new Textbox();  
tb.setStyle("color: red; font-style: oblique;");
```

Một vài thành phần như window, groupbox cần đặt style cho khối nội dung của nó nếu cần

```
<window title="below is content" contentStyle="background:yellow; overflow:auto "> Hello </window>
```

Sử dụng sclass

```
<window>  
  <style>  
    .red {  
      color: blue;  
      font-style: oblique;  
    }  
  </style>  
  <textbox sclass="red" /> <!-- first textbox -->  
  <textbox sclass="red" /> <!-- another textbox -->  
</window>
```

Sử dụng zclass: thay thế class css mặc định của zk

```
<button zclass="btn" />
```

được biên dịch ra:

```
<button class="btn" />
```

Khác với class mặc định của zk:

```
<button class="z-button" />
```

4. Một số thành phần giao diện cơ bản

- **Các thành phần cơ bản khác**

- **Timer:** là một thành phần ẩn thường được dùng kết hợp thành phần khác để xử lý sự kiện sau những khoảng thời gian nhất định

- Ví dụ: hiển thị thời gian thay đổi theo giây trên label

<zk>

<label id="now" />

<timer id="timer" delay="1000" repeats="true"

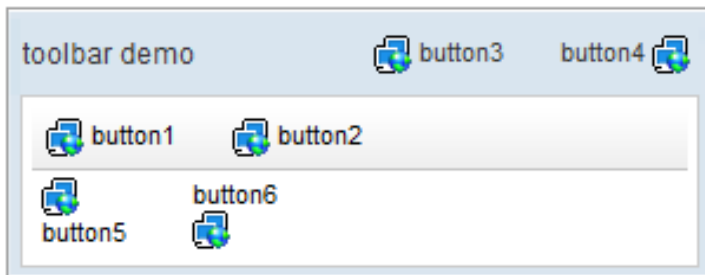
onTimer="now.setValue(new Date().toString())" />

</zk>

4. Một số thành phần giao diện cơ bản

- **Các thành phần cơ bản khác**

- **Toolbarbutton**: tương tự button, toolbarbutton có style hiển thị khác, đường viền nút bấm chỉ hiển thị khi toolbarbutton được focus, button được dịch ra thẻ button trong khi toolbarbutton được dịch ra thẻ div trong html



```
<window title="toolbar demo" border="normal" width="300px">
  <caption>
    <toolbarbutton label="button3" image="/img/network.gif" />
    <space />
    <toolbarbutton label="button4" image="/img/network.gif"
      dir="reverse" />
  </caption>
  <toolbar>
    <toolbarbutton label="button1" image="/img/network.gif" />
    <space />
    <toolbarbutton label="button2" image="/img/network.gif" />
  </toolbar>
  <hbox>
    <toolbarbutton label="button5" image="/img/network.gif"
      orient="vertical" />
    <space />
    <toolbarbutton label="button6" image="/img/network.gif"
      orient="vertical" dir="reverse" />
  </hbox>
</window>
```

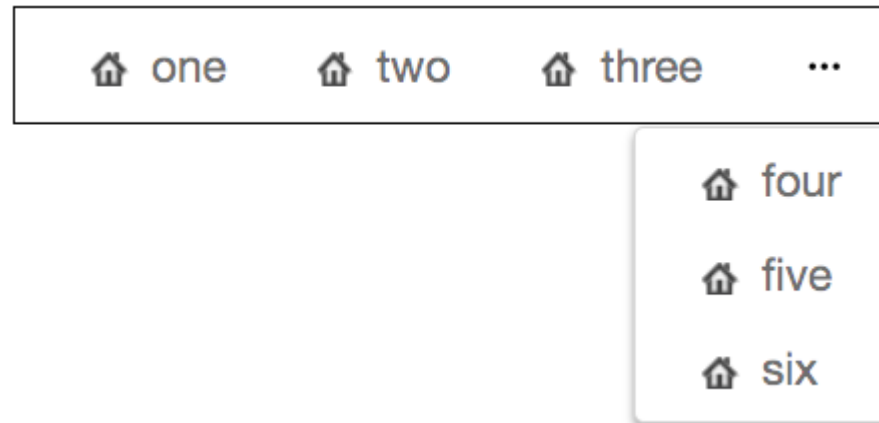
4. Một số thành phần giao diện cơ bản

- **Các thành phần cơ bản khác**

- **Toolbar:** được sử dụng để nhóm các button trong một khối.

- *Trong ví dụ về toolbarbutton, có thể thấy style của khối toolbar chứa các button khác style của khối hbox*
- *Nếu đặc tả thuộc tính overflowPopup="true" " khi số button vượt kích thước toolbar sẽ được thu bớt vào trong dấu ba chấm (...)*

```
<toolbar overflowPopup="true" width="350px" style="border: 1px black solid;">  
  <toolbarbutton label="one" iconSclass="z-icon-home"/>  
  <toolbarbutton label="two" iconSclass="z-icon-home"/>  
  <toolbarbutton label="three" iconSclass="z-icon-home"/>  
  <toolbarbutton label="four" iconSclass="z-icon-home"/>  
  <toolbarbutton label="five" iconSclass="z-icon-home"/>  
  <toolbarbutton label="six" iconSclass="z-icon-home"/>  
</toolbar>
```



4. Một số thành phần giao diện cơ bản

- **Các thành phần phụ trợ**

- Auxhead: thường được dùng trong grid để hiển thị tiêu đề của một nhóm các cột

<grid>

<auxhead>

<auxheader label="H1'07" colspan="6" />

<auxheader label="H2'07" colspan="6" />

</auxhead>

...

</grid>

H1'07						H2'07					
Q1			Q2			Q3			Q4		
Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
1,000	1,100	1,200	1,300	1,400	1,500	1,600	1,700	1,800	1,900	2,000	2,100

4. Một số thành phần giao diện cơ bản

- **Các thành phần phụ trợ**

- Paging: được dùng với grid, listbox, tree để phân trang

```
package demo.data;
import java.util.ArrayList;
public class BigList extends
ArrayList<Integer> {

    private int size;

    public BigList(int sz) {
        if (sz < 0)
            throw new
IllegalArgumentExcepion("Negative not
allowed: " + sz);
        size = sz;
    }

    public int size() {
        return size;
    }

    public Integer get(int j) {
        return Integer.valueOf(j);
    }
}
```

```
<zk>
<zscript><![CDATA[
    List items = new demo.data.BigList(1000); //a big list of Integer
]]></zscript>
<listbox id="listbox" mold="paging" pageSize="10">
    <listitem forEach="{items}">
        <listcell label="{each}-1" />
        <listcell label="{each}-2" />
        <listcell label="{each}-3" />
        <listcell label="{each}-4" />
    </listitem>
</listbox>
</zk>
```

10-1	10-2	10-3	10-4
11-1	11-2	11-3	11-4
12-1	12-2	12-3	12-4
13-1	13-2	13-3	13-4
14-1	14-2	14-3	14-4
15-1	15-2	15-3	15-4
16-1	16-2	16-3	16-4
17-1	17-2	17-3	17-4
18-1	18-2	18-3	18-4
19-1	19-2	19-3	19-4
« < 2 / 100 > » [11 - 20 / 1000]			

4. Một số thành phần giao diện cơ bản

- **Các thành phần phụ trợ**

- **Cell:** được sử dụng trong Hbox, Vbox để quản lý thành phần chứa trong nó. Cell trong Grid là các ô trong lưới, có thể áp dụng căn lề, rowspan, colspan nếu cần

<zk>

<grid>

<columns>

<column label="A" />

<column label="B" />

<column label="C" />

<column label="D" />

</columns>

A	B	C	D
item 1	item 2	item 3	
	item 4		item 5
	item 6	item 7	item 8
	item 9	item 10	item 11

</rows>

<row>

<cell rowspan="4" align="center" valign="bottom">

<label value="item 1" />

</cell>

<cell colspan="3">

<label value="item 2" />

</cell>

</row>

<row>

<cell colspan="2" align="center">

<label value="item 3" />

</cell>

<label value="item 4" />

</row>

<row>

<label value="item 5" />

<label value="item 6" />

<label value="item 7" />

</row>

<row>

<label value="item 8" />

<label value="item 9" />

<label value="item 10" />

</row>

</rows>

</grid>

</zk>

4. Một số thành phần giao diện cơ bản

- **Các thành phần phụ trợ**

- Frozen: tương tự thanh frozen trong MS excel được dùng để tạo chế độ thanh cuộn cho một số cột trong bảng. Thuộc tính columns cho biết số cột bỏ qua ko cuộn, thuộc tính start cho biết vị trí thanh cuộn khởi tạo ở vị trí cột (columns + start)

```
<grid width="600px">  
  <frozen columns="2" start="1"/>  
  <columns>  
    <column width="50px">ID</column>  
    <column width="50px">Priority</column>  
    <column width="50px">Status</column>  
    <column width="150px">Summary</column>  
    <column width="250px">Detail</column>  
    <column width="100px">Group</column>  
    <column width="50px">Assign</column>  
  </columns>
```

ID	Priority	Summary	Detail	Group
0001	1	Fix login issue	Login does not work at all	Account
0002	3	Button style broken	Check main.css	Styling
0003	2	Client search result	Search service returns incomplete result	Service

```
</rows>  
  <row>  
    <cell>0001</cell>  
    <cell>1</cell>  
    <cell>closed</cell>  
    <cell>Fix login issue</cell>  
    <cell>Login does not work at all</cell>  
    <cell>Account</cell>  
    <cell>Bob</cell>  
  </row>  
  <row>  
    <cell>0002</cell>  
    <cell>3</cell>  
    <cell>open</cell>  
    <cell>Button style broken</cell>  
    <cell>Check main.css</cell>  
    <cell>Styling</cell>  
    <cell>Alice</cell>  
  </row>  
  <row>  
    <cell>0003</cell>  
    <cell>2</cell>  
    <cell>open</cell>  
    <cell>Client search result</cell>  
    <cell>Search service returns incomplete result</cell>  
    <cell>Service</cell>  
    <cell>Bob</cell>  
  </row>  
</rows>  
</grid>
```

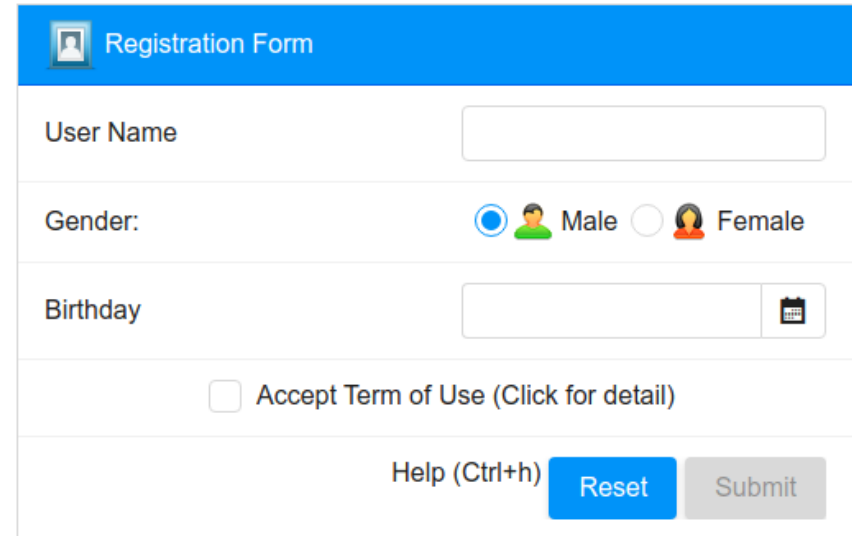
4. Một số thành phần giao diện cơ bản

- **Các thành phần Multimedia và Miscellaneous**
 - Audio
 - Video
 - Camera
 - Flash

5. Ứng dụng làm quen

- **Ứng dụng đăng ký user**

- Nếu người dùng không hiểu trường nào đó trong form, có thể click chuột vào form và bấm Ctrl+H để hiện bảng trợ giúp
- Nếu người dùng muốn đọc điều khoản, click chuột vào nhận Accept Term... Để hiện bảng điều khoản
- Nếu người dùng nhập thiếu một trường username, birthday cần có thông báo nhắc nhở nhập dữ liệu
- Khi người dùng check accept, nút Submit được enable, người dùng có thể nhấn Submit hoặc gõ Enter để gửi đi thông tin đăng ký
- Nếu người dùng bấm Reset, đưa các trường trong form về trạng thái ban đầu



The image shows a web form titled "Registration Form" with a blue header. It contains the following fields and controls:

- User Name:** A text input field.
- Gender:** Two radio buttons with icons. The first is selected (blue circle) and labeled "Male" with a green male icon. The second is unselected (white circle) and labeled "Female" with a red female icon.
- Birthday:** A date picker input field with a calendar icon on the right.
- Accept Term of Use:** A checkbox followed by the text "Accept Term of Use (Click for detail)".
- Buttons:** At the bottom right, there is a "Help (Ctrl+h)" link, a blue "Reset" button, and a grey "Submit" button.

Tài liệu tham khảo

- <https://www.zkoss.org/>