

Ajax4JSF

A [JavaBeat](#) portál [JSF](#) szakosztálya [Shunmuga Raja](#) személyében közzétett egy [Ajax4JSF gyorstalpalót](#). A leírás a fogalmak tisztázásával kezdődik, vagyis mi az AJAX, mi a JSF, mit jelent a kettő együtt, majd a lovak közé csap, és bemutatja a keretrendszer lényegi elemeit. Az egyik leglényegesebb elem a régiókba záras, amelyeknek törzse nem vesz részt az oldal felépítésében, viszont a megadott azonosító alapján a kliens oldali JavaScript vissza tud kérdezni a szerver oldalra:

```
<a4j:region id = "employeeRegion" actionListener = "#{manager.processEmpData}">
  <!-- Other Set of Components -->
</a4j:region>
```

A másik lényeges elem a `commandButton`, amely JSF esemény eltt egy kliensoldali AJAX eseményt vált ki:

```
<a4j:commandButton id = "clickButton" value = "Click Me" action = "#{Listener.processClickRequest}" oncomplete
= "jsFunction()">
  reRender = "CompId1, CompId2, CompId3"
</a4j:commandButton>
```

A harmadik lényegese elem a `poll`, amely rendszeres időközönként visszakérdez a szerver oldalra:

```
<a4j:poll id = "refreshId" reRender = "CompId1, CompId2" timeout = "50000" interval = "1000">
</a4j:poll>
```

Ezeknél túl sok apróság található meg a keretrendszerben, mint például a képek és stream-ek kezelésére szolgáló `mediaOutput`, amely a kliensoldali JavaScript működése alapján a szerverről letölti a megfelelő adatot:

```
<a4j:mediaOutput id = "telephone" element = "img" mimeType = "image/png" createContent = "#{ImagePainter.
drawImage}">
</a4j:mediaOutput>
```

A `keepAlive` életben tartja a kérés idejére szóló MBean-t:

```
<a4j:keepAlive beanName = "#{manager.empBean}"></a4j:keepAlive>
```

Kliensoldalon elálló ciklusszervezés alapján ismételtethet a JSF komponensek:

```
<a4j:repeat id = "repeater" value = "#{manager.empList}" var = "employee">
  <h:outputText id = "name" value = "#{employee.name}" />
  <h:outputText id = "age" value = "#{employee.age}" />
</a4j:repeat>
```

A státuszsor kezeléséhez szükséges `status` tag:

```
<a4j:status startText = "Request sent to the Server" stopText = "Request is completed">
</a4j:status>
```

Események összekötésére szükséges `support` tag, amellyel meghatározott JavaScript eseményekre akár több komponenst is újra tudunk tölteni:

```
<a4j:inputText id = "userInput">
  <a4j:support event = 'onblur' action = "#{bean.process}" binding = "bean.userInput" reRender = "CompId1,
CompId2" />
</a4j:inputText>
```

A leírás harmadik részében egy példaalkalmazás elkészítésébe tekinthetünk bele, természetesen bven dokumentálva és részletes leírásokkal...