

Pengenalan Framework Jakarta Struts

Struts merupakan aplikasi framework open source yang di design untuk membantu para developer dalam membangun arsitektur aplikasi berbasis pada Model-View-Controller (MVC) , dimana design MVC sudah merupakan standard umum dalam membangun aplikasi Web anda dengan penggunaan servlet Java dan Java Server Pages (JSP) sebagai teknologinya.

Penulis disini mengasumsikan bahwa para pembaca sudah mempunyai pengetahuan dasar mengenai JSP, servlets, tag library JSP dan XML. Jika belum, penulis menganjurkan anda untuk membaca beberapa tutorial dasar mengenai JSP dan XML yang pernah ditulis penulis dan telah diposting di Indo-Tektips.com.

The Apache Jakarta Project
<http://jakarta.apache.org/>

Struts

Struts

Welcome to Struts

Welcome to Struts! The goal of this project is to provide an open source framework for building Java web applications.

The core of the Struts framework is a flexible control layer based on standard technologies like Java Servlets, JavaBeans, ResourceBundles, and XML, as well as various Jakarta Commons packages. Struts encourages application architectures based on the Model 2 approach, a variation of the classic Model-View-Controller (MVC) design paradigm.

Struts provides its own Controller component and integrates with other technologies to provide the Model and the View. For the Model, Struts can interact with standard data access technologies, like JDBC and EJB, as well as most any third-party packages, like Hibernate, iBATIS, or Object Relational Bridge. For the View, Struts works well with JavaServer Pages, including JSTL and JSP, as well as Velocity Templates, XSLT, and other presentation systems.

The Struts framework provides the invisible underpinnings every professional web application needs to survive. Struts helps you create an extensible development environment for your application, based on published standards and proven design patterns.

Struts is part of the Apache Jakarta Project, sponsored by the Apache Software Foundation. The official Struts home page is at <http://jakarta.apache.org/struts>.

Struts in a Nutshell

A web application uses a deployment descriptor to minimize resources like servlets and tags. The deployment descriptor is formatted as a XML document and named "web.xml". Likewise, Struts uses a configuration file to initialize

Mari kita mulai.

Struts merupakan framework Model-View-Controller (MVC) , berbasis pada teknologi servlet dan JSP. Framework ini merupakan bagian dari proyek Apache Jakarta, dan untuk situs resmi Struts anda bisa akses langsung di sini : <http://jakarta.apache.org/struts>. Framework ini menarik perhatian khusus terutama bagi para developer yang banyak bermain di front-end development, terutama JSP dan Servlet dalam beberapa tahun terakhir ini, termasuk penulis sendiri tentunya. Ini tentunya tidak lain dikarenakan kemampuan Struts dalam membantu kita , memudahkan kita dalam merancang aplikasi berbasis MVC secara lebih mudah , cepat dan mengikuti semua aturan main framework ini. Penulis juga menjumpai beberapa design pattern yang diterapkan oleh framework ini, dan ini tentunya sangat menguntungkan bagi kita, karena pada layar ini, secara traspalansi dan tidak langsung telah kita implementasikan melalui framework Struts.

Pada saat penulis membuat tutorial perdana mengenai pengenalan Struts, versi 1.1 Struts telah direlease.

Framework Struts dapat dipecahkan menjadi menjadi 3 area utama, sesuai design MVC :

1.) Model

Dimana implementasinya disini berupa class Action, menyediakan logik bisnis yang harus ditulis oleh kita, developer. Bagaimana nantinya interaksi antara Controller dan class Action

yang kita progra, nantinya akan berbasis pada sebuah file konfigurasi yang dapat kita modifikasi secara mudah lewat file yang namanya : struts-config.xml.

2. View

View merupakan sekumpulan tag library dari JSP, untuk memudahkan kita dalam membuat form secara cepat. (anda bisa bayangkan form Struts disini hampir sama dengan form HTML yang biasa anda buat.)

3. Controller

Merupakan sebuah servlet, yang akan mengtransfer semua request yang datang dari HTTP ke Action yang telah diset melalui file xml yang sudah dijelaskan diatas.

Sekarang, mari kita mempelajari lebih detail berbagai bagian dari framework ini dan mendapat pemahaman bagaimana interaksi antar obyek class di dalam framework ini.

Konfigurasi (Configuration)

Sebelum bisa menggunakan Struts, anda harus mengsetup kontainer JSP anda, supaya tahu bagaimana harus memforward semua request yang datang melalui HTTP ke extensi file yang sesuai ke Controller Struts.

Ini bisa dilakukan lewat file web.xml yang dibaca pada saat sebuah kontainer JSP di start. (contoh : Tomcat).

Berikut dibawah ini, penulis memberikan contoh isi sebuah file web.xml yang mengdefinisikan sebuah konfigurasi sederhana

untuk mengsetting lingkungan kerja dengan Struts, berikut komentar dan penjelasan :

```
<web-app>
  <servlet>
    <!--
    Deklarasi > Indo_tektips_Action_Servlet sebagai tipe ActionServlet dari
    framework ini. Lokasi konfigurasi file juga diset melalui element param. -->
    <servlet-name> Indo_tektips_Action_Servlet</servlet-name>
    <servlet-class>org.apache.struts.action.ActionServlet</servlet-
class>
    <init-param>
      <param-name>config</param-name>
      <param-value>/WEB-INF/struts-config.xml</param-value>
    </init-param>
    <init-param>
      <param-name>debug</param-name>
      <param-value>2</param-value>
    </init-param>
    <init-param>
      <param-name>detail</param-name>
      <param-value>2</param-value>
    </init-param>
    <load-on-startup>2</load-on-startup>
  </servlet>
  <!--
  Semua HTTP request yang datang dan mempunyai file ekstensi .do, send akan
  ditransfer ke Controller Indo_tektips_Action_Servlet -->
  <servlet-mapping>
```

```
<servlet-name>> Indo_tektips_Action_Servlet</servlet-name>
<url-pattern>*.do</url-pattern>
</servlet-mapping>
<!--
  Default page
-->
<welcome-file-list><welcome-file>login.jsp</welcome-
file></welcome-file-list>
<!--
  Bagian dibawah ini adalah default, anda bisa copy dan paste, merupakan
  deklarasi untuk tag lig JSP Struts.
-->
<taglib>
  <taglib-uri>/WEB-INF/struts-bean.tld</taglib-uri>
  <taglib-location>/WEB-INF/struts-bean.tld</taglib-location>
</taglib>
<taglib>
  <taglib-uri>/WEB-INF/struts-html.tld</taglib-uri>
  <taglib-location>/WEB-INF/struts-html.tld</taglib-location>
</taglib>
<taglib>
  <taglib-uri>/WEB-INF/struts-logic.tld</taglib-uri>
  <taglib-location>/WEB-INF/struts-logic.tld</taglib-location>
</taglib>
</web-app>
```

Controller merupakan bagian penting dalam framework ini, dimana semua traffic dengan ekstensi file yang kita set diatas, akan dimonitor oleh Controller. Controller juga bertugas melakukan

packaging dan routing HTTP request ke obyek logik bisnis yang benar (bisa berupa halaman JSP atau class Action)

Controller merupakan representasi dari file web.xml file dalam bentuk instance dari `org.apache.struts.action.ActionServlet`. Pada contoh kita diatas, *Indo_tektips_Action_Servlet*. Ketika kontrol di inisialisasi, dia akan membaca file konfigurasi (`struts-config.xml`) yang mengdefinisikan mapping action untuk aplikasi yang kita buat. Controller menggunakan mapping action ini untuk menentukan langkah selanjutnya, dimana HTTP request akan di kirimkan (ditujukan ke class Action yang kita implementasikan). Kemudian class Action kita akan mengambil alih dan melakukan proses logik bisnis kita sebagaimana yang kita inginkan. Point yang cukup penting untuk diketahui, adalah bahwa obyek Action mempunyai akses ke method-method servlet.

Ini merupakan fitur yang cukup powerful, ketika obyek Action harus memforward kontrol ke obyek Action yang lain sebagai pemisalan.

Model

Merupakan obyek yang akan menyimpan request dari pengguna (user) selama proses berlangsung, biasanya implementasi berupa `JavaBean`. Obyek model biasanya bersifat application specific, karena implementasi daripada logik bisnis akan sangat tergantung kepada kepentingan tujuan dari aplikasi yang kita bangun.

Struts menyediakan class `ActionForm` dan `Action`. Melalui kedua model ini lah, dimana sebuah validasi form berlangsung, ataupun pra proses terhadap data berlangsung (misalkan : memeriksa apakah nama anggota sudah diisi, alamat emal, dll sbgnya.)

Struts-config.xml

struts-config.xml bisa anda anggap sebagai tempat dimana anda menempelkan semua implementasi class Action, model anda menjadi satu. Semua aturan main, dan flow daripada aplikasi yang anda bangun berbasis Struts ada disini ☺. Jika web.xml memberitahukan container anda dimana sebuah request harus ditransfer setelah sampai ke container anda, maka struts-config.xml memberitahukan kepada container apa yang harus dilakukan setelah mencapai layar ini. Siapa yang mengatur request yang datang dan harus di kirim ke siapa request tersebut untuk di proses ? Controller jawabannya. Bagaiman Controller tahu harus mengirimkan request ini kepada Action yang mana ? Mapping Action yang anda set di dalam struts-config.xml jawabannya ☺. Mudah bukan ?

Anda bisa lihat fleksibilitas Struts disini, dimana framewrok ini mencegah kita daripada melakukan hard code bagaimana cara memanggil, berinteraksi dengan component lain, lewat sebuah file konfigurasi. Ingin mengubah flow aplikasi anda nantinya ? Anda bisa ubah on the fly dari file konfigurasi ini.

Dibawah ini penulis memberitahukan sebuah contoh sederhana, bagaiman kemampuan Struts Controller mengatur traffic obyek model yang anda buat, lewat file konfigurasi yang telah kita sebutkan di atas.

```
<action-mappings>  
  <action path="/Login"
```

```
        type="com.indo-tektips.actions.LoginAction">  
        <forward name="sukses" path="/HalamanDepan.do"/>  
        <forward name="gagal" path="/login.jsp"/>  
    </action>  
</action-mappings>  
</struts-config>
```

Penjelasan :

Dari mapping diatas, jika katakanlah kontainer JSP anda dijalankan pada localhost:8080, dengan nama aplikasi web anda : portal_indonesia. Maka pemisalan user di browser mengetikan : http://localhost:8080/portal_indonesia/Login.jsp, dan setelah user mengisi informasi dan klik tombol "Submit", maka request ini pertama oleh kontainer anda akan di kirimkan kepada Controller Struts anda, dan Controller kemudian berdasarkan file konfigurasi diatas, akan melanjutkan proses ke class Action yang anda implementasikan (contoh diatas : LoginAction). Nantinya anda juga akan mempelajari bagaimana di dalam sebuah logik bisnis class Action anda, anda bisa melakukan suatu kondisi validasi, pemisalan : jika sesuatu gagal, maka anda menginginkan Controller untuk melanjutkan proses ke halaman /login.jsp, atau jika validasi berhasil, ke /HalamanDepan.do.

Pada tutorial lanjutan nantinya, penulis akan mengajak para pembaca untuk langsung praktek dalam bentuk contoh-contoh.

Sampai ketemu kembali di edisi berikutnya !

Jaya IT Indonesia ! - Anen – anenscwbs2002@yahoo.com

