


Capturing Input



Struts University Series




Capturing Input

- How do HTML forms capture input?
 - What do dynamic forms add?
 - How do we handle input errors?
 - How does input get to the database?
- 




Capturing Input

- HTML Forms
 - Dynamic Forms
 - Error Handling
 - Finding the Database
- 



HTML Forms

- Section of a document containing content, markup, controls and labels
 - checkboxes, radio buttons, menus
 - Users complete and submit a form
 - Agent processes input
 - CGI Program, Web Application, Web Service
- 

HTML Forms



HTML Forms

```
<FORM action="http://somesite.com/prog/adduser" method="post">
  <P>
    <LABEL for="firstname">First name: </LABEL>
      <INPUT type="text" id="firstname"><BR>
    <LABEL for="lastname">Last name: </LABEL>
      <INPUT type="text" id="lastname"><BR>
    <LABEL for="email">email: </LABEL>
      <INPUT type="text" id="email"><BR>
    <INPUT type="radio" name="gender" value="Male"> Male<BR>
    <INPUT type="radio" name="gender" value="Female"> Female<BR>
    <INPUT type="submit" value="Send"> <INPUT type="reset">
  </P>
</FORM>
```

HTML Forms

First name:

Last name:

email:

Male

Female



Control Types

 Button

 Checkbox

 Radio Button

 Menu

 Text Input

 File Select

 Hidden

 Object






FORM Element

<!ELEMENT FORM >

<!ATTLIST FORM

action	%URI ;
method	(GET POST)
enctype	%ContentType ;
accept	%ContentTypes ;
name	CDATA
onsubmit	%Script ;
onreset	%Script ;
accept-charset	%Charsets ;
>	





INPUT Element

 text

 password

 checkbox

 radio

 submit

 image

 reset

 button


 hidden

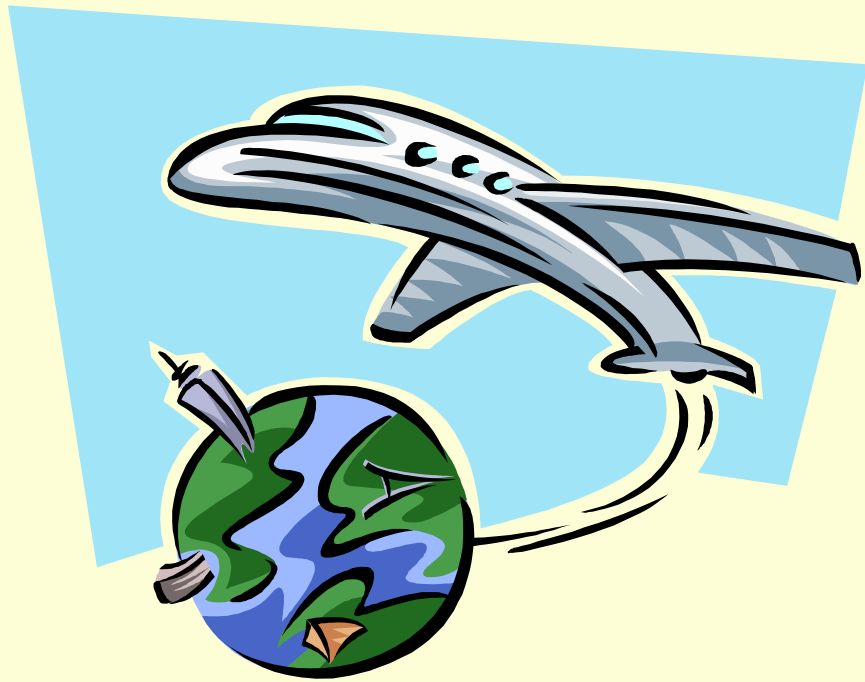
 file






Other Form Elements

- BUTTON
 - SELECT
 - OPTION
 - OPTGROUP
 - TEXTAREA
- 






Review

- The `*****` control does not submit a name or value when selected.
 - The `*****` control displays its value but the value cannot be changed.
 - Controls (can or cannot) have duplicate names.
- 




Review

- The checkbox control does not submit a name or value when selected.
 - The button control displays its value but the value cannot be changed.
 - Controls (can or cannot) have duplicate names (but should have unique IDs).
- 




Capturing Input

- HTML Forms
 - **Dynamic Forms**
 - Error Handling
 - Finding the Database
- 



Dynamic Forms

- Default input values
 - Default menu items
 - Set labels
 - Localize labels
 - Validate input
- 



Dynamic Form Tags

 form

 checkbox

 file

 hidden

 label

 password

 radio

 reset

 select

 submit

 textArea





Advanced Form Tags

 checkboxlist

 combobox

 datepicker

 doubleselect

 head

 optiontransferselect

 token


 updownselect

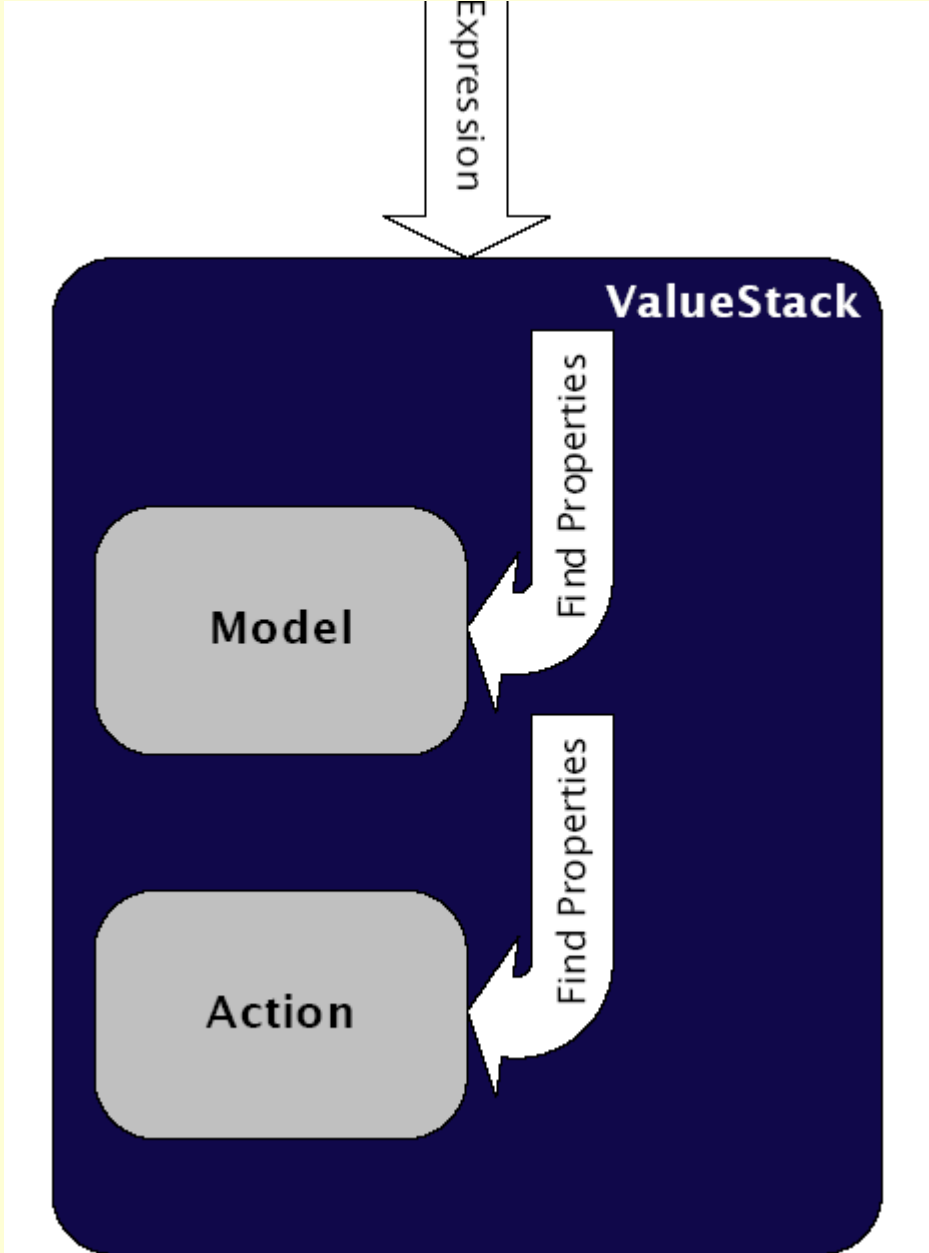
(Covered by another session)






Value Stack

- The Value Stack builds an ordered list of objects
 - Objects provide property values
 - The Value Stack enables expressions to find properties across multiple objects
- 






Value Stack

- The last object “pushed” on the stack has the nearest scope
 - Objects may be “pushed” and “popped” as a page is being rendered
- 




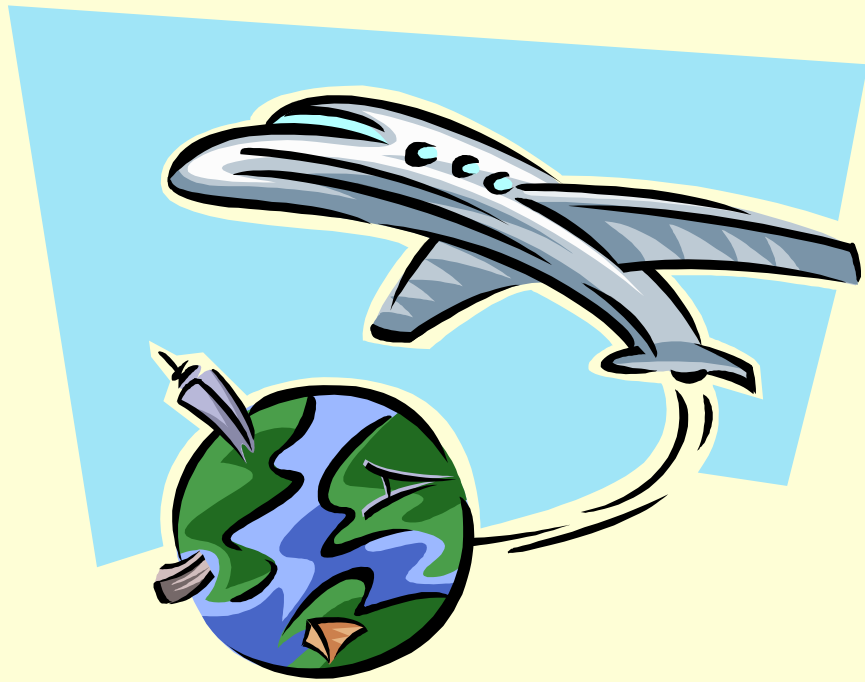
Tag Syntax

- Some tags may push and pop values from the stack at runtime
 - Some values not be available outside the scope of the tag
 - Initially, attributes are applied as Strings
 - Strings evaluated for `%{ ... }`
 - Anything in between the braces is evaluated against the value stack.
- 




Tag Syntax

- All String attribute types are parsed for the `%{ ... }` characters.
 - All non-String attribute types are not parsed, but instead evaluated directly
 - Except if the non-String attribute starts with `%{` and ends with `}`
 - Those characters are cut off before evaluating the expression.
- 






Review

- The Value Stack builds an ********* list of objects
 - Objects provide ********* values
 - The Value Stack enables expressions to find properties across ********* objects
 - Objects may be **“*****”** and **“*****”** as a page is being rendered
- 



Review

- The Value Stack builds an ordered list of objects
 - Objects provide property values
 - The Value Stack enables expressions to find properties across multiple objects
 - Objects may be “pushed” and “popped” as a page is being rendered
- 



Examples


```
<s:textfield label="%{getText("state.label")}" name="state"/>
```

```
<s:select label="%{getText("state.label")}" name="state"  
multiple="true"/>
```

```
<s:select label="%{getText("state.label")}" name="state"  
multiple="%{allowMultiple}"/>
```


```
<s:textfield label="%{getText("state.label")}" name="state"  
value="CA"/>
```

```
<s:textfield label="%{getText("state.label")}" name="state"  
value="%{'CA'}"/>
```





Capturing Input

- HTML Forms
 - Dynamic Forms
 - **Error Handling**
 - Finding the Database
- 



Error Handling

● Field Errors

- Error displays above field

● Action Errors

- Error displays at `actionError` tag

```
<s:actionerror />  
  <s:form . . . . >  
    . . . .  
  </s:form>
```





Bypassing Validation


```
<s:submit action="Register" method="cancel"  
  value="%{getText('button.cancel')}"  
  name="Cancel"  
  onclick="form.onsubmit=null"/>
```

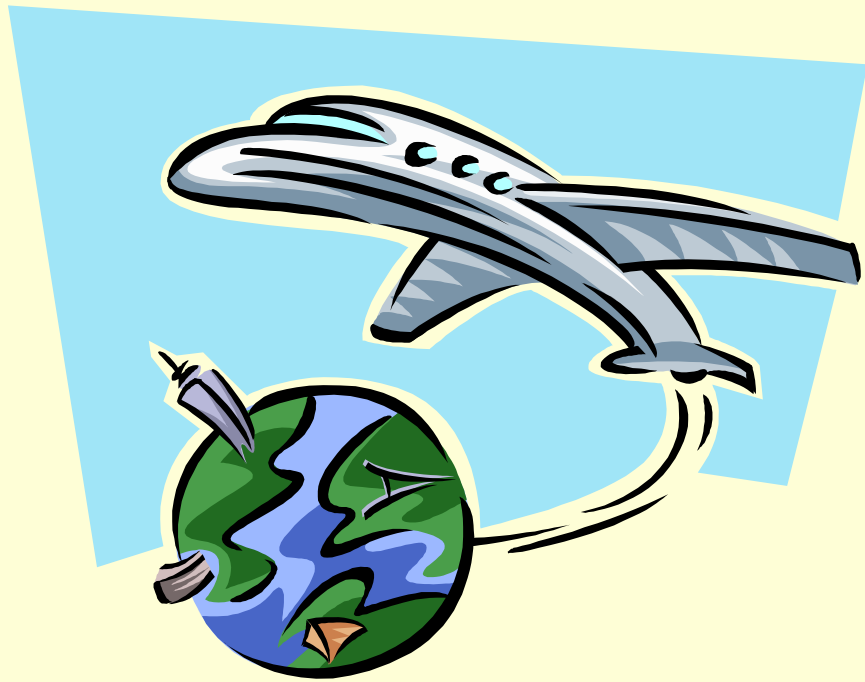




Bypassing Validation


```
<s:submit action="Register" method="cancel"  
  value="%{getText('button.cancel')}"  
  name="Cancel"  
  onclick="form.onsubmit=null"/>
```

- By default, four “magic” methods bypass validation
 - input, cancel, back, browse
- 






Review

- Initially, attributes are applied as `*****`
 - Anything in between the `%{ ... }` braces is evaluated against the `***** *****`.
 - The framework supports both `*****` errors and `*****` errors.
- 



Review

- Initially, attributes are applied as Strings.
 - Anything in between the `%{ ... }` braces is evaluated against the value stack.
 - The framework supports both Action errors and Field errors.
- 




Review

- One way to bypass validation is to submit to another `*****`.
 - Another way is to submit to one of the “magic” methods:
 - `i*****`, `c*****`, `b***`, `b*****`
 - By default, input returns the `“*****”` result code.
-




Review

- One way to bypass validation is to submit to another Action.
 - Another way is to submit to one of the “magic” methods:
 - input, cancel, back, browse
 - By default, input returns the “input” result code. (The other “magic” methods are not implemented.)
- 



Capturing Input

- HTML Forms
 - Dynamic Forms
 - Error Handling
 - Finding the Database
- 



Finding the Database

Listener

```
<!-- Application Listener for Mailreader database -->  
  <listener>  
    <listener-class>  
      DatabaseListener  
    </listener-class>  
  </listener>
```






Finding the Database

```
// ---- Database property ----

public UserDatabase getDatabase() {
    Object db = getApplication().get(DATABASE_KEY);
    if (db == null) {
        this.addActionError(getText(ERROR_DATABASE_MISSING));
    }
    return (UserDatabase) db;
}

public void setDatabase(UserDatabase database) {
    getApplication().put(DATABASE_KEY, database);
}
```





Finding the Database


Spring

- Autowire a data access object to an action property






Capturing Input

- How do HTML forms capture input?
 - What do dynamic forms add?
 - How do we handle input errors?
 - How does input get to the database?
- 




Capturing Input

- How do HTML forms capture input?
 - Simple name=value pairs
 - What do dynamic forms add?
 - How do we handle input errors?
 - How does input get to the database?
- 




Capturing Input

- How do HTML forms capture input?
 - Simple name=value pairs
 - What do dynamic forms add?
 - Prepopulation and validation
 - How do we handle input errors?
 - How does input get to the database?
- 



Capturing Input

- How do HTML forms capture input?
 - Simple name=value pairs
 - What do dynamic forms add?
 - Prepopulation and validation
 - How do we handle input errors?
 - `ActionError` or form tags
 - How does input get to the database?
- 



Capturing Input

- How do HTML forms capture input?
 - Simple name=value pairs
 - What do dynamic forms add?
 - Prepopulation and validation
 - How do we handle input errors?
 - `ActionError` or form tags
 - How does input get to the database?
 - Via the Action handler
-



Struts University Series