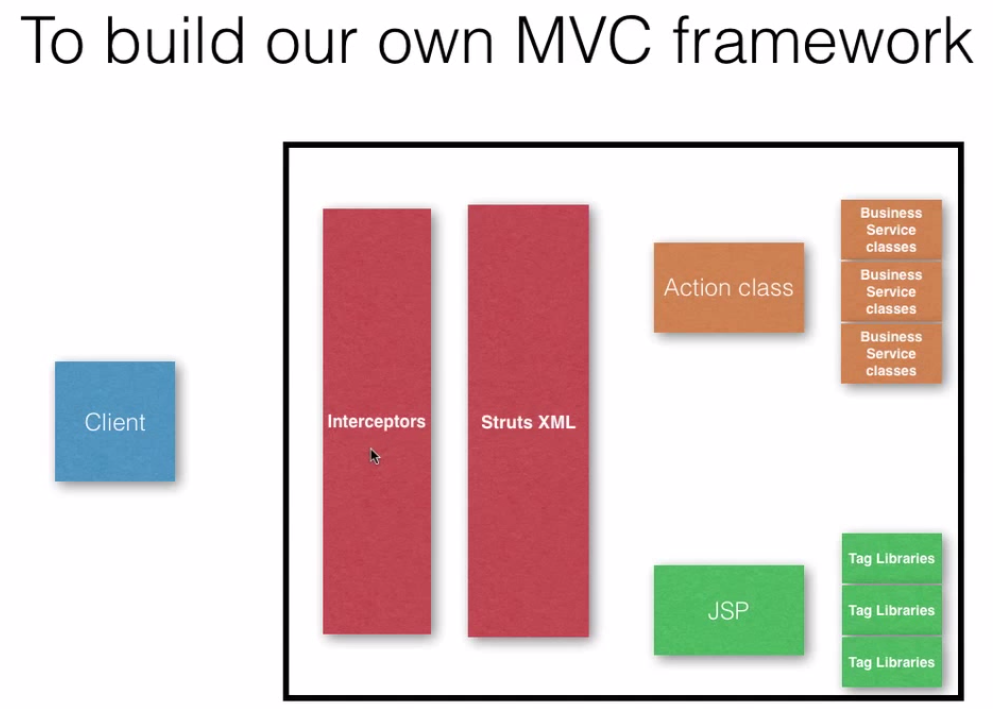
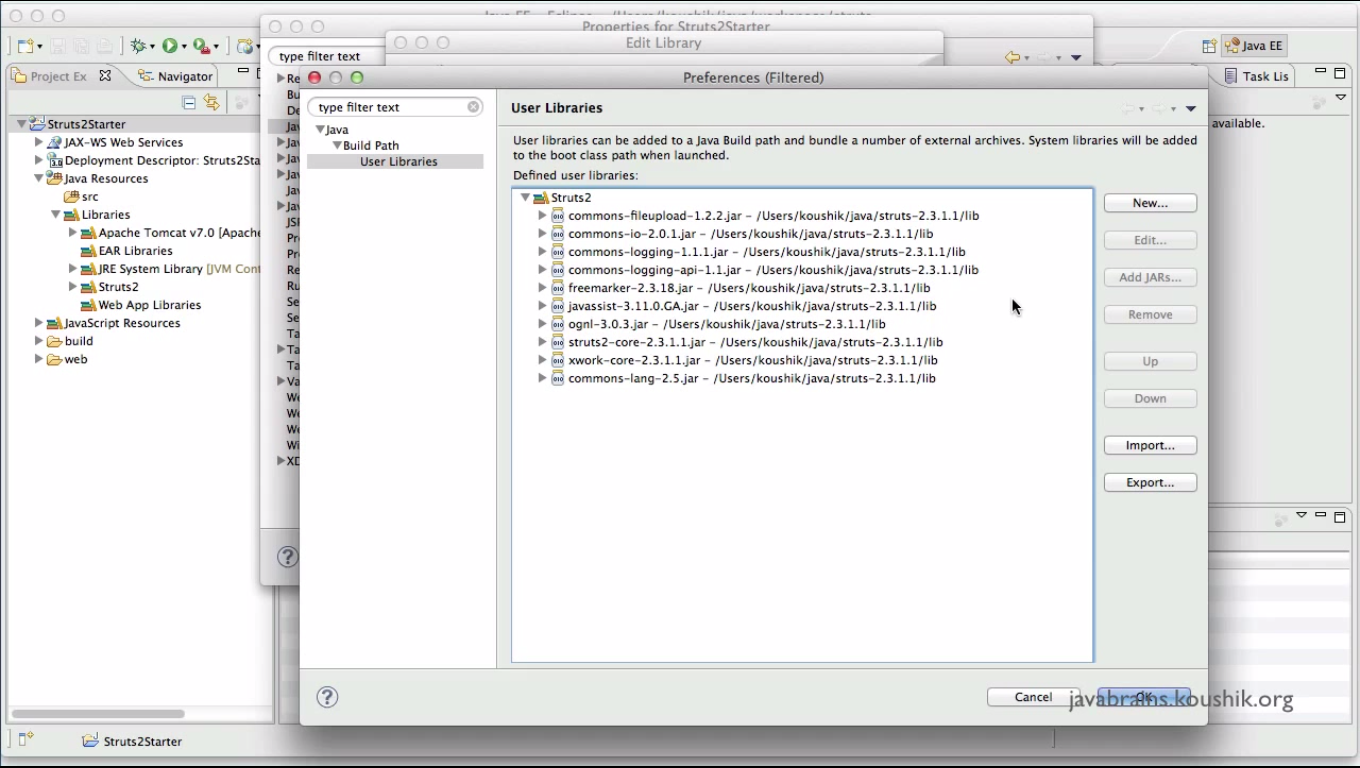
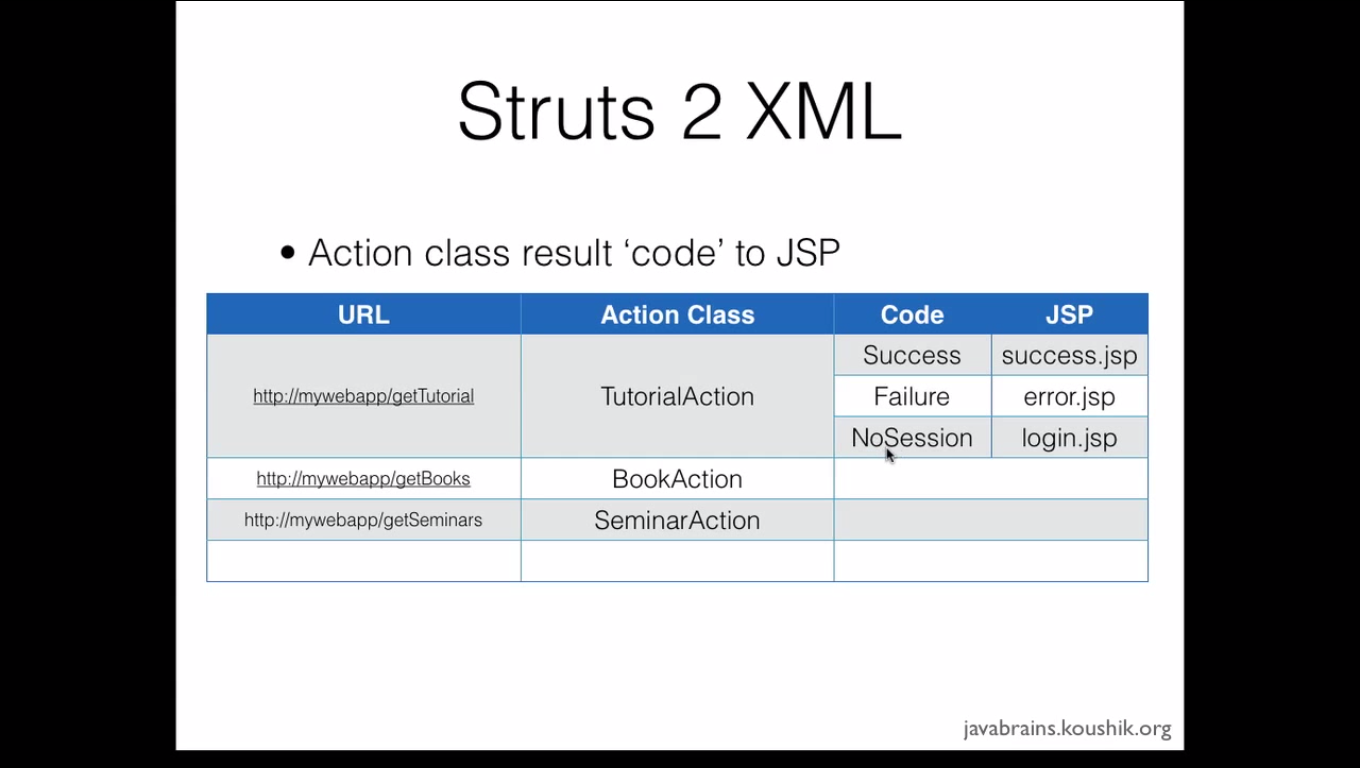
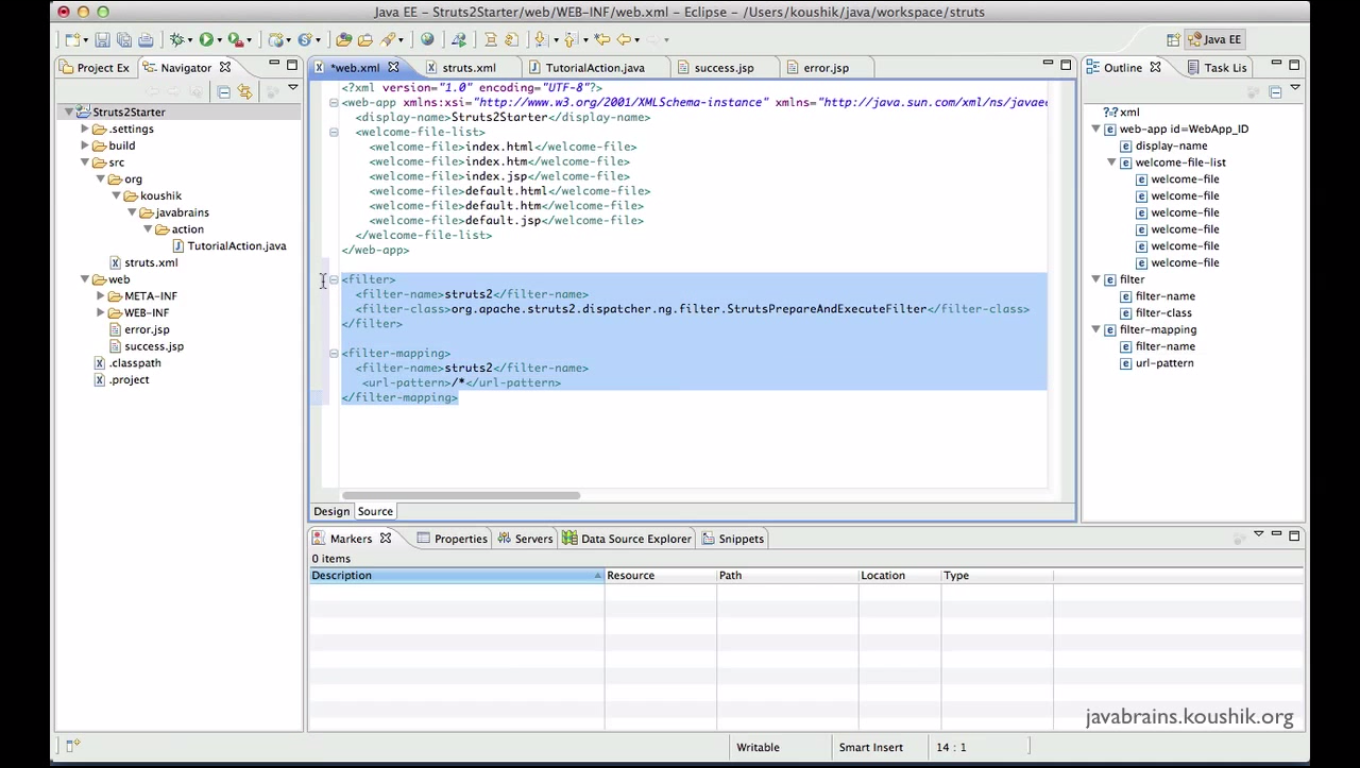
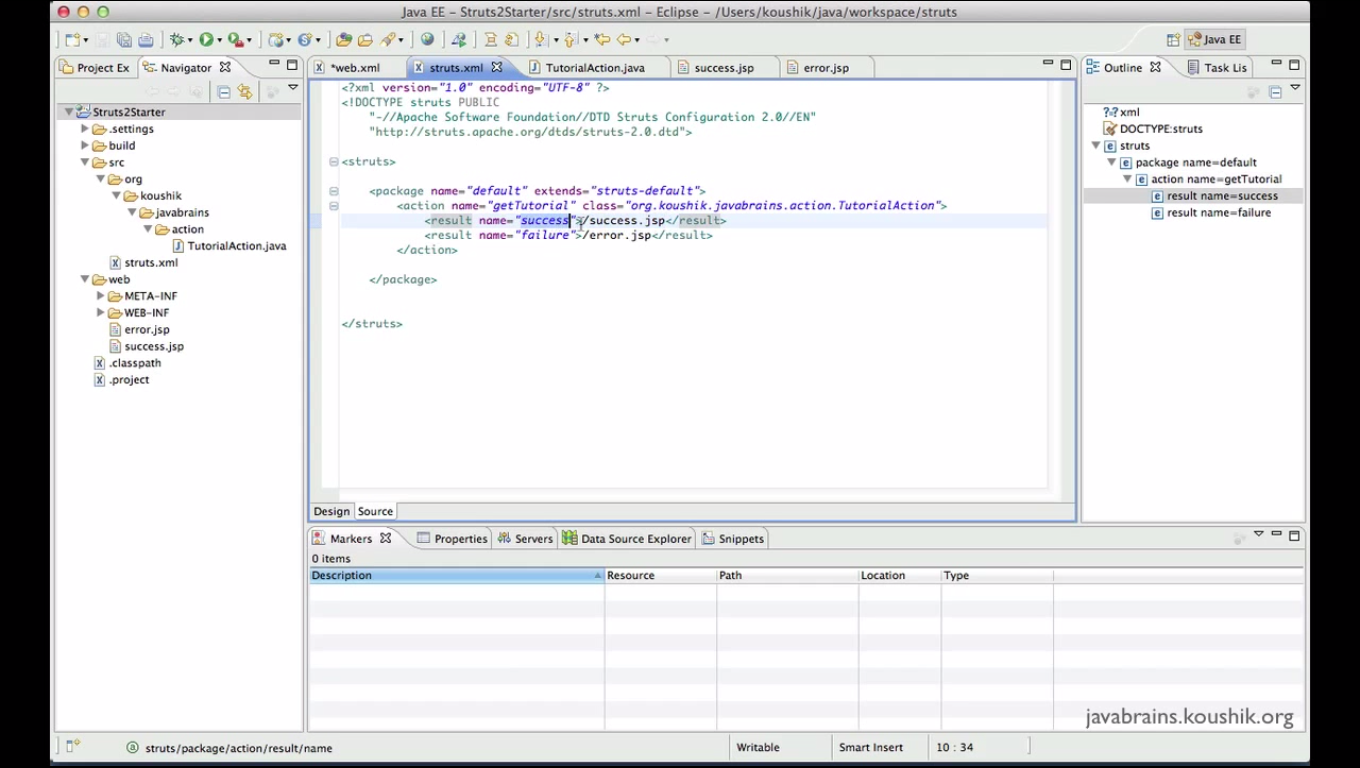
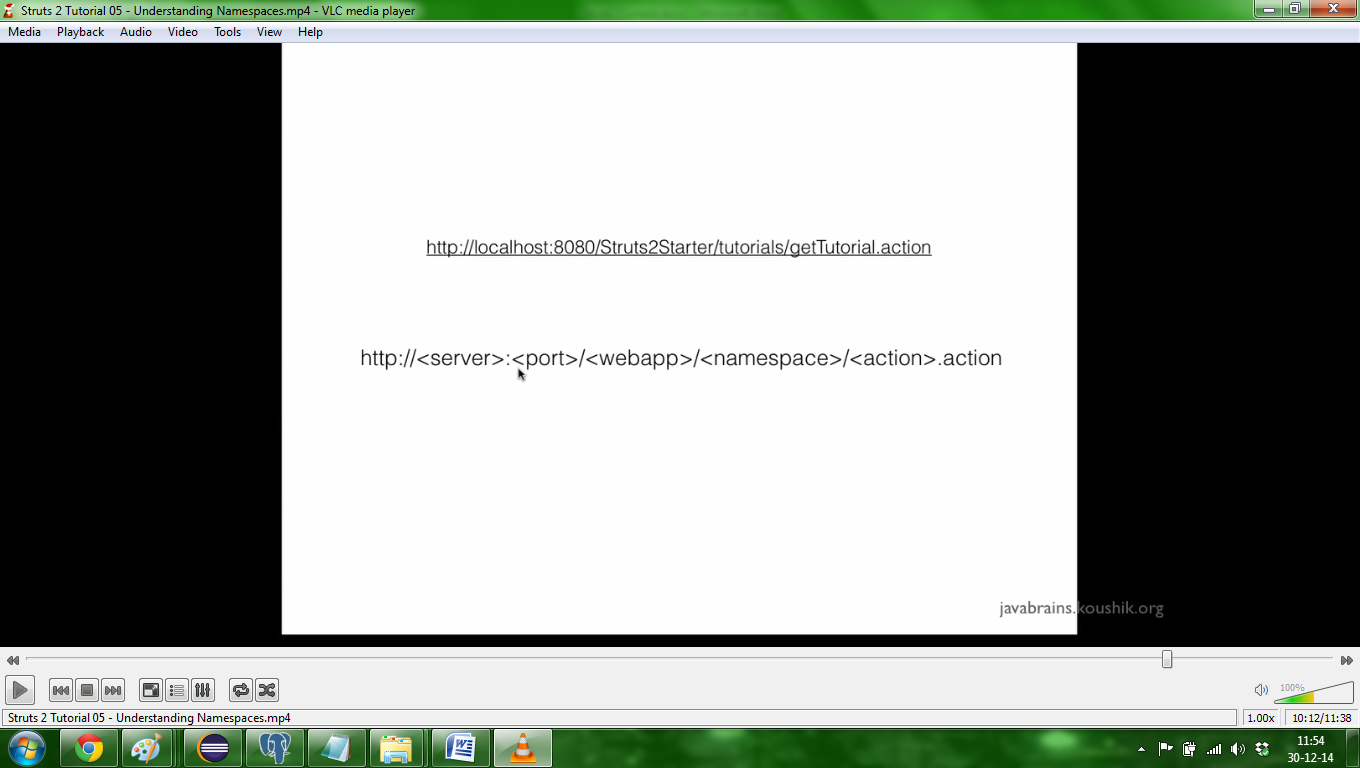
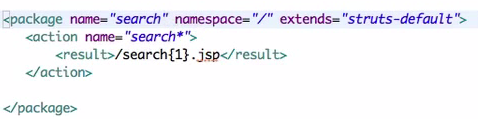
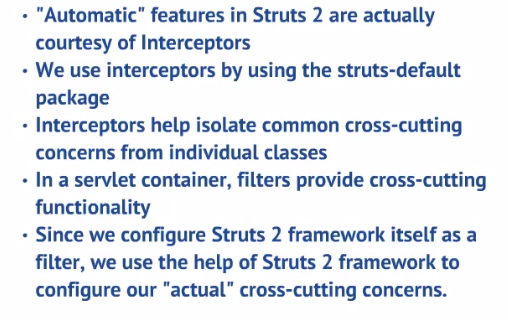
1. Introduction to MVC Pattern
2. Introduction to Structs 2
   1. Framework Vs Pattern
   2. How to build ur own MVC Pattern
   3. 
3. Environment setup(Download structs jar files)



1. Writing Application
   1. Structs XML, Action Class and JSP.
   2. Structs XML Configuration
      1. URL and its action class is mapped in the XML.
      2. Whenever a URL is entered the execute() method of the mapped class will gets executed.
      3. 
      4. 
      5. The filter and Filter-Mapping tag in Web.xml will redirect all the URL to look the structs.xml in the Application. So it need to be compulsorily placed in the web.xml.
      6. 
      7. Once the control is passed from web.xml to structs.xml, it looks for URL and go to the execute() method of the class mapped for that URL.
      8. From the return string from the execute() method, the controls move to the concordant JSP File mapped in the above XML.
2. Understanding Namespace
   1. Namespace Attribute can be added along with Package tag to specify the URL Namespace
   2. 
3. Tag and a business library
   1. <%@ taglib prefix ="s" uri="/structs-tag" %>
   2. The above tag the library for all structs classes. it needs to be added in all jsp.
4. The value stack
   1. the value stack is the place where the Structs 2 will store the objects and the member variable of different request
5. Accessing input parameter
   1. Interceptors will automatically set the parameter to the value of matching variable in class and set the value to it
6. Post request to action
   1. <s:form/>
   2. <s:textField/>
   3. <s:submit/>
7. Login action and best Practices
   1. Sample login page and configuration**(TO BE DONE)**
8. Login action and best Practices (Part 2)
   1. Implementing “Action” which contains key words for SUCCESS, NONE, ERROR, INPUT and LOGIN.
   2. We can configure the separate xml for the some action and add it in “structs.xml” using **“<include file=”Filename.xml”> </include>** ”
   3. Use dummy action in structs.xml to redirect the action becz if any file change will not affect the URL and configuration in client’s system.
9. Wildcards
   1. 
10. Action Support class.
    1. Structs 2 Actions
       1. Simple POGO Class
       2. Action Interface
       3. Action Support Interface
    2. Methods in action support method
       1. Validate()
       2. addFieldError(“FiledName”, “Message”)
11. Configuring methods in Action mappings
    1. Use “method” attribute in Action tag.
    2. We can also use wild cards
12. Model Objects
    1. Package of information that need to be send between different layers of Application
    2. We can directly map the member object to the Structs controls.
    3. Implementing Model driven and return the model object in the getModel() Overridden method.
    4. Can also implements generics.
13. Interceptors
    1. Automatic Features of structs 2
       1. Param transfer
       2. Validation
    2. These automation are done by interceptors
    3. Structs-default
       1. It is a abstract class, so it can be only extended.
       2. It has result-types, interceptors and interceptors-stack elements in it.
    4. Three Concepts



1. Types of Interceptors
   1. Nature of interceptor
      1. Configurable
      2. Java classes
      3. One or more interceptor
   2. Steps in creating an interceptor
      1. Declare an interceptor in structs.xml
      2. Declare the action
      3. Interceptor Stack
      4. We can specify a single interceptor or interceptor stack in action
      5. Refer structs-default.xml
      6. <Default-interceptor-ref name=”default-stack”> is used to specify the default stack.
2. Anatomy of interceptor
   1. Action Invocation
   2. Refer net for detailed interceptor

Notes:

Struts Framework consists of following classes:

* **Action Servlets:** used to control the response for each incoming request.
* **Action Class:** used to handle the request.
* **Action Form:** it is java bean, used to referred to forms and associated with action mapping
* **Action Mapping:** used for mapping between object and action.
* **Action Forward:** used to forward the result from controller to destination.

http://www.careerride.com/Interview-Questions-Java-Struts.aspx

Doubts

* Validation Framework consist of two XML configuration Files:  
  o Validator-Rules.xml file  
  o Validation.xml file

Exceptions are handled in struts by using any one of the following two ways:

* **Programmatically handling:** In this exception are handled by using try and catch block in program. Using this programmer can define how to handle the situation when exception arises.
* **Declarative handling:** In this exception handling is done by using the XML file. Programmer defines the exception handling logic in the XML file. There are two ways of defining the exception handling logic in the XML file:  
  -Global Action Specific Exception Handler Definition.  
  -Local Action Specific Exception Handler Definition.

The DispatchAction enable the programmer to combine together related function or class.

* Using Dispatch Action programmer can combine the user related action into a single UserAction. like add user, delete user and update user
* DispatchAction execute the action based on the parameter value it receives from the user.