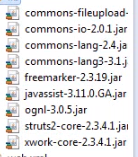
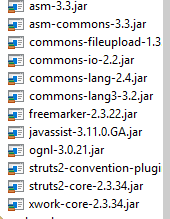
**STRUST 2 FRAMEWORK**

1. beginning

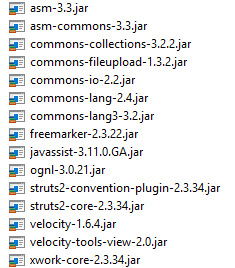
# Libraries struts 2:



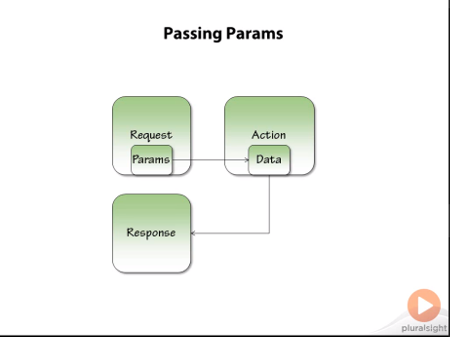
# Libraries use convention:

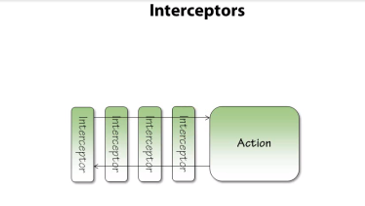


# Libraries use velocity



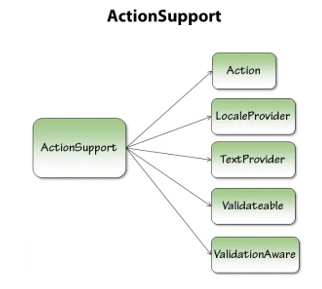
* Interceptors do play a crirital role in getting your data into your Action, look at the request look at the action , it will use something call OGNL to map your parameters to the data on your Action





* Class ActionSupport give us ability to localize messgages(get our text from somewhere else rather than hard coding in= property file) inside action.

**Ex:** LocaleProvider; Text Provider



* Action name have to same name of properties name(EnterProtein.java; EnterProtein.properties)
* **public** String getGoalText(){

**return** getText("goal.text");

}

* EnterProtein.properties

goal.text= Goal: 300 grams

* Jsp

<div><s:property value=*"goalText"*/></div>

* Using convention action name have to the same name of jsp

(EnterProtein.java; enter-protein,.jsp; Hello.java; hello.jsp)

* Result

Uses to redirect



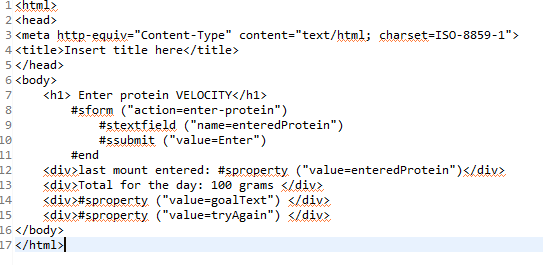


* Result template option: JSPs, FreeMarker, Velocity
* Result string map name of jsp [return ERROR(extend actionSupport);

enter-protein-error.jsp]

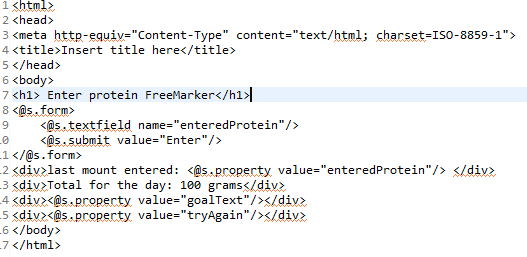
# Velocity

* to render result instead of use a jsp page
* create file: enter-protein.vm

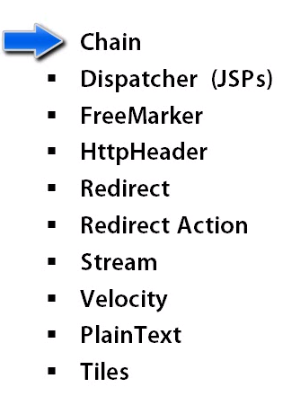


# FreeMarker

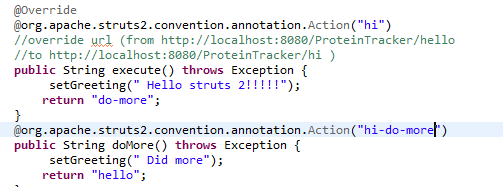
* To render result instead of use a jsp page
* Create file: enter-protein.ftl



# Default result types



* Chain: allow to chain actions together instead of displaying something immediately have the result go to another action



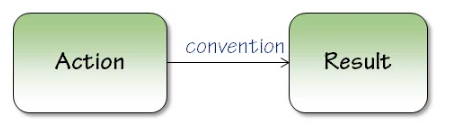
Result:

Did more (action1- action2-Didmore)

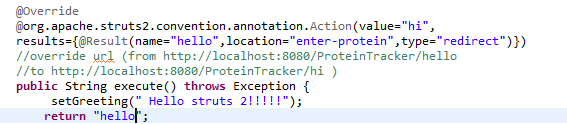
* HttpHeader: allow to send back an Http response, could use this one in order to send back an error reponse perhaps for a page that want to show as not being found (send back 404 or something )
* Redirect action: send the browser to a new action
* Stream: send direct data (file dowload)

# Annotation

Allow for overriding the default conventions



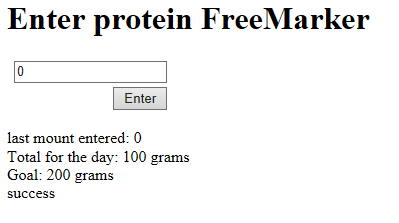
# Redirects



<http://localhost:8080/ProteinTracker/hi> automately stranffer

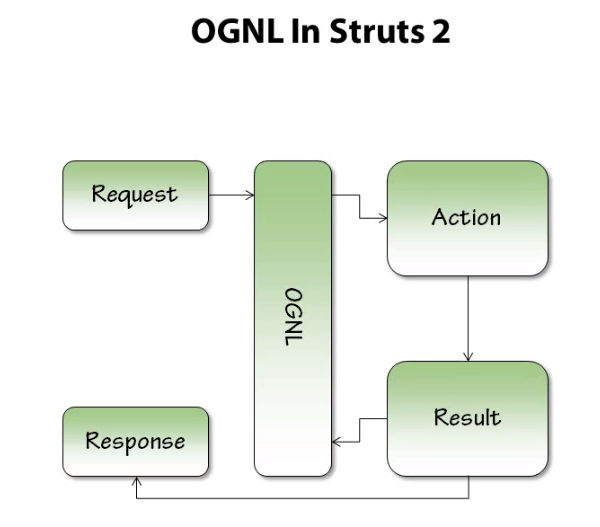
<http://localhost:8080/ProteinTracker/enter-protein>

appear:



GNL

OGNL stands for Object Graph Navigation Library

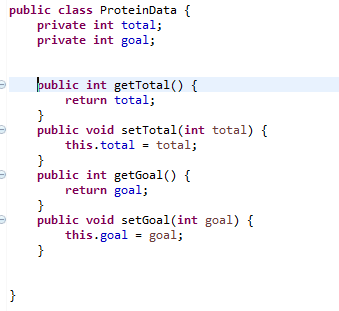


OGNL can actually map to more than just the action, more

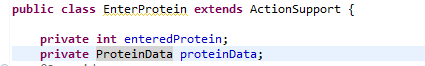
You can get data out of a session and you can get data out of several other objects that are part of what’s call the value stack that exist inside of the struts 2

When have the request, it has some OGNL string in it

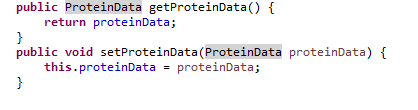
* Create OGNL



In action file



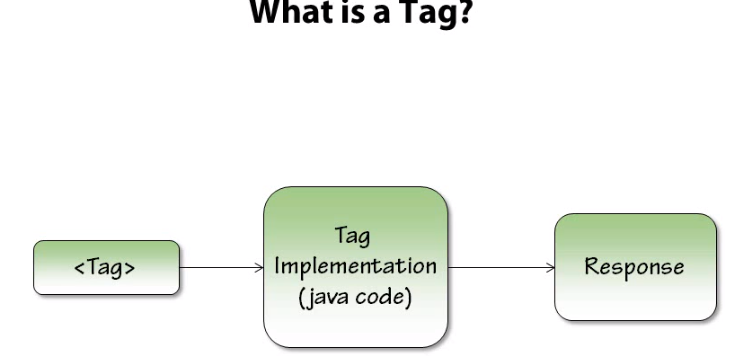
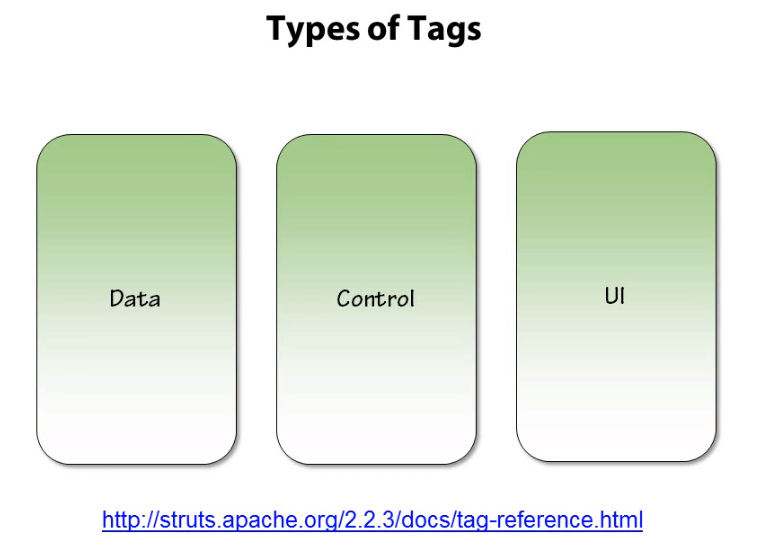
Just getter, setter proteinData



In jsp file



1. TAGS

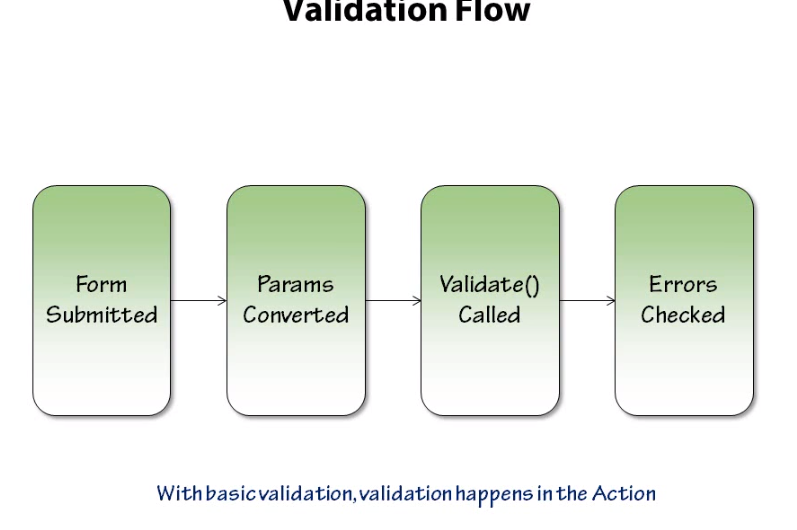




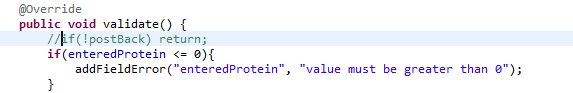




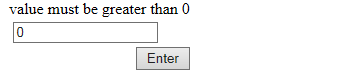
1. VALIDATION



To use validation just only override method validate in ActionSupport



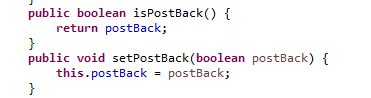
Note: result by default



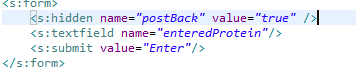
**Fix**

* Create postBack variable in action then getter,setter

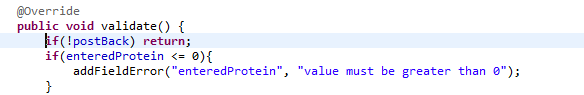




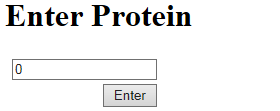
* In jsp add postBack value=true



* Then



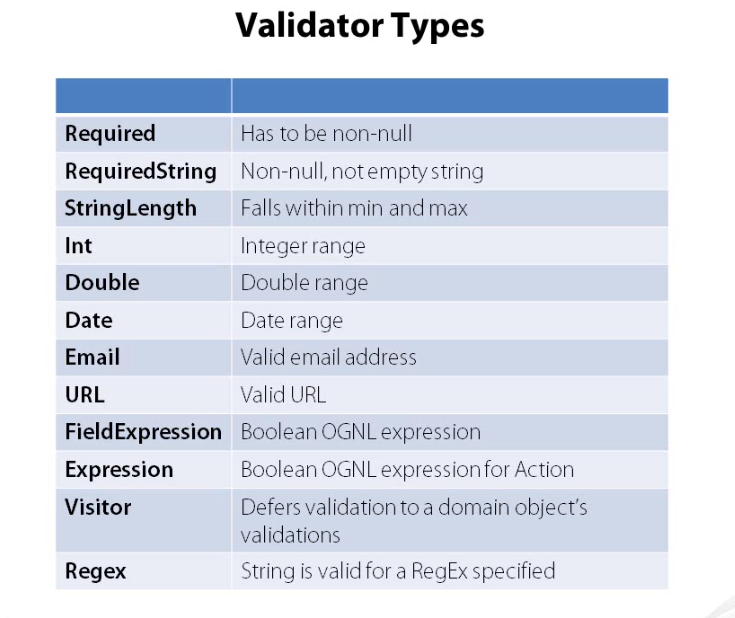
* Result:



**\*note**

* When load jsp page enter-protein data value of postBack is false
* When submit form value of postBack is true;

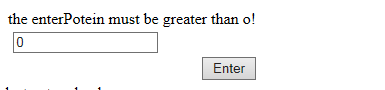
return meant Once return is executed, the rest of the code won't be executed.



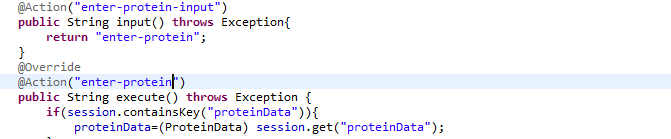
# Use validator



* But if again appear



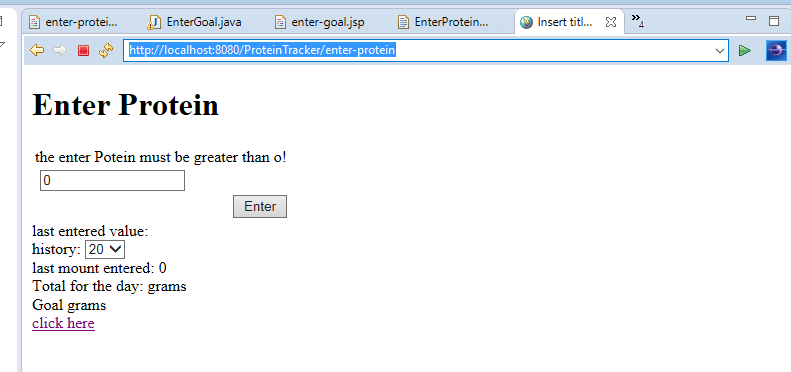
Fix:



<http://localhost:8080/ProteinTracker/enter-protein> -> it wipe out whole class EnterProtein,

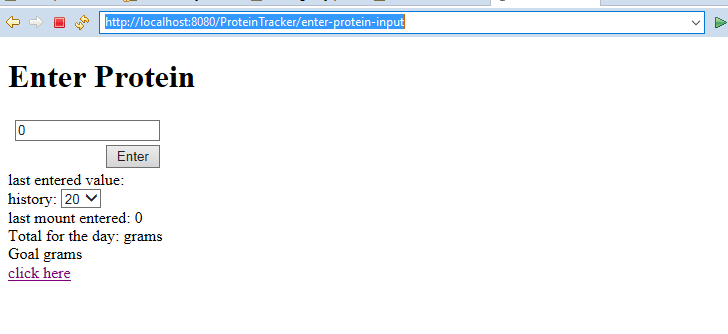
have method



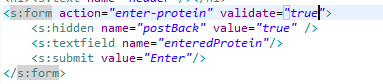


<http://localhost:8080/ProteinTracker/enter-protein-input> -> it just execute method execute(). Don’t wipe out method





In jsp add action, validate



Ajax