**14.1. Form Tag Text Fields – Overview**

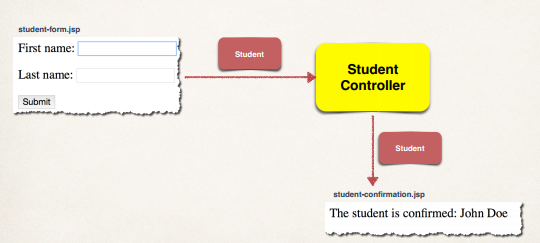
**14.1. Form Tag Text Fields (Spring MVC) – Overview**:

The Spring MVC form text field tag generates an HTML input tag using the bound value. By default, the type of the input tag is text.

**Syntax**:

**<form:input** path="name" **/>**

Here, path attribute binds the form field to the bean property.



**Showing Form**:

In our spring controller, before we show the form, we must add a model attribute. This is basically a bean that is used to hold the form data. That will give support for data binding.

* Before you show the form, you must add a model attribute
* This is a bean that will hold form data for the data binding

**Add Model Attribute**:

@RequestMapping("/showForm")public String showForm(Model theModel) { theModel.addAttribute("student", new Student()); return "student-form";}

1. Model is used to pass data between controllers and view
2. Here in "theModel.addAttribute("student", new Student());", "student" is attribute name and "new Student()" is attribute value. Effectively we are creating a empty Student object, that I will pass to the form for the form to make use of with data binding.

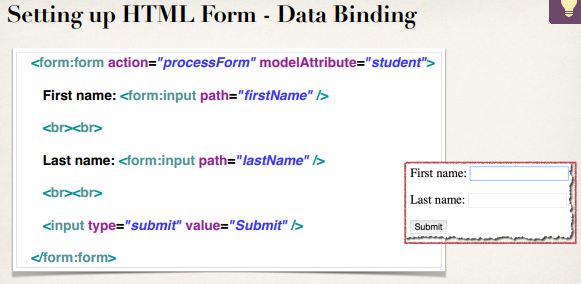
Here the important thing is the attribute name "student" is the same name that our form will use to reference this model attribute.

**Setting up HTML Form - Data Binding**:

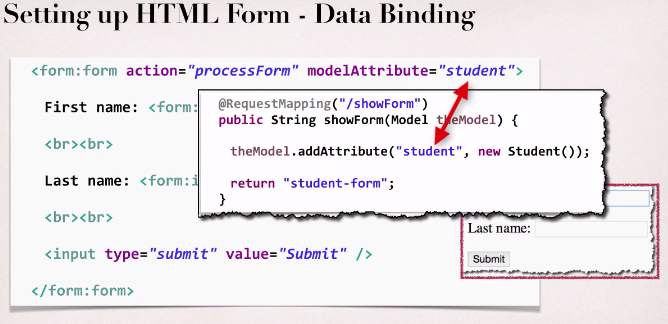
<form:form action=*"processForm"* modelAttribute=*"student"*>  
 First name: <form:input path=*"firstName"* />  
 <br><br>

Last name: <form:input path=*"lastName"* />  
 <br><br>

<input type=*"submit"* value=*"Submit"* />  
</form:form>



The important thing is that the modelAttribute name in form must be same with controller attribute name in controller class.



Here the **path=*"firstName"*** and **path=*"lastName"*** actually binds this form field to a property on the bean.

**When Form is Loaded … fields are populated**:

14.1. Form Tag Text Fields - Overview