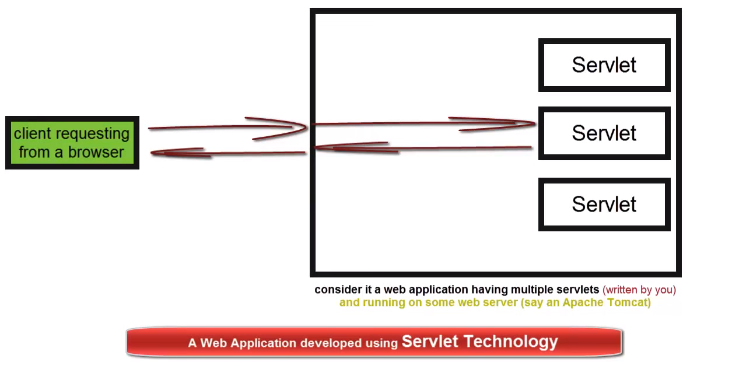
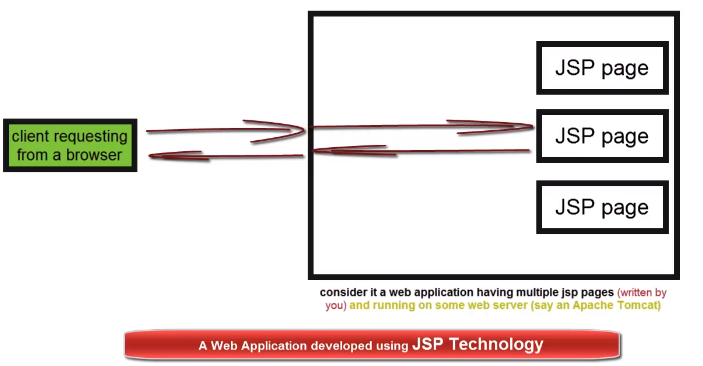
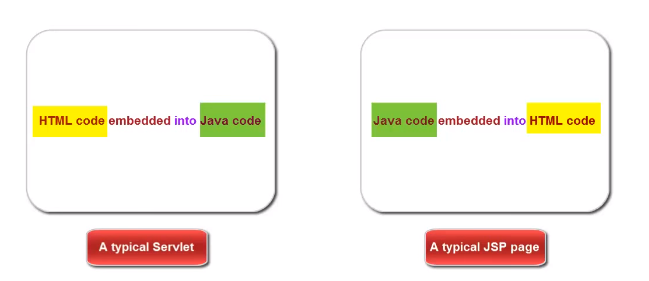
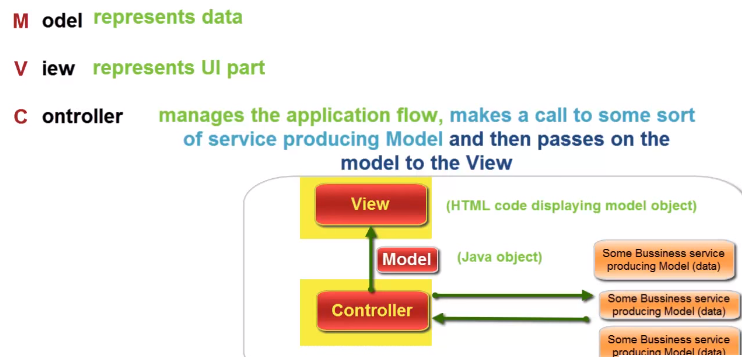
MVC

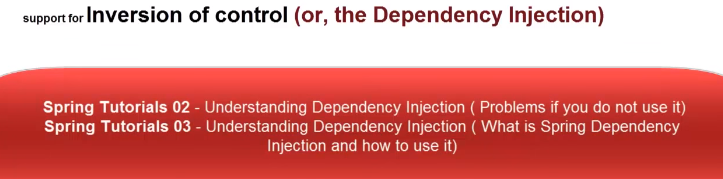


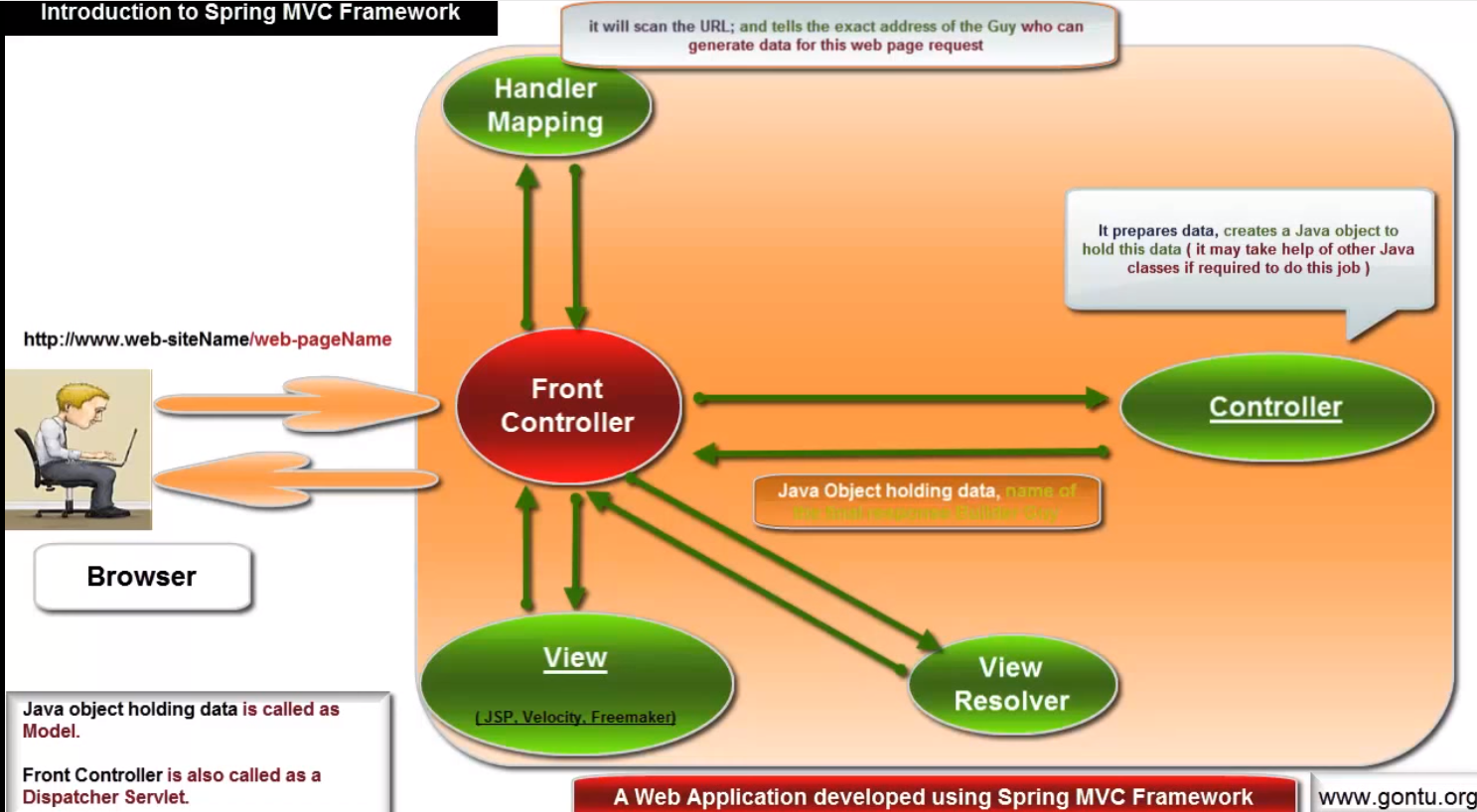




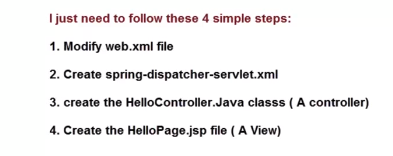








1. 建立一个Web dynamic web project
2. 在lib里导入要用到的包，spring和commonlogging
3. 右击projectname，property里面runtime选apache

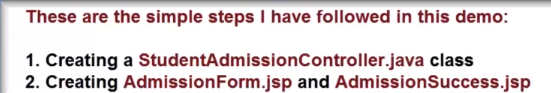
4

另一种方式：writing an annotation based controller class -@Controller, @RequestMapping

在spring-dispatcher-servlet.xml中加上：

<context:component-scan base-package="com.yang.hellocontroller" />

# Handling an HTML form using @RequestParam annotation



@Controller

**public** **class** StudentAdmissionController {

@RequestMapping(value="/admissionForm.html",method=RequestMethod.***GET***)

**public** ModelAndView getAdmissionForm(){

ModelAndView model = **new** ModelAndView("AdmissionForm");

**return** model;

}

@RequestMapping(value="/submitAdmissionForm.html", method=RequestMethod.***POST***)

//public ModelAndView submitAdmissionForm(@RequestParam(value="studentName",defaultValue="Mr.ABCDXYZ") String name,@RequestParam("studentName") String hobby){

**public** ModelAndView submitAdmissionForm(@RequestParam Map<String,String> reqPar){

String name = reqPar.get("studentName");

String hobby = reqPar.get("studentHobby");

**if**(name.isEmpty()){

name="MR.ABCD";

}

ModelAndView model = **new** ModelAndView("AdmissionSuccess");

model.addObject("msg","Details submitted by you:: Name: "+name+", Hobby: "+hobby);

**return** model;

}

}

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=ISO-8859-1"*>

<title>Insert title here</title>

</head>

<body>

<h1>STUDENT ADMISSION FORM FOR ENGINEERING COURSES</h1>

<form action = *"/myFirstMVCWeb/submitAdmissionForm.html"* method=*"post"*>

<p>

Student's Name: <input type=*"text"* name=*"studentName"*>

</p>

<p>

Student's Hobby: <input type=*"text"* name=*"studentHobby"*>

</p>

<input type=*"submit"* value=*"Submit this form by clicking here"*>

</form>

</body>

</html>

# Understanding @ModelAttribute Annotation 01 (using on a method argument)

Create a Student Class-----a model

Using Student Model in the controller directly.. 名字一定要一致。

**public** ModelAndView submitAdmissionForm(@ModelAttribute("student1") Student student1){

ModelAndView model = **new** ModelAndView("AdmissionSuccess");

model.addObject("headerMessage","College of Engineering");

//model.addObject("student1",student1);

**return** model;

}

# Understanding @ModelAttribute annotation 02 ( using at a method level )

@ModelAttribute

**public** **void** addingCommonObjects(Model model){

model.addAttribute("headerMessage","College of Engineering");

}

在Controller中加ModelAttribute Method，对其他所有此controller中的method共用，要注意，这个method会被第一个被调用

# Data Binding with a User-Defined Type, BindingResult (concept)

Controller:

**public** ModelAndView submitAdmissionForm(@ModelAttribute("student1") Student student1,BindingResult result){

**if**(result.hasErrors()){

ModelAndView model= **new** ModelAndView("AdmissionForm");

**return** model;

}

ModelAndView model = **new** ModelAndView("AdmissionSuccess");

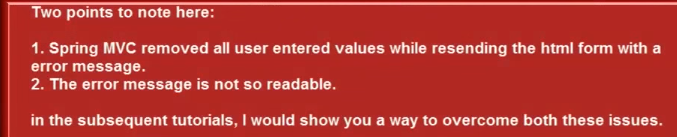
**return** model;

}

JSP file:

<%@ taglib prefix=*"form"* url=*"http://www.springframework.org/tags/form"* %>

<form:errors path=*"student1.\*"*/>



# @InitBinder annotation, WebDataBinder, CustomDateEditor (concept)

@InitBinder

**public** **void** initBinder(WebDataBinder binder){

binder.setDisallowedFields(**new** String[]{"studentMobile"});

SimpleDateFormat dateformat = **new** SimpleDateFormat("yyyy\*\*\*mm\*\*\*dd");

binder.registerCustomEditor(Date.**class**,"studentDOB", **new** CustomDateEditor(dateformat,**false**));

}

# Writing your own custom property editor class

The custom property editor class:

**import** java.beans.PropertyEditorSupport;

**public** **class** StudentNameEditor **extends** PropertyEditorSupport {

//when you will submit the admission form

//Spring MVC will run setAsText function of this class

//before performing data binding task for studentName property of student obejct

@Override

**public** **void** setAsText(String studentName) **throws** IllegalArgumentException{

**if**(studentName.contains("Mr.") || studentName.contains("Ms.")){

setValue(studentName);

}**else**{

studentName="Ms."+studentName;

setValue(studentName);

}

}

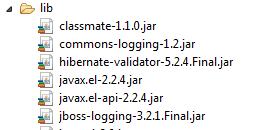
}

Controller:

binder.registerCustomEditor(String.**class**,"studentName", **new** StudentNameEditor());

# Form Validations 01 ( using JSR 303/349 provided annotations )

@size and @valid annotations

Add jars in lib: 

“Shift”+”Ctrl”+”O”

@Size(min=2,max=20)

**private** String studentHobby;

**public** ModelAndView submitAdmissionForm(@Valid @ModelAttribute("student1") Student student1,BindingResult result)

<mvc:annotation-driven/> in dispatcher file

# Form Validation 02 (customizing error messages using Spring MessageSource)

1. 在Student class里 @Size中加message

@Size(min=2,max=20,message="Please enter a value for studentHobby field between {min} and {max} characters.")

1. 在WEB-INFOz下添加studentmessages.properties file

加上Size.student1.studentHobby,value为要显示的message

在dispacther中添加

<bean id=*"messageSource"*

class=*"org.springframework.context.support.ReloadableResourceBundleMessageSource"*>

<property name=*"basename"* value=*"/WEB-INF/studentmessages"*/>

</bean>

