

## Controller Classes of MVC application

In the project JobFinder we controllers HomeController.java and VisitorController.java  
In HomeController.java we defined only one method which maps to the Home page of our application.

### HomeController.java

```
@Controller
public class HomeController {
    @RequestMapping("/")
    public String showPage() {

        return "main-menu";
    }
}
```

We have also declared VisitorConstructor.java which has all the initialization and processing of the visitor form and details.

### VisitorController.java

```
@Controller
@RequestMapping("/visitor")
public class VisitorController {

    //add an initbinder ...to convert trim input strings
    //remove leading and trailing whitespace
    //resolve issue for validation

    @InitBinder
    public void initBinder(WebDataBinder dataBinder) {

        StringTrimmerEditor stringTrimmerEditor = new
StringTrimmerEditor(true);
        dataBinder.registerCustomEditor(String.class, stringTrimmerEditor);
    }
}
```

The above method is used to trim the whitespaces from the input done by the visitor.

```
@RequestMapping("/showForm")
public String showForm(Model theModel) {

    /*
     * //create customer object Customer theCustomer = new Customer();
     */

    //add customer object to the model
    theModel.addAttribute("visitor", new Visitor());

    return "visitor-form";
}
```

The above method is for initializing visitor details. It maps to showForm and returns visitorForm.

```
@RequestMapping("/processForm")
```

## Controller Classes of MVC application

```
public String processForm(@Valid @ModelAttribute("visitor") Visitor
theVisitor, BindingResult theBindingResult) {

    if(theBindingResult.hasErrors()) {
        return "visitor-form";
    }
    else {
        /*
        * //log the input data System.out.println("theCustomer"
        * +theCustomer.getFirstName() + theCustomer.getLastName());;
        */
        return "visitor-details";
    }
}
```

The above form is to process the details filled by the Visitor. It maps to processForm and returns visitor-details .jsp file.

Controller handles the request sent by the client and sends back the response as a view page which is .jsp files in this case.

Previous step: [Entity Class](#)

Next Step: [jsp files](#)