

```

+++++
Sending Data from Controller to UI
+++++
spring.mvc.view.prefix = WEB-INF/view
spring.mvc.view.suffix = .jsp
server.port = 9999
server.servlet.context-path=/SecondApp

```

```

Controller
+++++
@GetMapping("/show")
public String showMsg(Model model) {
    Employee employee = new Employee();
    employee.setEmpId(10);
    employee.setEmpName("sachin");
    employee.setEmpSal(3454.0);

    model.addAttribute("emp", employee);

    return "empData";
}

```

```

+++++
empData.jsp
+++++
<h1>EMPLOYEE OBJECT IS :: ${emp }</h1>
<h1>READ VALUE FROM OBJECT :: ${emp.empId }, ${emp.empName },${emp.empSal }</h1>

```

```

+++++
Sending Data from UI To Controller
+++++
using
1. @ModelAttribute

```

Note: SpringContainer convert the data to Object format on clicking submit button after entering data in below steps

- a. Container creates the object for Model class[Here objectName is className,firstLetter lower case]
- b. Container reads the data from HTML form inputs[request.getParamter("")]
- c. Container parses the data[Using Wrapper classes as per the datatype]
- d. Finally set the value to ModelAttribute.

```

+++++
Controller
+++++
@PostMapping("/reg")
public String register(@ModelAttribute Employee employee, Model model) {
    System.out.println("EmployeeController.register()");
    System.out.println(employee);
    model.addAttribute("employee", employee);
    return "emp";
}

```

```

+++++
View[emp.jsp]
+++++
<h1>EMPLOYEE DATA</h1>
<br>

```

```

<table class="table table-striped">
  <thead>
    <tr>
      <th scope="col">EMPID</th>
      <th scope="col">EMPNAME</th>
      <th scope="col">EMPSAL</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>${employee.empId }</td>
      <td>${employee.empName }</td>
      <td>${employee.empSal }</td>
    </tr>
  </tbody>
</table>

```

+++++

Employee.java

+++++

```
package in.pwskills.nitin.entity;
```

```

public class Employee {

    private Integer empId;
    private String empName;
    private Double empSal;

    public Employee() {
        System.out.println("Employee Object is created...");
    }

    public Integer getEmpId() {
        System.out.println("Employee.getEmpId()");
        return empId;
    }

    public void setEmpId(Integer empId) {
        this.empId = empId;
        System.out.println("Employee.setEmpId()");
    }

    public String getEmpName() {
        System.out.println("Employee.getEmpName()");
        return empName;
    }

    public void setEmpName(String empName) {
        this.empName = empName;
        System.out.println("Employee.setEmpName()");
    }

    public Double getEmpSal() {
        System.out.println("Employee.getEmpSal()");
        return empSal;
    }

    public void setEmpSal(Double empSal) {
        this.empSal = empSal;
    }
}

```

```

        System.out.println("Employee.setEmpSal()");
    }

    @Override
    public String toString() {
        return "Employee [empId=" + empId + ", empName=" + empName + ",
empSal=" + empSal + "]\n";
    }
}

```

++++++
TwoWay-Binding
++++++

=> If we create a form using html, then the object created by the container can be used only in getting the data from Controller to UI, but it is not possible to send the data from UI to Controller.
=> To do 2-way binding of data we need to use "Spring supplied form tags".
=> <%@ taglib prefix="form" uri="http://www.springframework.org/tags/form" %>
=> form should be created with a new attribute called "modelAttribute" as shown below

```
<form:form method="POST" modelAttribute="employee">
```

=> Inside the controller we need to have the following code with the model attribute attached as shown below.

```

@GetMapping("/")
public String homePage(Model model) {
    model.addAttribute("employee", new Employee());
    return "register";
}

```

Note: When the form is loaded by the container, SpringBoot internally calls "getter method on the modelAttribute object" and loads the value from the entity.

When the form is submitted by the user, container will internally calls "setter method on the modelAttribute object" and injects the value to the entity.

++++++
Creating LibraryManagement Application
++++++

1. new-book.jsp -----> save(Book)-----> books.jsp
2. /add -----> List<Book> findAll() -----> books.jsp

Controller

++++++

@Controller

```
public class LibraryController {
```

```

    @Autowired
    private IBookService service;

```

```

    @GetMapping("/")
    public String showPage() {
        return "redirect:new-book";
    }

```

```

    @GetMapping("/new-book")
    public String showBookCreationForm(Model model) {

```

```

        model.addAttribute("book", new Book());
        return "new-book";
    }

    @PostMapping("/add")
    public String addNewBook(@ModelAttribute Book book, Model model) {
        System.out.println(book);
        service.saveBook(book);
        model.addAttribute("books", service.findAllBooks());
        return "books";
    }
}

+++++
View[books.jsp]
+++++
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<%@ taglib prefix="form" uri="http://www.springframework.org/tags/form"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>

<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Book Library</title>
</head>
<body>
    <div>
        <h2>LIBRARY</h2>
        <br/>
        <a href='/new-book'>
            <button type='submit'>ADD NEW BOOK</button>
        </a>
        <br/><br/>
    </div>
    <div>
        <div>BOOK LIST</div>
        <div>
            <table border='1'>
                <tr>
                    <th>ID</th>
                    <th>Author</th>
                    <th>Name</th>
                </tr>
                <c:forEach var="book" items="${books }">
                    <tr>
                        <td>${book.id }</td>
                        <td>${book.author }</td>
                        <td>${book.name }</td>
                    </tr>
                </c:forEach>
            </table>
        </div>
    </div>
</body>
</html>

```

```

+++++++
new-book.jsp
+++++++
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<%@ taglib prefix="form" uri="http://www.springframework.org/tags/form" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Book Library</title>
</head>
<body>
    <div>
        <h2> NEW USER</h2>
    </div>
    <div>
        <form:form action ="/add" modelAttribute="book" method="post">
            <div>
                <div>
                    <form:label path="author">AUTHOR</form:label>
                    <form:input path="author"/>
                </div>
                <div>
                    <form:label path="name">NAME</form:label>
                    <form:input path="name"/>
                </div>
                <div>
                    <input type='submit' value='ADD USERS'>
                </div>
            </div>
        </form:form>
    </div>
</body>
</html>

```

