

Searching for a Specific User and Updating the User Information.

All Codes..

App.java

```
package com.example;
```

```
import org.springframework.context.annotation.ComponentScan;
```

```
import org.springframework.context.annotation.Configuration;
```

```
@Configuration
```

```
@ComponentScan(basePackages = {"com.example"})
```

```
public class App {
```

```
}
```

Employee.java

```
package com.example;
```

```
import javax.persistence.Entity;
```

```
import javax.persistence.GeneratedValue;
```

```
import javax.persistence.GenerationType;
```

```
import javax.persistence.Id;
```

```
@Entity
```

```
public class Employee {
```

```
    @Id
```

```
    @GeneratedValue(strategy = GenerationType.AUTO)
```

```
        private int eid;
```

```
        private String ename;
```

```
        private String eemail;
```

```
public int getId() {  
    return eid;  
}  
  
public void setId(int eid) {  
    this.eid = eid;  
}  
  
public String getName() {  
    return ename;  
}  
  
public void setName(String ename) {  
    this.ename = ename;  
}  
  
public String getEmail() {  
    return eemail;  
}  
  
public void setEmail(String eemail) {  
    this.eemail = eemail;  
}
```

```
}
```

```
=====
```

Empdao.java

```
package com.example;
```

```
import java.util.List;
```

```
import org.springframework.orm.hibernate3.HibernateTemplate;
```

```
public class Empdao {
```

```
    HibernateTemplate temp;
```

```
    public void setTemp(HibernateTemplate temp) {
```

```
        this.temp = temp;
```

```
}
```

```
public int insert(Employee e) {  
    return (Integer) temp.save(e);  
}
```

```
//update and delete
```

```
public List<Employee> getall(){  
    String sql="from Employee";  
    return (List<Employee>)temp.find(sql);  
}
```

```
public String update(Employee e) {  
    temp.update(e);  
  
    return "Success";    }
```

```
}
```

```
=====
```

Initializer.java

```
package com.example;
```

```
import
```

```
org.springframework.web.servlet.support.AbstractAnnotationConfigDispatcherServletInitializer;
```

```
public class Intializer extends
```

```
AbstractAnnotationConfigDispatcherServletInitializer  
{
```

```
@Override
```

```
protected Class<?>[] getRootConfigClasses() {  
    // TODO Auto-generated method stub  
    return null;  
}
```

```
@Override
```

```
protected Class<?>[] getServletConfigClasses() {  
    // TODO Auto-generated method stub  
    return new Class[] {App.class};  
}
```

```
@Override
```

```
protected String[] getServletMappings() {  
    // TODO Auto-generated method stub  
    return new String[] {"/"};  
}
```

```
}
```

MainController.java

```
package com.example;
```

```
import java.util.List;
```

```
import javax.servlet.http.HttpServletRequest;
```

```
import javax.servlet.http.HttpServletResponse;
```

```
import org.springframework.context.ApplicationContext;
```

```
import  
org.springframework.context.support.ClassPathXmlApplicationContex  
t;
```

```
import org.springframework.stereotype.Controller;
```

```
import org.springframework.web.bind.annotation.RequestMapping;
```

```
import org.springframework.web.servlet.ModelAndView;
```

```
@Controller
```



```
public class MainController {

    @RequestMapping("/insert")

    public ModelAndView insertion(HttpServletRequest req, HttpServletResponse res) {

        ModelAndView mv=new ModelAndView();

        ApplicationContext ac=new
        ClassPathXmlApplicationContext("spring.xml");

        Employee e=ac.getBean(Employee.class);

        Empdao dao=ac.getBean(Empdao.class);

        e.setEname(req.getParameter("ename"));

        e.setEemail(req.getParameter("eemail"));

        if(dao.insert(e)>0) {

            mv.setViewName("success.jsp");

        }

        else {

            mv.setViewName("fail.jsp");

        }

        return mv;

    }

}
```

```
@RequestMapping("/get")

public ModelAndView getall(HttpServletRequest req, HttpServletResponse res) {

    ModelAndView mv=new ModelAndView();

    ApplicationContext ac=new
ClassPathXmlApplicationContext("spring.xml");

    Employee e=ac.getBean(Employee.class);

    Empdao dao=ac.getBean(Empdao.class);

    List<Employee> list=dao.getall();

    if(list!=null) {

        mv.setViewName("display.jsp");

        mv.addObject("list", list);

    }

    return mv;

}
```

```
@RequestMapping("/update")
```

```
public ModelAndView update(HttpServletRequest req,
    HttpServletResponse res) {

    ModelAndView mv = new ModelAndView();

    ApplicationContext ac = new
    ClassPathXmlApplicationContext("spring.xml");

    Employee e = ac.getBean(Employee.class);
    Empdao dao = ac.getBean(Empdao.class);
    e.setEname(req.getParameter("ename"));
    e.setEemail(req.getParameter("eemail"));
    e.setEid(Integer.parseInt(req.getParameter("eid")));

    String r = dao.update(e);

    if(r.equals("Success")) {
        mv.setViewName("updateSeccuss.jsp");
    }
}
```

```
        else
        {
            mv.setViewName("updateFail.jsp");
        }
        return mv;

//    if(dao.update(e)>0) {
//
//        mv.setViewName("updateSeccuss.jsp");
//    }
//    else {
//
//        mv.setViewName("updateFail.jsp");
//    }
//    return mv;
//
//    }

}
```

index.jsp

```
<html>
<body>
<h2>Hello World!</h2>

<form action="insert">
Ename<input type="text" name="ename"><br>
Eemail<input type="email" name="eemail"><br>
<input type="submit">

</form>
</body>
</html>
```

update.jsp

```
<html>

<body>

<h2>Hello World!</h2>


<form action="update">
Eid<input type="text" name="eid"><br>
Ename<input type="text" name="ename"><br>
Eemail<input type="email" name="email"><br>
<input type="submit">


</form>

</body>

</html>
```

Name :- Vishnu Avtar