NOM POUPA

Prénom Adrien

Promo 071 2018

Date 07/01/17

12,75

POUPA Adrien
M1 - 2016

MATIÈRE JEE

1. "Java Enterprise Edition" is a programming language NO developped by Oracle and based on Java. In includes more than 32 specifications as of 2013. It is made for enterprises due to its easy scalability and its modular approach, allowing an easy replacement of each vendor (switching from GlassFish to JBoss if needed for example), thanks to the strict conventions it has. ?

2. A Java Bean is a simple POJO class. It has to meet the following requirements:
- public class
- private attributes, no public attributes
- private attribute can be seen or modified using with public getters and setters
- public default constructor: no argument required
- can implement Serializable for persistance

It is a Model in the MVC architecture (working logique / "logique métier")

Example:

```
public class User {
    private String _name;
    public User() {
    }
    public String getName() {
        return _name;
    }
    public void setName (String name) {
    } _name = name;
}
```

✓ ②

3. A Java Bean is a simple Model class as explained above while an Enterprise Java Bean has to meet JEE's strict specifications. It can implement JPA for example. It is still a model in the MVC architecture and still contains the working logic. Confusing but not false (1)

4. The answer b) $bean.getName() is not correct. ✓ (1)

5. To show "wind" on our page, we would do:
${Key_nature['2']} ✓ (2)

6. a) This instruction will throw an error (exception)

b) We cannot set the same attribute twice in the same scope (request scope).

7. a) We have used the following syntaxes to forward:
In JSPs:
- <jsp:forward page="myURL or File"></jsp:forward>
- <c:redirect url="myURL or File"></c:redirect>
In controller:
getRequestDispatcher("WEB-INF/path/to/jsp").forward(request, response);

What is the Java code behind that? ✓ (1)

b) Both <jsp:forward and the method in the controller use

the Request Dispatcher while the <c:redirect approach? works on the client side, asking for the browser to redirect.

8. a) To make "main.jsp" the entry point of our app, we would add the following to web.xml:

(1)

```
<welcome-file-list>
   <welcome-file>main.jsp</welcome-file>
</welcome-file-list>
```

b) For the servlet, in the web.xml:

```
<servlet>
   <servlet-name>controller</servlet-name>
   <servlet-class>package.name.Controller</servlet-class>
</servlet>

<servlet-mapping>
   <servlet-name>controller</servlet-name>
   <url-pattern>(.*)</url-pattern>
</servlet-mapping>
```

(0.75)

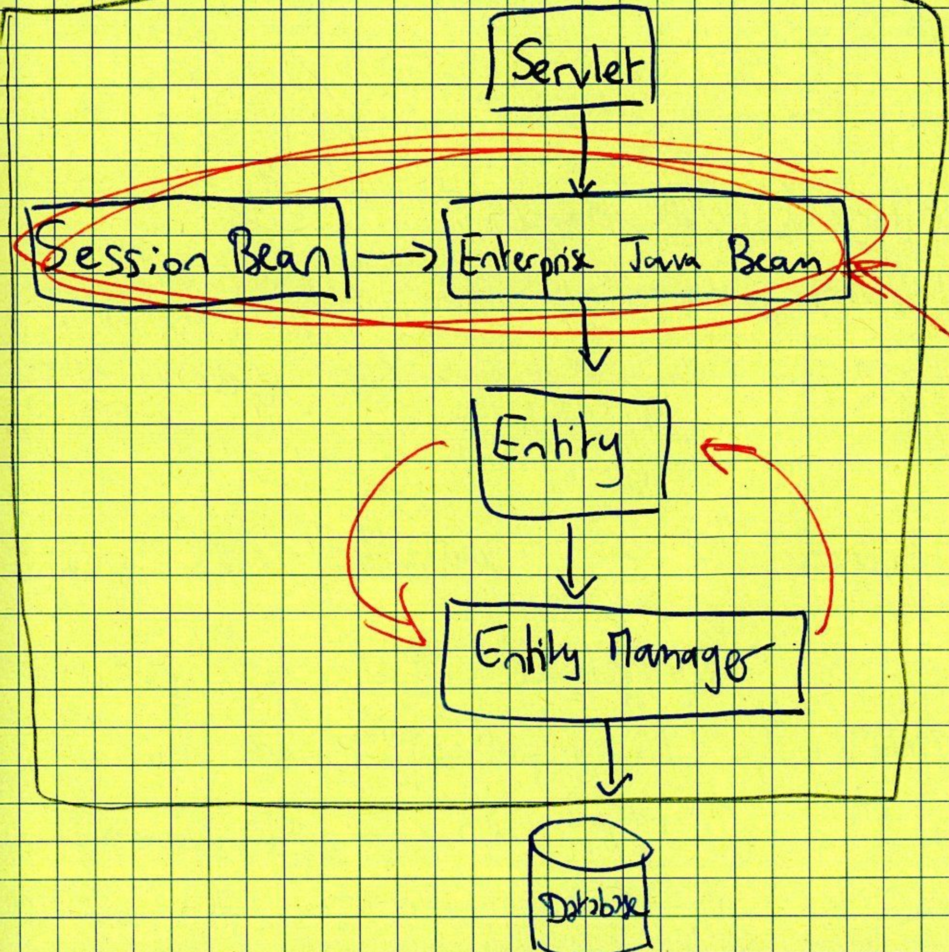
yes

Note: (.*) is a wildcard so this controller will answer to all the requests. We can make it more specific (e.g: <url-pattern>/myRoute</url-pattern>) or make it respond to the index: <url-pattern>/</url-pattern>. The package name should be changed: mn.efrei.jee.controller etc

9. The instruction request.getAttribute("cField") will not produce the same result because it only works on the server-side, meaning that an user cannot access it directly. We would have to do a request.setAttribute ("cField", value) in the Servlet to make it work. On the other hand, request.getParameter works for user input and that is what we should use. ✓ 2

10. Properties file are useful in real life because it is easier to tell deployment teams to edit only one file and not to touch XML or Java source code. This way, they only have to change a few lines in the properties file (username and password) to deploy the project. ✓ 1

11. JPA contains the following components: SessionBean, EJB, Entity



Incomplete 1