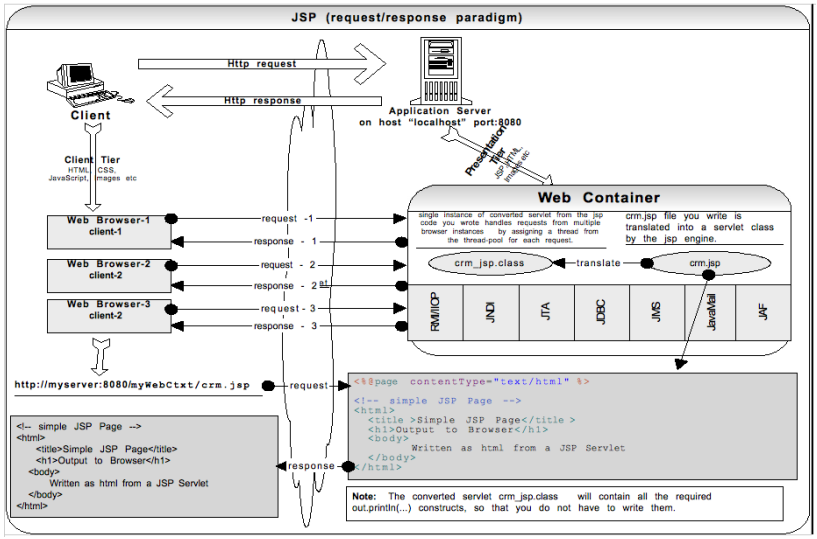
06: ♦ MVC0, MVC1, MVC2, MVP, MVVM, and MVW web design patterns Java examples

Posted on November 25, 2014 by Arulkumaran Kumaraswamipillai — No Comments ↓

Q1. What's wrong with Servlets? What is a JSP? What is it used for? What do you know about model 0 (aka **MVC0**), model 1 (aka **MVC1**) and model 2 (aka **MVC2**) patterns? In "model 2" architecture, if you set a request attribute in your JSP, would you be able to access it in your subsequent request within your servlet code? How do you prevent multiple submits due to repeated "refresh button" clicks? What do you understand by the term JSP translation phase or compilation phase?

A1. Writing **out.println (…)** statements using servlet is cumbersome and hard to maintain, especially if you need to send a long HTML page with little dynamic code content. Worse still, every single change requires recompilation of your Servlet.



MVC-0

Q2. Did JSPs make servlets obsolete?

A2. No. JSPs did not make Servlets obsolete. Both Servlets and JSPs are complementary technologies. You can look at the JSP technology from an HTML designer's perspective as an extension to HTML with embedded dynamic content and from a Java developer's as an extension of the Java Servlet technology. JSP is commonly used as the presentation layer for combining HTML and Java code. While Java Servlet technology is capable of generating HTML with out.println("<html>….. </html>") statements, where "out" is a PrintWriter. This process of embedding HTML code with escape characters is cumbersome and hard to maintain. The JSP technology solves this by providing a level of abstraction so that the developer can use custom tags and action elements, which can speed up Web development and are easier to maintain.

Q3. What is a model 0 pattern (i.e. model-less pattern) and why is it not recommended? What is a model-2 or MVC architecture?

⊞ SQL, XML, UML, JSC
⊞ Hadoop & BigData Int
⊞ Java Architecture Inte
⊞ Scala Interview Q&As
⊞ Spring, Hibernate, & M
⊞ Testing & Profiling/Sa
⊞ Other Interview Q&A f
⊞ ▶ Free Java Interviev

16 Technical Key Areas

open all | close all
⊞ Best Practice (6)
⊞ Coding (26)
⊞ Concurrency (6)
⊞ Design Concepts (7)
⊞ Design Patterns (11)
⊞ Exception Handling (3
⊞ Java Debugging (21)
⊞ Judging Experience In
⊞ Low Latency (7)
⊞ Memory Managemen
⊞ Performance (13)
⊞ QoS (8)
⊞ Scalability (4)
⊞ SDLC (6)
⊞ Security (13)
⊞ Transaction Managen

80+ step by step Java Tutorials

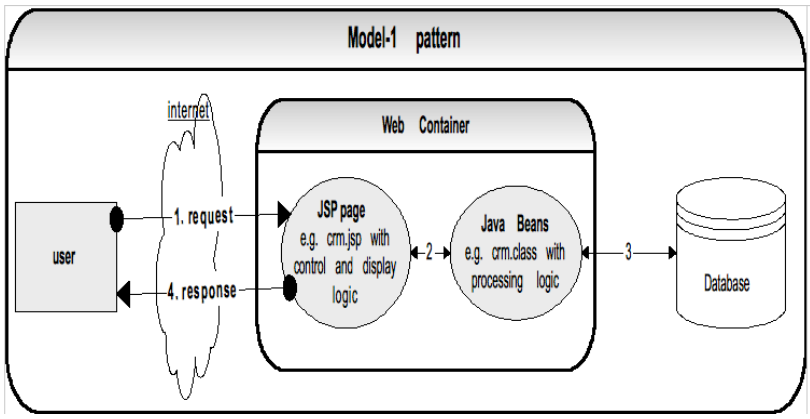
open all | close all
⊞ Setting up Tutorial (6)
⊞ Tutorial - Diagnosis (2
⊞ Akka Tutorial (9)
⊞ Core Java Tutorials (2

A3.

Problem: The example shown above is based on a model 0 (MVC0) pattern. The model 0 pattern is fine for a very basic Jsp page as shown above. But real web applications would have business logic, data access logic etc, which would make the above code hard to read, difficult to maintain, difficult to refactor, and untestable. It is also not recommended to embed business logic and data access logic in a JSP page since it is protocol dependent (i.e. HTTP protocol) and makes it unable to be reused elsewhere like a wireless application using a WAP protocol, a standalone XML based messaging application etc.

Solution: You can refactor the processing code containing business logic and data access logic into Java classes, which adhered to certain standards. This approach provides better testability, reuse and reduced the size of the JSP pages. This is known as the “model 1” pattern where JSPs retain the responsibility of a controller, and view renderer with display logic but delegates the business processing to java classes known as Java Beans. The Java Beans are Java classes, which adhere to following items:

1) Implement java.io.Serializable or java.io.Externalizable interface.
2) Provide a no-arguments constructor.
3) Private properties must have corresponding getXXX/setXXX methods.



MVC1

Hadoop & Spark Tuto
JEE Tutorials (19)
Scala Tutorials (1)
Spring & HIbernate Tu
Tools Tutorials (19)
Other Tutorials (45)

100+ Java pre-interview coding tests

open all | close all
Can you write code? (
♦ Complete the given
Converting from A to I
Designing your classe
Java Data Structures
Passing the unit tests
What is wrong with th
Writing Code Home A
Written Test Core Jav
Written Test JEE (1)

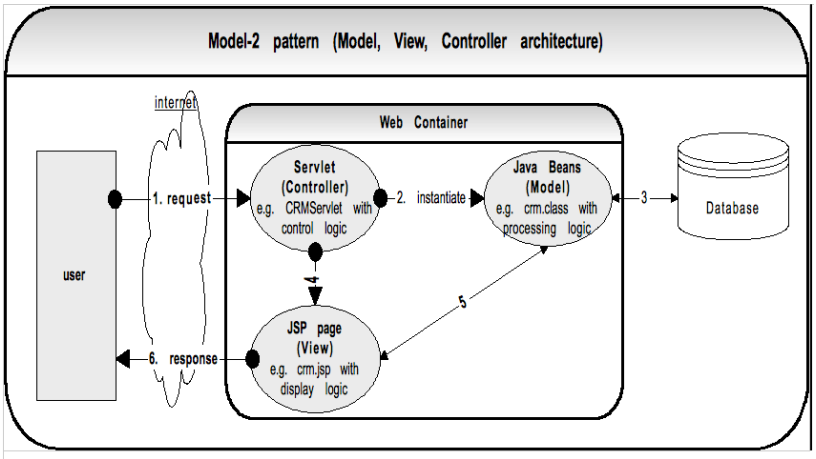
How good are your .....?

open all | close all
Career Making Know-
Job Hunting & Resum

The above model provides a great improvement from the **model 0** or **model-less pattern**, but there are still some problems and limitations.

Problem: In the **model 1** architecture the JSP page is alone responsible for processing the incoming request and replying back to the user. This architecture may be suitable for simple applications, but complex applications will end up with significant amount of Java code embedded within your JSP page, especially when there is significant amount of data processing to be performed. This is a problem not only for java developers due to design ugliness but also a problem for web designers when you have large amount of Java code in your JSP pages. In many cases, the page receiving the request is not the page, which renders the response as an HTML output because decisions need to be made based on the submitted data to determine the most appropriate page to be displayed. This would require your pages to be redirected (ie. sendRedirect (…)) or forwarded to each other resulting in a messy flow of control and design ugliness for the application. So, why should you use a JSP page as a controller, which is mainly designed to be used as a template?

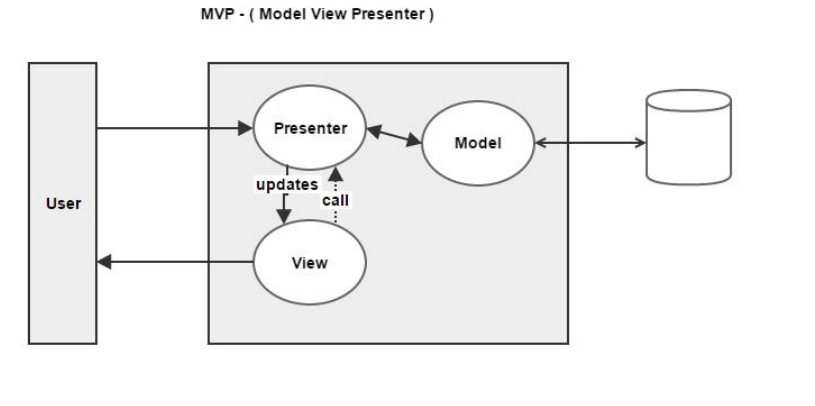
Solution: You can use the **Model 2** architecture (MVC – Model, View, Controller architecture), which is a hybrid approach for serving dynamic content, since it combines the use of both Servlets and JSPs. It takes advantage of the predominant strengths of both technologies where a Servlet is the target for submitting a request and performing flow-control tasks and using JSPs to generate the presentation layer. As shown in the diagram below, the servlet acts as the controller and is responsible for request processing and the creation of any beans or objects used by the JSP as well as deciding, which JSP page to forward or redirect the request to (i.e. flow control) depending on the data submitted by the user. The JSP page is responsible for retrieving any objects or beans that may have been previously created by the servlet, and as a template for rendering the view as a response to be sent to the user as an HTML.



MVC-2

Q4. What is an MVP architecture, and how does it differ from MVC?

A4. MVP stands for "**M**odel **V**iew **P**resenter". MVP is derived from MVC, where a "Presenter" assumes the role of a middle man, and a view is responsible for handing the UI events like button click, etc which used to be the controller's job in MVC. The "Model" in MVP is strictly a domain model.



MVP

Presenter will have a 2 way communication with the view (i.e. Presenter **<--->** View) whereas in MVC a controller has one way communication with the view (i.e. Controller **–>** View). The presenter takes care of the user tasks by communicating with the view by talking to an interface implemented by the view. At times, a view communicates with the presenter by directly calling functions on an instance of the presenter. There will be a single presenter for each view, whereas in

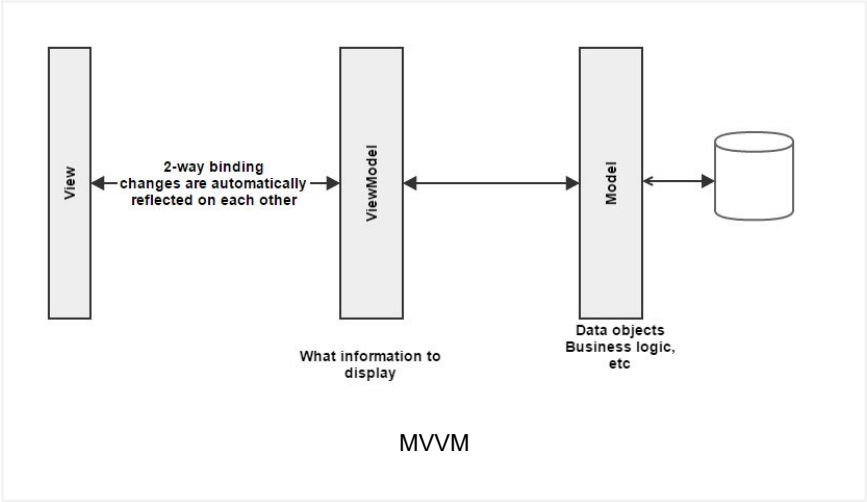
MVC a single controller will manage multiple views. This means that the Controller and the Views are more 'tightly coupled'.

Now a days, JavaScript based Web frameworks like Backbone.js, AngularJS, Ember.JS, Knockout, etc are popular and they make use of the MVW (**M**odel **V**iew **W**hatever) design patterns. "Whatever" means whatever works for you like MVC, MVP, MVVM (**M**odel **V**iew **V**iew**M**odel), etc.

The main goal with MVP is decoupling of different aspects in the code. A JavaScript based MVP architecture will have:

1) EventHandling = **Presenter**
2) DOM Manipulation = **View**
3) AJAX calls = **Model**

Q5. What is an MVVM architecture, and how does it differ from MVP?
A5. MVVM stands for **M**odel View **V**iew**M**odel. Popular JavaScript based frameworks like angularjs and knockoutjs use this pattern with so called 2 way binding.



MVVM

The ViewModel will have 2 way communication with the view, which means the fields in a view model usually match up more closely with the view than with the model. There will be a single ViewModel for each view.

View binds directly to the ViewModel. Because of the binding, changes in the view are automatically reflected in the ViewModel and changes in the ViewModel are automatically reflected in the view.

So, don't get too bogged down with all these different patterns, and implement "**Whatever**" works for you with **MVW**. The key is to separate concerns without tightly coupling them. Your goal is not to make an MVP, MVVM, or MVC system. Your goal is to separate the view, the model, and the logic that governs both of them. Knowing the semantics help you in job interviews and understanding the frameworks a bit better.

Popular Posts

♦ 11 Spring boot interview questions & answers

825 views

♦ Q11-Q23: Top 50+ Core on Java OOP Interview Questions & Answers

766 views

18 Java scenarios based interview Questions and Answers

400 views

001A: ♦ 7+ Java integration styles & patterns interview questions & answers

388 views

01b: ♦ 13 Spring basics Q8 – Q13 interview questions & answers

295 views

♦ 7 Java debugging interview questions & answers

293 views

01: ♦ 15 Ice breaker questions asked 90% of the time in Java job interviews with hints

285 views

♦ 10 ERD (Entity-Relationship Diagrams) Interview Questions and Answers

279 views

♦ Q24-Q36: Top 50+ Core on Java classes, interfaces and generics interview questions & answers

239 views

001B: ♦ Java architecture & design concepts interview questions & answers

201 views

Bio | Latest Posts

Arulkumaran Kumaraswamipillai

Mechanical Eng to freelance Java developer in 3 yrs. Contracting since 2003, and attended 150+ Java job interviews, and often got 4 - 7 job offers to choose from. It pays to prepare. So, published Java interview Q&A books via Amazon.com in 2005, and sold 35,000+ copies. Books are outdated and replaced with this subscription based site.**945+** paid members. join my LinkedIn Group. **Reviews**

About Arulkumaran Kumaraswamipillai

Mechanical Eng to freelance Java developer in 3 yrs. Contracting since 2003, and attended 150+ Java job interviews, and often got 4 - 7 job offers to choose from. It pays to prepare. So, published Java interview Q&A books via Amazon.com in 2005, and sold 35,000+ copies. Books are outdated and replaced with this subscription based site.**945+** paid members. join my LinkedIn Group. **Reviews**

‹ 06: ♦ 10+ Atomicity, Visibility, and Ordering interview Q&A in Java multi-threading

Top 7+ Java productivity tools that make your life easier as a Java developer ›

Posted in FAQ JEE Job Interview Q&A Essentials, JEE Patterns, member-paid, Web basics

Tags: Architect FAQs

Leave a Reply

Logged in as geethika. Log out?

Comment

Post Comment

Empowers you to open more doors, and fast-track

Technical Know Hows

☀ Java generics in no time ☀ Top 6 tips to transforming your thinking from OOP to FP ☀ How does a HashMap internally work? What is a hashing function?
☀ 10+ Java String class interview Q&As ☀ Java auto un/boxing benefits & caveats ☀ Top 11 slacknesses that can come back and bite you as an experienced Java developer or architect

Non-Technical Know Hows

☀ 6 Aspects that can motivate you to fast-track your career & go places ☀ Are you reinventing yourself as a Java developer? ☀ 8 tips to safeguard your Java career against offshoring ☀ My top 5 career mistakes

Prepare to succeed

☀ Turn readers of your Java CV go from “Blah blah” to “Wow”? ☀ How to prepare for Java job interviews? ☀ 16 Technical Key Areas ☀ How to choose from multiple Java job offers?

Select Category

© Disclaimer

The contents in this Java-Success are copy righted. The author has the right to correct or enhance the current content without any prior notice.

These are general advice only, and one needs to take his/her own circumstances into consideration. The author will not be held liable for any damages caused or alleged to be caused either directly or indirectly by these materials and resources. Any trademarked names or labels used in this blog remain the property of their respective trademark owners. No guarantees are made regarding the accuracy or usefulness of content, though I do make an effort to be accurate. Links to external sites do not imply endorsement of the linked-to sites.

↑