

[Home](#) › [Interview](#) › [Spring, Hibernate, & Maven Interview Q&A](#) › [Git & SVN](#) › 6 more Git interview Q&A

6 more Git interview Q&A

Posted on [June 11, 2015](#) by [Arulkumaran Kumaraswamipillai](#)

Extending [8 Git Source control system interview questions & answers](#).

Q1. What do you understand by the term patch or pull request?

A1. A patch means change sets you want to communicate and apply to another repository. Nowadays, the GitHub “pull request” makes it really easy to apply patches on GitHub repos, which is useful when you aren’t a direct contributor. A patch is a small file that indicates what was changed in a repository. It’s generally used when someone from outside your team has read-only access but had a good code change available. He/she then creates a patch and sends it to you.

Q2. How do you create a “pull request” (i.e. a patch) in Git?

A2. You can create a “pull” request from Git repository management systems like **Stash** or **GitHub** to merge changes from your bug fixes branch say “bugfix/ISSUE-161” to myapp-integration branch. Someone has to review and approve the changes. Once the changes are approved, they can be merged. Git repository management systems provide

600+ Full Stack Java/JEE Interview Q&As ♥Free ♦FAQs

[open all](#) | [close all](#)

[Ice Breaker Interview](#)

[Core Java Interview C](#)

[JEE Interview Q&A \(3](#)

[Pressed for time? Jav](#)

[SQL, XML, UML, JSC](#)

[Hadoop & BigData Int](#)

[Java Architecture Inte](#)

[Scala Interview Q&As](#)

[Spring, Hibernate, & I](#)

[Spring \(18\)](#)

[Hibernate \(13\)](#)

[AngularJS \(2\)](#)

[Git & SVN \(6\)](#)

[♥ Git & Maven fc](#)

[♥ Merging Vs rel](#)

[♥ Understanding](#)

[6 more Git interv](#)

[8 Git Source cor](#)

[Setting up Cygw](#)

[JMeter \(2\)](#)

[JSF \(2\)](#)

[Maven \(3\)](#)

the facility to review the changes and approve them. At times, if there are any conflicting artifacts in the from (i.e. bugfix/ISSUE-161) and target (i.e. myapp-integration) branches, you will not be able to merge it, and you need to follow the steps described below.

Q3. When applying a patch or executing a pull request, if you get merge conflicts, how will you go about resolving them?

A3. They can be resolved applying the following steps.

Merging conflicting changes in Git

Step 1: Fetch the changes (saving the target branch as FETCH_HEAD).

```
1 git fetch origin myapp-integration
```

Step 2: Checkout the source branch “bugfix/ISSUE-161” and merge in the changes from the target branch “myapp-integration”. Resolve any conflicts manually if they exist.

```
1 git checkout bugfix/ISSUE-161
2 git merge FETCH_HEAD
```




Step 3: After the merge conflicts are resolved, stage the changes accordingly, commit the changes and push.

```
1 git commit
2 git push origin HEAD
```

Step 4: Merge the updated pull request in a Git repository management system like Stash or GitHub.

















Q4. How will you go about rebasing your changes with someone else’s changes?

A4. Assume that you are currently working on the “myapp-integration” branch and when you try to push your changes to the origin repository, you get an error “error: failed to push some refs to ‘...’”. This is because Git notices that the

-  [Testing & Profiling/Sa](#)
-  [Other Interview Q&A 1](#)
-  [Free Java Interview](#)

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 I](#)
-  [Low Latency \(7\)](#)
-  [Memory Management](#)
-  [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 \(2\)](#)
-  [Hadoop & Spark Tuto](#)
-  [JEE Tutorials \(19\)](#)
-  [Scala Tutorials \(1\)](#)
-  [Spring & Hibernate T](#)
-  [Tools Tutorials \(19\)](#)

“myapp-integration” branch in the origin repository no longer matches ‘origin/myapp-integration’ so it won’t allow the push. Some artifacts have been changed by others and “pushed” to the origin. So, you need to rebase your local working repository with the “origin/master”

Step 1: Fetch someone else’s changes from the origin repository.

```
1 git fetch
2 git log --graph --all
3
```

Step 2: Rebase the changes.

```
1 git rebase origin/myapp-integration
```

Step 3: If no conflicts, push it to “origin/myapp-integration”

```
1 git push
```

If there are conflicts

Step 4: Manually resolve the conflicts and stage the changes. After staging, continue with the rebasing.

```
1 git add pom.xml
2 git rebase --continue
3
```

Step 5: Once rebased, you can push it to the “origin/myapp-integration”

```
1 git push
```

Q5. Can you describe the terms origin, master, head, fetch_head, etc relating to Git?

A5. Understanding Git terms origin, master, and head

[Other Tutorials \(45\)](#)

100+ Java pre-interview coding tests

[open all](#) | [close all](#)

- [Can you write code? \(1\)](#)
- [♦ 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](#)

Q6. How does merging differ from rebasing?

A6. [Merging Vs rebasing on Git questions and answers](#)

Popular Member Posts

♦ [11 Spring boot interview questions & answers](#)

904 views

♦ [Q11-Q23: Top 50+ Core on Java OOP Interview Questions & Answers](#)

816 views

001A: ♦ [7+ Java integration styles & patterns interview questions & answers](#)

427 views

[18 Java scenarios based interview Questions and Answers](#)

408 views

♦ [7 Java debugging interview questions & answers](#)

324 views

01b: ♦ [13 Spring basics Q8 – Q13 interview questions & answers](#)

311 views

01: ♦ [15 Ice breaker questions asked 90% of the time in Java job interviews with hints](#)

304 views

♦ [10 ERD \(Entity-Relationship Diagrams\) Interview Questions and Answers](#)

301 views

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

251 views

♦ [Object equals Vs == and pass by reference Vs value](#)

234 views

0

Like

Share

Tweet

↑

submit

↓

reddit

0

G+1

Share

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](https://www.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](https://www.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](#)

◀ ♥ Understanding Git terms origin, master, and head

Top 5 tips for software developers on blogging for additional income ▶

Posted in Git & SVN, member-paid

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.