

# Java-Success.com

Industrial strength Java Career Companion


[Home](#)
[Java FAQs](#)
[600+ Java Q&As](#)
[Career](#)
[Tutorials](#)
[Member](#)
[Why?](#)
[Can u Debug?](#)
[Java 8 ready?](#)
[Top X](#)
[Productivity Tools](#)
[Judging Experience?](#)

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

Understanding Git terms origin, master, and head

## ♥ Understanding Git terms origin, master, and head

Posted on [June 10, 2015](#) by [Arulkumaran Kumaraswamipillai](#)

**Q1.** What is a .git folder?

**A1.** Like .SVN folder for subversion holding the metat data, .git folder holds the Git meta data like remote repository, branch names, etc.

```
0 Jun 9 11:27 ./
0 Jun 5 18:43 ../
46 Jun 5 18:47 COMMIT_EDITMSG
134 Jun 9 11:33 FETCH_HEAD
34 Jun 5 10:32 HEAD
41 Jun 5 10:31 ORIG_HEAD
525 Jun 5 10:32 config
73 Jun 4 12:12 description
0 Jun 4 12:12 hooks/
29K Jun 9 11:27 index
0 Jun 4 12:12 info/
0 Jun 4 12:16 logs/
0 Jun 9 11:33 objects/
6.8K Jun 4 12:16 packed-refs
0 Jun 4 12:16 refs/
```

.git folder when did a listing with "ls -la"

If you do

### 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 \(3\)](#)

```
1 cat head
```

it shows that you are on the branch “**myapp-integration**”

```
1 ref: refs/heads/myapp-integration
```

take note of the “**config**, **HEAD**, **MERGE\_HEAD**, and **FETCH\_HEAD**” files discussed later on.

**Q2.** Can you explain the terms “origin”, “master”, and “head” that gets used very often in git?

**A2.**

**origin** is a name commonly given to the main remote. remote is another repository that you can pull from and push to. Usually it's on some server, like github. On any git project, if you change directory to “.git” and then do “cat config”. Cloning with “git clone url” sets up the [remote-name], [branch-name], etc in the “**.git/config**” file.

```
1 cd .git # change directory to
2 cat config
3
```

```
1
2 [core]
3     repositoryformatversion = 0
4     filemode = true
5     bare = false
6     logallrefupdates = true
7     ignorecase = true
8 [remote "origin"]
9     fetch = +refs/heads/*:refs/remotes/origin/*
10    url = ssh://git@stash.internal.myhost.com:2222/git
11 [branch "master"]
12     remote = origin
13     merge = refs/heads/master
14     rebase = true
15 [branch "myapp-integration"]
16     remote = origin
17     merge = refs/heads/myapp-integration
18     rebase = true
19 [branch "bugfix/ISSUE-161"]
20     remote = origin
21     merge = refs/heads/bugfix/ISSUE-161
22     rebase = true
23
```

- ✚ 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 In
- ✚ 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 Ti
- ✚ Tools Tutorials (19)

**master** is the name of the default branch that git creates for you when first creating a repository. It is a name commonly given to the main branch, but it could be called anything else (or there could be no main branch). A typical command to push your local repository changes to your remote repository would be “git push [remote-name] [branch-name]”

```
1 git push origin master
```

**Note:** The above command works only if you cloned from a server. If you want to see more information about a particular remote, you can use the git remote show [remote-name]

```
1 git remote show origin
```

**HEAD** is the current revision. Most of the time HEAD points to the latest commit in your branch, but that doesn't have to be the case.

```
1 git checkout master #checkouts the latest
2 git checkout HEAD~2 #go back two revisions
3 git checkout <previous version hash> #go back to
4
```

Resetting is a way to move the tip of a branch to a different commit. For example, the following command moves the “bugfix/ISSUE-161” branch backwards by two commits.

```
1 git checkout bugfix/ISSUE-161
2 git reset HEAD~2
3
```

**HEAD is like a symbolic reference**

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

## 100+ Java pre-interview coding tests

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

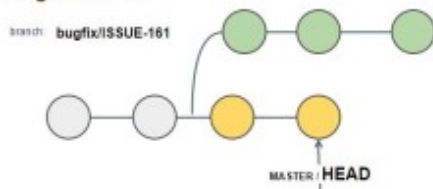
- [Can you write code? \(](#)
- [♦ Complete the given](#)
- [Converting from A to I](#)
- [Designing your classe](#)
- [Java Data Structures](#)
- [Passing the unit tests](#)
- [What is wrong with th](#)
- [Writing Code Home A](#)
- [Written Test Core Jav](#)
- [Written Test JEE \(1\)](#)

## How good are your .....?

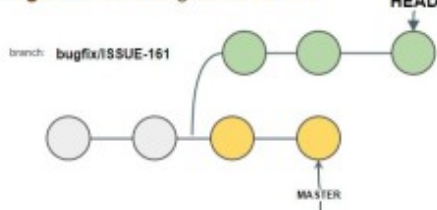
[open all](#) | [close all](#)

- [Career Making Know-](#)
- [Job Hunting & Resum](#)

After: **git clone url**



After: **git checkout bugfix/ISSUE-161**



You can always cat the “head” file in the **.git** folder.

```
1 cat head
```

## What are ORIG\_HEAD, FETCH\_HEAD, and MERGE\_HEAD?

You can see this file “ORIG\_HEAD” and “FETCH\_HEAD” in the **.git** folder shown above.

**FETCH\_HEAD** is a short-lived reference to keep track of what has just been fetched from the remote repository. “git pull” first invokes “git fetch” to fetch a branch from the remote.

```
1 git fetch origin master      #fetch from the ori
2 git checkout bugfix/ISSUE-161 #checkout bugfix/IS
3 git merge FETCH_HEAD         #merge changes in m
4
```

**ORIG\_HEAD** is previous state of HEAD. “pull” or “merge” leaves the original tip of the current branch in ORIG\_HEAD. Git “merge” always sets ‘.git/ORIG\_HEAD’ to the original state of HEAD so a problematic merge can be removed by using

```
1 git reset ORIG_HEAD
2
```

**MERGE\_HEAD** records the commit(s) which you are merging into your branch when you run git merge. This is the SHA of the branch you're merging from.

Undoing the merge

```
1 git merge --abort #[Since git version 1.7.4]
2 git reset --merge #[prior git versions]
3
```

“git merge –abort” is equivalent to “git reset –merge” when **MERGE\_HEAD** is present. “git diff” will show a three-way diff, highlighting changes from both the **HEAD** and **MERGE\_HEAD** versions.

## 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

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](#)

◀ What are good real life Java inheritance (or use of abstract classes)

examples ?

6 more Git interview Q&A ▶

**Posted in** Git & SVN

## 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.