

Using Git Branches with Your Team



Craig Golightly

SENIOR SOFTWARE CONSULTANT

@seethatgo www.seethatgo.com



Overview



Remotes, pull, push, diff

Avoiding conflicts

- .gitignore file

Best practices for team members

- Feature flags

What can go wrong?

- Whitespace errors
- Break the build

Pull requests

Resolve difficult merge conflicts



Remotes



Location where project is hosted

- GitHub
- Bitbucket
- GitLab
- Anywhere

Reference for project

- Team milestones
- Testing
- Code reviews



```
https://service.com/project/a.git
```

```
git clone <remote-url>
```

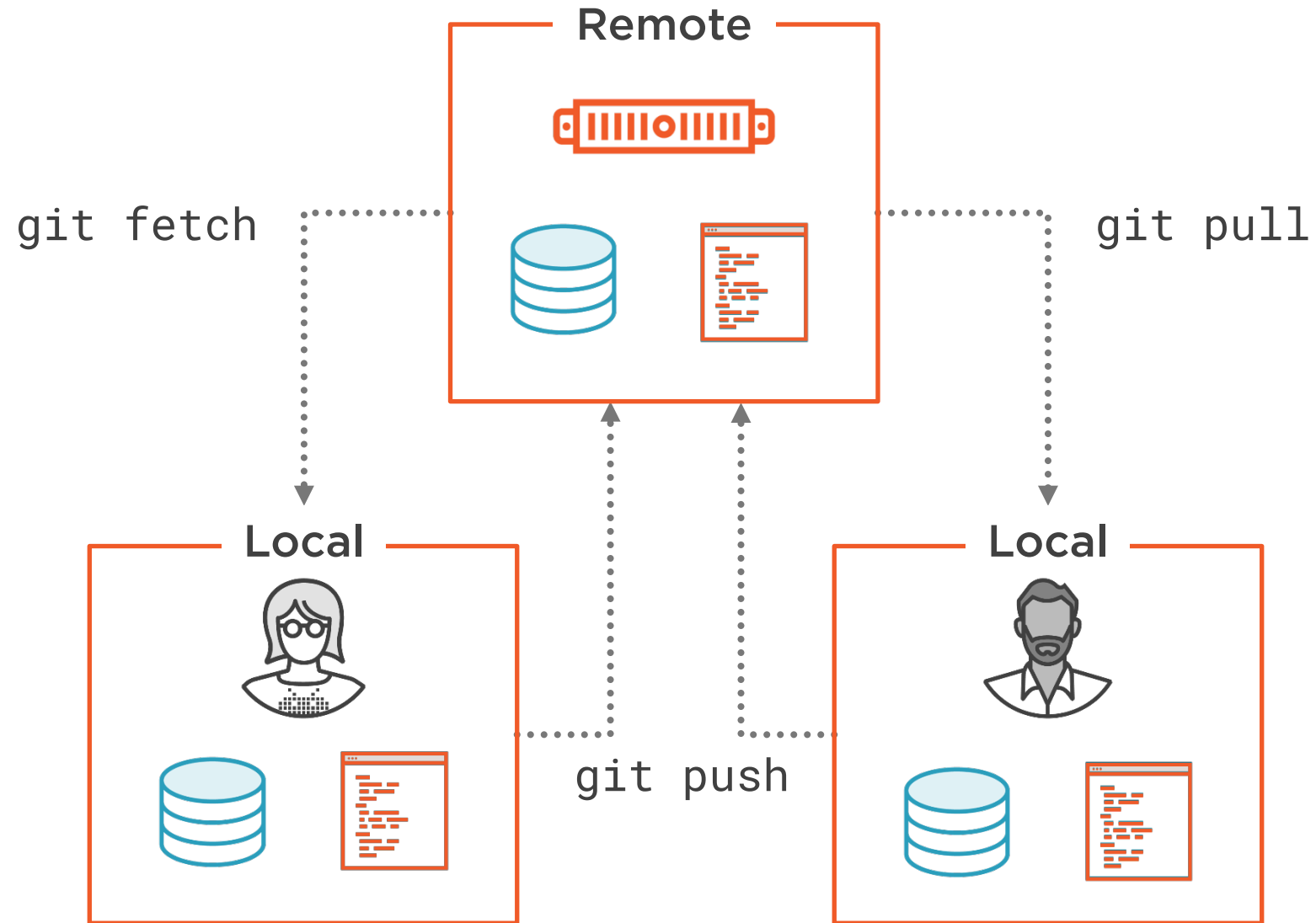
```
origin
```

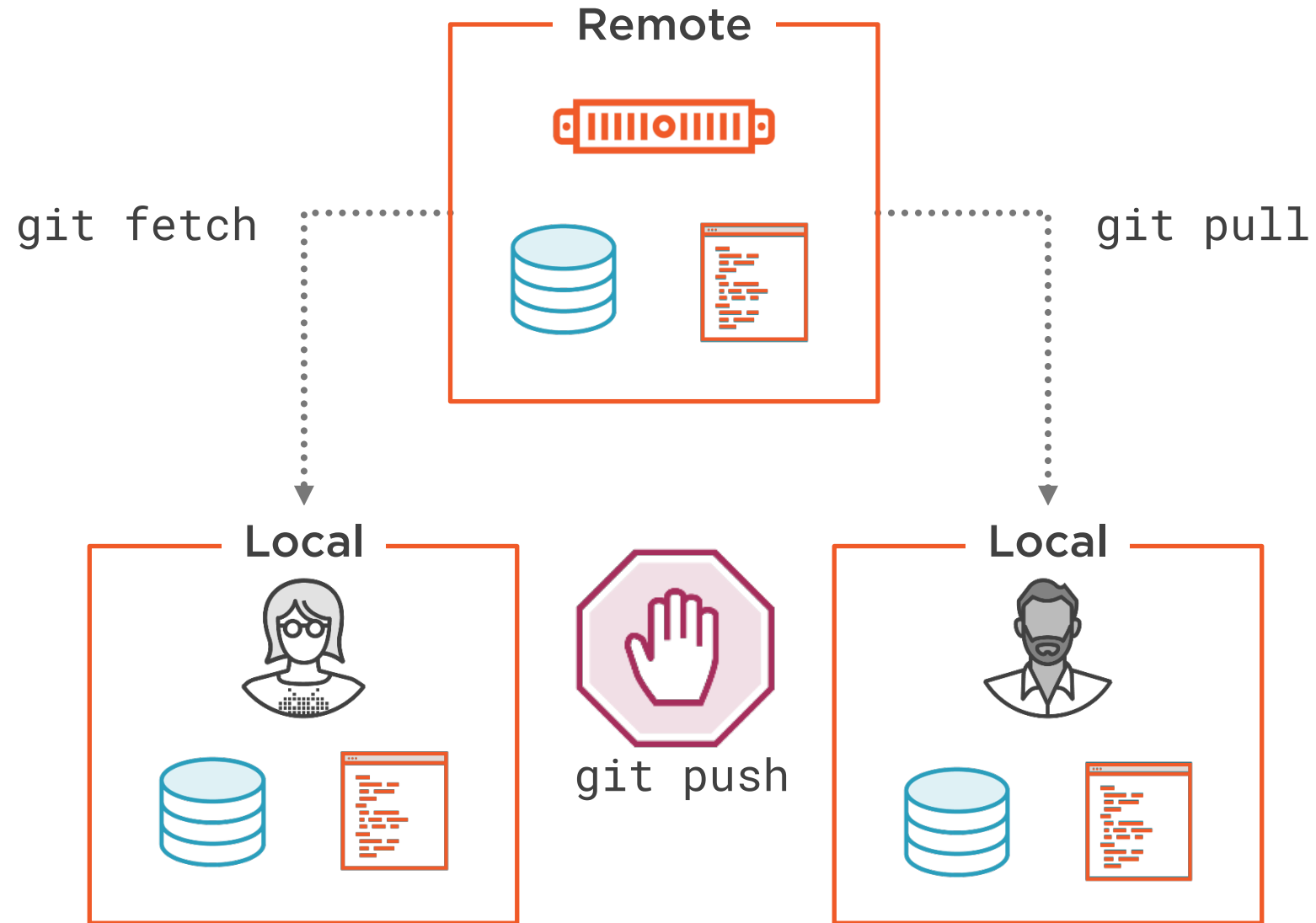
```
git remote add <name> <remote-url>
```

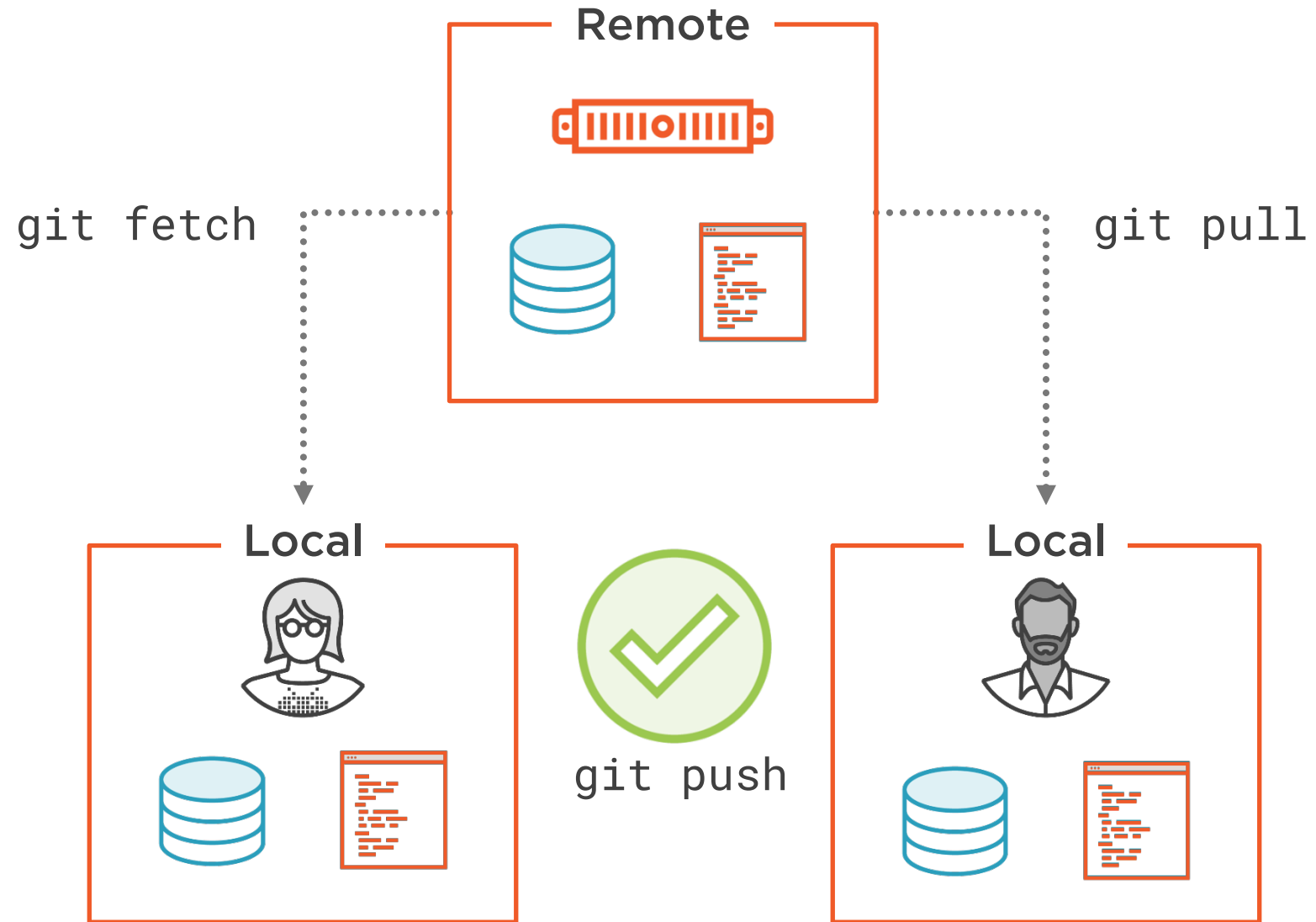
```
git remote -v
```

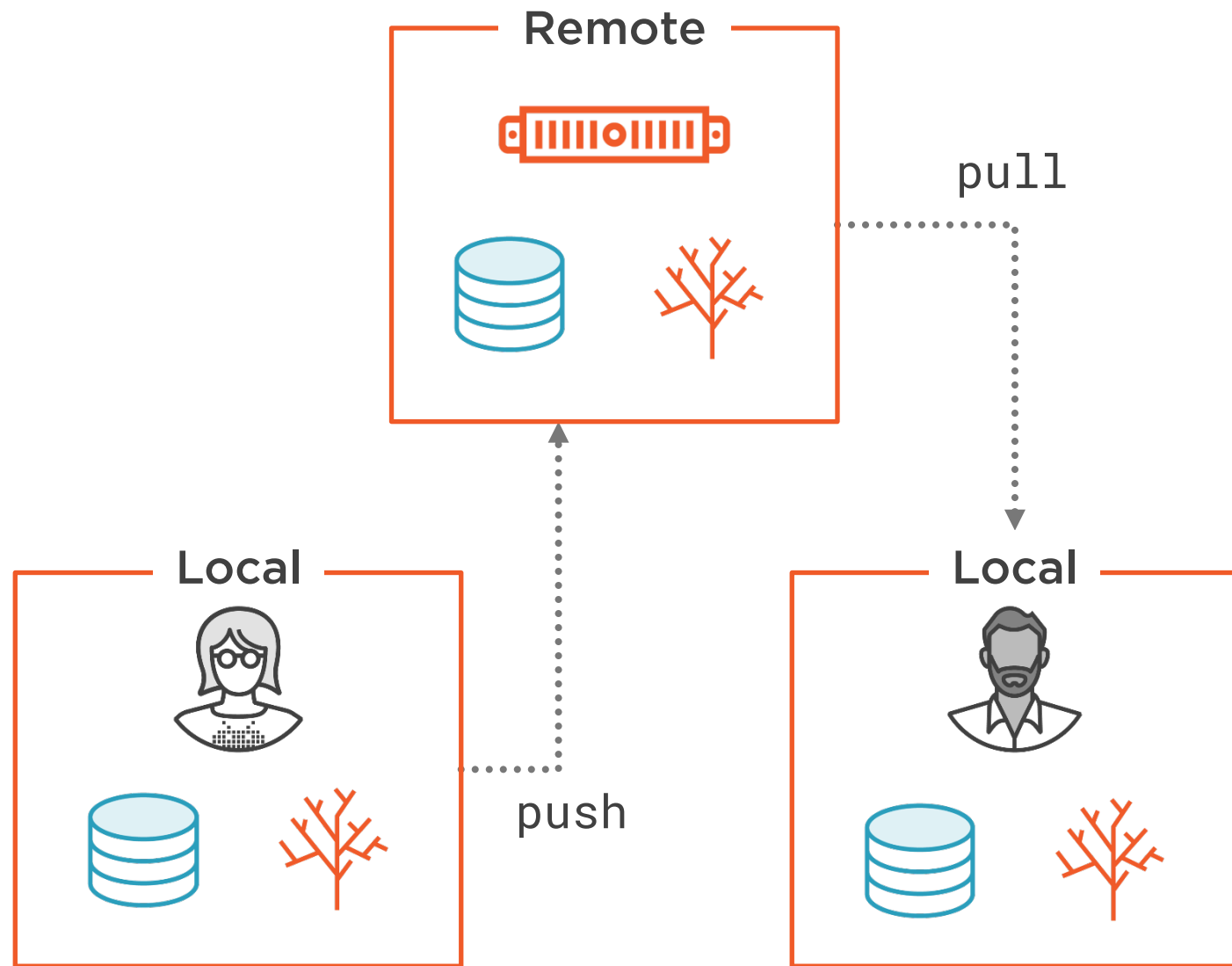
- ◀ Remote URL
- ◀ Pull code and set up local branch
- ◀ Default name for remote server
- ◀ Provide the name for a remote server
- ◀ List remotes and URLs











Demo



Remote branches

Create a local branch with changes

Push the branch to remote

List remote branches

Fetch branch

Checkout branch



```
git diff
git diff --cached    (git commit)
git diff HEAD        (git commit -a)
```

```
git diff <commit>
git diff --cached <commit>
git diff <commit> <commit>
```

```
git diff feature master
git diff feature...master
git diff feature master file.txt
```

- ◀ Changed and not staged for commit?
- ◀ Changed and staged for commit?
- ◀ Changed since last commit?

- ◀ Specific commit and current?
- ◀ Specific commit and staged?
- ◀ Difference between two commits?

- ◀ Difference between tips of branches?
- ◀ Changed in master since feature was started off of it
- ◀ Difference in file.txt on two branches?



```
diff --git a/example.txt b/example.txt
```

```
index 747e2b3..8ef0a69 100644
```

```
--- a/example.txt
```

```
+++ b/example.txt
```

```
@@ -3,6 +3,7 @@
```

yellow

green

blue

-purple

+violet

brown

black

+gray

◀ Label a and b for files being compared

◀ Git hashes of file before and after

◀ `git show <hash>` to see file

◀ a file marker (---)

◀ b file marker (+++)

◀ `/dev/null` if new or deleted file

◀ @@ are just markers for chunk header

◀ 6 lines of file a starting at line 3

◀ 7 lines of file b starting at line 3

◀ Lines with no file marker are the same in both a and b



```
diff --git a/example.txt b/example.txt
```

```
index 747e2b3..8ef0a69 100644
```

```
--- a/example.txt
```

```
+++ b/example.txt
```

```
@@ -3,6 +3,7 @@
```

```
yellow
```

```
green
```

```
blue
```

```
-purple
```

```
+violet
```

```
brown
```

```
black
```

```
+gray
```

◀ Label a and b for files being compared

◀ Git hashes of file before and after

◀ `git show <hash>` to see file

◀ a file marker (---)

◀ b file marker (+++)

◀ `/dev/null` if new or deleted file

◀ @@ are just markers for chunk header

◀ 6 lines of file a starting at line 3

◀ 7 lines of file b starting at line 3

◀ Lines with no file marker are the same in both a and b



Ignore File



.gitignore

Files are

- Tracked
- Untracked
- Ignored

Generated files are usually ignored

Ignored files not added to branches

* ? ! / [a-zA-Z]

#this is a comment

** /bin

*.zip

/bin

git rm --cached <filename>

git rm <filename>

.git/info/exclude

- ◀ anything, one character, negator, directory separator, range
- ◀ # for comments
- ◀ ** matches any directory in repository
- ◀ * matches any file in repository
- ◀ relative to .gitignore directory
Put a .gitignore at your project root!
- ◀ Delete file from Git repo or delete from repo and local filesystem
- ◀ Ignore patterns for your system only



Prevention and Treatment



Code conflicts



Merging tools



**Team conventions and
best practices**



Team Conventions



.gitignore file



README.md



Frequent smaller commits vs. one giant commit



Coordination of large refactors, renames, reformats



Don't Break the Build



Automated Tests

Timer on master
Pull request merge



Errors

Syntax errors
Unit test failures



“On it.”

Get help
Report when fixed



Commits

Commit related changes

If cherry-picked would it make sense?

Good subject line

< 50 characters

Add a body (when appropriate)

WHY change was made



```
if (flag)
{
    //new code
}
else
{
    //existing code
}

flag = false
```

- ◀ If feature flag is true, run the new code
- ◀ New functionality is separated while in progress, easy to identify
- ◀ Run existing code by default
- ◀ Buildable code each commit - easier to commit frequently
- ◀ Rollback is easy - change flag to false
Also used for gradual rollout



```
//int function foo() {  
//  int x = 1;  
//  int y = 2;  
//  return x + y;  
//}
```

Commented Out Code

If it was committed, Git will remember it

Costs time of everyone who reads it

Let Git do its job of keeping a history



```

535 src/main/java/rocks/nt/apm/jmeter/JMeterInfluxDBBackendListenerClient.java
... @@ -1,286 +1,277 @@
1  - package rocks.nt.apm.jmeter;
2  -
3  - import java.util.ArrayList;
4  - import java.util.HashSet;
5  - import java.util.List;
6  - import java.util.Set;
7  - import java.util.Random;
8  - import java.util.concurrent.Executors;
9  - import java.util.concurrent.ScheduledExecutorService;
10 - import java.util.concurrent.TimeUnit;
11 -
12 - import org.apache.jmeter.config.Arguments;
13 - import org.apache.jmeter.samplers.SampleResult;
14 - import org.apache.jmeter.threads.JMeterContextService;
15 - import org.apache.jmeter.threads.JMeterContextService.ThreadCounts;
16 - import org.apache.jmeter.visualizers.backend.AbstractBackendListenerClient;
17 - import org.apache.jmeter.visualizers.backend.BackendListenerContext;
18 - import org.apache.jorphan.logging.LoggingManager;
19 - import org.apache.log.Logger;
20 - import org.influxdb.InfluxDB;
21 - import org.influxdb.InfluxDBFactory;
22 - import org.influxdb.dto.Point;
23 - import org.influxdb.dto.Point.Builder;
24 -
25 - import rocks.nt.apm.jmeter.config.influxdb.InfluxDBConfig;
26 - import rocks.nt.apm.jmeter.config.influxdb.RequestMeasurement;
27 - import rocks.nt.apm.jmeter.config.influxdb.TestStartEndMeasurement;
28 - import rocks.nt.apm.jmeter.config.influxdb.VirtualUsersMeasurement;
29
30 + public class JMeterInfluxDBBackendListenerClientpackage rocks.nt.apm.jmeter;
31 +
32 + import java.util.ArrayList;
33 + import java.util.HashSet;
34 + import java.util.List;
35 + import java.util.Set;
36 + import java.util.Random;
37 + import java.util.concurrent.Executors;
38 + import java.util.concurrent.ScheduledExecutorService;
39 + import java.util.concurrent.TimeUnit;
40 +
41 + import org.apache.jmeter.config.Arguments;
42 + import org.apache.jmeter.samplers.SampleResult;
43 + import org.apache.jmeter.threads.JMeterContextService;
44 + import org.apache.jmeter.threads.JMeterContextService.ThreadCounts;
45 + import org.apache.jmeter.visualizers.backend.AbstractBackendListenerClient;
46 + import org.apache.jmeter.visualizers.backend.BackendListenerContext;
47 + import org.apache.jorphan.logging.LoggingManager;
48 + import org.apache.log.Logger;
49 + import org.influxdb.InfluxDB;
50 + import org.influxdb.InfluxDBFactory;
51 + import org.influxdb.dto.Point;
52 +

```

Whitespace is invisible, right?

- Not to Git

Can drown out real changes in code

Pick a formatting standard

- Can automate it

Line endings

- Differences can cause problems
- Pick one and enforce it



Pull Requests

**Push branch
to remote**

Open pull request

Add reviewers

Discuss changes

**Make fixes in
local branch**

Push to remote

Do not rebase

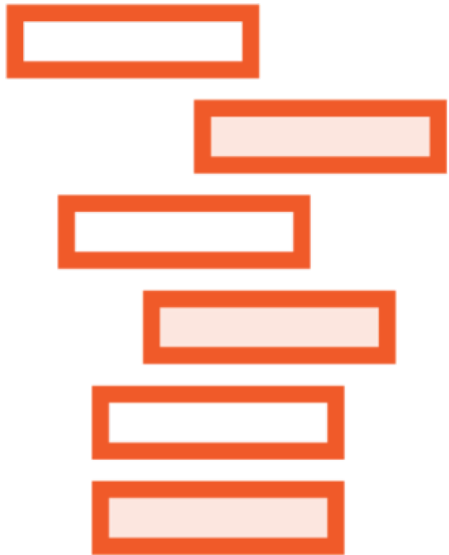
**After pushing
to remote**

Merge to master

Delete branch



Merging Branches



Conflict

- Changes that occurred in both branches
- Git needs you to decide how to combine

File markers indicate conflicting changes

Edit the file to combine the content

Add and commit to create merge commit

Merge commit has 2 parent commits

Merge Tips

Update from remote

Before starting merge

Combine existing content

Add new content as a separate commit in a branch before the merge




```
<<<<<< HEAD  
yellow  
green  
light purple  
=====  
bright yellow  
green  
dark purple  
>>>>>> master
```

```
bright yellow  
green  
light purple
```

- ◀ This is the information in your branch
- ◀ Separator for other file content
- ◀ This is the information in the file from the branch you are merging in
- ◀ Resolve conflicts, save and commit file
- ◀ Be sure to remove file markers



Demo



Start merge

Resolve conflicts

Complete Merge



Demo



Abort merge

Add commit

Restart and complete merge



Summary



Use and manage remotes

Resolve merge conflicts

- git diff

Avoid conflicts

- .gitignore
- Feature flags
- Update branches often

Pull request

- Easier code reviews

Git to work

- Comfortable using branches

