Strive

SCHOOL

GIT

Hello Strivers!

about me

Riccardo
Gulin
Teacher

- / Master’s degree in Computer Engineering
- / Freelance developer
- / Senior software architect @ Nucleode srl
- / NodeJS services certified developer
- / MongoDB certified developer

nucleode

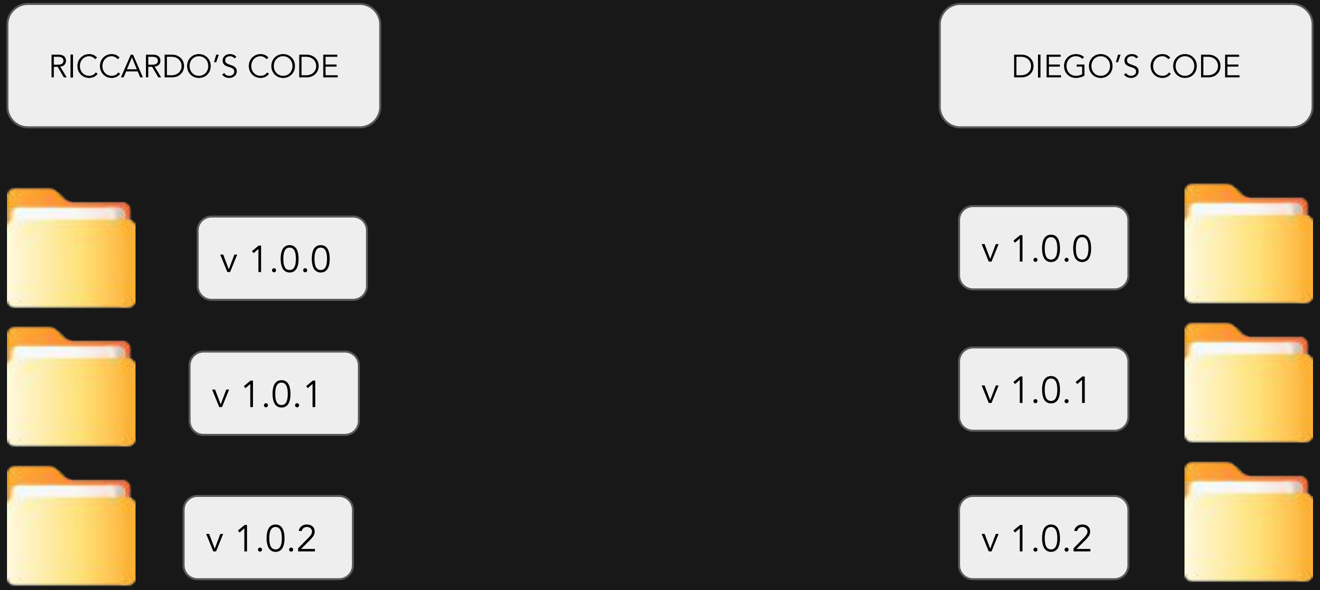
Git

/ This morning will be focused on understanding and working with GIT, one of the most widely used collaborative tools for developers

Git

- Git is a DVCS (Distributed Version Control System)
- Git lets you work together
- Git lets you track the full history of your projects
- Git lets you save your precious projects!

Without Git



RICCARDO’S CODE

v 1.0.0

v 1.0.1

v 1.0.2

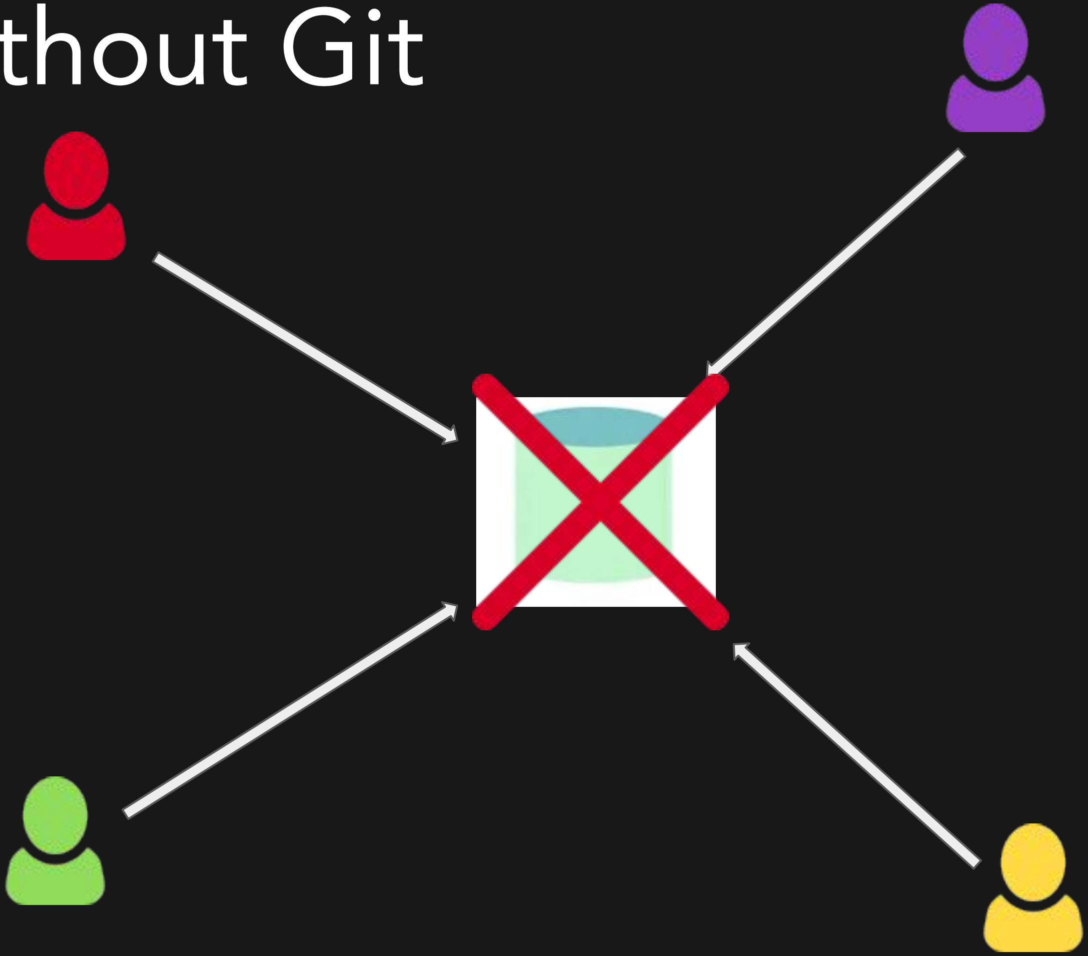
DIEGO’S CODE

v 1.0.0

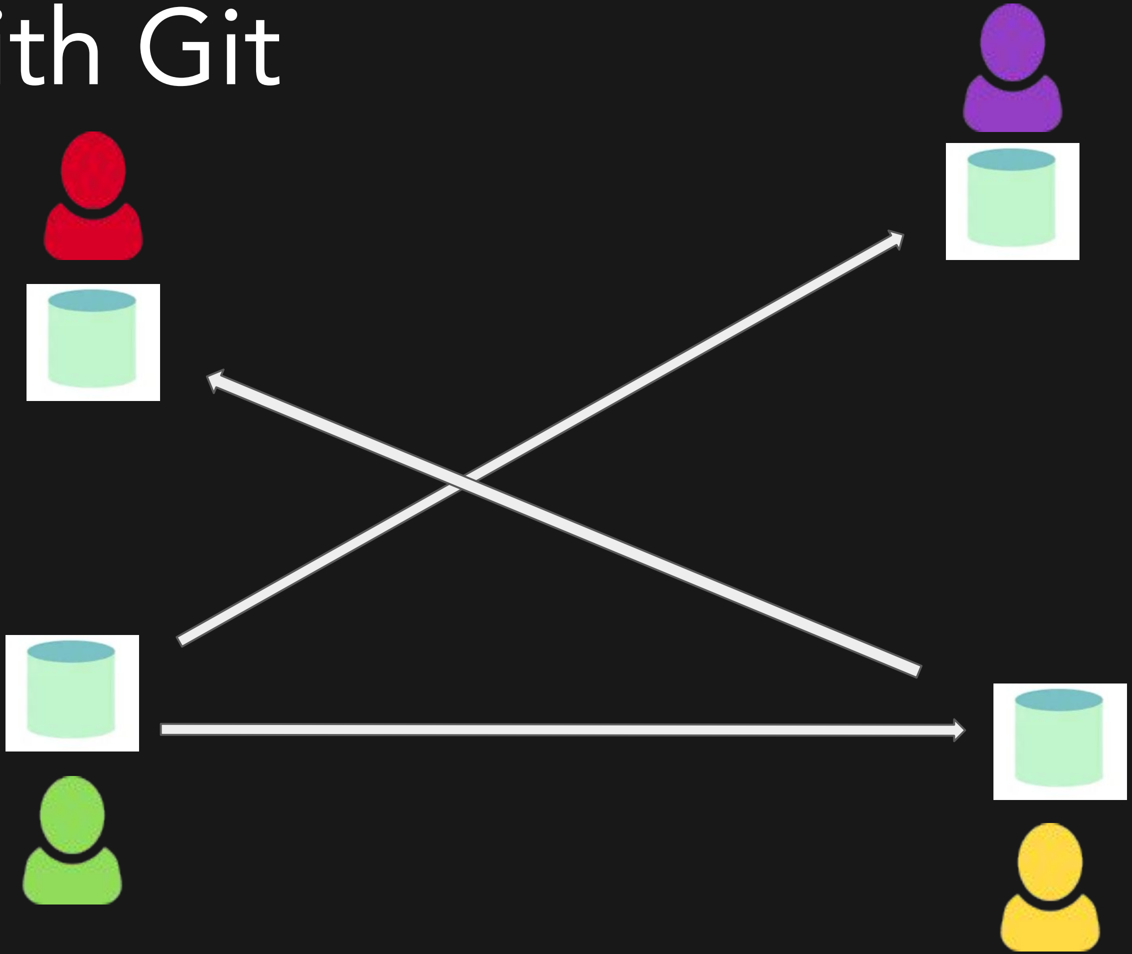
v 1.0.1

v 1.0.2

Without Git



With Git



How can we use Git?

- From Command Line
- From Code Editors
- Git GUIs (GitHub Desktop, GitKraken, SourceTree, GitFork, ...)

Verify if it is already installed

- / Mac → command + space open terminal
- / Windows → windows + r cmd
- / Linux → bash

```
$ git --version
```

Install Git

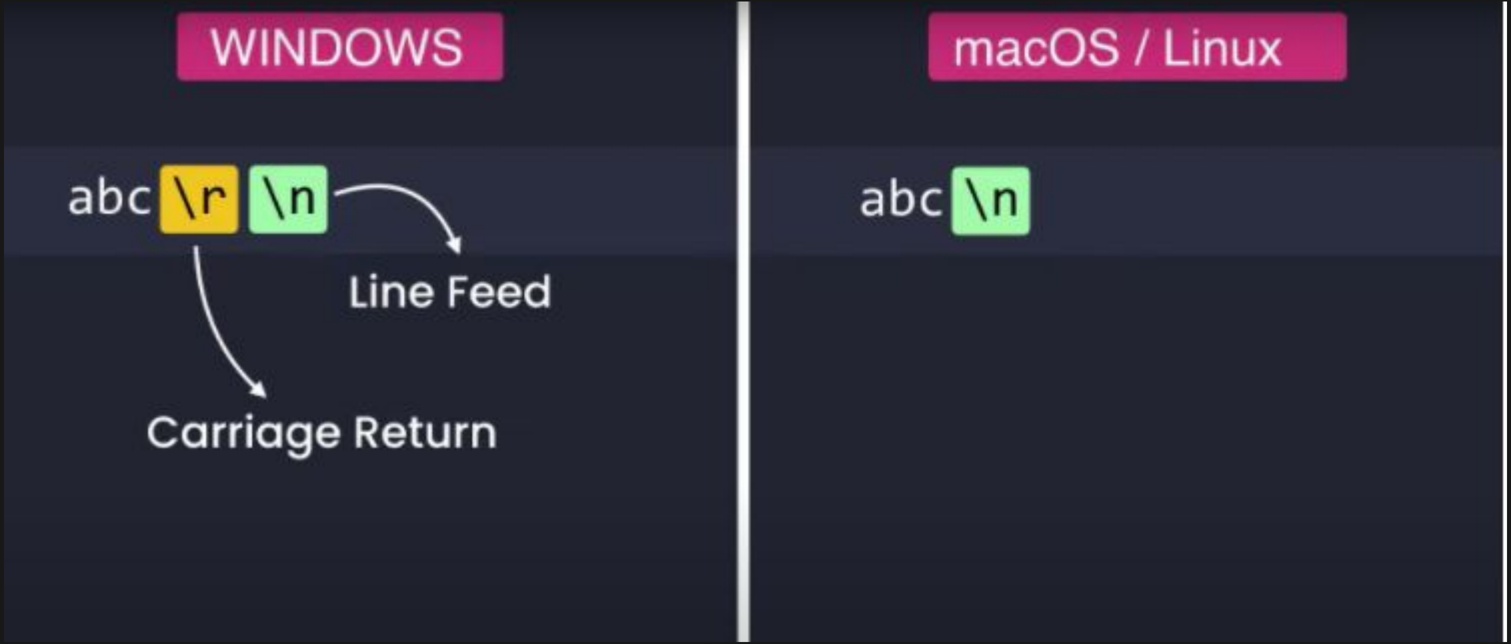
- It's really easy on all the 3 main OS
- https://git-scm.com/book/en/v2/Getting-Started-Installing-Git
- On Windows you should also install GitBash

Git config

```
/ $ git config --global user.name “Riccardo Gulin”
```

```
/ $ git config --global user.email “useyour@email.com”
```

Git config

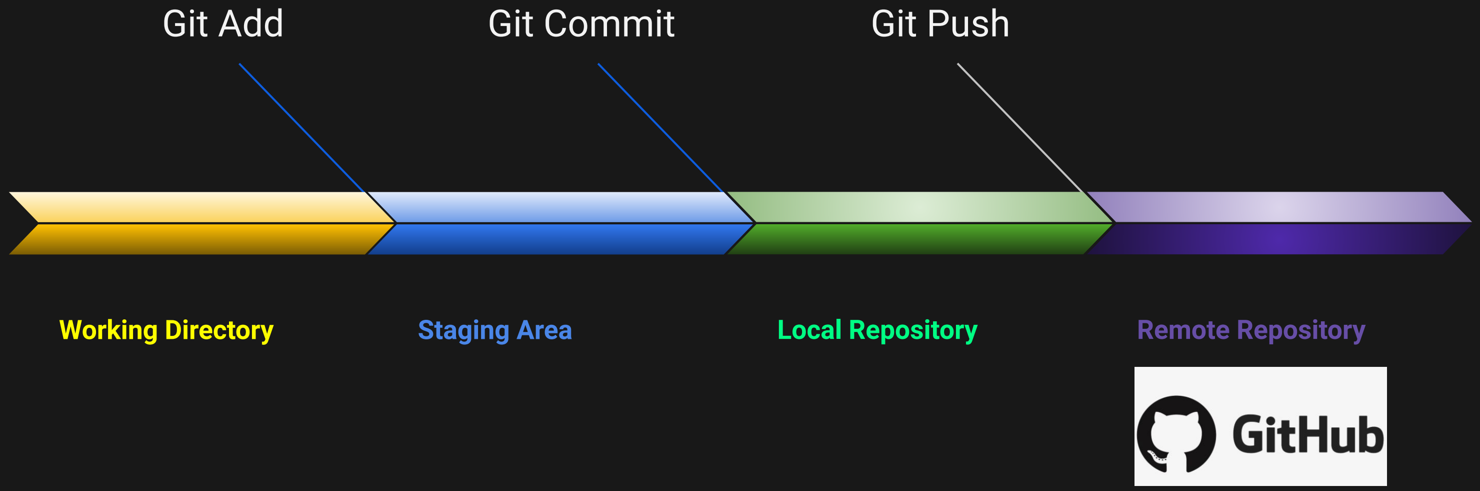


Git config

/ On Windows → `$ git config --global core.autocrlf true`

/ On Linux/MacOS → `$ git config --global core.autocrlf input`

Git flow



Git Add

Git Commit

Git Push

Working Directory

Staging Area

Local Repository

Remote Repository

Staging Area

- Special place in which we bring all the files we are proposing to be saved
- Here we can review changes
- If everything is fine we can proceed
- Or we can just remove one or more files and select just the ones we care

```
$ git add file1.js
```

Local Repository

/ Repository → “Git project”

/ Contains the entire collection of files and folders associated with a project

/ It is organized like a sequence of *snapshots* of your files

/ Every snapshot is a *commit* with an id, message, date, author,…

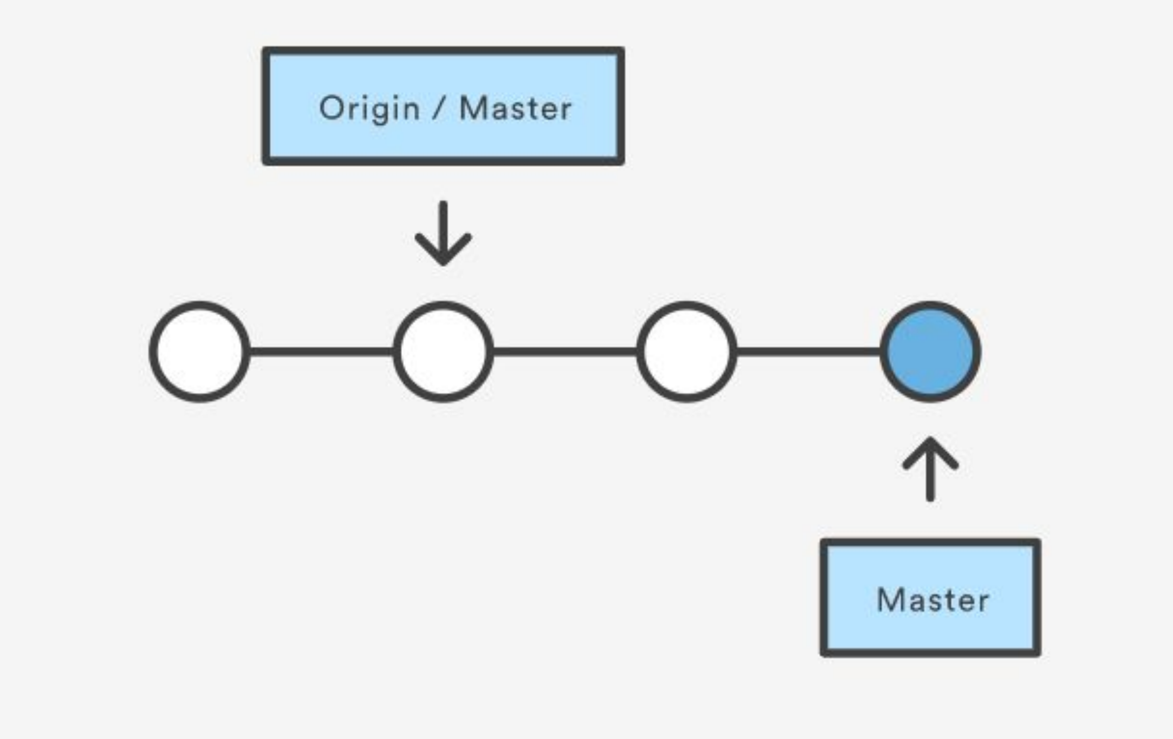
```
$ git commit -m "my first commit"
```

Remote Repository

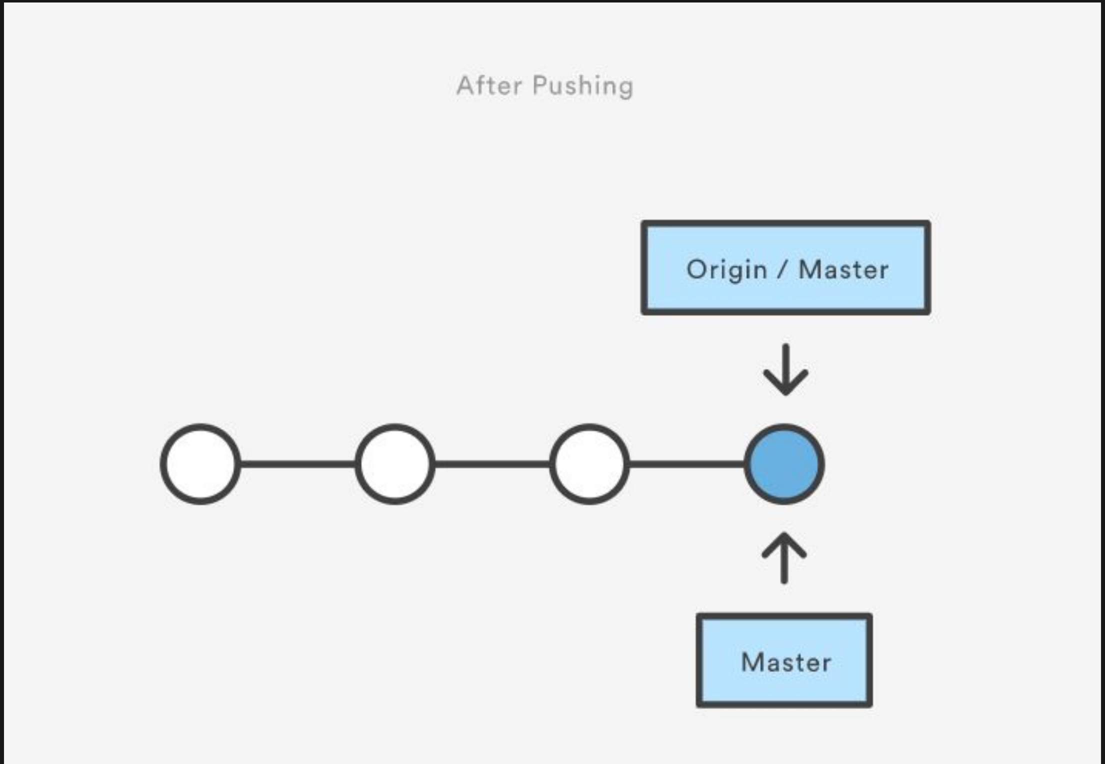
- Remote Repository → same as before but hosted on GitHub, GitLab,….
- It is accessible from all team members
- You can *push* all your local commits there
- This will probably become your production code!

```
$ git push
```

Before pushing



After pushing



$ git init / $ git clone

Let’s go to practice

```
$ git init
```

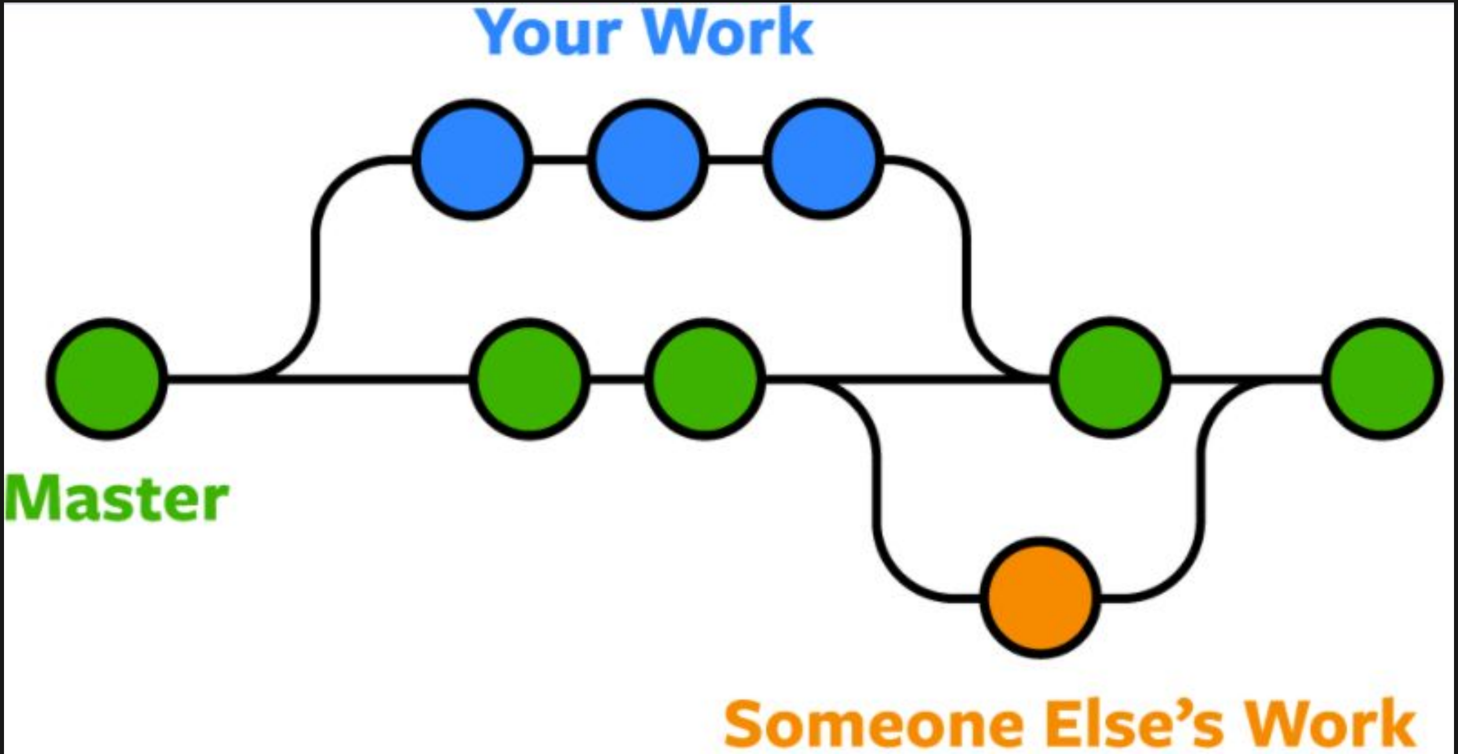
Let’s go to practice

Git fetch & pull

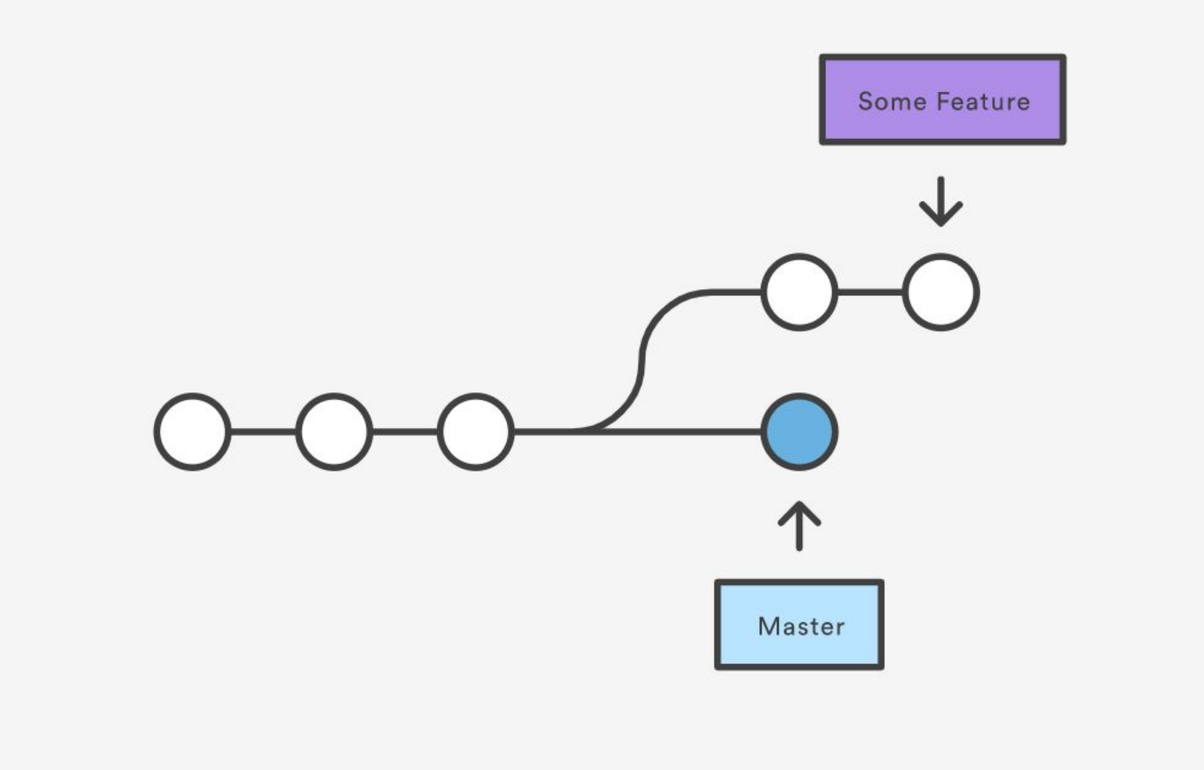
- Fetch downloads newest commits from remote repository without updating local repository
- Pull downloads remote commits and merges them to the local repository

```
$ git fetch
$ git pull
```

Cooperating



Git branch



Git branch

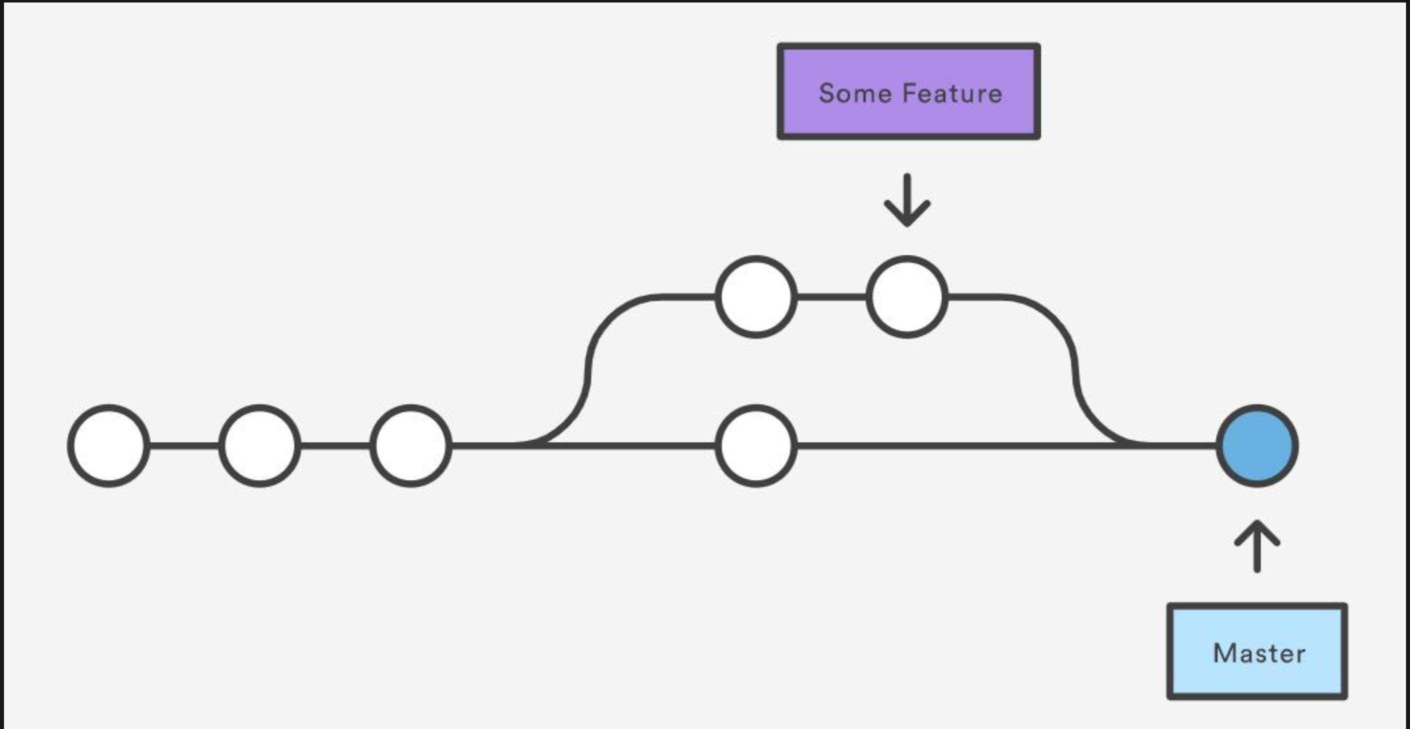
/ A branch is a different path detached from the main one (master branch)

/ Very useful technique to have multiple parallel flows

/ In a branch you can push new features or bugfixes without involving the main part of the project

/ When you are done you can eventually merge those commits in the main branch

Git merge



Git merge

/ We can merge any changes made on a separate branch to the current branch with the merge command

```
$ git merge branchName
```

Git merge

Git merge

Merge Conflicts

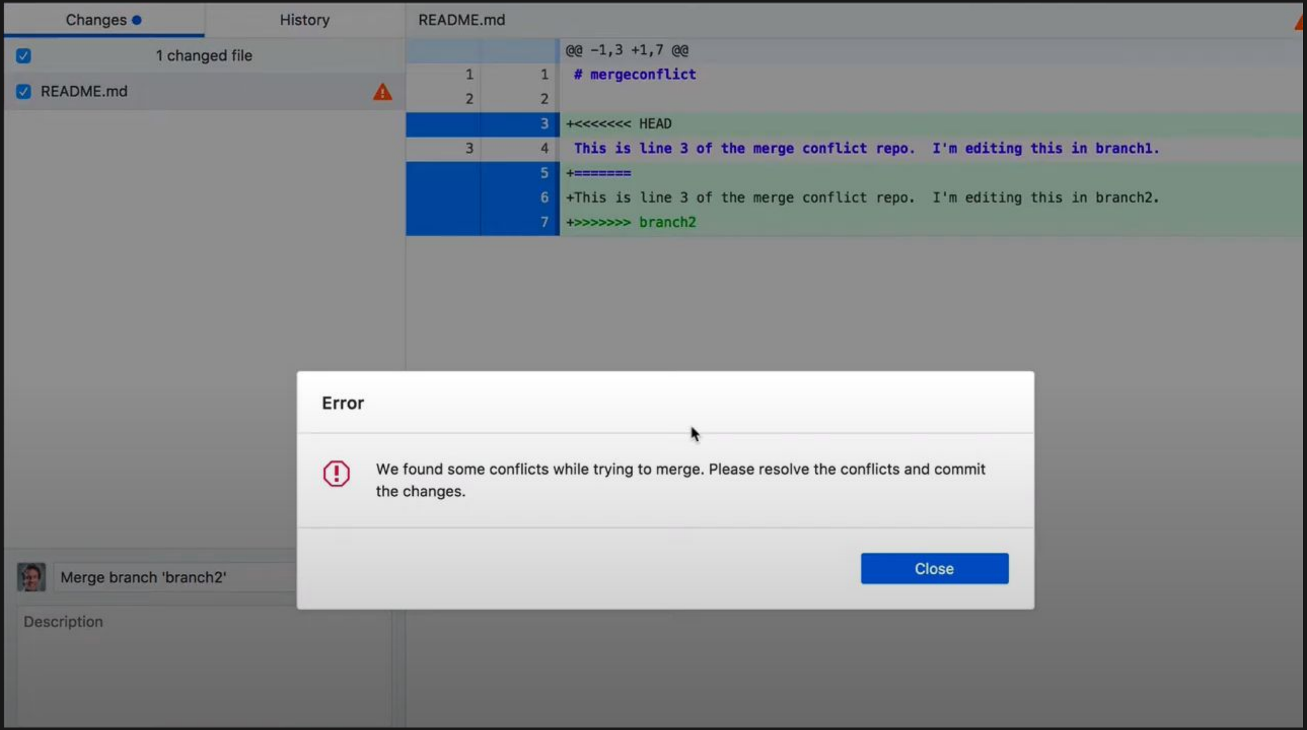
/ Common part of the Git experience

/ Lot of times Git is able to automagically merge and integrate new changes

/ Git magical abilities can’t do nothing when two people worked on the same lines of a file or if one dev deleted a file while another dev was modifying it

/ In these situations the developer trying to merge is the one responsible of solving conflicts manually before being able to merge

Merge Conflicts

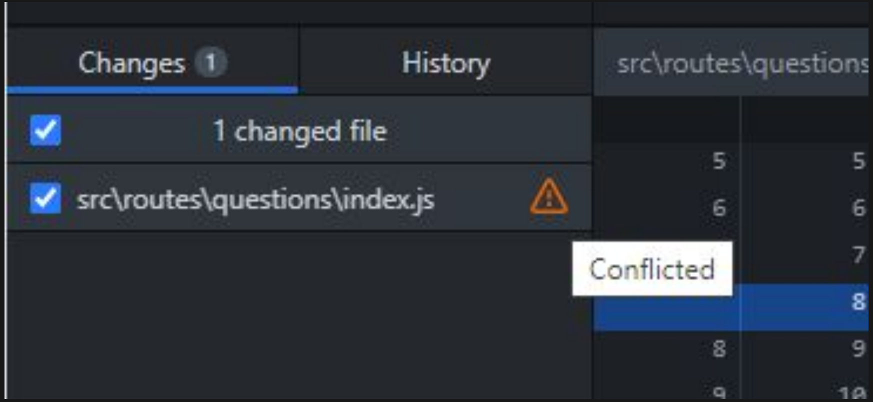


```
@@ -1,3 +1,7 @@
1 1   # mergeconflict
2 2
  3  +<<<<<<< HEAD
3 4   This is line 3 of the merge conflict repo.  I'm editing this in branch1.
  5  +=======
  6  +This is line 3 of the merge conflict repo.  I'm editing this in branch2.
  7  +>>>>>>> branch2
```

Error

We found some conflicts while trying to merge. Please resolve the conflicts and commit the changes.

Merge Conflicts

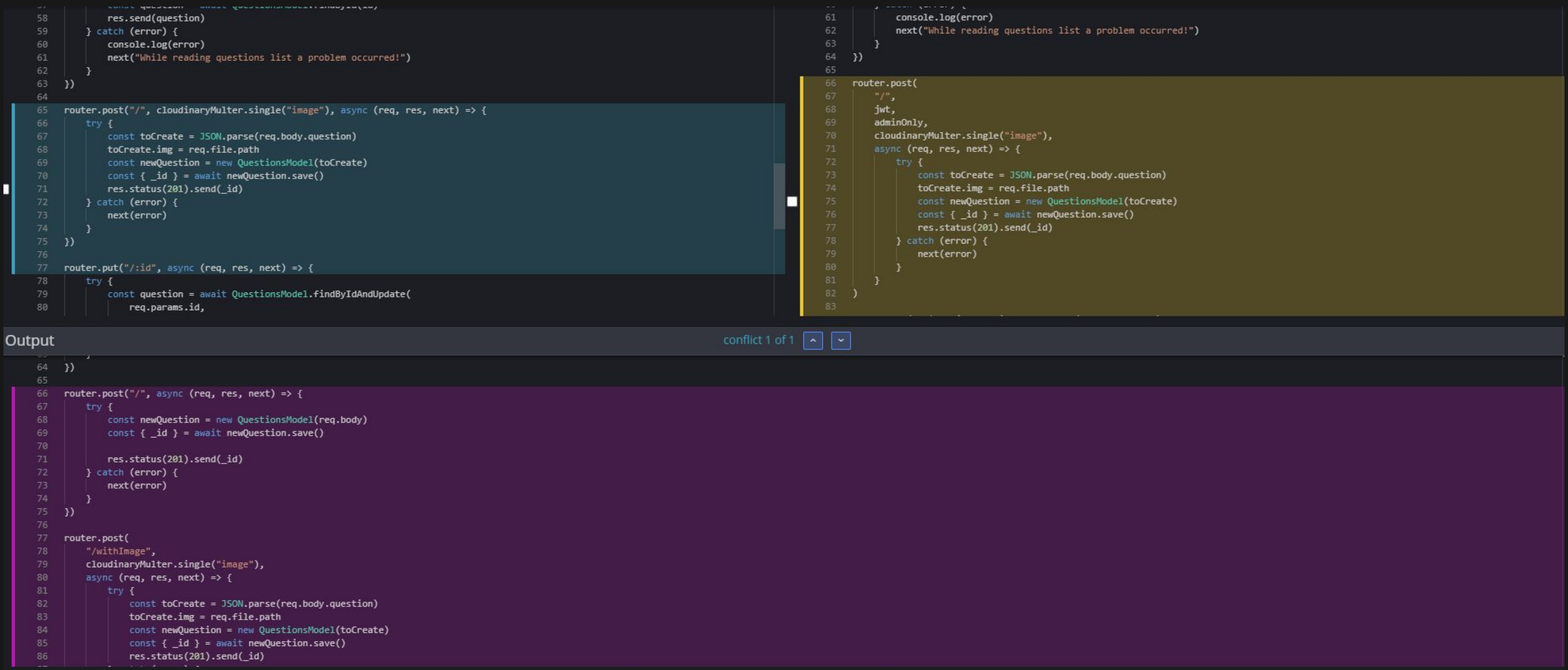


☑ 1 changed file

☑ src\routes\questions\index.js ⚠

Conflicted

How to solve Merge Conflicts



```
        const question = await QuestionsModel.findById(id)
        res.send(question)
    } catch (error) {
        console.log(error)
        next("While reading questions list a problem occurred!")
    }
})

router.post("/", cloudinaryMulter.single("image"), async (req, res, next) => {
    try {
        const toCreate = JSON.parse(req.body.question)
        toCreate.img = req.file.path
        const newQuestion = new QuestionsModel(toCreate)
        const { _id } = await newQuestion.save()
        res.status(201).send(_id)
    } catch (error) {
        next(error)
    }
})

router.put("/:id", async (req, res, next) => {
    try {
        const question = await QuestionsModel.findByIdAndUpdate(
            req.params.id,
```

```
    } catch (error) {
        console.log(error)
        next("While reading questions list a problem occurred!")
    }
})

router.post(
    "/",
    jwt,
    adminOnly,
    cloudinaryMulter.single("image"),
    async (req, res, next) => {
        try {
            const toCreate = JSON.parse(req.body.question)
            toCreate.img = req.file.path
            const newQuestion = new QuestionsModel(toCreate)
            const { _id } = await newQuestion.save()
            res.status(201).send(_id)
        } catch (error) {
            next(error)
        }
    }
)
```

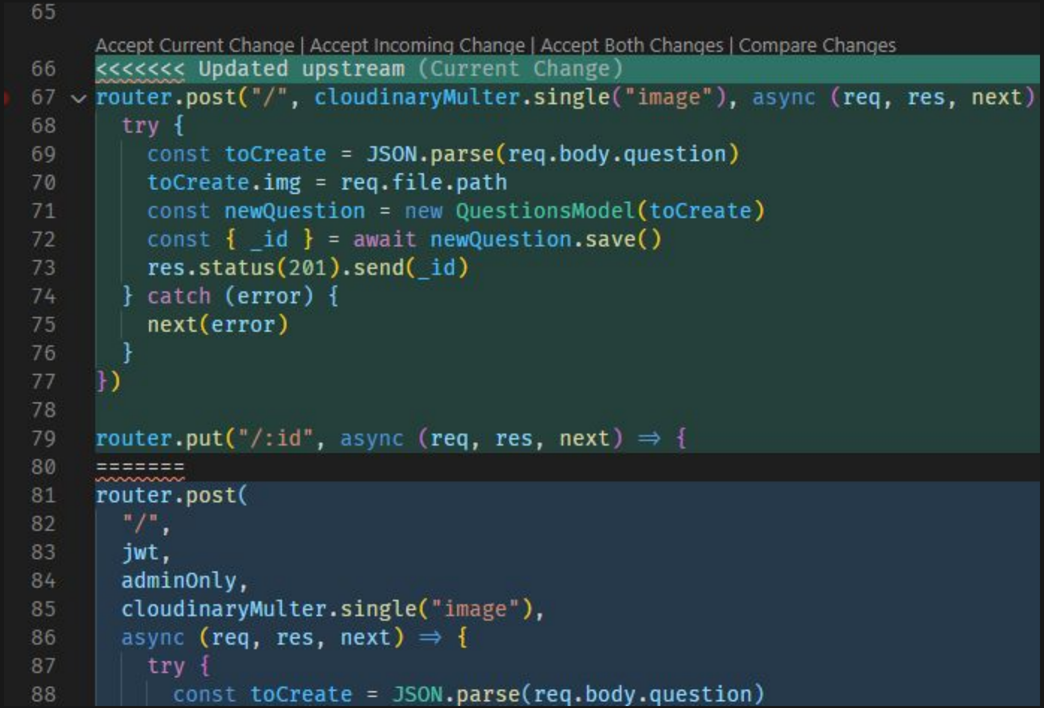
```
})

router.post("/", async (req, res, next) => {
    try {
        const newQuestion = new QuestionsModel(req.body)
        const { _id } = await newQuestion.save()

        res.status(201).send(_id)
    } catch (error) {
        next(error)
    }
})

router.post(
    "/withImage",
    cloudinaryMulter.single("image"),
    async (req, res, next) => {
        try {
            const toCreate = JSON.parse(req.body.question)
            toCreate.img = req.file.path
            const newQuestion = new QuestionsModel(toCreate)
            const { _id } = await newQuestion.save()
            res.status(201).send(_id)
```

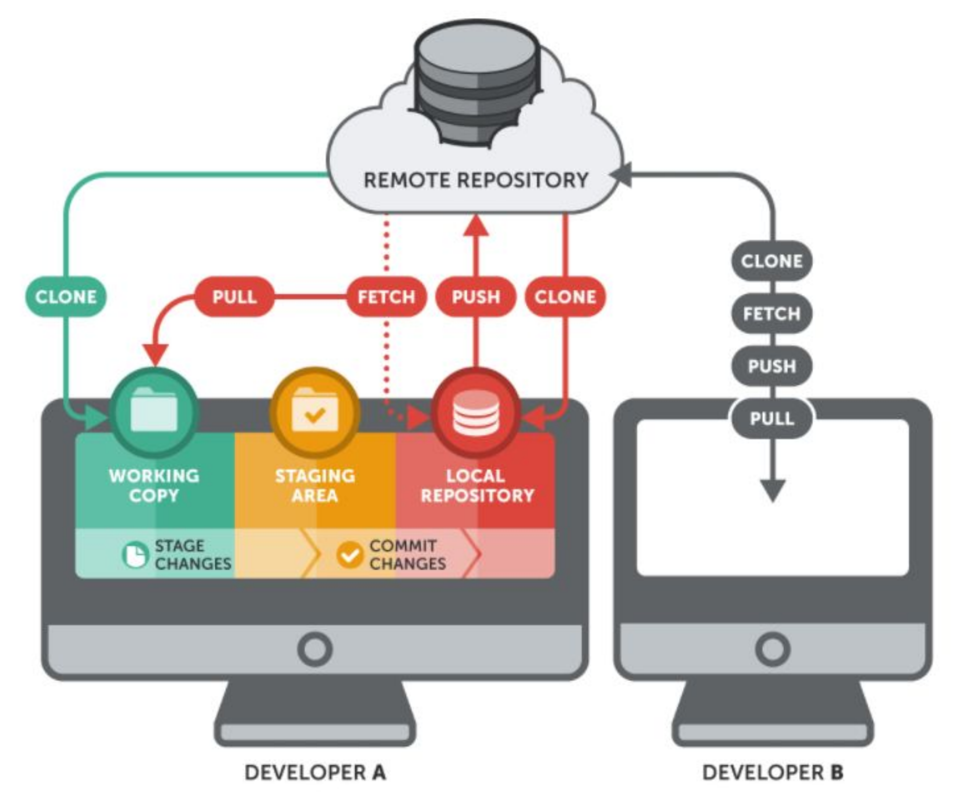
How to solve Merge Conflicts


Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes

```
<<<<<<< Updated upstream (Current Change)
router.post("/", cloudinaryMulter.single("image"), async (req, res, next)
  try {
    const toCreate = JSON.parse(req.body.question)
    toCreate.img = req.file.path
    const newQuestion = new QuestionsModel(toCreate)
    const { _id } = await newQuestion.save()
    res.status(201).send(_id)
  } catch (error) {
    next(error)
  }
})

router.put("/:id", async (req, res, next) => {
=======
router.post(
  "/",
  jwt,
  adminOnly,
  cloudinaryMulter.single("image"),
  async (req, res, next) => {
    try {
      const toCreate = JSON.parse(req.body.question)
```

Summary


REMOTE REPOSITORY
CLONE
PULL
FETCH
PUSH
CLONE
CLONE
FETCH
PUSH
PULL
WORKING COPY
STAGING AREA
LOCAL REPOSITORY
STAGE CHANGES
COMMIT CHANGES
DEVELOPER A
DEVELOPER B

I HATED GIT
NOW I HATE IT LESS

Q&A

Don’t be SHY