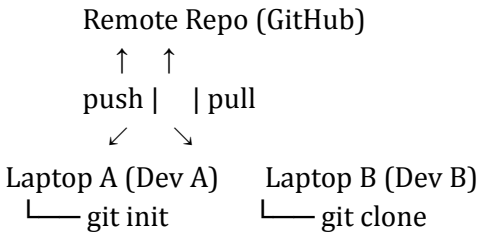
Scenario: Two Developers Working on the Same GitHub Repository

Real-World Context

Dev A and Dev B are collaborating on the same GitHub repository while working from different locations. They are editing the same file (e.g., Myjavafile). Git and GitHub are used to synchronize their changes.

Visual Workflow



Editing the Same File – Sync Strategy

If multiple developers edit the same file sequentially and pull the latest changes before pushing, each contribution will be preserved. If the same lines are modified without pulling updates first, merge conflicts can occur.

Example:
Myjavafile:
1st line by Dev A
2nd line by Dev B
3rd line by Dev C

Step-by-Step Instructions

Dev A (Initial Setup)

- # Setup Git identity
- git config --global user.name "Dev A"
- git config --global user.email "deva@example.com"
- # Create project folder and initialize
- mkdir myproject
- cd myproject
- git init

- echo "1st line by Dev A" > Myjavafile
- git add .
- git commit -m "Dev A - First line"
- git remote add origin https://github.com/user/mytestRepo.git
- git push -u origin main

Dev B (Another Laptop)

- # Setup identity
- git config --global user.name "Dev B"
- git config --global user.email "devb@example.com"
- # Clone the repo
- git clone https://github.com/user/mytestRepo.git
- cd mytestRepo
- # Pull latest changes (optional if just cloned)
- git pull
- # Edit file
- echo "2nd line by Dev B" >> Myjavafile
- # Push changes
- git add Myjavafile
- git commit -m "Dev B - Added 2nd line"
- git push

Daily Git Workflow

- git pull # Always pull first to get latest changes
- # Make your changes
- git add .
- git commit -m "your message"
- git push # Push to GitHub

Step-by-Step Workflow for GitHub Setup

1. Sign Up and Setup GitHub

- Go to github.com → Sign up with email
- Create a username and password
- Verify email → Done

2. Create a Repository

- Click "New" → Create a repo (e.g., mytestRepo30Dec)
- Choose public/private
- Optionally initialize with README

3. Create a Token (for HTTPS push)

- GitHub Profile → Settings → Developer settings
- Personal access tokens → Generate new token
- Select scopes (e.g., repo), set No expiry → Generate and copy token

4. Install Git on Your Laptop

- Search "Git for Windows" → Download and install
- Use Git Bash for commands

5. Initialize Local Repo (Dev A)

- mkdir myrepo
- cd myrepo
- git init
- echo "# myTestRepo30Dec" >> README.md
- git add README.md
- git commit -m "first commit"
- git branch -M main
- git remote add origin https://github.com/<username>/mytestRepo30Dec.git
- git push -u origin main

6. Configure Your Identity

- git config --global user.name "John Doe"
- git config --global user.email "john@example.com"

7. Clone Repo (Dev B)

- git clone https://github.com/<username>/mytestRepo30Dec.git
- cd mytestRepo30Dec

8. Typical Git Workflow

- git pull # Get latest changes
- # make changes to file (e.g., Myjavafile)
- git add .
- git commit -m "added new feature"
- git push # Push to GitHub
- To receive updates made by others: git pull

Summary

- git init: Used by Dev A to initialize the local repository.
- git clone: Used by Dev B to copy the remote repository.
- git push: Uploads local changes to the remote GitHub repository.
- git pull: Fetches and merges changes from the remote repository.

Purpose of Each Git Command

Command:

```
git config --global user.name "Your Name"
git config --global user.email "your@example.com"
```

Purpose:

Sets your Git identity (name and email) globally. This info will be associated with your commits.

Command:

```
mkdir myproject
cd myproject
git init
```

Purpose:

Creates a new folder, navigates into it, and initializes an empty Git repository.

Command:

```
echo "1st line by Dev A" > Myjavafile
git add .
git commit -m "Dev A - First line"
```

Purpose:

Creates a file, stages it, and commits the change with a message.

Command:

```
git remote add origin https://github.com/user/mytestRepo.git
git push -u origin main
```

Purpose:

Links the local repo to a remote GitHub repository and pushes the local main branch for the first time.

Command:

```
git clone https://github.com/user/mytestRepo.git
cd mytestRepo
```

Purpose:

Clones the remote repository to your local system and navigates into it.

Command:

```
git pull
```

Purpose:

Fetches and merges the latest changes from the remote repository.

Command:

```
echo "2nd line by Dev B" >> Myjavafile
git add Myjavafile
git commit -m "Dev B - Added 2nd line"
git push
```

Purpose:

Appends a line to the file, stages, commits, and pushes the change to GitHub.

Command:

```
git pull
# make changes
git add .
git commit -m "update message"
git push
```

Purpose:

Standard daily workflow to pull, make changes, stage, commit, and push updates.