Tooling for Java EE applications

PA165

Jiří Uhlíř, Martin Kotala

26. 9. 2017

Contents

Contents

Version control

- Motivation
- History
  - One file at a time
  - Centralized (CVS, Subversion)
  - Distributed (Git, Mercurial)

Git history

- Created in 2005 by Linus Torvalds
  - described by himself as "stupid content tracker"
  - Originally created for linux kernel development
- Inspired by BitKeeper, aiming to be performant and free
- CVS taken as example of what *not to do*
- git - no exact meaning
  - random three-letter combination that is pronounceable, and not actually used by any common UNIX command. The fact that it is a mispronunciation of "get"may or may not be relevant.
  - "global information tracker": you're in a good mood, and it actually works for you. Angels sing, and a light suddenly fills the room.
  - "g*dd*mn idiotic truckload of sh*t": when it breaks
  - `https://github.com/git/git/blob/master/README.md`

Git characteristics

- Strong support for non-linear development
  - Rapid branching and merging
  - Tools for visualisation and navigation in development history
  - Lightweight branches
- Distributed development
  - Each developer has full history
    - Prevents data loss
    - Subteams can share reposities without access to central repository
  - No need to have access to central repository all the time
  - Changes are commited locally and then pushed to central repository

Git characteristics

- Variety of protocols supported
  - HTTP/HTTPS
  - FTP
  - SSH
- Efficient handling of large projects
  - Fast (when applying patches)
  - Scalable
  - Fetching version history from locally stored repository is faster then from remote
- Allows various workflows
  - Centralized (enterprise companies)
  - Hierarchical (Linux kernel)
  - Distributed (open source projects, pull requests)

Git Basics - commands

git init

- Initializes empty local repository

git status

- Shows current file differences between HEAD commit and current working copy

git add <filename>

- Adds a file/directory into commit checklist
- -A (all files not versioned, or not ignored), -u (only updated files already under version control)

git commit -m <message>

- Records working copy changes into repository

.gitignore

- File for specifying files not to be tracked under version control (binary files, log files, temporary build files, etc.)

Git Basics - commands

git log

- Shows latest commits for local repository
- –oneline (condensed view), –graph (includes branches)

git diff

- Shows code difference between HEAD commit and current working copy

git checkout / git reset

- Removes local uncommited changes

git reset –soft HEAD 1 / –hard <commithash>

- Reverts working copy to given commit (soft keeps changes as *to be commited*, hard removes them completely)

Git Basics - commands

git tag

- Annotates current version of local repository with tag (such as version)

git clone

- Clones remote repository into local repository and fetches latest changes

git push

- Pushes local commited changes into remote repository
- –tags Pushes tags into remote repository

Git Basics - Demo