Git: Comprehensive Local Repository Maintenance

Course Summary

Description

This course is for those who work with Git and need to learn techniques to manage their local repository. This course looks at some familiar commands in greater detail and adds in others which may be new to students. Promoting the Git workflow of leveraging branches, the course begins by re-examining branches with a focus on gaining experience with the various merge strategies: what they are and how to choose the right one for the job. Central to Git are commits, but finding the commit object needed can be difficult. In this training students will learn about some useful commands to navigate the local repository, and practice working directly with git objects to gain control over the stored resources. With mastering the skill to find and identify Git objects, a number of key commands will be covered to empower the students with the ability to organize and manage commits to ensure their local repository is optimal and ready to share. Problems will and do arise, so it is important to practice working with Git tools to troubleshoot and fix issues. The training concludes with strategies for managing large and growing projects, as well as options for automating predictable and repetitive tasks.

Objectives

At the end of this course, students will be able to:

- Describe merge strategies and choose the correct option for the task at hand
- Navigate and work directly with Git objects
- Manage commits through amending, updating, and deleting commits
- Keeping the local repository clean and ready to share appropriately with team members
- Work with troubleshooting tools to fix problems
- Setup support for code dependencies
- Preserve project history through archiving
- Leverage hooks to ensure compliance with project policies
- Script options to simplify repetitive work

Topics

- Working with Branches
- Understanding Commits
- Troubleshooting in Git
- Controlling Commits
- Scripting for Git
- Enterprise Git Repositories

Audience

This course is designed for developers regularly using Git who want to deepen their knowledge and ability to maintain their local repository in an optimal state.

Prerequisites

Foundational Git experience and knowledge are required including:

- core concepts of how git works
- creating and adding files to the local repository
- creating and navigating branches

Familiar with how to run git commands is assumed. Comfortable working in bash (unix shell) is highly beneficial

Duration

Three days

Course Outline

I. Working with Branches

A. Navigating Branches
1. Moving Between Branches
2. Switch vs Checkout
3. Git Switch Command
4. Activity: Switch Options
5. Activity: Switch Options - Results
6. Untracked Files Shared
7. Git Stash Command Basics
8. Working with Git Stash
9. Useful Commands
10. Activity: Work with Git Stash
11. Activity: Work with Git Stash - Results

B. Updating Branches
1. What Is Merging?
2. Types of Merges
3. Resolve Merge Strategy
4. Octopus Merge Strategy
5. Octopus Successes
6. Activity: Octopus Merge
7. Activity: Octopus Merge - Results
8. Octopus Merge - Visual
9. Ours Merge Strategy
10. Activity: Ours Merge
11. Activity: Ours Merge - Results
12. Ours Merge - Visual
13. Recursive Merge Strategy
14. Activity: Recursive Merge With Options
15. Activity: Recursive Merge With Options - Results

C. Working with Branches - Summary

II. Understanding Commits

A. Finding Commits
1. Finding Commits
2. Git Rev List Command
3. Rev List Options
4. Activity: Using Rev List
5. Activity: Using Rev List - Result
6. Git Reflog Command
7. Reflog Options
8. Activity: Using Reflog
9. Activity: Using Reflog - Results
10. Finding Commits For Files
11. Activity: Using Log
12. Activity: Using Log – Result

B. Reading Commits
1. Git LS Files Command
2. Git LS Tree Command
3. Git Cat File Command
4. Activity: View Git Objects
5. Activity: View Git Objects - Results

C. Immutable Commits
1. Versioning the Repository
2. Tag types
3. Git Tag Command
4. Working with Tags
5. Activity: Version the Repository
6. Activity: Version the Repository - Results

D. Understanding Commits - Summary

III. Troubleshooting in Git

A. Debugging
1. Planning for Problems
2. Debugging
3. Useful Debugging Commands
4. Useful Debugging Commands
5. Activity: Troubleshoot a Problem
6. Activity: Troubleshoot a Problem
7. Activity: Troubleshoot a Problem - Results

B. Cherry-Picking
1. Git Cherry-Picking Scenarios
2. Git Cherry-Pick Command
3. Activity: Cherry-pick to Move a File
4. Activity: Cherry-pick to Move File - Results
5. Git Cherry-Pick - Visual

C. Troubleshooting in Git - Summary

IV. Controlling Commits

A. Altering Commits
1. Amending Commits
2. Git Amend Command
3. Activity: Amend a Commit
4. Activity: Amend a Commit - Results
5. Rebase Interactively
6. Activity: Clean Up Older Commits
7. Activity: Clean Up Older Commits - Results

Course Outline (cont'd)

B. Undoing Commits
   1. Undoing Commits - Options
   2. Git Revert Command
   3. Activity: Fix Reverting
   4. Activity: Fix Reverting - Results
   5. Git Revert - Visual
   6. Git Reset Command
   7. Activity: Undo a Reset
   8. Activity: Reset a Commit - Results
   9. Git Reset - Visual
   10. Recover Files
   11. Activity: Manage Files

C. Controlling Commits - Summary

V. Scripting in Git

A. Hooks
   1. What Are Hooks?
   2. Built-in Hooks - Client (Local) Side
   3. Built-in Hooks - Server Side
   4. Using Hooks
   5. Standardize Hooks for Repository
   6. Activity: Use a Hook
   7. Activity: Use a Hook - Results
   8. Simplify Local Work

B. Git Aliases
   1. Using Git Aliases
   2. Create an Aliases
   3. Activity: Using Aliases
   4. Activity: Using Aliases - Results

C. Scripting in Git – Summary

VI. Enterprise Git Projects

A. Project Dependencies
   1. Application Dependencies
   2. Git Submodules
   3. Working with Git Submodules
   4. Activity: Add a Submodule
   5. Activity: Add a Submodule - Result
   6. Git Subtrees
   7. Working with Git Subtrees
   8. Activity: Add a Subtree
   9. Git Submodules vs Git Subtrees
   10. Git Submodules vs Git Subtrees - Visual

B. Archiving
   1. Archiving Repositories
   2. Git Archive Command
   3. Activity: Archive Project Code
   4. Activity: Archive Project Code - Result

C. Very Large Projects
   1. Types of Large Projects
   2. Cloning for Large Projects
   3. 'Weight Loss' for Large Projects
   4. Large Binary Files

D. Enterprise Git Projects - Summary