Solve real problems and enhance your skills with browser based hands on labs without any downloads or configuration

Scenario 1 - Committing Files

Learn how to initalise a repository and start committing files.

Repeat Scenario

Scenario 2 - Committing Changes

Learn how to compare and commit changes.

Repeat Scenario

Scenario 3 - Working Remotely

Learn how to share your changes with others and access other people's changes.

Repeat Scenario

Scenario 4 - Undoing Changes

Learn how to undo changes when required.

Repeat Scenario

Scenario 5 - Fixing Merge Conflicts

Learn how to fix merge conflicts then they occur.

Repeat Scenario

Scenario 6 - Experiments Using Branches

Learn how to create branches of master for experimenting and prototyping ideas.

Repeat Scenario

Scenario 7 - Finding Bugs

Learn how to find commits related to bugs and issues with code.

Repeat Scenario

Scenario 8 - Being Picky With Git

Learn how to pick certain commits and changes from other repositories.

Repeat Scenario

Scenario 9 - Re-writing History

Learn how to re-write history when required.

Repeat Scenario

Playground

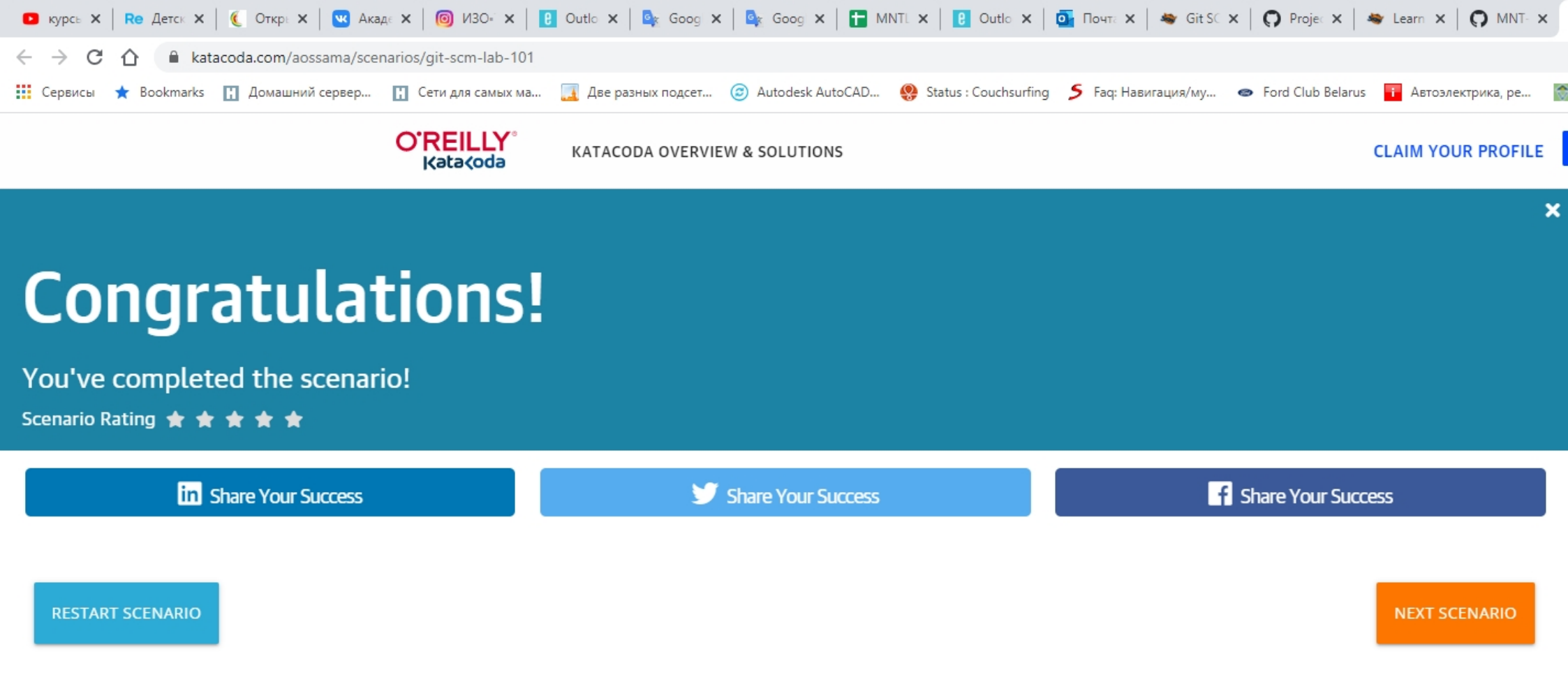
Use Git in a safe playground environment

Explore Playground

Your Content Here

Add your own content to Katacoda and share your experience or product with the community

Create Content


katacoda.com/aossama/scenarios/git-scm-lab-101
O'REILLY
Katacoda
KATACODA OVERVIEW & SOLUTIONS
CLAIM YOUR PROFILE
Congratulations!
You've completed the scenario!
Scenario Rating
Share Your Success
Share Your Success
Share Your Success
RESTART SCENARIO
NEXT SCENARIO

Congratulations!

You've completed the scenario!

Scenario Rating ★ ★ ★ ★ ★

The most important takeaways from this lab are:

- `git clone` is used to create a copy of a target repo
- `git remote` is used to create, view, and delete connections to other repositories
- `git push` is used to propagate changes on the local repository to remote repository
- `git fetch` is used to download objects and refs from another repository
- `git pull` is used to fetch from and integrate with another repository or a local branch

Share Your Success | Share Your Success | Share Your Success

RESTART SCENARIO

NEXT SCENARIO

Congratulations!

You've completed the scenario!

Scenario Rating ★ ★ ★ ★ ★

The most important takeaways from this lab are:

- `git checkout` can be used to create branches, switch branches, and checkout remote branches
- `git branch` commands primary functions are to create, list, rename and delete branches
- `git tag` is used to create semantic version number identifier tags that correspond to software release cycles
- `git merge` is used to combine multiple sequences of commits into one unified history
- `git rebase`
- `git reset`

Share Your Success | Share Your Success | Share Your Success

RESTART SCENARIO

NEXT SCENARIO

Congratulations!

You've completed the scenario!

Scenario Rating ★ ★ ★ ★ ★

Now that you have an understanding of the projects you will use throughout this course, let's get started!

Share Your Success | Share Your Success | Share Your Success

RESTART SCENARIO

NEXT SCENARIO