First contribution

Contribution Workflow

The codebase and documentation are maintained using the same "contributor workflow" where everyone without exception contributes changes proposals using "pull-requests".

This facilitates social contribution, easy testing, and peer review.

To contribute changes, use the following workflow:

1. **Fork** the repository.
2. **Clone** your fork to your computer.
3. **Create a topic branch** and name it appropriately. Starting the branch name with the issue number is a good practice and a reminder to fix only one issue in a Pull-Request (PR).
4. **Make your changes** adhering to the coding conventions described below. In general a commit serves a single purpose and diffs should be easily comprehensible. For this reason do not mix any formatting fixes or code moves with actual code changes.
5. **Commit your changes** see How to Write a Git Commit Message article by Chris Beams.
   a. Make sure to add DCO message to each commit, more information here
6. **Test your changes** locally before pushing to ensure that what you are proposing is not breaking another part of the software. Running the **gradlew clean check test** command locally will help you to be confident that your changes will pass CI tests once pushed as a Pull Request.
7. **Push your changes** to your remote fork (usually labeled as origin).
8. **Create a pull-request** (PR) on the Besu repository. If the PR addresses an existing issue, include the issue number in the PR title in square brackets (for example, [PAN-2374]).
9. **Add labels** to identify the type of your PR. For example, if your PR is not ready to validate, add the "work-in-progress" label. If it fixes a bug, add the "bug" label.
10. If the PR address an existing issue, comment in the issue with the PR number.
11. **Ensure your changes are reviewed**. Select the reviewers you would like to review your PR. If you don't know who to choose, simply select the reviewers proposed by GitHub or leave blank. If you are not a maintainer, you won't be able to choose reviewers, this will be done by one of the maintainers.
12. **Make any required changes** on your contribution from the reviewers feedback. Make the changes, commit to your branch, and push to your remote fork.
13. When your PR is validated, all tests passed and your branch has no conflicts with the target branch, you can "squash and merge" your PR and you're done.

You contributed to Besu! Thanks!