Contributing to SolrNet

=======================

For those without any GitHub knowledge: [Paul Bouwer has written an excellent guide for contributors](http://blog.paulbouwer.com/2010/12/27/git-github-and-an-open-source-net-project-introduction/) starting from scratch.

You can contribute to SolrNet with issues and PRs. Simply filing issues for problems you encounter is a great way to contribute and of course contributing implementations is greatly appreciated.

## Workflow

We use and recommend the standard GitHub [Fork and Pull](https://gist.github.com/Chaser324/ce0505fbed06b947d962]) workflow:

1. Create an issue for your work.

- You can skip this step for trivial changes.

- Reuse an existing issue on the topic, if there is one.

- State that you are going to take on implementing it, if that's the case. You can request that the issue be assigned to you. Note: The issue filer and the implementer don't have to be the same person.

2. Create a personal fork of the repository on GitHub (if you don't already have one).

3. Create a branch off of master

- Name the branch so that it clearly communicates your intentions, such as issue-123 or githubhandle-issue.

- Branches are useful since they isolate your changes from incoming changes from upstream. They also enable you to create multiple PRs from the same fork.

``` bash

git checkout -b issue-123

```

4. Make and commit your changes.

5. Add new tests corresponding to your change, if applicable.

6. Build the repository with your changes.

- Make sure that the builds are clean.

- Make sure that the tests are all passing, including your new tests.

7. If you haven't already, push your changes to your fork on GitHub in order to start the process of opening a Pull Request.

``` bash

git push origin my-branch

```

8. Create a pull request (PR) against the upstream repository's \*\*master\*\* branch.

9. Discuss and update: you might get feedback or requests for changes to your pull request. To make changes to an existing Pull Request, make the changes to your local branch, add a new commit with those changes, and push those to your fork. GitHub will automatically update the Pull Request.

10. If all is good, the SolrNet team will approve and merge your Pull Request. Thank you for helping us all out!

Note: It is OK for your PR to include a large number of commits. Once your change is accepted, you will be asked to squash your commits into one or some appropriately small number of commits before your PR is merged.

Note: It is OK to create your PR as "[WIP]" on the upstream repo before the implementation is done. This can be useful if you'd like to start the feedback process concurrent with your implementation. State that this is the case in the initial PR comment.