

# Push.Default Strategies

I was accidentally updating other branches when pushing my feature branch, and didn't realise why.

Ever pushed to the wrong branch by mistake? what happens when you `git push`? Is your expectation that your local branch is pushed to the remote? This might not be the case:

What happens depends on your `push.default` configuration setting! List push default strategy that is in use `$ git config push.default` for your project and `$ git config --global push.default` for the fallback. To set the value, append it to the command.

## Push.Default Strategies

There are 5 options, my understanding is that:

1. `simple` pushes the current branch to the **remote branch with the same name, but doesn't create the branch** if it doesn't already exist. This is the default if you've recently updated git.
2. `current` works like simple but **creates the remote branch**. Handy unless you're accidentally pushing to the wrong remote.
3. `upstream` works like current, but pushes to the **tracking branch which might not have the same name**. To illustrate this, you can set the tracking branch to a different destination, for example: `git branch --set-upstream-to=origin/my-experiment CI-3911-husky`

There are two more strategies to be aware of:

4. `nothing` **does not push** anything. I don't know when you'd want to use this.
5. `matching` pushes **ALL branches that exist remotely!**  
Realising that you're pushing *dev/stage/any other branch* with changes when pushing your feature branch, and having to reset them, is not a fun day at the office. This gets more complex when using multiple remotes! I would avoid this strategy.

## Tracking branch vs remote branch

The distinction is that all remote branches are potentially tracking branches, but the tracking branch for your local branch does not have to be on a remote.

You can track local master for example for your feature work.

Note that you can find out your tracking branch through `git status`, in general it should contain the remote to avoid unintended consequences:

```
$ git status
On branch CI-3911-husky
Your branch is up to date with 'origin/CI-3911-husky'.
```

## Tips

Finally a tip: please specify a remote after `git push` in your `npm / composer` scripts, so that if the remote is set incorrectly for reasons your team is pushing to the right place.

Finally, Finally 111 latest: if you're not sure about any git command, append `--dry-run` to test it non-destructively. And if you get stuck, I found the [Flight Rules](#) helpful.

## Corrections

Feel free to comment with any corrections or knowledge, everyone is learning :D

Git going!

---

Revision #1

Created 20 August 2021 13:05:07 by Sander

Updated 20 August 2021 13:05:31 by Sander