

PULL REQUEST STRATEGY

1. Switching the branch where we need to fix bugs or create functions

Git Checkout - To start working in a different branch, use *git checkout* to switch branches.

```
(14:36:17)[vanshikis-MacBook-Pro] ~/code/vfde-wave5 [feature/VFDED2W60S-10-change-label-for-create-contracts-button-and-warnings]
$ git checkout develop-wave5
Switched to branch 'develop-wave5'
Your branch is up to date with 'origin/develop-wave5'.
(15:00:38)[vanshikis-MacBook-Pro] ~/code/vfde-wave5 [develop-wave5]
```

Use: - Here switching to develop-wave5

2. Checking the status of the Repository

Git Status- This command returns the current state of the repository. *Git status* will return the current working branch. If a file is in the staging area, but not committed, it shows with *git status*. Or, if there are no changes it'll return *nothing to commit, working directory clean*.

```
(15:00:38)[vanshikis-MacBook-Pro] ~/code/vfde-wave5 [develop-wave5]
$ git status
On branch develop-wave5
Your branch is up to date with 'origin/develop-wave5'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   app/code/local/Vfde/Helper/Panel.php

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    app/code/community/Omnibus/Proxy/controllers/
    shell/catalogExport_2020-08-28_ose_st3.tar.gz

It took 4.00 seconds to enumerate untracked files. 'status -uno'
may speed it up, but you have to be careful not to forget to add
new files yourself (see 'git help status').
no changes added to commit (use "git add" and/or "git commit -a")
(15:10:50)[vanshikis-MacBook-Pro] ~/code/vfde-wave5 [develop-wave5]
```

Use: - Here we can see the file Panel.php is highlighted in red as it is not committed.

3. Checking the difference

Git Diff- It is used in git to track the difference between the changes made on a file. Diff command takes two inputs and reflects the differences between them. It is not necessary that these inputs are files only.

```
(15:10:50)[vanshikis-MacBook-Pro] ~/code/vfde-wave5 [develop-wave5]
$ git diff
diff --git a/app/code/local/Vfde/Helper/Panel.php b/app/code/local/Vfde/Helper/Panel.php
index 50a3fab8e6..06e6a3b779 100644
--- a/app/code/local/Vfde/Helper/Panel.php
+++ b/app/code/local/Vfde/Helper/Panel.php
@@ -50,6 +50,7 @@ class Vfde_Customer_Helper_Panel extends Omnium_Customer_Helper_Panel
    list($componentsAdditionalData, $componentType, $endDate, $addonId, $startDate, $serviceCodeAvailable) = $params;

    if ($endDate) {
+       echo "Test";
        $terminationTo = $this->getFnComponentTerminationTo($componentsAdditionalData,
            $componentType, $endDate);
        $terminationDate = $this->getFnAddonTerminationDate($componentsAdditionalData,
(15:10:59)[vanshikis-MacBook-Pro] ~/code/vfde-wave5
```

4. Create bugfix branch - bugfix/ticket_number-Section-Wave-5

```
(15:10:59)[vanshikis-MacBook-Pro] ~/code/vfde-wave5
$ git checkout -b bugfix/VFDED2W51S-1656-HW-payment-section-wave5
```

Use: - Created branch for given ticket number.

5. Adding the file to the branch and checking the status for file that we have added.

Git Add- Adds files in the to the staging area for Git. Before a file is available to commit to a repository, the file needs to be added to the Git index (staging area). There are a few different ways to use git add, by adding entire directories, specific files, or all unstaged files.

```
$ git add app/code/local/Vfde/Helper/Panel.php
(15:13:29)[vanshikis-MacBook-Pro] ~/code/vfde-wave5 [bugfix/VFDED2W51S-1656-HW-payment-section-wave5]
$ git status
On branch bugfix/VFDED2W51S-1656-HW-payment-section-wave5
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   app/code/local/Vfde/Helper/Panel.php

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    app/code/community/Omnium/Proxy/controllers/
    shell/catalogExport_2020-08-28_ose_st3.tar.gz

It took 3.92 seconds to enumerate untracked files. 'status -uno'
may speed it up, but you have to be careful not to forget to add
new files yourself (see 'git help status').
(15:13:37)[vanshikis-MacBook-Pro] ~/code/vfde-wave5 [bugfix/VFDED2W51S-1656-HW-payment-section-wave5]
```

Use: - Added the Panel.php file

6. Committing the file to the branch

Git commit- Record the changes made to the files to a local repository. For easy reference, each commit has a unique ID.

It's best practice to include a message with each commit explaining the changes made in a commit. Adding a commit message helps to find a particular change or understanding the changes.

```
(15:13:37)[vanshikis-MacBook-Pro] ~/code/vfde-wave5 [bugfix/VFDED2W51S-1656-HW-payment-section-wave5]
$ git commit -m "VFDED2W51S-1656 fixed for displaying HW payment section for mobile"
[bugfix/VFDED2W51S-1656-HW-payment-section-wave5 485ad3a368] VFDED2W51S-1656 fixed for displaying HW payment section for mobile
1 file changed, 1 insertion(+)
(15:15:06)[vanshikis-MacBook-Pro] ~/code/vfde-wave5 [bugfix/VFDED2W51S-1656-HW-payment-section-wave5]
```

Use: - Committed with the message using tag -m

7. Pushing the file to the branch

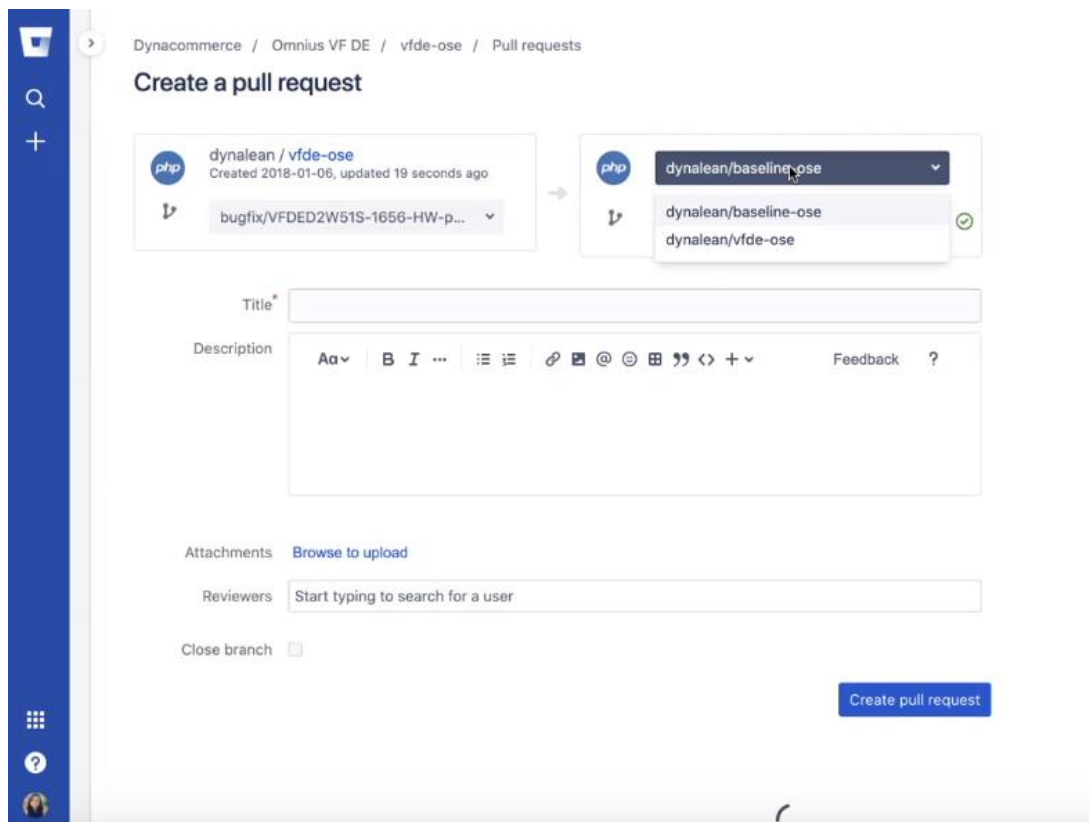
Git Push- Sends local commits to the remote repository. *Git push* requires two parameters: the remote repository and the branch that the push is for.

```
(15:15:32)[vanshikis-MacBook-Pro] ~/code/vfde-wave5 [bugfix/VFDED2W51S-1656-HW-payment-section-wave5]
$ git push --set-upstream origin bugfix/VFDED2W51S-1656-HW-payment-section-wave5
Enumerating objects: 17, done.
Counting objects: 100% (17/17), done.
Delta compression using up to 8 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (9/9), 822 bytes | 822.00 KiB/s, done.
Total 9 (delta 6), reused 0 (delta 0)
remote:
remote: Create pull request for bugfix/VFDED2W51S-1656-HW-payment-section-wave5:
remote: https://bitbucket.org/dynalean/vfde-ose/pull-requests/new?source=bugfix/VFDED2W51S-1656-HW-payment-section-wave5&t=1
remote:
To bitbucket.org:dynalean/vfde-ose.git
 * [new branch]      bugfix/VFDED2W51S-1656-HW-payment-section-wave5 -> bugfix/VFDED2W51S-1656-HW-payment-section-wave5
Branch 'bugfix/VFDED2W51S-1656-HW-payment-section-wave5' set up to track remote branch 'bugfix/VFDED2W51S-1656-HW-payment-section-wave5' from 'origin'.
(15:15:46)[vanshikis-MacBook-Pro] ~/code/vfde-wave5 [bugfix/VFDED2W51S-1656-HW-payment-section-wave5]
```

Use: - Here we got the remote link for PR

8. Link that we got from previous step

Fill particular wave branch in the given space and must read all the instructions to specifying labels and attachments in the given space.



Follow PSR2 standard coding norms:

- Code MUST follow a “coding style guide” PSR [[PSR-1](#)].
- Code MUST use 4 spaces for indenting, not tabs.
- There MUST NOT be a hard limit on line length; the soft limit MUST be 120 characters; lines SHOULD be 80 characters or less.
- There MUST be one blank line after the `namespace` declaration, and there MUST be one blank line after the block of `use` declarations.
- Opening braces for classes MUST go on the next line, and closing braces MUST go on the next line after the body.
- Opening braces for methods MUST go on the next line, and closing braces MUST go on the next line after the body.
- Visibility MUST be declared on all properties and methods; `abstract` and `final` MUST be declared before the visibility; `static` MUST be declared after the visibility.
- Control structure keywords MUST have one space after them; method and function calls MUST NOT.
- Opening braces for control structures MUST go on the same line, and closing braces MUST go on the next line after the body.

- Opening parentheses for control structures MUST NOT have a space after them, and closing parentheses for control structures MUST NOT have a space before.

9. Checking for Unit Test Cases and Jenkins Pipelines

All dependencies will be installed and coding standards will be checked

vodafone-de / ose < 5 Pipeline Changes Tests Artifacts Logout

Pull Request: PR-14626 2m 14s No changes
Commit: a3effbd 15 hours ago Branch indexing

Install dependencies

No steps This stage has no steps

```

1  Branch indexing
2  Checking out git@bitbucket.org:dynalean/vfde-ose.git git@bitbucket.org:dynalean/vfde-ose.git into /var/lib/jenkins/workspace/vodafone-de_ose_PR-14626@script to read Jenkinsfile
3  using credential jenkins
4  using credential jenkins
5  > git rev-parse --is-inside-work-tree # timeout=10
6  Fetching changes from 2 remote Git repositories
7  > git config remote.origin.url git@bitbucket.org:dynalean/vfde-ose.git # timeout=10
8  Fetching without tags
9  Fetching upstream changes from git@bitbucket.org:dynalean/vfde-ose.git
10 > git --version # timeout=10
11 using GIT_SSH to set credentials Jenkins bitbucket ssh key
12 > git fetch --no-tags --progress git@bitbucket.org:dynalean/vfde-ose.git +refs/heads/bugfix/VFDED2WS25-391:refs/remotes/origin/PR-14626
13 > git config remote.upstream.url git@bitbucket.org:dynalean/vfde-ose.git # timeout=10
14 Fetching without tags
15 Fetching upstream changes from git@bitbucket.org:dynalean/vfde-ose.git
16 using GIT_SSH to set credentials Jenkins bitbucket ssh key
17 > git fetch --no-tags --progress git@bitbucket.org:dynalean/vfde-ose.git +refs/heads/develop-wave5:refs/remotes/upstream/develop-wave5
18 Merging remotes/upstream/develop-wave5 commit df065fd95e282bfd4a337e347bd4d3bclafce0a7 into PR head commit a3effbda60c47e256d70d58ef2189cd041e37c85
19 > git config core.sparsecheckout # timeout=10
20 > git checkout -f a3effbda60c47e256d70d58ef2189cd041e37c85
  
```

Activate Windows
Go to Settings to activate Windows.