

Notes

INDEX

[INDEX](#)

[STORIES WORKFLOW](#)

[GENERAL](#)

[DEVELOPMENT](#)

[Launching On Pod Branch](#)

[Merging](#)

[Asking for a Deploy](#)

[BUILD COMMERCE-PROJECT](#)

[PROJECT](#)

[SERVICES](#)

[WEB](#)

[JBEHAVES](#)

[LOG IN](#)

[DOCUMENTATION COMMERCE-PROJECT](#)

[Styleguide](#)

[CSS Colors](#)

[GIT](#)

[FILE STATUS](#)

[COMMANDS](#)

[TAGS](#)

[AGILE PROCESS](#)

[STAND-UP](#)

[GROOMING](#)

[PLANNING](#)

[RETROSPECTIVE](#)

[SHORTCUTS](#)

[UBUNTU](#)

[WEBSTORM](#)

[CHROME](#)

[EXPLORING COMMERCE-PROJECT](#)

[PATHS](#)

[Account Bar](#)

[Controller](#)

[SHTML](#)

[JSPF](#)

[JBehaves](#)

[Toggle Features](#)

[Decorators](#)

[Schedule a Flight](#)

[JSP](#)

[JSPF](#)

[Controller](#)

[FIT State - Incidents](#)

[SHTML](#)

STORIES WORKFLOW

GENERAL

- Analysis → Speak with everyone (including mkt.)
 - Before Planning, everyone should be familiar with every story that's gonna be covered (in planning meetings on wednesdays).
 - Before SME Signoff (product owner or mkt. reviews) everything should be green and defined.
 - Story Presentation → All the team presents.
 - BA Val. → Review acceptance criteria.
 - Defects have less phases (last one is closed).
 - Test Dev.: BA o QC review the Acceptance Criteria.
 - Commit and Deploy → Between Code Rev. y BA Val.
 - If the execution passes, the story is passed to Ready for Certification. From now on, only Pod Bugs.
-

DEVELOPMENT

1. Jira Story is assigned to Developer.

2. Pull Rebase:

```
git pull --rebase origin master
```

3. Branch creation using the Story's identifier (e.g. OPS 13...). storyID = branchname

```
git checkout -b <branchname>
```

4. (Coding)

5. Move modifications to staging area:

```
git add <file name or directory name>
```

6. Commit:

```
git commit -m "<devname 1 / ... / devname n> - <storyID> - <description>"
```

7. Switch to master:

```
git checkout master
```

8. Pull rebase:

```
git pull --rebase origin master
```

9. Switch to my branch:

```
git rebase master
```

10. Build to verify change

11. Push to Stash

```
git push origin <branchname>
```

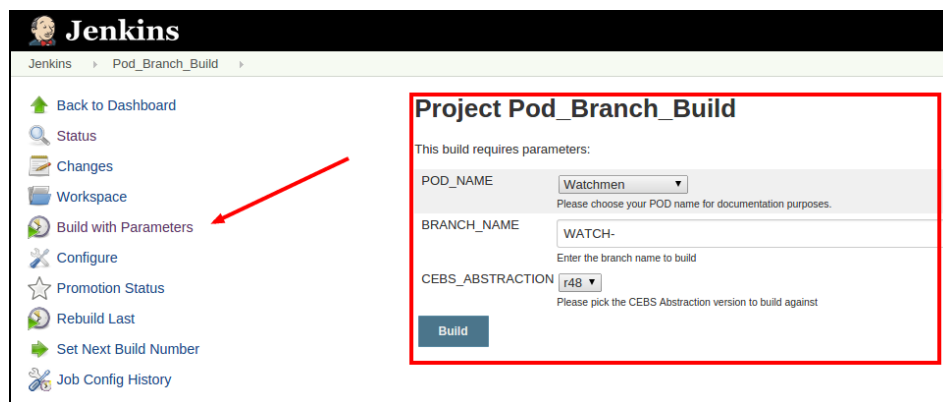
12. Enter Stash and create Pull Request: Select branch so that people I determine review the code (Reviewers).

13. Diff changes:

```
git diff HEAD^1 HEAD [> fileaddress]
```

Launching On Pod Branch

- Go to http://lfjenkins.swacorp.com/jenkins/job/pod_branch_Build/
- Log In
- Build With Parameters



Merging

- Get 2 Approval in Code Review.
- Wait for Jenkins to be green.

Asking for a Deploy

How to fill the fields:

Build (i.e. Core)	Trunk
BUILD NUMBER	#22396 (Check it out on Jenkins)
Environment	(i.e. DEV-B) DEV-I
Summary	Deploy del Trunk_Build #22396
Description	Deploy del Trunk_Build #22396

BUILD COMMERCE-PROJECT

Open 3-window Terminal:

- Project
- Services
- Web

1. PROJECT

- Explore make process
`./make.sh -- help`
- Build (Skipping Tests):
`./make.sh -UCKF -Sv47`
`./make.sh -CKF -s48`
- Build with Tests:
`./make.sh -UCKF -v47`

* Version number (47) May change.

2. SERVICES

- Start Dynamic services:
`./start.sh -d47`
- Start Real services:
`./start.sh -r47`

3. WEB

- Start web process:
`./start.sh`

4. JBEHAVES

- In terminal:
`cd commerce-project/jbehave-webdriver/air`

```
mvn3 -e clean install -Pjbehave-story-run -DDYNA_STUBS  
-DstoryFilter=<jbehave_story_name.story>  
-DJBEHAVE_WEBDRIVER_FIREFOX_PROFILE=firefox-10
```

5. LOG IN

- *UserId:* 500 321 334
- *Password:* test123

DOCUMENTATION COMMERCE-PROJECT

Styleguide

- https://www.southwest.com/assets/pdfs/online_styleguide/SWA_Digital_Styleguide.pdf

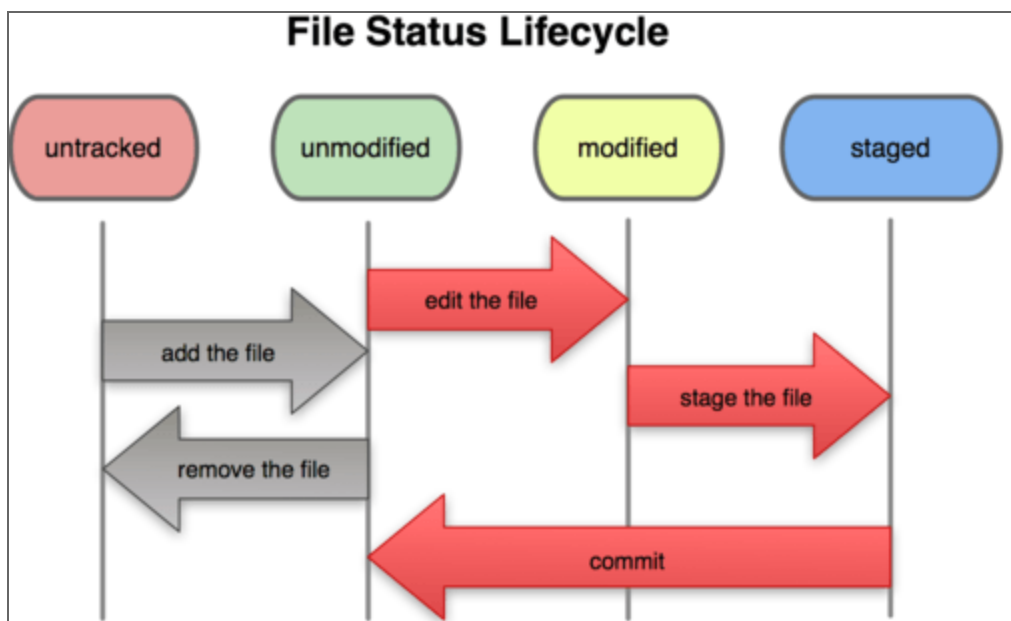
CSS Colors

- https://stash.swacorp.com/projects/UI/repos/swa-ui-common/browse/sass-globals/_variables.scss

GIT

FILE STATUS

- **UNTRACKED**
- **TRACKED**
 - **Modified:** you have changed the file but have not committed it to your database yet.
 - **Staged:** you have marked a modified file in its current version to go into your next commit snapshot.
 - **Committed:** Data is safely stored in your local database. (*Unmodified*).



- **Working Directory:** copy of one version of the project. What's in here is already committed.
- **Staging Area:** what will go into my next commit. Modified but not committed and added to the stage area (Staged).
- **Git Directory:** it's not in my computer, it is what is copied when I clone a repository from another computer. Modified since checked out, but not staged.

COMMANDS

`git status`

- *Checks files' state.*
- *Indicates current branch.*
- *Shows:*
 - *changes to be committed.*
 - *changes not staged for commit.*

`git add [filename]`

- *Begins tracking a new file.*
- *Moves a file to staging area. The version added is the one we're going to commit when we do so.*
- *If its a directory, it adds contents recursively.*

`git diff`

- *What was changed but not staged?*
- *What have you changed that you are about to commit?*
- *Compares what's in your working directory with what is in your staging area.*

`git diff --staged`

- *Shows what will go into your next commit.*

`git commit -m "[message]"`

- *Commits whatever is staged (files that you have run git add before).*

`git commit -a -m "[message]"`

- *Stages first and then commits.*

`git rm [filename]`

- *Removes a file.*

```
git mv [filenameFrom] [filenameTo]
```

- *Renames a file.*

```
git log
```

- *Shows what has happened before.*

```
git clone
```

- *Creates a copy of the Repository in a specific directory.*

```
git tag
```

- Lists available tags.

```
git tag -a v1.4 -m '[message]'
```

- Creates an annotated tag. "m" stands for the tagging message.

```
git branch [branchname]
```

- Creates a new branch.

```
git branch
```

- Show all local branches

```
git branch -d [branchname]
```

- Deletes a local branch.

```
git push origin --delete [branchname]
```

- Deletes a remote branch.

```
git checkout [branchname]
```

- Moves through branches, setting the current one.

```
git checkout -b [branchname]
```

- Creates a new branch and moves to it. Equivalent to both previous commands.

```
git stash
```

- Saves the current changes on a auxiliar repository.

```
git stash save [name]
```

- Stashes with a name for that bundle.

```
git stash list
```

- Shows the list of previously stashed bundles.

TAGS

Functionality used to mark release points.

- **Lightweight:** are just a pointer to a specific commit
- **Annotated:** are stored as full objects in the Git database. They're checksummed; contain the tagger name, e-mail, and date; have a tagging message.

AGILE PROCESS

Sprints last 2 weeks and start on Wednesdays.

STAND-UP

- Daily Meetings at 9:00 Dallas Time. (11:00 Cordoba).
- Inform what you worked on yesterday, and what you're going to do today.

GROOMING

- Takes place in the middle of the Sprint.
- Issues of the **following sprint** are covered.
- Estimate **Complexity**, using *Fibonacci*; 13 would be a very complex story.

PLANNING

- Estimate **Hours (Effort)**. In Increments of 3.

RETROSPECTIVE

- What went fine in the last sprint?
 - What went wrong in the last sprint?
 - Questions?
-

Things to Estimate

- Complexity
- Effort
- Doubts/Risks

* A top level of points can be determined and “subtract” these points in order to resolve this story in the next Sprint.

SHORTCUTS

Add any shortcuts here:

UBUNTU

- *Show Desktop:* Command + Tab
- *New Terminal Tab:* Ctrl + Shift + T
- *Switch Workspace:* Ctrl + Alt + Arrow Key

WEBSTORM

- *Search Document:* Ctrl + Shift + N
- *Recent Documents:* Ctrl + Shift + E
- *Search Content:* Ctrl + Shift + F
- *Close Tab:* Shift + Click or Ctrl + W
- *Go to Page:* Ctrl + G

CHROME

- *Search Bar:* Ctrl + K
- *Address Bar:* Alt + D

EXPLORING COMMERCE-PROJECT

PATHS

commerce-project/

Account Bar

Controller

web/src/main/java/com/swacorp/dotcom/account/accountbar

SHTML

web/src/main/public/fragments/app/page_components/account_bar.shtml

JSPF

web/src/main/webapp/WEB-INF/jsp/accountBar

JBehaves

jbehave-webdriver/air/src/main/groovy/ (keep browsing and find groovy files)

Toggle Features

common/togglefeatures/features.xml

Decorators

web/src/main/webapp/decorators

Schedule a Flight

JSP

web/src/main/webapp/WEB-INF/jsp/flightschedules/flightSchedulesResult.jsp

web/src/main/webapp/WEB-INF/jsp/flightschedules/flightSchedulesForm.jsp

JSPF

web/src/main/webapp/WEB-INF/jsp/WEB-INF/jsp/homepage/booking-form-air.jspf

Controller

web/src/main/java/com/swacorp/dotcom/flightschedules/controller/flightSchedulesController.java

FIT State - Incidents

SHTML

web/src/main/public/fragments/generated/incident