

—— Le 06/11/2023

Développement mobile

Enseirb-Matmeca,
Filière Info, GL 2023



Enseirb-
Matmeca

Sommaire

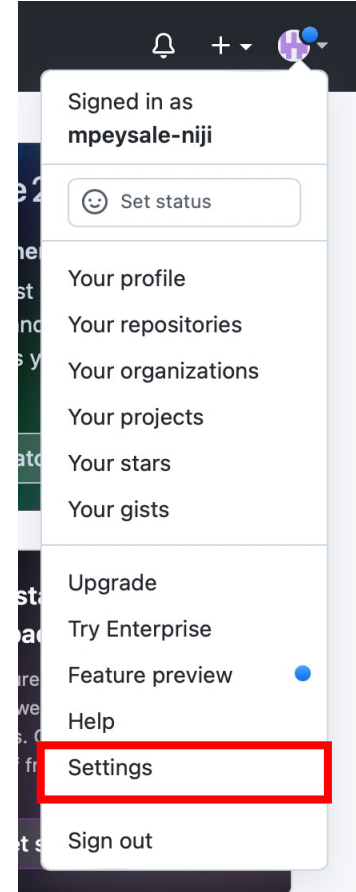
1. Projet



GitHub

<https://github.com>

-> Créer son compte



GitHub

<https://github.com>

-> Créer sa clé SSH



mpeysale-niji

Your personal account [Switch 1](#)



Public profile



Account



Appearance



Accessibility



Notifications

Access



Billing and plans



Emails



Password and authentication



SSH and GPG keys



Organizations



Moderation



GitHub

<https://github.com>

-> Créer sa clé SSH

```
ssh-keygen -t rsa
```

Pas de passphrase

```
cd ~/.ssh  
cat id_rsa.pub
```

GitHub

SSH keys

New SSH key

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

Authentication Keys



SSH

MacBook Pro de Niji

SHA256:9pNWofADndhS3rU9csveSw09VxCBKoPqt9TsBtDZzAc

Added on 27 days ago

Last used within the last week — Read/write

Delete

Check out our guide to [generating SSH keys](#) or troubleshoot [common SSH problems](#).

GitHub

Copier la clé publique

[SSH keys](#) / Add new

Title

Key type

Authentication Key ▾

Key

Begins with 'ssh-rsa', 'ecdsa-sha2-nistp256', 'ecdsa-sha2-nistp384', 'ecdsa-sha2-nistp521', 'ssh-ed25519', 'sk-ecdsa-sha2-nistp256@openssh.com', or 'sk-ssh-ed25519@openssh.com'

Add SSH key

Git

Pratique

```
mkdir git_example  
cd git_example  
git init
```

```
git status
```

```
[marc.peysale@RVF62P24RQ git_example % git status  
On branch main  
  
No commits yet  
  
nothing to commit (create/copy files and use "git add" to track)
```


Git

Pratique

```
git flow init
```

```
> Branch name for production releases: main
```

```
> Branch name for "next release" development: [develop]
```

```
How to name your supporting branch prefixes?
```

```
Feature branches? [feature/]
```

```
Release branches? [release/]
```

```
Hotfix branches? [hotfix/]
```

```
Support branches? [support/]
```

```
Version tag prefix? []
```






—

Git

Pratique

git status

```
[marc.peysale@RVF62P24RQ git_example % git status  
On branch develop  
nothing to commit, working tree clean _
```

Graph	Description	Commit	Author	Date	
	  develop  main Initial commit	c2bfacd	Marc PEYSALE <marc.peysale@niji.fr>	Today at 18:03	

Git

Pratique

`git flow feature start feature1`

ou

`git checkout -b feature/feature1`

```
marc.peysale@RVF62P24RQ git_example % git flow feature start feature1
Switched to a new branch 'feature/feature1'
```

Summary of actions:

- A new branch 'feature/feature1' was created, based on 'develop'
- You are now on branch 'feature/feature1'

Now, start committing on your feature. When done, use:

```
git flow feature finish feature1
```

Git

Pratique

```
touch changelog.txt
```

```
git add changelog.txt  
git commit -m "Created file"
```

Git

Pratique

```
git flow feature finish feature1
```

ou

```
git checkout develop
```

```
git merge feature/feature1
```

```
marc.peysale@RVF62P24RQ git_example % git flow feature finish feature1
```

```
Switched to branch 'develop'
```

```
Updating c2bfacd..cd875b9
```

```
Fast-forward
```

```
 changelog.txt | 0
```

```
 1 file changed, 0 insertions(+), 0 deletions(-)
```

```
 create mode 100644 changelog.txt
```

```
Deleted branch feature/feature1 (was cd875b9).
```

```
Summary of actions:
```

- The feature branch 'feature/feature1' was merged into 'develop'
- Feature branch 'feature/feature1' has been removed
- You are now on branch 'develop'

Git



Conflits

```
git flow feature start feature2  
echo "feature 2" > changelog.txt  
git add changelog.txt  
git commit -m "Feature 2"
```

```
git flow feature start feature3  
echo "feature 3" > changelog.txt  
git add changelog.txt  
git commit -m "Feature 3"
```

Git

Conflits

	feature/feature3 Feature 3	6354625	Marc PEYSALE <marc.peysale@niji.fr>	Today at 18:19	
	feature/feature2 Feature 2	09bc80f	Marc PEYSALE <marc.peysale@niji.fr>	Today at 18:18	
	develop Created file	cd875b9	Marc PEYSALE <marc.peysale@niji.fr>	Today at 18:11	
	main Initial commit	c2bfacd	Marc PEYSALE <marc.peysale@niji.fr>	Today at 18:03	

Git

Conflits

git flow feature finish feature2

```
marc.peysale@RVF62P24RQ git_example % git flow feature finish feature2
```

```
Switched to branch 'develop'
```

```
Updating cd875b9..09bc80f
```

```
Fast-forward
```

```
 changelog.txt | 1 +
```

```
 1 file changed, 1 insertion(+)
```

```
Deleted branch feature/feature2 (was 09bc80f).
```

```
Summary of actions:
```

- The feature branch 'feature/feature2' was merged into 'develop'
- Feature branch 'feature/feature2' has been removed
- You are now on branch 'develop'

Git

Conflits

git flow feature finish feature3

```
marc.peysale@RVF62P24RQ git_example % git flow feature finish feature3
```

```
Already on 'develop'
```

```
Auto-merging changelog.txt
```

```
CONFLICT (content): Merge conflict in changelog.txt
```

```
Automatic merge failed; fix conflicts and then commit the result.
```

```
mkdir: /.gitflow: Read-only file system
```

```
/opt/homebrew/Cellar/git-flow/0.4.1_1/libexec/bin/git-flow-feature: line 320: /.gitflow/MERGE_BASE: No such file or directory
```

There were merge conflicts. To resolve the merge conflict manually, use:

```
git mergetool
```

```
git commit
```


You can then complete the finish by running it again:

```
git flow feature finish feature3
```

Git

Conflits

All Branches Show Remote Branches Ancestor Order Jump to:

Graph	Description	Commit	Author	Date
	Uncommitted changes *	*		Today at 18:21 +
	feature/feature3 Feature 3	6354625	Marc PEYSALE <marc.peysale@niji.fr>	Today at 18:19
	develop Feature 2	09bc80f	Marc PEYSALE <marc.peysale@niji.fr>	Today at 18:18
	Created file	cd875b9	Marc PEYSALE <marc.peysale@niji.fr>	Today at 18:11
	main Initial commit	c2bfacd	Marc PEYSALE <marc.peysale@niji.fr>	Today at 18:03

☒ Pending files, sorted by path

☒ changelog.txt

changelog.txt

Hunk 1 : Lines 1-5

1

+ <<<<<< HEAD

feature 2

+ =====

+ feature 3

+ >>>>>> feature/feature3

Stage hunk

Discard hunk

Git

Conflicts

```
[marc.peysale@RVF62P24RQ git_example % cat changelog.txt
<<<<<<< HEAD
feature 2
=====
feature 3
>>>>>>> feature/feature3
```

Git

Conflicts

```
[marc.peysale@RVF62P24RQ git_example % cat changelog.txt
<<<<<<< HEAD
feature 2
=====
feature 3
>>>>>>> feature/feature3
```

emacs changelog.txt

```
File Edit Options Buffers Tools Text Help
```

```
feature 2
feature 3
```

Git

Conflits

git commit

```
Merge branch 'feature/feature3' into develop
```

```
# Conflicts:
```

```
#     changelog.txt
```

```
#
```

```
# It looks like you may be committing a merge.
```

```
# If this is not correct, please run
```

```
#     git update-ref -d MERGE_HEAD
```

```
# and try again.
```

```
# Please enter the commit message for your changes. Lines starting
```

```
# with '#' will be ignored, and an empty message aborts the commit.
```

```
#
```

```
# On branch develop
```

```
# All conflicts fixed but you are still merging.
```

```
#
```

```
# Changes to be committed:
```

```
#     modified:   changelog.txt
```

```
#
```

```
~
```

```
~
```

```
~
```

```
"~/Documents/workspace/git_example/.git/COMMIT_EDITMSG" 20L, 476B
```

```
[marc.peysale@RVF62P24RQ git_example % git commit  
[develop 3ccd227] Merge branch 'feature/feature2' into develop
```

Git

Conflits



A vertical line of blue dots represents the commit history. An orange line branches off from the 'main' branch (blue dot) to create the 'feature/feature3' branch. The 'feature/feature3' branch has three commits (orange dots) before merging back into the 'develop' branch (blue dot). The 'main' branch has one commit (blue dot).

🔗 develop	Merge branch 'feature/feature2' into develop	3ccd227	Marc PEYSALE <marc.peysale@niji.fr>	Today at 18:24	+
🔗 feature/feature3	Feature 3	6354625	Marc PEYSALE <marc.peysale@niji.fr>	Today at 18:19	
	Feature 2	09bc80f	Marc PEYSALE <marc.peysale@niji.fr>	Today at 18:18	
	Created file	cd875b9	Marc PEYSALE <marc.peysale@niji.fr>	Today at 18:11	
🔗 main	Initial commit	c2bfacd	Marc PEYSALE <marc.peysale@niji.fr>	Today at 18:03	

```
marc.peysale@RVF62P24RQ git_example % git status
On branch develop
nothing to commit, working tree clean
```

```
marc.peysale@RVF62P24RQ git_example % cat changelog.txt
feature 2
feature 3
```

Git

Tuto Git Flow

<https://www.atlassian.com/fr/git/tutorials/comparing-workflows/gitflow-workflow>

Projet

Application de cocktails

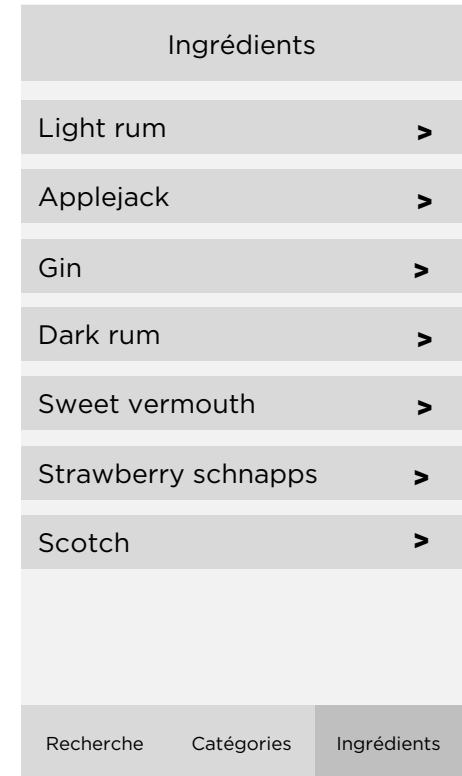
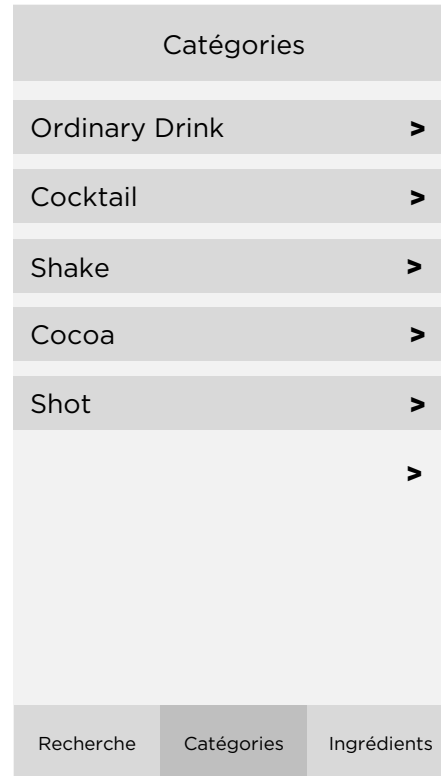
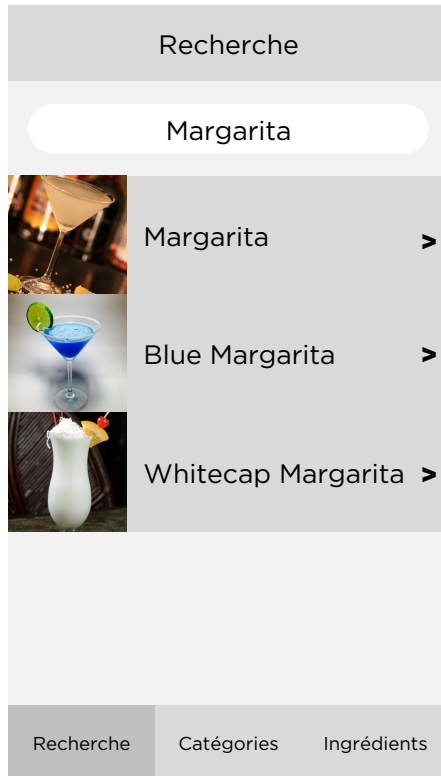
- The Cocktail DB
 - API publique et gratuite
 - Liste de cocktails
 - Recherche
 - Catégories
 - Ingrédients...

Features minimales

- App Android, écrite en Kotlin
 - Tab Bar
 - Premier onglet : Recherche
 - Deuxième onglet : liste des catégories
 - Troisième onglet : liste des ingrédients
 - Liste des recettes associées aux catégories/recettes
 - Détail de recette

Features minimales

Wireframes



Features minimales

Détails de recette

Smashed Watermelon Margarita



Category: **Cocktail**

Served in: **Collins glass**

Instructions

In a mason jar muddle the watermelon and 5 mint leaves together into a puree and strain. Next add the grapefruit juice, juice of half a lime and the tequila as well as some ice. Put a lid on the jar and shake. Pour into a glass and add more ice. Garnish with fresh mint and a small slice of watermelon.

Smashed Watermelon Margarita

into a glass and add more ice. Garnish with fresh mint and a small slice of watermelon.

Ingredients

- **Watermelon** (1/2 cup)
- **Mint** (5)
- **Grapefruit Juice** (1/3 cup)
- **Lime** (Juice of 1/2)
- **Tequila** (1 shot)
- **Watermelon** (Garnish with)
- **Mint** (Garnish with)

Features optionnelles

- Gestion des favoris
 - Via les SharedPreferences
- Recette de cocktail aléatoire
- Gestion cocktails alcoolisés/non alcoolisés
- A votre imagination !

API

- Liste des catégories

www.thecocktaildb.com/api/json/v1/1/list.php?c=list

- Liste des ingrédients

www.thecocktaildb.com/api/json/v1/1/list.php?i=list

- Recherche par nom

www.thecocktaildb.com/api/json/v1/1/search.php?s=margarita

- Filtre par ingrédients/catégories

www.thecocktaildb.com/api/json/v1/1/filter.php?i=Gin
www.thecocktaildb.com/api/json/v1/1/filter.php?c=Cocktail

- Détail de recette

www.thecocktaildb.com/api/json/v1/1/lookup.php?i=11007

Rendu

- Projet à rendre sous forme de Github public
- A effectuer en **binôme**
- Le projet doit contenir à la racine un mini-rapport au format Markdown (Readme.md)
 - Nom des étudiants
 - Fonctionnalités développées
 - Difficultés rencontrées
- Mail avec l'URL du projet au plus tard **le 08/01/2024 à 18h**
>>>> marc.peysale@niji.fr <<<<

Notation

- Le projet sera évalué selon les critères suivants :
 - L'ensemble des features obligatoires est implémenté
 - Le code est découpé proprement
 - Le code compile
 - Pas de crash à l'exécution
 - Les deux membres du binôme ont participé (commits Git)
 - Présence de loaders, gestion des erreurs
- Des points seront attribués si des fonctionnalités additionnelles sont ajoutées
- Le design de l'app ne sera pas pris en compte, mais faites en sorte que l'app donne envie d'être utilisée

Tips

- Utiliser des composants Material Design
- Utiliser la lib Picasso pour l'affichage d'images provenant d'une URL

Marc PEYSALE

Mobile Team Lead

M : +33 6 31 77 86 65

@ : marc.peysale@niji.fr

niji.fr

