

Role Play Git : gestionnaire de version

-

Alpha Team

Lundi 24 Février

Contents

1	Preparation de l'espace de travail	1
1.1	Introduction	1
1.2	Création d'un compte framagit	2
1.3	Constitution des binômes	2
1.4	Recuperation du depot distant: git clone	3
1.5	Récuperation du projet de page web "initialWebPage"	4
1.6	Premier commit	5
1.7	Prêts à travailler en équipe !	5
2	Premiers pas chacun de votre côté : Partie Alpha Team	6
2.1	Préparez (récupérez) votre espace de travail	6
2.2	Exercice 1 - Afficher plus d'informations sur la page HTML - Alpha Team	6
2.2.1	Travail individuel	6
2.2.2	Synchronisation avec votre binôme	7
2.3	Exercice 2 - Premiers conflits potentiels	7
2.4	Exercice 3 : Travail d'équipe - temporiser	8
3	Hot fix	9

1 Preparation de l'espace de travail

1.1 Introduction

"Un gestionnaire de version est un système qui enregistre l'évolution d'un fichier ou d'un ensemble de fichiers au cours du temps de manière à ce qu'on puisse rappeler une version antérieure d'un fichier à tout moment. [...] Nous utiliserons des fichiers sources de logiciel

comme fichiers sous gestion de version, bien qu'en réalité on puisse l'utiliser avec pratiquement tous les types de fichiers d'un ordinateur."¹

En pratique, disposer d'un dépôt git présente plusieurs avantages :

- garder une copie de votre travail sur un serveur distant (backup);
- faciliter le travail de développement en équipe ;
- revenir à une version antérieure ("si, je te le promets, chez moi ça marchait...") ;
- tracer l'évolution des développements (log, branching...)

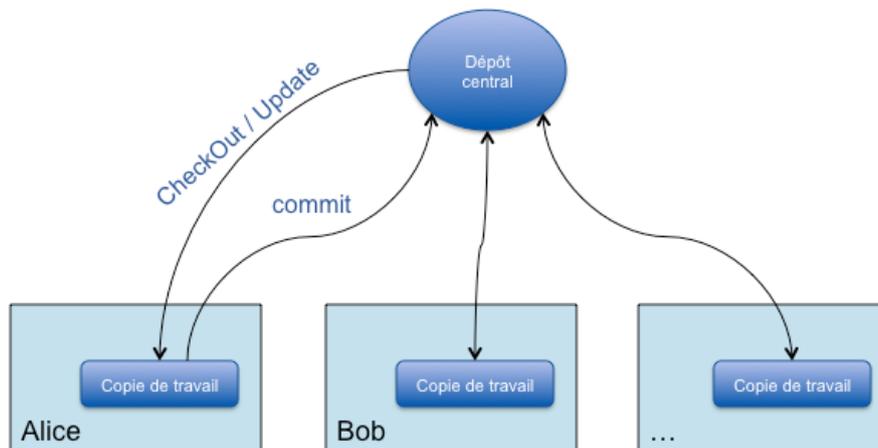


Figure 1: Fonctionnement classique d'un gestionnaire de version (centralisé)²

²<https://blog.octo.com/introduction-a-la-gestion-de-version-distribuee/>

1.2 Création d'un compte framagit

- Créez un compte sur <https://framagit.org/>

1.3 Constitution des binômes

- mettez vous en binôme
- choisissez un nom de groupe contenant vos 2 noms de famille, et créez le sur l'un de vos compte framagit comme suit :

¹<https://git-scm.com/book/fr/v2/D%C3%A9marrage-rapide-%C3%A0-propos-de-la-gestion-de-version>

- cliquez sur le petit plus en haut de la page
 - créez le groupe "groupeNom1Nom2"
 - positionnez vous sur la page du groupe que vous venez de créer
 - invitez votre binome dessus (members dans le menu de gauche)
- vérifiez que vous êtes positionné sur le groupe "groupeNom1Nom2" et créez un nouveau projet ProjectNom1Nom2 :
 - cliquez sur "new project"
 - supposons pour la suite que vous le nommiez ProjectNom1Nom2
 - cochez l'option Initialize repository with a README
 - cliquez sur "create project"
- vérifiez que vous faites tous les deux parti du même groupe
 - vérifiez que votre framagit contient bien le projet ProjectNom1Nom2 et entrez dans la page correspondante : il contient toutes les informations liées aux git dont vous aurez besoin (page <https://framagit.org/groupeNom1Nom2/ProjectNom1Nom2>)
 - accordez vous : l'un représentera l'équipe **Alpha**, l'autre, l'équipe **Bravo**
 - gardez la feuille de TP qui correspond à votre équipe (celle-ci étant l'équipe **Alpha**)
 - convenez de moyens de communications "silencieux" (e-mail, texto, chat...)

1.4 Recuperation du depot distant: git clone

```

1 #T1> Dans un terminal
2 #preparez votre espace de travail
3
4 cd
5 mkdir workspace
6 cd workspace
7
8 #cloner le depot distant que vous allez partager avec votre binome
9 git clone git@framagit.org:groupeNom1Nom2/ProjectNom1Nom2.git
10
11
12 #allez dans le nouveau repertoire
13 cd ProjectNom1Nom2
14
15 #verifiez le contenu
16 ls
17
18 #a ce stade vous devriez avoir qu'un fichier readme.md dans votre projet
    "ProjectNom1Nom2"

```

```
19
20 #placez vous a l'interieur
21 cd ProjectNom1Nom2
```

1.5 Récupération du projet de page web "initialWebPage"

Pour le membre de l'équipe Alpha **uniquement** :

Récupérez le projet sur lequel vous allez travailler. Il s'inspire fortement des TP que vous avez déjà fait.

- récupérez l'archive en suivant ce lien : <https://sourceforge.isae.fr/attachments/download/2770/initialWebPage.zip>
- décompressez là racine de votre répertoire git afin d'obtenir l'arborescence suivante:

```
1 contenu de "~/workspace/ProjectNom1Nom2/" :
2
3 readme.md
4 initialWebPage
5 |- d3.v4.min.js
6 |- data.csv
7 |- edt.html
8 |- jquery.min.js
9 |- script.js
10 |- style.css
```

- vous devriez visualiser un dossier "initialWebPage" contenant une version allégée de votre TP matinal.
- Vérifiez que vous disposez d'un projet fonctionnel.

Rappel:

```
1 #T1> Dans un terminal, placez vous dans le dossier initialWebPage
2 cd ~/workspace/ProjectNom1Nom2/initialWebPage
3
4 #lancer un serveur HTTP
5 python3 -m http.server
6
7 #visualisez sous un browser la page
8 http://localhost:8000/edt.html
9
10 #le texte "Un emploi du temps" doit s'afficher dans une grande page
    blanche...
```

1.6 Premier commit

A deux sur la même machine, pour le membre de l'équipe Alpha **uniquement** (la machine sera donc celle de l'équipe alpha) :

```
1 #T1> Dans un terminal, placez vous dans le dossier ProjectNom1Nom2
2 cd ~/workspace/ProjectNom1Nom2/
3
4 #ajouter le repertoire (et ainsi les fichiers) que vous souhaitez sur le
   depot
5 git add initialWebPage
6
7 #commitez ce simple ajout, en fournissant un message explicite, par
   exemple :
8 git commit -m "[dev] initialisation de l'espace de travail"
9
10 #pushez vers le depot distant
11 git push origin master
12
13 #quelques details:
14 #le mot-cle "origin" indique le nom par default du depot git que vous
   avez clone
15 #le mot-cle "master" indique la branche sur laquelle vous travaillez.
   Ici, il s'agit de la branche master, la principale.
```

1.7 Prêts à travailler en équipe !

- separez-vous (chacun dans une salle, le membre de l'équipe Bravo change de salle)

2 Premiers pas chacun de votre côté : Partie Alpha Team

2.1 Préparez (récupérez) votre espace de travail

- Pour la suite, gardez ouverts:
 - Un éditeur pour les fichiers `script.js` et `edt.html`
 - un terminal qui fait tourner le serveur python
 - un terminal pour gérer git
 - un browser pour visualiser vos modifications

2.2 Exercice 1 - Afficher plus d'informations sur la page HTML - Alpha Team

2.2.1 Travail individuel

- modifier l'affichage HTML pour avoir la somme totale des heures de cours (en dessous de l'EDT)
 - si possible, insérez votre code entre les commentaires
- ```
1 script.js
2 //BEGIN Team Alpha code
3
4 //END Team Alpha code
5
6 OU
7
8 edt.html
9 <!-- BEGIN Team Alpha code-->
10
11 <!-- END Team Alpha code-->
```
- dans le `script.js`, définissez à cet effet une fonction `displayHours()`
  - ajouter une ligne plutôt vers la fin de la page `edt.html` (ex: `<div id="sum"></div>`)
  - n'oubliez pas d'appeler `displayHours()` dans la fonction `fetch_courses()` !
  - visualisez le résultat sur votre browser

- dès que tout semble correcte, commitez votre travail

```
1 # dans le repertoire ~/workspace/ProjectNom1Nom2/ :
2 git commit -m "[dev]affichage de la somme des heures de cours en
 dessous de l'EDT"
```

## 2.2.2 Synchronisation avec votre binôme

- vérifiez si votre binôme a avancé de son côté par un pull, pour mettre à jour votre repertoire :

```
1 # dans le repertoire ~/workspace/ProjectNom1Nom2/ :
2 git pull origin master
```

- et enfin, poussez votre travail sur le dépôt distant, afin que votre binôme puisse en bénéficier :

```
1 # dans le repertoire ~/workspace/ProjectNom1Nom2/ :
2 git push origin master
```

- à partir de ce stade, vérifiez régulièrement si votre binôme a avancé

```
1 # pour recuperer ses contributions :
2 git pull origin master
3 # pour visualiser ses commentaires (et les votres !) :
4 git log
```

- naturellement, sur votre browser, rechargez votre page web pour visualiser les contributions de votre binôme. Pensez également à recharger la page de votre éditeur de texte. En effet `script.js` et `edt.html` ont été modifié ! Visualisez ces modifications.
- si aucun message n'apparaît sur le log de votre binôme, vous pouvez râler par messagerie interposée pour lui demander ce qu'il a modifié. Rappelez lui qu'il devrait veiller à laisser un message de commit explicite.

## 2.3 Exercice 2 - Premiers conflits potentiels

- Votre code `script.js` doit contenir une fonction `displayHours()`. Si ce n'est pas le cas, aidez votre binôme pour que ça le soit.
- ajoutez un (ou plusieurs) argument(s) à cette fonction pour permettre de choisir de quel "tutor" vous souhaitez visualiser la somme des heures de cours.
- adaptez votre code en conséquence pour visualiser et valider votre modification
- dès que tout semble correcte, commitez votre travail

```
1 # dans le repertoire ~/workspace/ProjectNom1Nom2/ :
2 git commit -m "[dev] amelioration, possibilite d'affichage de la
 somme des heures de cours en dessous de l'EDT par tutor"
```

- vérifiez si votre binôme a avancé de son côté par un pull, pour mettre à jour votre repertoire :

```
1 # dans le repertoire ~/workspace/ProjectNom1Nom2/ :
2 git pull origin master
```

- **ATTENTION** : à ce stade un message de git peut apparaitre, du type :

```
1 CONFLICT (content): Merge conflict in ProjectNom1Nom2/script.js
```

- Si c'est le cas, reportez vous à la documentation [https://githowto.com/resolving\\_conflicts](https://githowto.com/resolving_conflicts) et résolvez la situation ! Sinon poussez votre travail sur le dépôt distant, afin que votre binôme puisse en bénéficier :

```
1 # dans le repertoire ~/workspace/ProjectNom1Nom2/ :
2 git push origin master
```

## 2.4 Exercice 3 : Travail d'équipe - temporiser

- ajouter un menu déroulant qui permet de sélectionner l'intervenant et le module
- pour ce faire, vous pouvez utiliser:
  - pour ce faire, vous pouvez utiliser les liste dropdown [https://www.w3schools.com/tags/tag\\_select.asp](https://www.w3schools.com/tags/tag_select.asp) :

```
1 dans edt.html
2
3 <select id="dropDownEx">
4
5 <select id="test">
6 <option value="volvo">Volvo</option>
7 <option value="saab">Saab</option>
8 <option value="mercedes">Mercedes</option>
9 <option value="audi">Audi</option>
10 </select>
```

- et ajouter dans le javascript code proche de celui ci:

```
1 dans script.js
2 d3.select("#dropDownEx").selectAll(".tutors").data(tutors).
 enter()
3 .append("option").attr("value", function(d) {return d;})
4 .attr("class", "tutors").text(function(d) {return d;})
```

```
1 # dans le repertoire ~/workspace/ProjectNom1Nom2/ :
2 git commit -m "[dev] ajout d'une liste dropdown pour la selection
 des tuteurs
```

- vérifiez si votre binôme a avancé de son côté par un pull, pour mettre à jour votre repertoire :

```
1 # dans le repertoire ~/workspace/ProjectNom1Nom2/ :
2 git pull origin master
```

- **ATTENTION** : à ce stade un message de git peut apparaitre, du type :

```
1 CONFLICT (content): Merge conflict in ProjectNom1Nom2/script.js
```

- Si c'est le cas, reportez vous à la documentation [https://githowto.com/resolving\\_conflicts](https://githowto.com/resolving_conflicts) et résolvez la situation ! Sinon poussez votre travail sur le dépôt distant, afin que votre binôme puisse en bénéficier :

```
1 # dans le repertoire ~/workspace/ProjectNom1Nom2/ :
2 git push origin master
```

### 3 Hot fix