

Bosser sur la zone avec git

vendredi 22 septembre 2017, par [placido](#)

```
1. #####
   ##
2. ## Création d'un nouveau plugin sur la zone #####
3. #####
   ##
4.
5. # On commence par récupérer les plugins de la zone, mais sans prendre le contenu des
   dossiers
6. svn checkout --depth=immediates svn://zone.spip.org/spip-zone/_plugins_
7.
8. # On crée ensuite son plugin et un trunk
9. cd _plugins_
10. mkdir mon_plug
11. mkdir mon_plug/trunk
12.
13. # puis on le commit
14. svn add mon_plug
15. svn add mon_plug/trunk
16. svn commit -m "ajout du nouveau plugin mon_plug"
17.
18. # On peut ensuite effacer tout ça, pour la suite on bossera avec git svn
19.
20. #####
   ###
21. ## importer un dépôt git existant dans la zone #####
22. #####
   ###
23.
24. # On crée le dossier du plugin sur la zone comme ci-dessus,
25. # puis on clone ce dossier en local
26.
27. # l'option -s à la fin est utile si le dépôt SVN a la structure trunk branches tags
28. git svn clone --prefix=origin/ svn://zone.spip.org/spip-zone/_plugins_/mon_plug -s
29.
30. # On crée une branche temporaire
31. git branch svn
32.
33. # On importe le dépôt git existant (p.ex ici sur github
34. git remote add github git@github.com:mon_compte_github/mon_plug.git
35. git pull github master
36.
37. # Puis on passe tous ces commits git au-dessus du premier commit svn
38. git rebase svn
39.
```

```
40. # On peut alors effacer la branche temporaire...
41. git branch -d svn
42.
43.
44. #####
   ###
45. ## Travailler sur un plugin existant déjà sur la zone #####
46. #####
   ###
47.
48. # on clone le projet à l'endroit souhaité
49. # l'option -s à la fin est utile si le dépôt SVN a la structure trunk branches tags
50. git svn clone --prefix=origin/ svn://zone.spip.org/spip-zone/_plugins_/mon_plug -s
51.
52. # le mieux est ensuite de se créer une branche locale
53. git branch local
54. # et de travailler sur cette branche
55. git checkout local
56.
57. # Pour récupérer les nouveautés sur la Zone,
58. # on repasse sur la branche principale
59. git checkout master
60. # et on chope les nouveautés
61. git svn rebase
62. # qu'on merge ensuite sur la branche locale
63. git checkout local
64. git merge master
65.
66. # Pour publier ses modifs, on commence par récupérer les
67. # nouveautés comme ci-dessus...
68. # Puis on merge la branche locale sur le master
69. git checkout master
70. git merge local
71. # Et on envoie nos commits sur le SVN, chaque commit git va
72. # créer un commit svn correspondant
73. git svn dcommit
74.
75. # On peut ensuite travailler normalement avec git, sauf qu'au lieu de faire des
76. # "git push", on fera
77. git svn dcommit
```

[Télécharger](#)

Voir en ligne : <https://gist.github.com/bystrano/5755464>