

Lab 0 : Starter lab

Ajouter Cygwin à VSCode

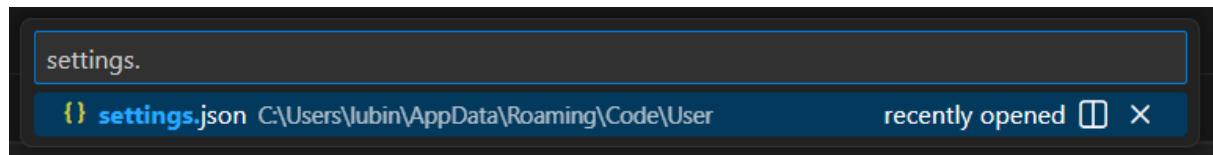
Objectif de l'étape : Permettre l'utilisation de commandes Bash (nécessaires pour le cours) directement dans l'éditeur VS Code sur un système Windows.

Protocole suivi :

Ouvrir VSCode.

Dans la barre de recherche.

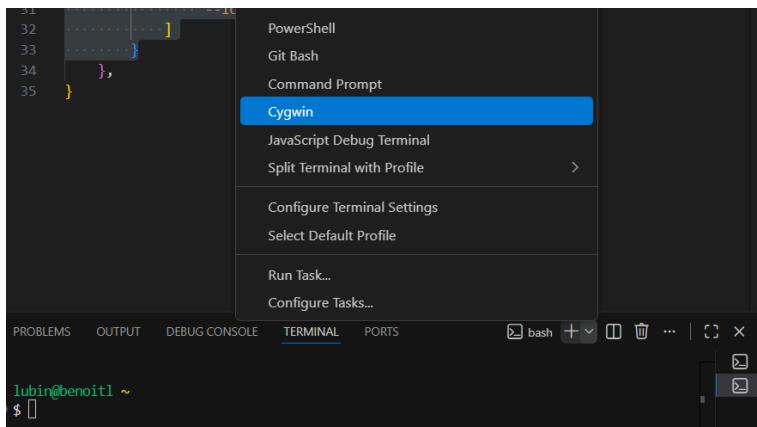
Rechercher et sélectionner settings.json.



Ajouter une configuration Cygwin dans la section terminal.integrated.profiles.windows.

```
"terminal.integrated.profiles.windows": {  
    "PowerShell": {  
        "source": "PowerShell",  
        "icon": "terminal-powershell"  
    },  
    "Command Prompt": {  
        "path": [  
            "${env:windir}\\\\Sysnative\\\\cmd.exe",  
            "${env:windir}\\\\System32\\\\cmd.exe"  
        ],  
        "args": [],  
        "icon": "terminal-cmd"  
    },  
    "Git Bash": {  
        "source": "Git Bash"  
    },  
    "Cygwin": {  
        "path": "C:\\cygwin64\\bin\\bash.exe",  
        "args": [  
            "--login"  
        ]  
    },  
},  
}
```

Lancer le terminal Cygwin via le menu déroulant du terminal intégré à VS Code.



Explication : L'invite de commande Windows ne supporte pas nativement les commandes Bash. Donc on a configuré VS Code pour utiliser l'exécutable Cygwin comme shell alternatif. Il nous permettra dans la suite des labs d'avoir un environnement proche de celui de Linux pour exécuter des commandes et des scripts Bash.

Configuration de Node.js pour l'environnement Cygwin

Objectif de l'étape : Rendre les outils Node.js et npm installés sur Windows accessibles et fonctionnels depuis le noyau Cygwin.

Protocole suivi :

Installer Node.js.

Démarrer un terminal PowerShell et vérifier que Node.js et npm sont présents.

```
● PS C:\Users\lubin> node -v
v25.2.1
● PS C:\Users\lubin> npm -v
11.6.2
```

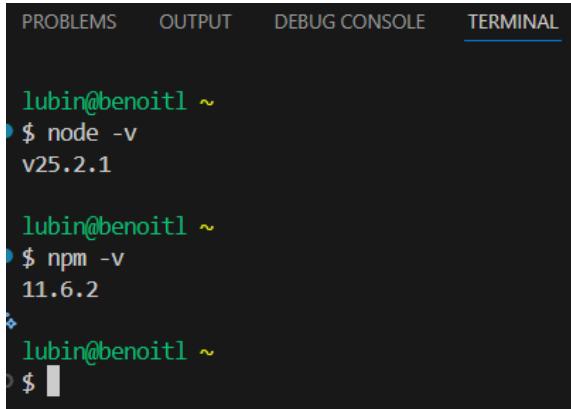
Dans C:/cygwin64/home/<user>/, modifier le fichier .bash_profile afin de définir les variables d'environnement node et npm dans le profil utilisateur Cygwin.

The screenshot shows the VS Code interface with the terminal window active. The tab bar at the top has three tabs: 'Welcome', 'settings.json', and '\$.bash_profile'. The terminal window displays the contents of the .bash_profile file:

```
C: > cygwin64 > home > lubin > $ .bash_profile
42  # if [ -d "${HOME}/info" ]; then
43  #   INFOPATH="${HOME}/info:${INFOPATH}"
44  # fi
45  # write at the end : (or echo "" >> ~/.bash_profile)
46  alias npm="/cygdrive/c/Program Files/nodejs/npm.cmd"
47  alias node="/cygdrive/c/Program Files/nodejs/node.exe"
48
```

Recharger la configuration en exécutant source ~/.bash_profile.

Vérifier la présence de node et npm dans l'environnement Cygwin sur le terminal VS Code.



The screenshot shows the VS Code interface with the 'TERMINAL' tab selected. The terminal window displays the following command-line session:

```
lubin@benoitl ~
$ node -v
v25.2.1

lubin@benoitl ~
$ npm -v
11.6.2

lubin@benoitl ~
$
```

Explication : Nous avons établi un lien via des alias entre l'environnement linux simulé de Cygwin et le chemin réel des exécutables Windows, accessible via /cygdrive/c/.

Résumé (ce que nous avons retenu de ce lab) :

Nous avons ajouté un profil Cygwin dans settings.json de VSCode afin de disposer d'un terminal compatible Unix sur Windows. Ensuite, nous avons configuré des alias dans le fichier ~/.bash_profile pour lier les outils Node.js et npm à ce terminal, créant ainsi un environnement de développement complet pour débuter les laboratoires.