



Documentation TerriSTORY®
docs.terristory.fr

Debugger avec VSCode

Debugger avec VSCode

Utiliser `print()` ou `console.log()` peut fonctionner pour debugger son code, mais utiliser les fonctionnalités de debuggage de son IDE s'avère souvent plus efficace. Ces fonctionnalités permettent entre autre de définir des points d'arrêt (breakpoints) dans le code à partir desquels on peut exécuter le code ligne par ligne, inspecter les variables etc.

Pour VSCode, l'introduction à l'utilisation du mode debug peut se retrouver [ici](#). Il y a notamment quelques librairies et extension à installer.

Des fonctionnalités similaires existent dans la plupart des IDE.

Débugger le back

La config `lauch.json` pour lancer le serveur back en mode debug peut ressembler à :

```
{
  "version": "0.2.0",
  "configurations": [
    {
      "name": "Run TerriAPI Server",
      "type": "debugpy",
      "request": "launch",
      "program": "/path/to/terristory/terristory/.env/bin/terriapi-serve",
      "args": ["--cors", "--reload"],
      "console": "integratedTerminal",
      "cwd": "${workspaceFolder}/terriapi"
    }
  ]
}
```

Il faut adapter le chemin vers `terriapi-serve` et éventuellement le `cwd`.

Débugger le front

VSCode permet également de débbugger le front en s'attachant à un navigateur ouvert. L'utilisation de cette fonctionnalité est détaillée [ici](#).

La config `lauch.json` peut, pour firefox, ressembler à :

```
{
  "version": "0.2.0",
  "configurations": [
    {
      "name": "Run localhost debugger",
      "type": "firefox",
      "request": "attach",
      "url": "http://localhost:3000",
      "webRoot": "${workspaceFolder}"
    }
  ]
}
```