Votre ami le débugger

C’est quoi ?

Un outil génial permettant de :

- stopper l’exécution
- inspecter le contexte

Pourquoi faire ?

- comprendre ce qui se passe à l’exécution

…plutôt que d’ajouter des logs

Je trouve ça où ?

- dans le navigateur
- dans l’IDE

Comment je m’en sers ?

- activer le debugger dans l’application
- placer un point d’arrêt dans le code (breakpoint)
- exécuter le code
  - via un test unitaire
  - manuellement en navigant dans l’application

Qu’est ce que je peux faire avec ?

Mettre le programme en pause

- à la volée
- à une ligne spécifique du programme
- lorsqu’une erreur est déclenchée
- en fonction d’une valeur d’une variable
- en fonction d’un autre breakpoint

Inspecter l’état du programme

- inspecter les valeurs
- inspecter la pile d’appels
- appeler une méthode pour voir son résultat

Comprendre le déroulement d’un algorithme

- dérouler pas à pas
- entrer/sortir d’une fonction
- revenir en arrière

Influer sur le programme

- modifier une valeur
- déclencher des erreurs
- changer le flot d’exécution

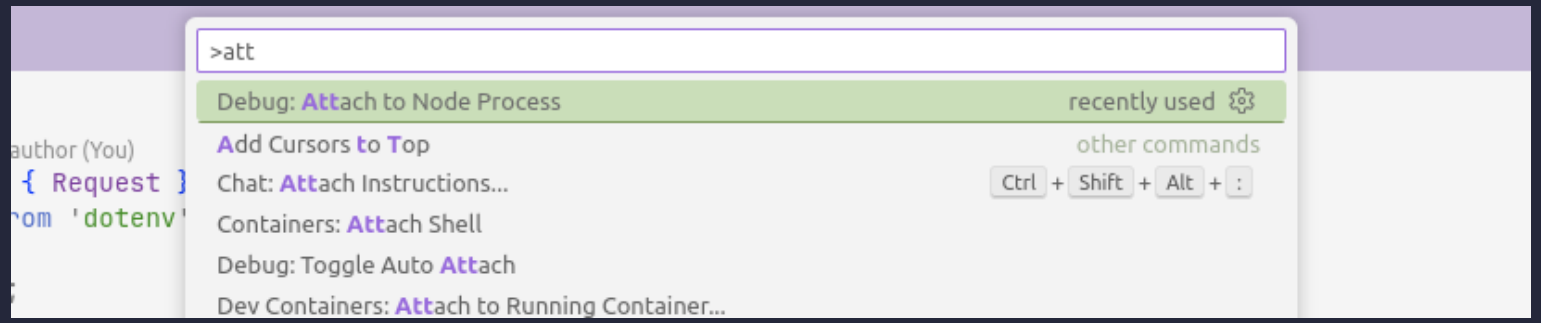
Avec Node.js

Lancer le serveur avec l'option `--inspect` :

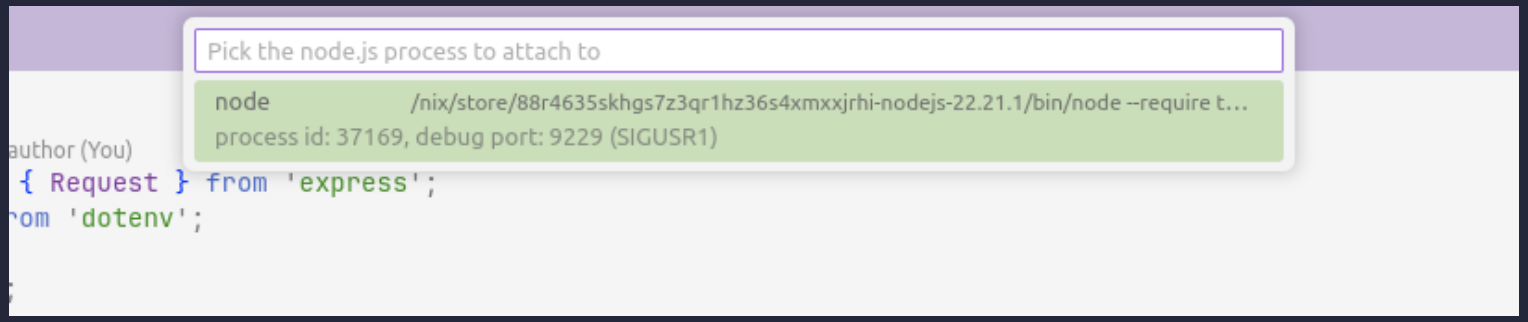
```
{
  "scripts": {
    "start:debug": "node --require ts-node/register --inspect src/index.ts"
  }
}
```

Définir un point d'arrêt

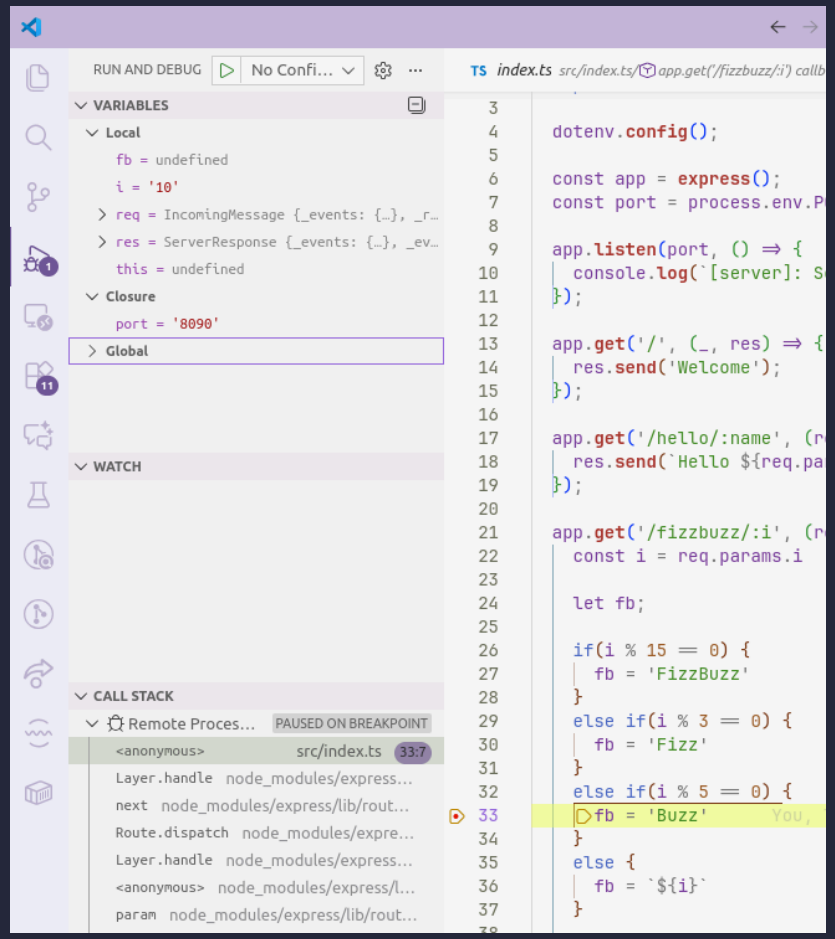
Démarrer le debugger



Sélectionner le serveur



Déclencher le point d'arrêt

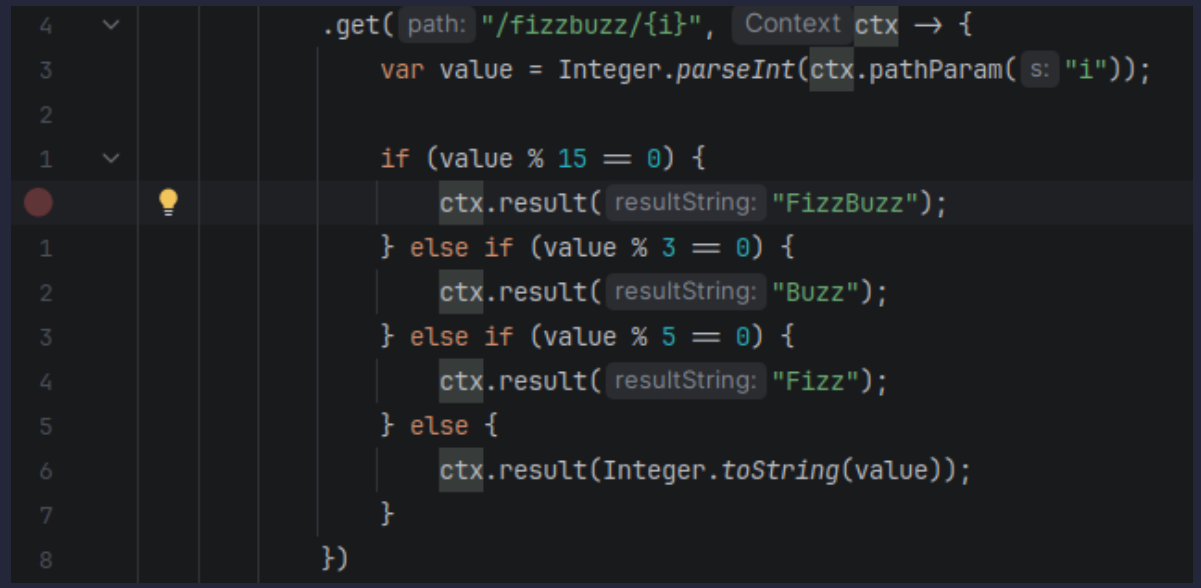


Avec Java

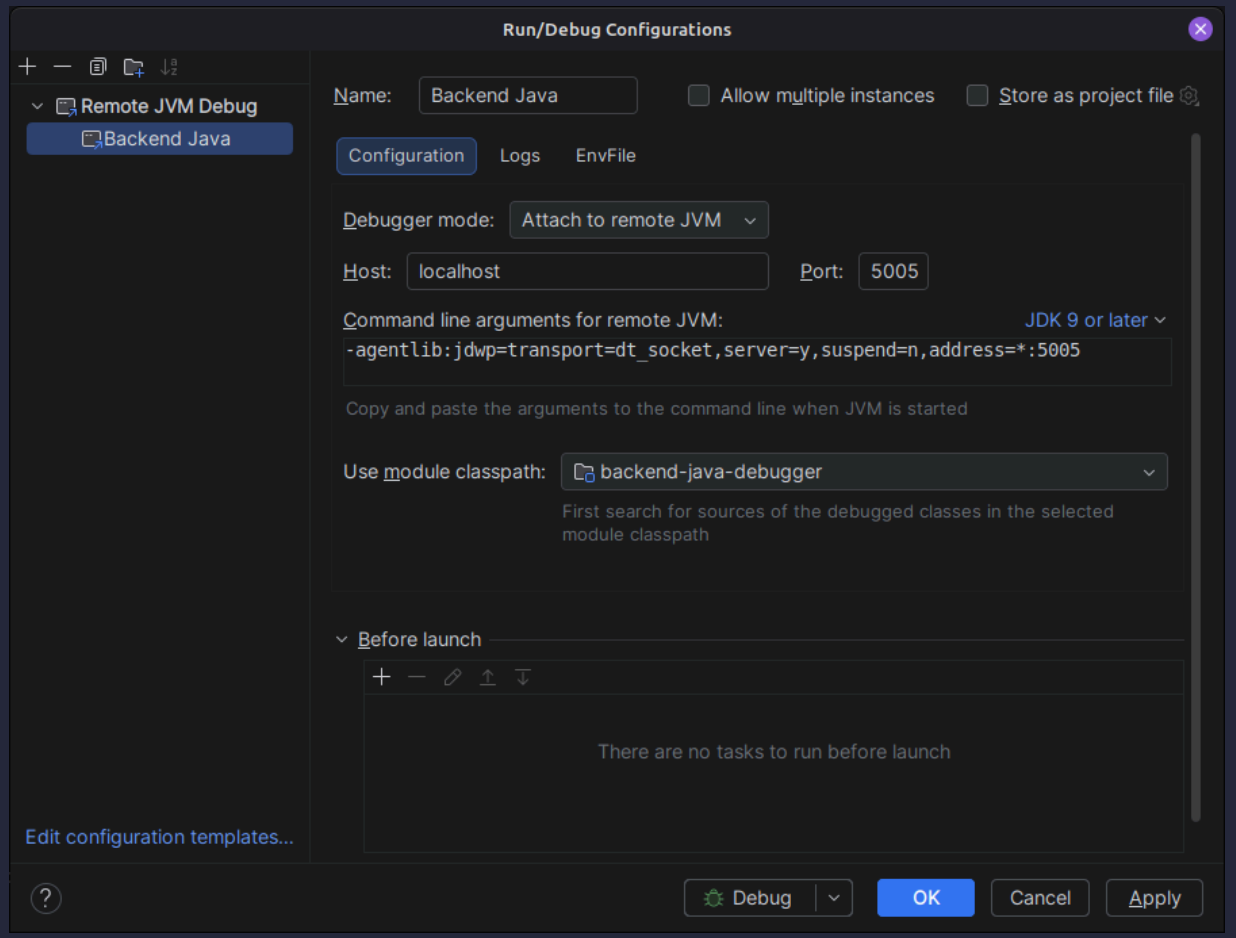
Ajouter l’agent au démarrage de l’application via l’option suivante :

```
-agentlib:jdwp=transport=dt_socket,server=y,suspend=n,address=*:5005
```

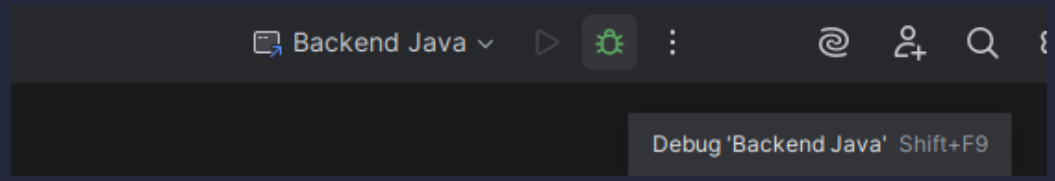
Définir un point d'arrêt



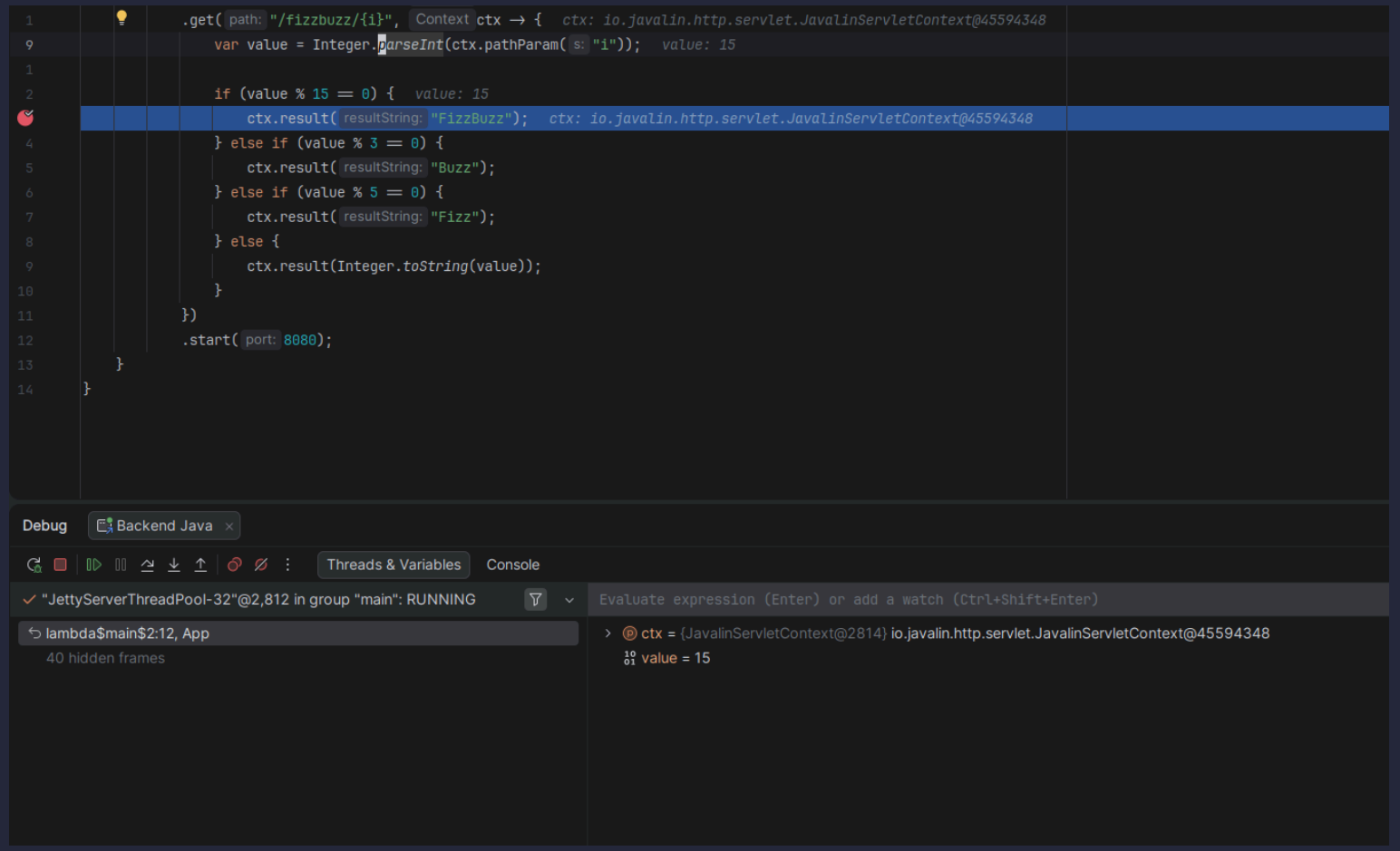
Configurer le debugger



Lancer le debugger



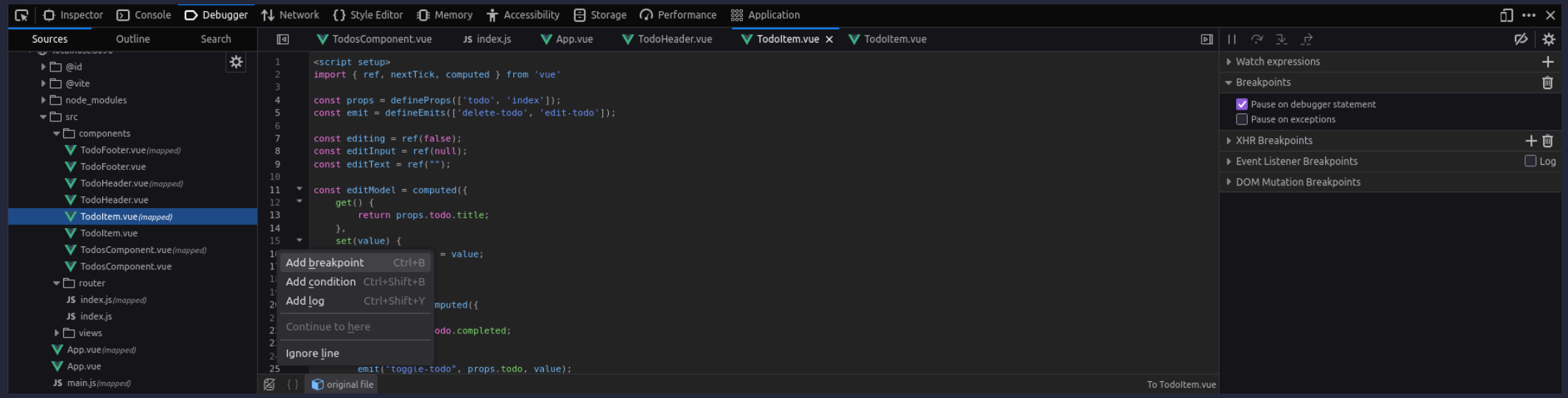
Déclencher le point d'arrêt



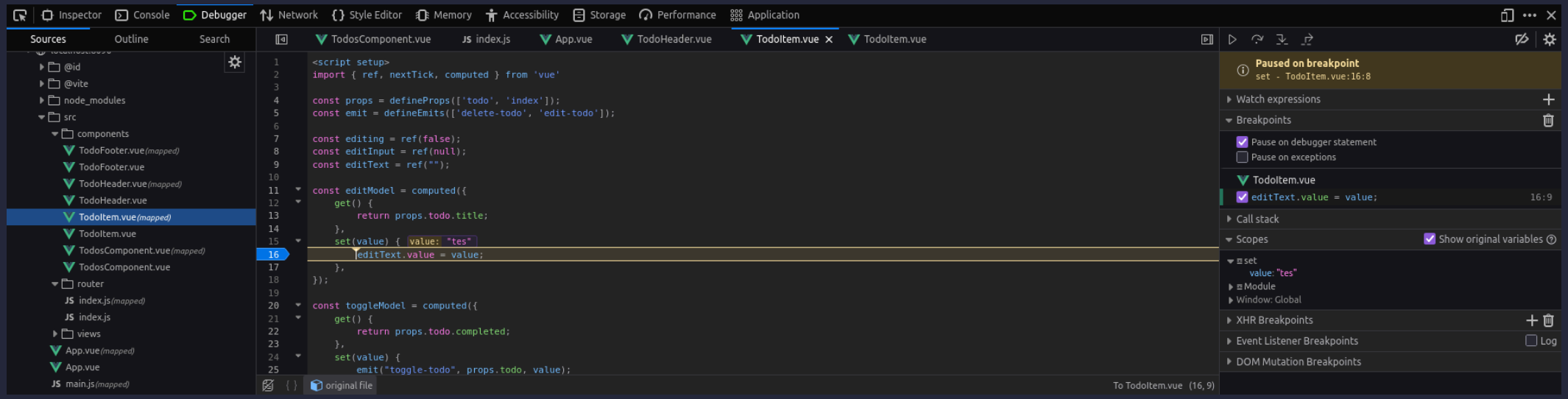
Avec Javascript

Ouvrir les outils développeur : F12

Définir un point d'arrêt



Déclencher le point d'arrêt



Merci de votre attention