

# Atelier: coder avec une IA générative

Utilisation de Large Language Models pour développer un logiciel  
en respectant les standards de l'industrie

Partie 3

Nicolas Debeissat  
[nicolas.debeissat@mail-formateur.net](mailto:nicolas.debeissat@mail-formateur.net)

# Plan du module

## **Installation de l'extension continue.dev**

Dans VSCode, avec ollama en local

## **Utiliser la configuration de base**

Création d'une stack React + FastAPI + Postgresql

## **Configuration avancée et codebase**

Comment adapter ses réponses

## **RAG avec Langchain**

API de génération de requêtes ElasticSearch

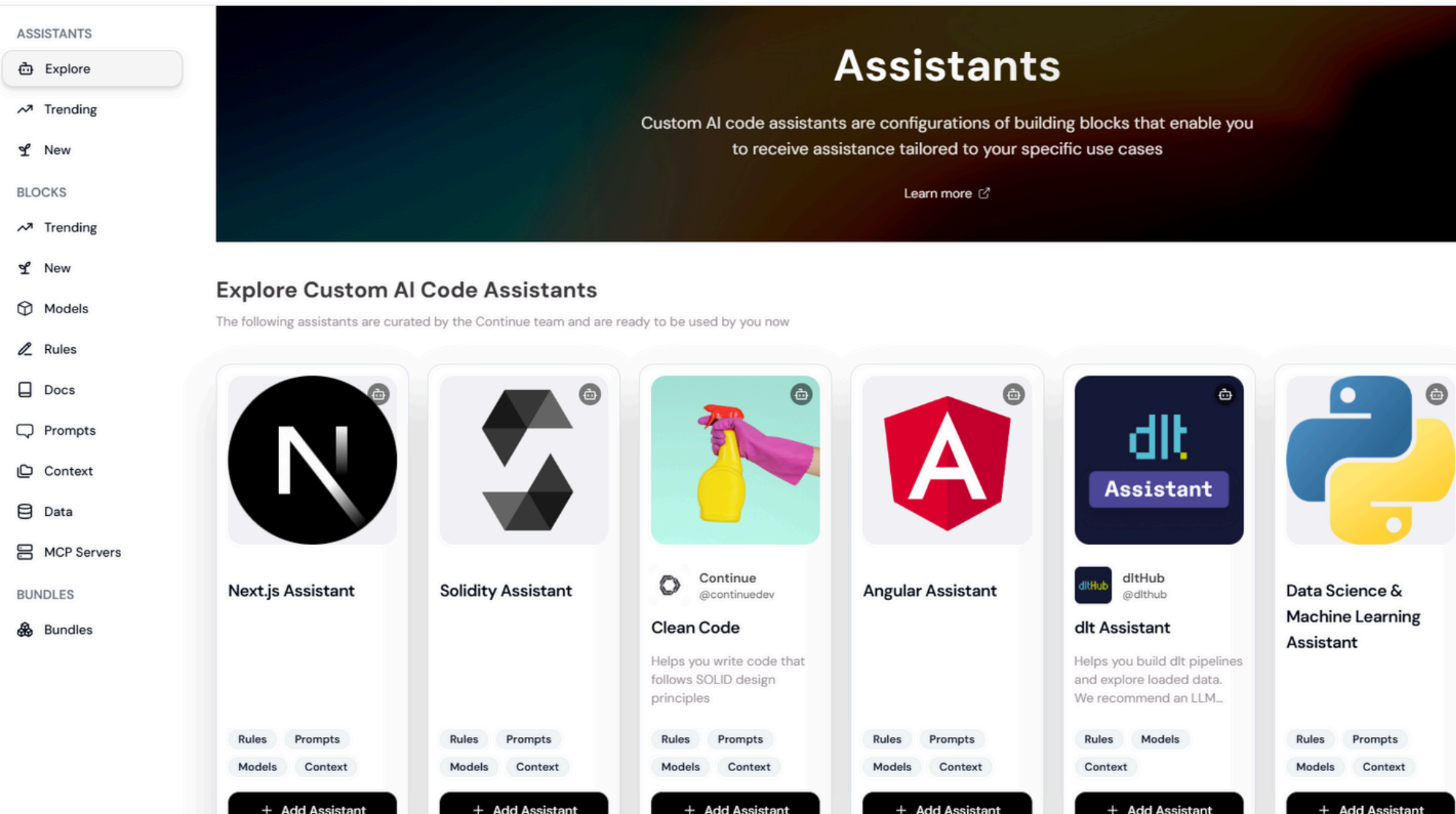
## **Model Context Protocol (mode Agent)**

Création d'outils et connexion à son agent

## **Evaluation**

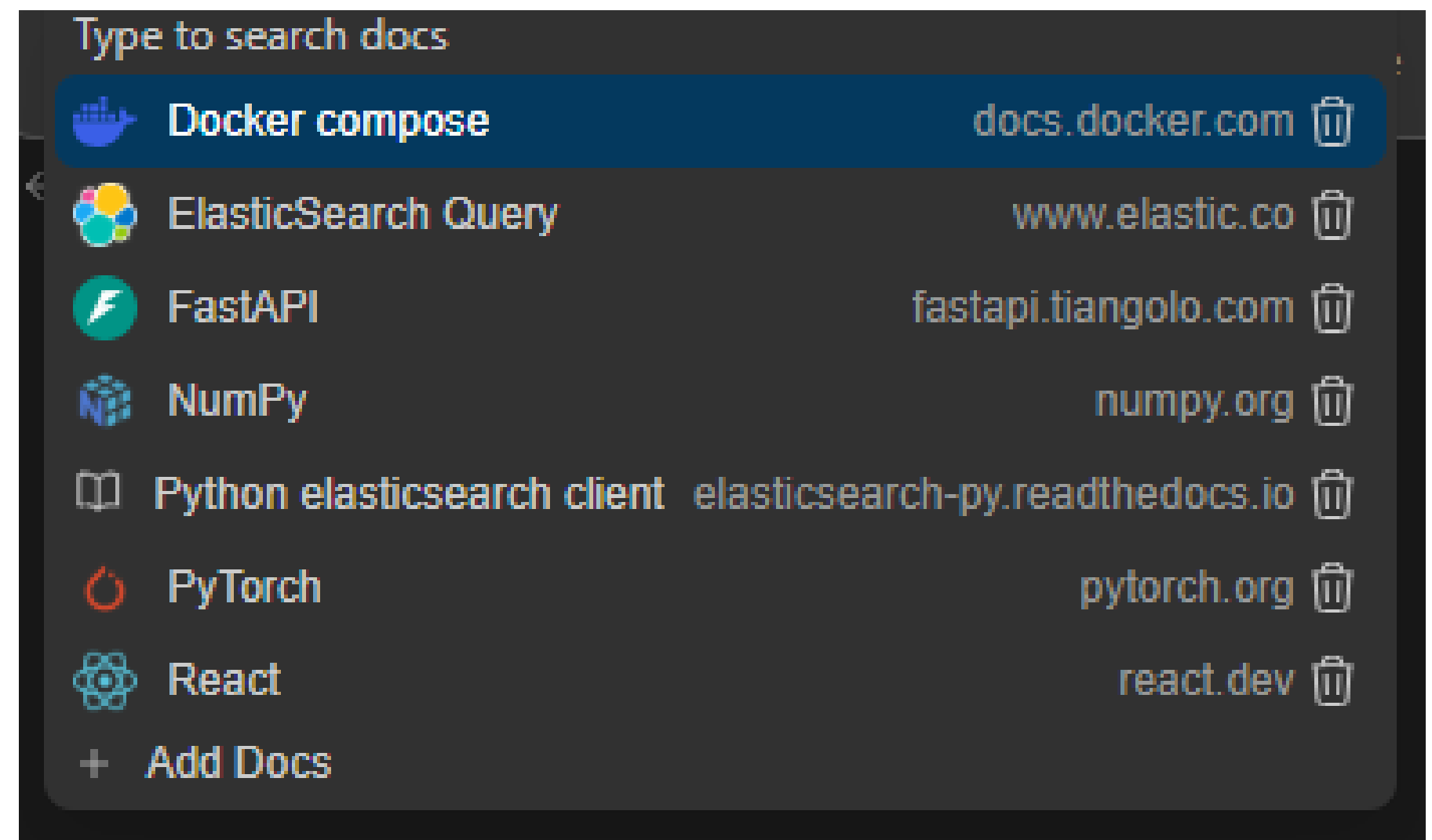
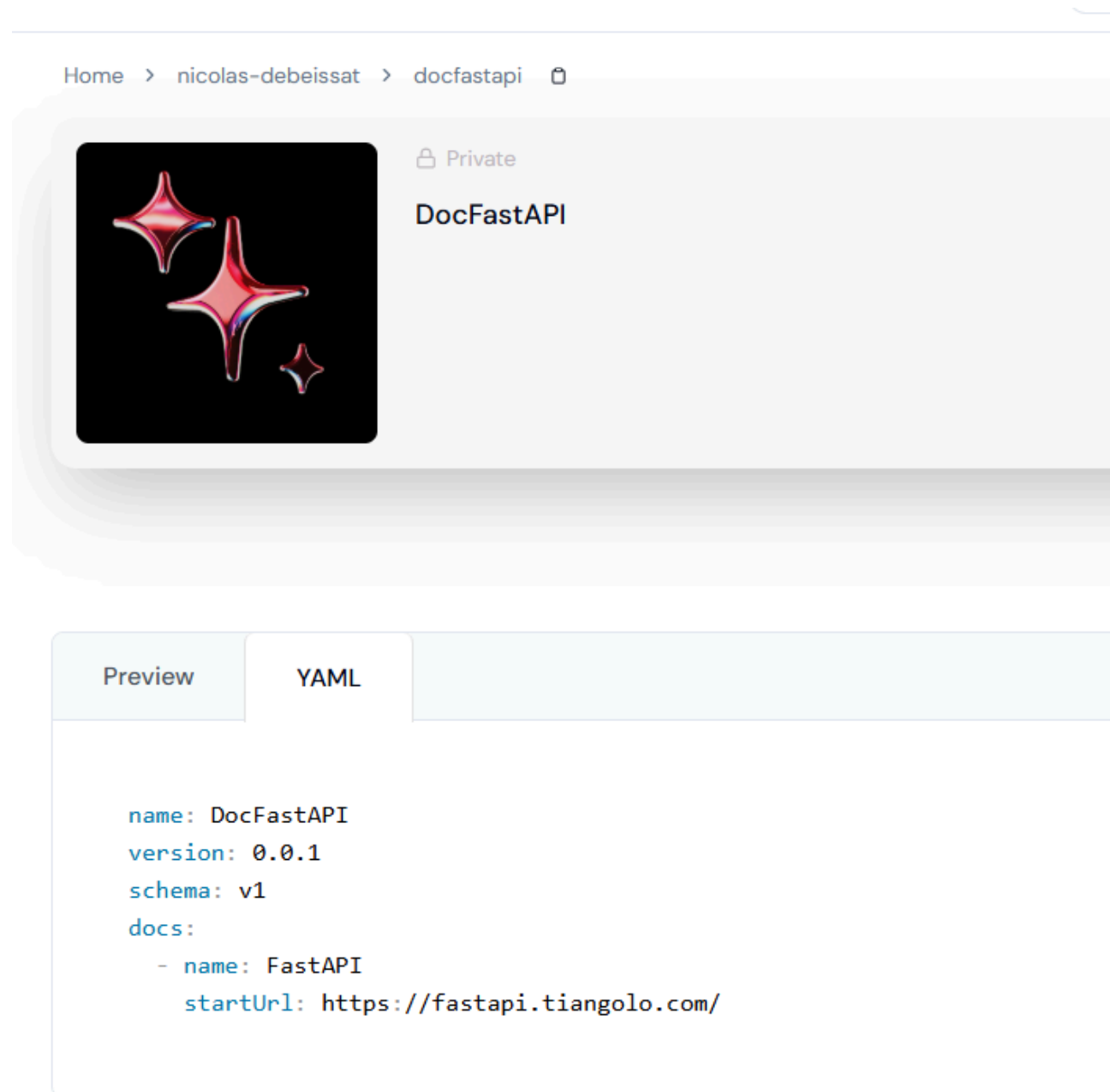
Quizz et rendu du projet effectué en cours

# Configurer un assistant sur <https://hub.continue.dev/explore/context>



- Utilisation d'assistants partagés par la communauté
- Ajout de blocs partagés sur votre assistant
- Changement d'assistant dans le plugin

# Ajouter une source de documentation



# Ajouter des “rules”

Dans les logs ollama :

```
time=2025-03-28T10:36:54.048+01:00 level=DEBUG source=routes.go:1521 msg="chat request" images=0 prompt="
<important_rules>\n Always include the language and file name in the info string when you write code blocks. If you
are editing \"src/main.py\" for example, your code block should start with ```python
src/main.py'.\n</important_rules>\n\nDo not comment the code you generate, only return the code\n###
Instruction:
```

```
name: FastAPIpython
version: 0.0.2
schema: v1
rules:
- >
```


```
## Build & Development Commands
```

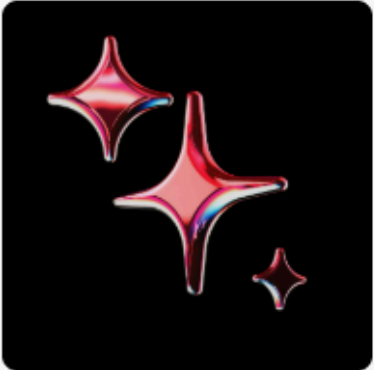
```
This is a Python FastAPI project.
```


```
## Documentation Guidelines
```

```
Adds a docstring at start of each method which explains what the code is
doing
```

# Ajouter des prompts

Home > nicolas-debeissat > prompt-document 



 Private

Document that method with a docstring

Prompts

Preview

YAML

```
name: Document that method with a docstring
version: 0.0.1
schema: v1
prompts:
  - name: d
    description: ""
    prompt: >+
      Adds a docstring at start of the method which explains what the code is
      doing.
```

# Utiliser le contexte

Contexte fichier (ajoute le contenu du fichier au contexte) :

@query.sql write sql query for returning the oldest employee

Contexte documentaire (fais une recherche dans la documentation indexée) :

@React create a function component to display a counter

Contexte codebase (inspire-toi du code du projet) :

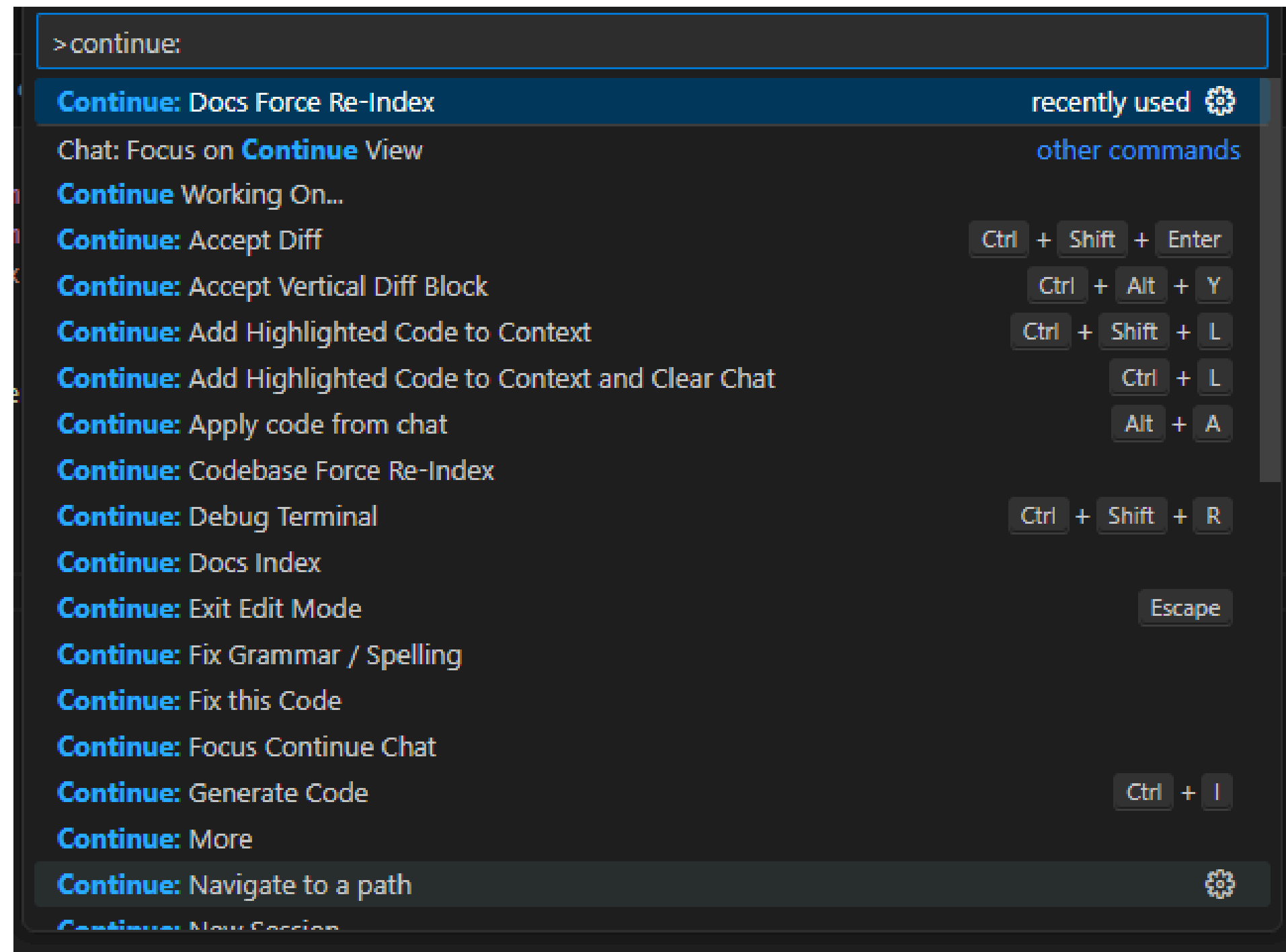
@Codebase write code in order to divide two numbers in php

Contexte terminal (ajoute le contenu du terminal) :

@Terminal correct that error

... et beaucoup d'autres

# Commandes dispo dans Ctrl+Shift+P



# Ouvrir le projet continue

- <https://github.com/continuedev/continue/blob/main/CONTRIBUTING.md>
- Cloner le projet github <https://github.com/continuedev/continue.git>
- Ouvrir ce projet dans VSCode
- dans PowerShell : `Set-ExecutionPolicy -ExecutionPolicy Bypass -Scope CurrentUser`
- Ctrl+Shift+P “Tasks: run task” “Install all dependencies”
- Dans Run and Debug : “Launch Extension”
- Point d’arrêt sur :
  - `return this.createLanceDbRows(chunkMap, embeddings);`
- Forcer la reconstruction de l’index Ctrl+Shift+P :
  - Continue : Rebuild codebase index

# Explorer la base de code sqlite

```
const sqlite3 = require('/home/nico/repos/continue/core/node_modules/sqlite3');
const db = new sqlite3.Database('/home/nico/.continue/index/index.sqlite')
db.all("select * from sqlite_master where type='table'", [], (err, tables) => {
  tables.forEach(table => {
    console.log(table.name)
  });
})
db.all("select * from chunks where path like '%.js' limit 10", [], (err, chunks) => {
  chunks.forEach(chunk => {
    console.log(chunk)
  });
})
```

# Explorer la base de code lancedb

```
const lancedb = require('/home/nico/repos/continue/core/node_modules/vectordb');
lancedb.connect('/home/nico/.continue/index/lancedb/').then(db => {
  db.tableNames().then(tableNames => {
    tableNames.forEach(tableName => {
      console.log(tableName);
    })
  })
  db.openTable("docsOllamaEmbeddingsProvider__nomic-embed-text_latest").then(tbl => {
    tbl.search().where("starturl = 'https://fr.react.dev/learn'").limit(10).execute().then(rows => {
      rows.forEach(row => {
        console.log(row);
      })
    })
  })
});
});
```