Creating a New Hardhat Project

Here's a step-by-step guide to setting up a new project with Hardhat:

Create a new project directory: Create a new, empty folder for your project. Make sure this folder isn't nested inside another project directory to avoid potential conflicts. For example, create a folder named my-hardhat-project.

Navigate to the project directory: Open your terminal and navigate into the newly created project folder using the cd command. For example: cd my-hardhat-project

Initialize a new npm project: Use the command npm init. This will create a package.json file in your project directory, which will track your project's dependencies. You can press enter to accept the default settings for each prompt.

Install Hardhat: Install Hardhat as a development dependency using the command: npm install --save-dev hardhat

Run Hardhat initialization: Execute npx hardhat. You'll be presented with several project setup options. Choose "Create a JavaScript project" if you prefer to code tests using Javascript or "Create a TypeScript project" for using TypeScript. Also choose to install the example project or not. It is usually best to not install it.

Create Hardhat config file (if not autogenerated): Based on your selection in the previous step a config file called hardhat.config.js or hardhat.config.ts will be created at the root of your project directory. If for some reason one is not automatically created, manually create the file in the root directory of your project.

Test the setup: Run npx hardhat compile to ensure Hardhat can compile contracts. You should see a message indicating either a successful compilation or that there are no contracts to compile. Then, run npx hardhat test to verify your testing environment is set up correctly. This is a good initial check to confirm that the setup is correct.

Troubleshooting

Conflicting Peer Dependency Errors: If you encounter errors related to conflicting peer dependencies, the recommended approach is to start over in a brand new directory. Initialize a new project within this directory and reinstall Hardhat. This can often resolve dependency conflicts that may have arisen from previous installations.

Error H203: This error typically indicates that you're attempting to initialize a Hardhat project within an existing Hardhat project directory. Ensure your project folder is not nested within another project folder and try again.

No tests found: Check that you have a .mocharc.json configuration file, your VS Code is opened in the correct project folder, and the test directory contains your test files (.js or .ts depending on whether you chose a JavaScript or TypeScript project when initializing Hardhat).

By following these steps and troubleshooting tips, you should be able to set up a new Hardhat project successfully. If you continue to have problems, please provide specific error messages or details about where you're getting stuck for more assistance.