Here’s a step-by-step guide on how to **install Angular extensions**, **attach the Visual Studio Code (VS Code) debugger**, use **breakpoints and watches**, and **debug state management** in Angular applications.

**✅ 1. Installing Angular Extension in VS Code**

**Recommended Extensions:**

* **Angular Essentials (by John Papa)** – Includes Angular Language Service, snippets, and tools.
* **Angular Language Service** – Provides error checking, completions, and navigation.
* **Debugger for Chrome** – For debugging Angular apps in Chrome.

**Steps:**

1. Open **VS Code**.
2. Go to the **Extensions** view (Ctrl + Shift + X or Cmd + Shift + X).
3. Search for:
   * Angular Essentials
   * Angular Language Service
   * Debugger for Chrome
4. Click **Install** on each.

**🐞 2. Attaching the Debugger**

To debug Angular apps in the browser (usually Chrome), you’ll use the **Debugger for Chrome** extension.

**Steps:**

1. Open .vscode/launch.json (create it if it doesn’t exist).
2. Add the following configuration:

{

"version": "0.2.0",

"configurations": [

{

"name": "Debug Angular App",

"type": "chrome",

"request": "launch",

"url": "http://localhost:4200",

"webRoot": "${workspaceFolder}",

"sourceMaps": true,

"sourceMapPathOverrides": {

"webpack:///./\*": "${webRoot}/\*"

}

}

]

}

1. Make sure your app is running with ng serve.
2. Press F5 to start debugging.

**⛔ 3. Setting Breakpoints and Using Watches**

**Breakpoints:**

* Open a .ts file (like a component or service).
* Click to the left of the line number to set a **breakpoint**.
* Once you hit a breakpoint, VS Code pauses execution.

**Watches:**

* Go to the **Run and Debug** panel (left sidebar or Ctrl + Shift + D).
* In the **WATCH** section, click **+** and enter a variable or expression (e.g., this.user, myVar.length > 0).
* It will evaluate the expression during pause/breakpoint.

**🔄 4. Debugging State Management (e.g., NgRx)**

If you're using **NgRx** for state management:

**Tools:**

* Install the **Redux DevTools Extension** in Chrome.
* Use the @ngrx/store-devtools package.

**Steps:**

1. Install store devtools:
2. npm install @ngrx/store-devtools
3. Import in AppModule:
4. import { StoreDevtoolsModule } from '@ngrx/store-devtools';
5. import { environment } from '../environments/environment';
6. @NgModule({
7. imports: [
8. StoreDevtoolsModule.instrument({
9. maxAge: 25,
10. logOnly: environment.production,
11. }),
12. ],
13. })
14. Open **Redux DevTools** in Chrome to inspect dispatched actions, state changes, etc.
15. You can also pause at **effects**, **reducers**, or **selectors** using breakpoints.

**🧠 Tips**

* Use **console.log()** as a quick debug tool before stepping into breakpoints.
* Use **conditional breakpoints** (right-click on the red dot) to pause only when a condition is true.
* Use the **Call Stack** and **Variables** sections in the Debug sidebar to trace logic and view state.

Let me know if you want help setting this up for a specific Angular project or using a specific state management library (like Akita or NGXS).