Essential Steps and Considerations:

1. Environment Setup:

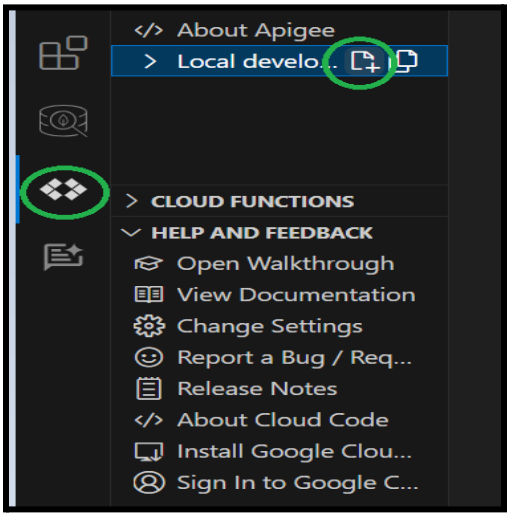
- **Prerequisites:**
    - **JDK 8 or 11:** Install and configure.
    - **Node.js and npm:** Download and install the appropriate version (https://nodejs.org/).
    - **Docker Desktop:** Install and configure.
- **Apigee Cloud Tools Extension:**
    - **VS Code:** Download and install VS Code (https://code.visualstudio.com/).
    - **Install the extension:**
        - Open VS Code, go to the Extensions tab (Ctrl/Cmd+Shift+X).
        - Search for "Apigee Cloud Tools" by GoogleCloudTools.
        - Click "Install" and restart VS Code.
- **Apigee Emulator:**
    - **Download and install:** Refer to Apigee documentation for the latest version (https://cloud.google.com/apigee/docs/api-platform/local-development).

      (https://cloud.google.com/apigee/docs/api-platform/local-development/vscode/manage-apigee-emulator#install).

2. Project Creation and Initialization:

- **Create a new API Proxy:**
    - Open VS Code.
    - Go to the Apigee section (the cloud icon with an umbrella).
    - Click "Local Development".

-Create Workspace:



-same way you can

- Click "Add Environment" and select "Default" or create a new one.
- Click "Add API Proxy" and provide a name, description, and base path.

- **Initialize the Project Directory:**
  - Choose the appropriate option from the "Initialize Project Directory" prompt:
    - **Start from scratch:** Create a new directory structure and files.
    - **Use existing directory:** Use an existing directory with basic Apigee files.
    - **Use existing Git repository:** Use an existing Git repository containing Apigee files.

3. Develop and Deploy:

- **Write and Configure Policies:**
  - Use the Apigee Visual editor or code directly in VS Code for policies in JavaScript (policies.js), XSLT (policies.xsl), or Java (Java policies).

    - Refer to Apigee policy documentation for specific syntax and usage.
- **Run Locally:**
    - Right-click your API Proxy in the Apigee section and select "Deploy to Emulator."
    - Make API calls using a tool like Postman or cURL, specifying the local emulator's host (`http://localhost:8080`) and base path.
    - Debug using Apigee JavaScript Debugger, breakpoints, and logging.
- **Deploy to Apigee Edge:**
    - Make sure you have the required permissions.
    - Click "Deploy to Apigee Edge" or use the `gcloud` command-line tool

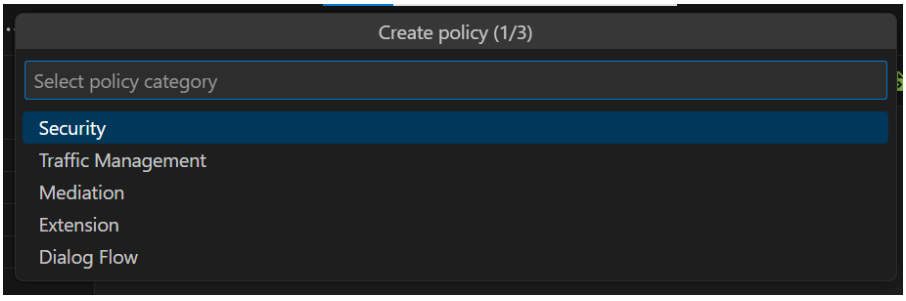
4. Testing and Debugging:

- **Local Testing:**
    - Use postman or cURL to make API calls to your locally deployed API.
    - Verify responses, headers, and error handling.
- **Debugging Locally:**
    - Use Apigee JavaScript Debugger to step through code execution in local policies.
    - Set breakpoints, inspect variables, and log statements.
- **Testing in Apigee Edge:**
    - Deploy your API to Apigee Edge.
    - Use Apigee UI or `gcloud` commands to manage environments and deployments.
    - Configure test settings, mock responses, and assertions.
    - Run tests manually or through CI/CD pipelines.

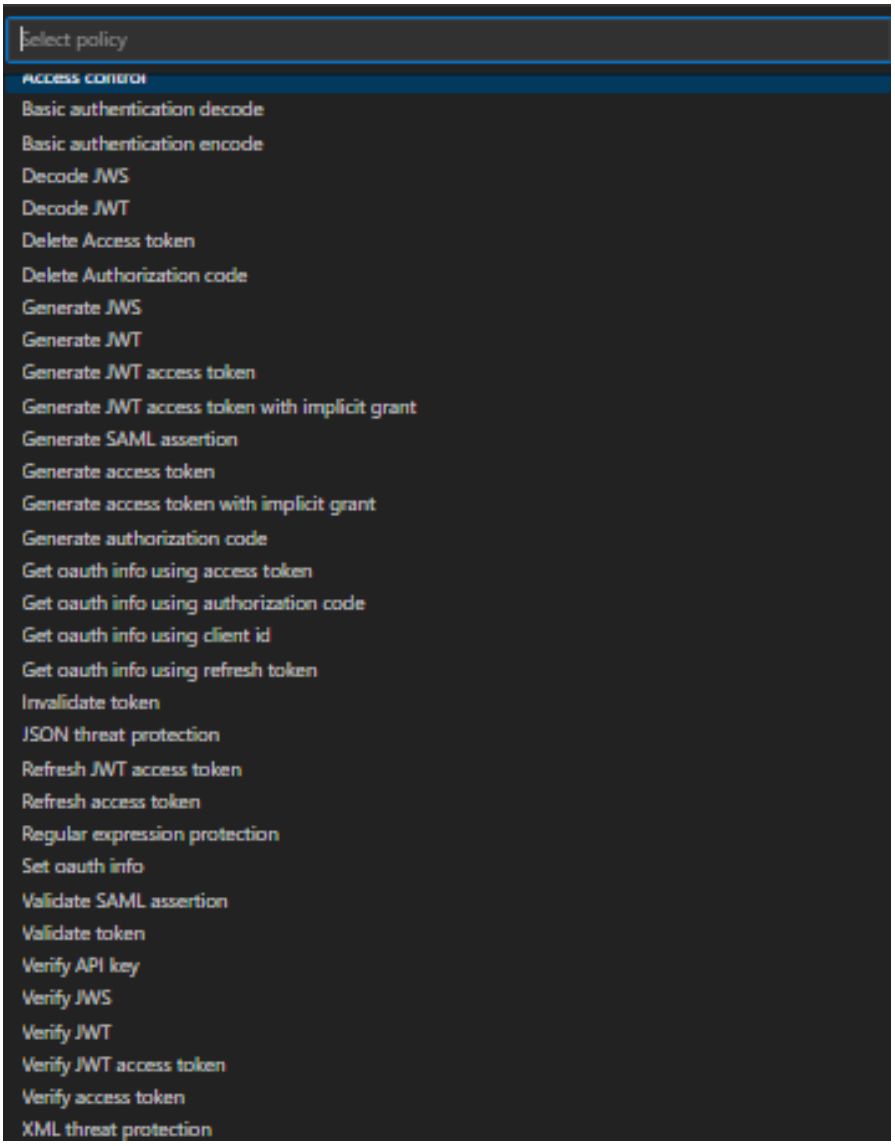
Key Tips and Best Practices:

- **Leverage Local Development Benefits:**
    - Rapidly develop and test locally without needing live Apigee Edge access.
    - Debug more effectively using breakpoints and logging.
    - Iterate faster and save time before deploying to production.
- **Start with Simple APIs:**
    - Familiarize yourself with the local development workflow on small, well-defined APIs before tackling complex ones.
- **Organize Your Project:**
    - Use appropriate directory structures and version control to maintain clarity and track changes.
- **Test Thoroughly:**
    - Cover positive and negative cases, edge scenarios, and potential errors.
    - Test both locally and in Apigee Edge environments.
- **Use Policies Appropriately:**
    - Understand the purpose and behavior of each policy you use.

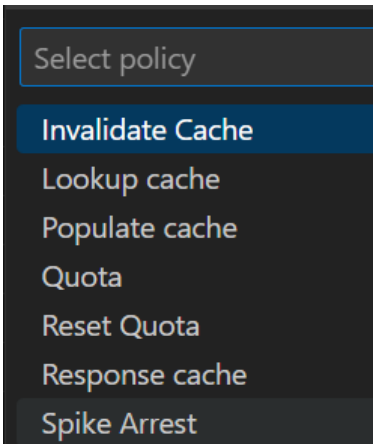
Available policy types:



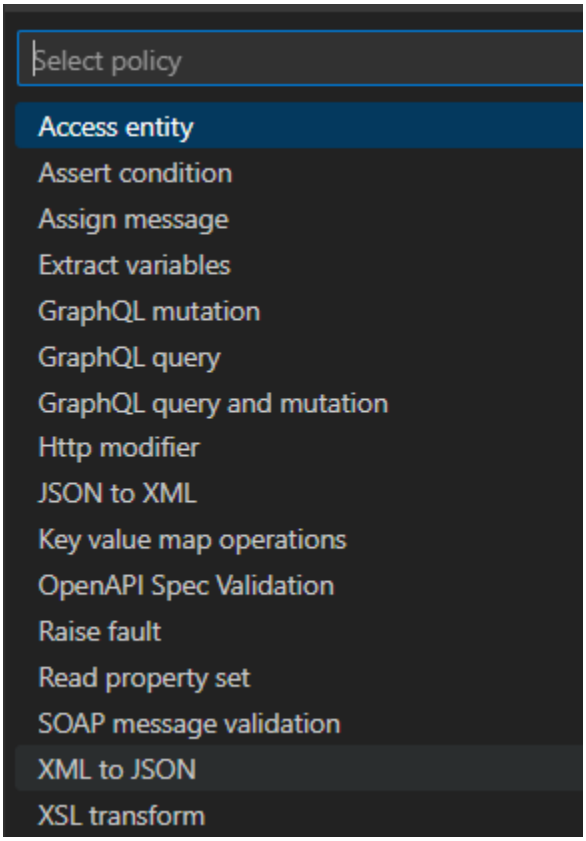
- Security



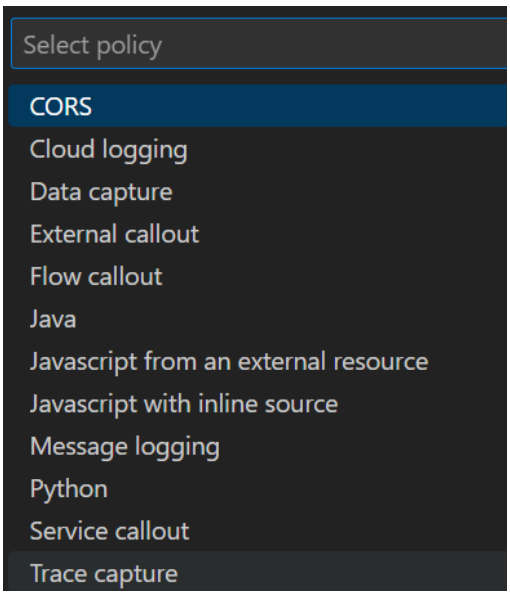
- Traffic Management



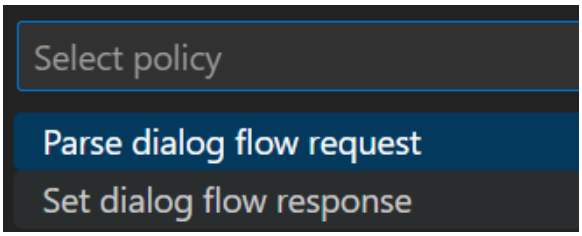
- Mediation



- Extension



- Dialog Flow


Select policy
Parse dialog flow request
Set dialog flow response