

# Basic Tooling



# Agenda

## VS Code

- Basics
- Extensions
- Hands-on
  - Setup
  - Install extensions manually
  - install extensions via command-line

## Postman

- Basics
- Hands-on
  - Setup
  - Create GET request
  - Create POST request
  - Using Variables
  - Exporting Collections & Environment



# VS Code

# VS Code

- Available for macOS, Linux, and Windows
- Sampling of other code editors (in no particular order)
  - Atom, Sublime, Vim, Brackets
- Built-in support for Git
- Public extensibility model that lets developers build and use extensions

# VS Code Extensions

## We're using:

- [ES Lint](#) - Integrates [ESLint](#) into VS Code.
- [npm](#) - Run npm scripts defined in the package.json file and validating the installed modules against the dependencies defined in the package.json.
- [NPM IntelliSense](#) - autocompletes npm modules in import statements
- [DotENV](#) - Adds support for dotenv file syntax
- [EditorConfig](#) - EditorConfig Support for Visual Studio Code
- [GraphQL for VSCode](#) - GraphQL syntax highlighting, linting, auto-complete, and more!
- [Debugger for Chrome](#) - Debug your JavaScript code in the Chrome browser, or any other target that supports the Chrome Debugger protocol.
- [ES7 React/Redux/GraphQL/React-Native snippets](#) - Snippets for React, Redux and GraphQL in JS/TS with ES7 syntax
- [SQL Server \(mssql\)](#) - Develop Microsoft SQL Server, Azure SQL Database and SQL Data Warehouse everywhere
- [TDD Snippets](#) - TDD Snippets for writing tests quickly

# Exercise 1: VS Code

## Goal: Set up VS Code

[Download VSCode](#) and install if you haven't already

- Set Default Shell to bash - for Windows Users only
- Open Command Palette by pressing (Ctrl + Shift + P) or (F1).
- Type and select Terminal: Select Default Shell.
- Choose your preferred terminal (bash) from the list.
- **Note:** In any open terminal you can switch it to another shell by typing a keyword: powershell, bash, cmd, node.



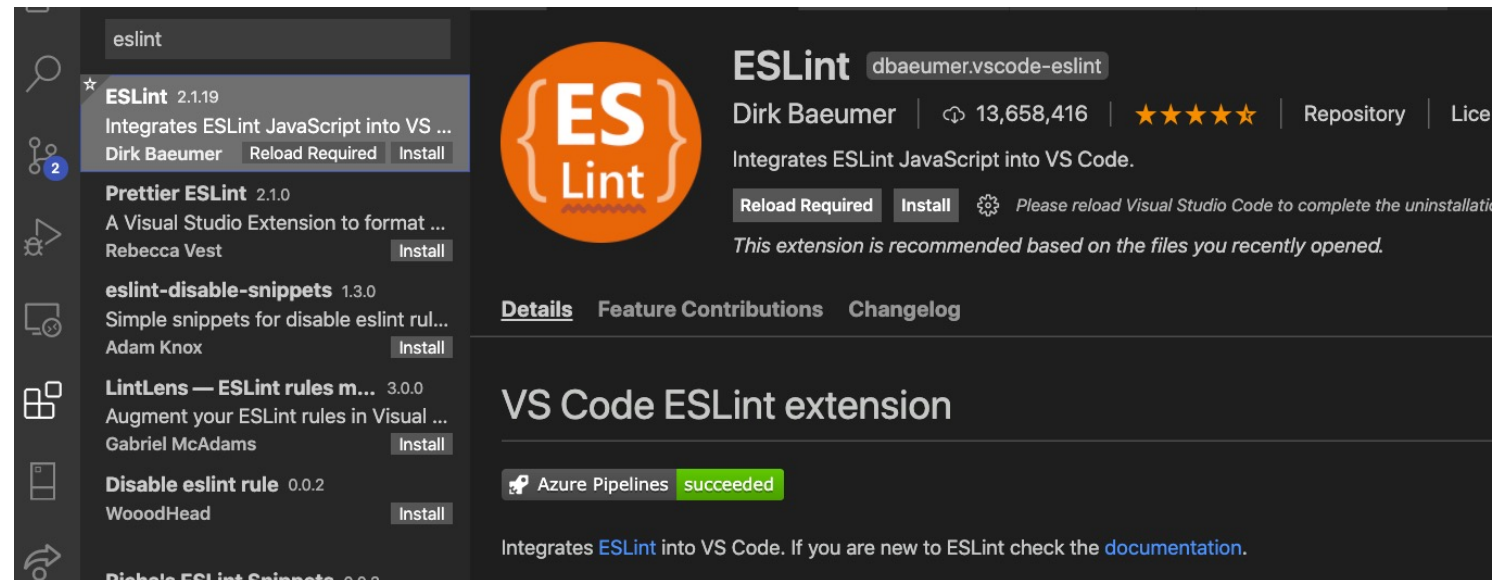
# Exercise 1: VS Code

## Goal: Set up VS Code extensions

### Option #1:

### Manual install

- Click on the *Extensions* tile on VS code's sideNav
- search the extensions (e.g. eslint)
- Click on the *Install* button



# Exercise 1: VS Code

**Goal:** Set up VS Code extensions

Option #2:

## **Install via command-line:**

- Users on macOS must first open the **Command Palette** (⇧⌘P) and type 'shell command' to find the **Shell Command: Install 'code' command in PATH** command. This is to add VS Code executable to the PATH environment variable. Read the [macOS setup guide](#) for help.
- Windows and Linux installations should add the VS Code binaries location to your system path. If this isn't the case, you can manually add the location to the Path environment variable (\$PATH on Linux). For example, on Windows, VS Code is installed under AppData\Local\Programs\Microsoft VS Code\bin. To review platform specific setup instructions, see [Setup](#)



# Exercise 1: VS Code

## Goal: Set up VS Code extensions

### Option #2:

### Install via command-line:

- `code --install-extension dbaeumer.vscode-eslint \ --install-extension eg2.vscode-npm-script \ --install-extension christian-kohler.npm-intellisense \ --install-extension mikestead.dotenv \ --install-extension EditorConfig.EditorConfig \ --install-extension kumar-harsh.graphql-for-vscode \ --install-extension msjsdiag.debugger-for-chrome \ --install-extension dsznajder.es7-react-js-snippets \ --install-extension ms-mssql.mssql \ --install-extension haihua.tdd-snippets`
- *If you are getting errors for the above (something like '--install-extension' is not recognized as an internal or external command operable program or batch file.), try run the command line by line for each extension i.e.:*
  - `code --install-extension dbaeumer.vscode-eslint`
  - `code --install-extension eg2.vscode-npm-script`
  - Etc.

# Postman

# Postman

- API Development
- Sending a request
- Setting up a collection
- Useful to segregate multiple projects
- Easy to export for other team members
- Using variables
- Endpoints usually change between environments
- Also handy for sending authorization tokens
- Exporting a collection

# Exercise 2

## Postman

**Goal:** Set up Postman & Create a new GET request

- [Download Postman](#) and install if you haven't already.
- Open up Postman.
- Click on New -> New Collection and name it *postman exercises*.
- Click on New -> New Request and name it GET /postman-test. Select the *postman exercises* as the collection to save it to.
- In the request URL field, add:
  - `http://postman-echo.com/get?foo1=bar1&foo2=bar2`
- Click SEND to submit the request and review the results

# Exercise 2

## Postman

### **Goal:** Create a new POST request

- Create a new request and name it POST /postman-test. Select the *postman exercises* as the collection to save it to.
- Set the value to the left of the URL to POST. Add the URL:
  - `http://postman-echo.com/post`
- Click the Body tab below the URL.
- Select the raw option.
- Select 'JSON (application/json)' from the dropdown (default is set to Text).
- Add this to the Body:
  - `{ "testObject": { "testProp": "valueA" } }`
- Click SEND to submit the request and review the results.

# Exercise 2

## Postman

### Goal: Using Variables

- Let's define a variable for the host name.
- Click the gear icon under 'Sign-In' in the top right of Postman
- Click the 'Add' button once in the modal window
- Type 'postman-echo-dev' for the environment name.
- Define a new key (variable column) postmanHostUrl with an initial value of: `http://postman-echo.com`.
- Click the 'Add' button once more to complete the setup. Exit the modal.
- Ensure the 'postman-echo-dev' environment is selected (top right dropdown, default may be No Environment).
- Update your requests to use the variable
  - Saved Request: `postman-echo.com/get`
    - `{{postmanHostUrl}}/get?foo1=bar1&foo2=bar2`
  - Saved Request: `postman-echo.com/post`
    - `{{postmanHostUrl}}/post`
- Use SEND to test both the GET and POST requests to prove the variable works.



# Exercise 2

## Postman

### **Goal:** Exporting the Collection & Environment

- In the Collections tab, find your collection.
  - Click on ellipsis at the bottom right of that collection.
  - Choose Export, select the recommended Collection v2.1 option and export the collection to a JSON file
  - Click the gear icon under 'Sign-In' in the top left of Postman
  - Click the download arrow to download the environments to export the environment to a JSON file.
- 
- **Reference material for Postman**
    - [Postman Docs](#)
    - [Postman Videos](#)