EventStoreDB Node.js Environment Installation

Overview

Welcome to the Node.js example of Event Store's **From Scratch** series. This series allows you to quickly overcome the common challenges of setting up and configuring a new development environment, and focus on advancing your EventStoreDB skills.

The **From Scratch** series provides working code examples for basic reads and writes to EventStoreDB, a tested environment to run the code, and instructions that clearly describe the steps required to run the code successfully.

Each **From Scratch** repository provides the following:

- A working Github Codespaces environment
- Instructions on running EventStoreDB locally
- Instructions to set up a similar project on your own

We recommend you progress through the **From Scratch** projects in the following order:

1. Run the code in Codespaces
2. Clone the For Scratch GitHub repo, and follow the instructions to run it locally
3. Build your own project

This document provides detailed instructions for **setting up your own Node.js project in EventStoreDB.** ***This is the recommended third stage in Event Store's From Scratch Node.js series.***

Other clients in the **From Scratch** series include:

- .NET
- Java
- Python

Topics covered

1. Installing Node.js
2. Installing Yarn
3. Initialize a Yarn project
4. Add EventStoreDB dependencies to the project
5. Modify package.json to set the type to "module"
6. Start a local Docker container running EventStoreDB
7. Run some sample code
8. Initialize a local git repository
9. Configure the gitignore file
10. Push to GitHub

This is intended as a baseline working example of an EventStoreDB Node.js project. Your Node.js projects may be significantly more complex.

1. Install Node.js

Instructions for installing Node.js are available at
https://nodejs.org/en/learn/getting-started/how-to-install-nodejs

On a Mac, open a terminal window and run the following command to install Node.js.

```
brew install node
```

Verify that Node.js is installed by running the following command.

```
node -v
```

Note that Node.js has an interactive shell known as a REPL. You can access the shell by typing the following.

```
node
```

2. Install Yarn

The documentation for installing more recent yarn versions is available at
https://yarnpkg.com/getting-started/install.

The following link hosts more classic versions:
https://classic.yarnpkg.com/lang/en/docs/install/

If using a Mac, you could install Yarn and Node using homebrew.

```
brew install yarn
```

3. Initialize a Yarn Project

Create a directory called “myEventStoreDB_project.”, switch to it, and run “yarn init”

```
mkdir my_EventStoreDB_project
```

Navigate to the new directory.

```
cd my_EventStoreDB_project
```

Running `yarn init` will prompt you to answer some questions about the project.

```
yarn init
```

You will be prompted to name your project, set the version number, description, entry point, repository URL, and other questions. If you are creating a practice project, you can hit <return> for each prompt. Once you have completed the questions, yarn will create a package. json file.

The package.json file defines the Node.js project. Future dependencies added to the project will be reflected in the package.json file.

Install jq (Optional)

"jq" is a tool that formats JSON on the command line for easy viewing and filtering. It can be useful when working with JSON. Instructions for installing it are at https://jqlang.github.io/jq/download/.

4. Add EventStoreDB dependencies to the project

To connect to EventStoreDB, you must add dependencies, including EventStoreDB dependencies, to your project directory. Yarn takes care of this with the following terminal commands.

```
yarn add @eventstore/db-client
```

This updates the package.json file and modifies or creates `yarn.lock` in the same directory.

package.json should now contain:

```
"dependencies": {
   "@eventstore/db-client": "^6.1.0"
 }
```

Note: If using an IDE, there should be an interface for managing dependencies.

5. Set type to Module in package.json

Generally, demonstration projects (*and educational services teams*) want code to be as clear as possible. Inline imports help clarify execution requirements.

To have inline import statements in the code, the project type must be set to "module."

Use a text editor of your choice or a vi editor to add a field to your package.json that specifies the “type” : “module”

Once completed, your package.json should look similar to the following.

```
{
 "name": "developers_class_rough_draft",
 "version": "1.0.0",
 "main": "index.js",
 "type": "module",
 "author": "Tom Hanlon",
 "license": "MIT",
 "scripts": {
  "start": "node index.js"
 },
 "dependencies": {
  "@eventstore/db-client": "^6.1.0"
 }
}
```

6. Start a local Docker container running EventStoreDB

If you have yet to do so, install Docker: https://docs.docker.com/engine/install/.

Run the following command to initiate an unsecured single instance EventStoreDB cluster locally.

```
docker run -d --name esdb-node -it -p 2113:2113 -p 1113:1113 \
     eventstore/eventstore:lts --insecure --run-projections=All \
     --enable-external-tcp --enable-atom-pub-over-http
```

Like earlier examples in the From Scratch series, you can view the EventStore WebUI by pointing your browser to http://localhost:2113. Select the “Stream Browser” tab to view the list of streams.

7. Run some sample code

Create a file named “sample_append.js” and add the following code to the file. This code will append an event to the local, unsecured instance of EventStoreDB.

```
import { EventStoreDBClient } from "@eventstore/db-client";
import { jsonEvent } from '@eventstore/db-client';

const client =
EventStoreDBClient.connectionString("esdb://localhost:2113?tls=fal
se");

const stream_data = "Sample Data";

const event = jsonEvent({
  type: "Sample_Type",
  data: {
     stream_data
   },
});

await client.appendToStream(stream_data, event);
client.dispose();
```

Execute the following command to run "sample_append.js."

```
node sample_append.js
```

View the Stream Browser in the EventStoreDB WebUI at http://localhost:2113 to verify a successful append.

8. Initialize a local git repository

Run the following command to create a local git repository.

```
git init
```

This creates a .git folder used to manage the versioning of the directory's content.

9. Create a .gitignore file

Most projects have folders or files required locally but should not be pushed to GitHub.

In this example, the dependencies are included in the node_modules folder. **Do not store this folder in GitHub**. If another user clones or downloads your GitHub repo, the contents of that folder can be generated locally by running `yarn install`.

Create a .gitignore file with the following content. To learn more about this process, please see the GitHub documentation.

```
node_modules
```

That is right, a single line. The format for gitignore is one file (directory is also a file) per line. Creating your gitignore before the first push is best practice,

10. Push to GitHub

Create a repo on GitHub and push your directory as the first commit

Navigate to your GitHub repositories and create a new repo titled "FromScratch_node.js" (or a name of your choosing) by selecting the green "New" button in the upper right of the screen.

After creating the repo, GitHub presents a page with instructions. Follow the instructions titled "**...or push an existing repository from the command line**." You will see two steps below that are not included in GitHub. Ensure you follow the steps below.

Run the following commands to link and push your local Git repo to your GitHub repo. Remember to replace `<YOUR REPO NAME>` with the name of your repository.

```
git remote add https://github.com/<YOUR REPO NAME>.git
```

```
git branch -M main
```

```
git add -A
```

```
git commit -m "first commit"
```

```
git push -u origin main
```

Congratulations! You now have a working Node.js project that includes a basic write to EventStoreDB. This can be the foundation for building more advanced and complete projects.

Next Steps

Now that you have successfully created a Node.js project, you have completed the From Scratch Node.js series. Please feel free to venture into another **From Scratch** series. Event Store offers similar content for Python, Java, and .NET.

Or continue your learning, you can find additional examples in the following repo:

https://github.com/EventStore/samples

In particular, we recommend the Quickstart examples here:

https://github.com/EventStore/samples/tree/main/Quickstart