An Inaugural AI Journey with Langchain and

The Monthly dev: April 30 2024

About Michael Dawson

Node.js lead for Red Hat and IBM

Active Node.js community member

- Node.js Collaborator, Node.js Technical Steering Committee,
- Active in a number of Working group(s)

Active OpenJS Foundation member

- Voting Cross Project Council Member
- Community Director 2020-2022

Twitter: @mhdawson1

GitHub: @mhdawson

Linkedin: https://www.linkedin.com/in/michael-dawson-6051282

Overview

- Why Node.js ?
- My journey
  - Starting as a Newbie
  - Running a model locally
  - Leveraging a GPU
  - Retrieval Augmented Generation
  - Working with different model serving options

Why Node.js ?

- Python often seen as runtime for AI
- But !!!
  - Not all applications will move to Python
  - Emerging AI client libraries often support TypeScript/JavaScript

What does it mean to me as a Node.js Dev

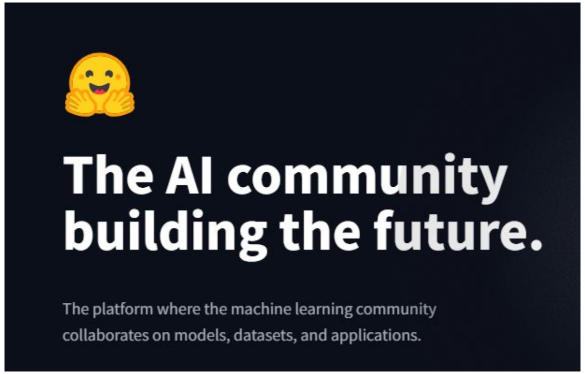
- Until recently
  - https calls to bespoke service
- But, libraries are now emerging
  - langchain.com
  - llamaindex.ai
  -
  - .. add growing list here

Running your first Langchain.js application

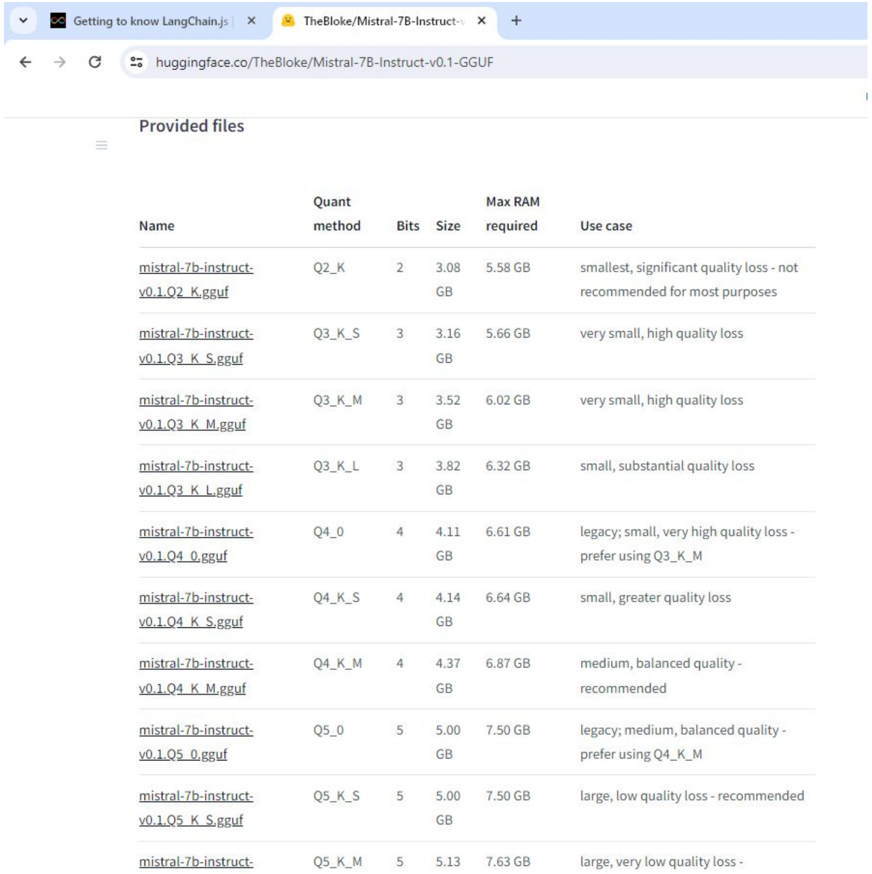
- Need a model or remote API service
  - Need to be cautious with proprietary info
  - Choose to start by running locally

Where do I get a model?

- [HuggingFace]


The AI community building the future.

The platform where the machine learning community collaborates on models, datasets, and applications.



huggingface.co/TheBloke/Mistral-7B-Instruct-v0.1-GGUF

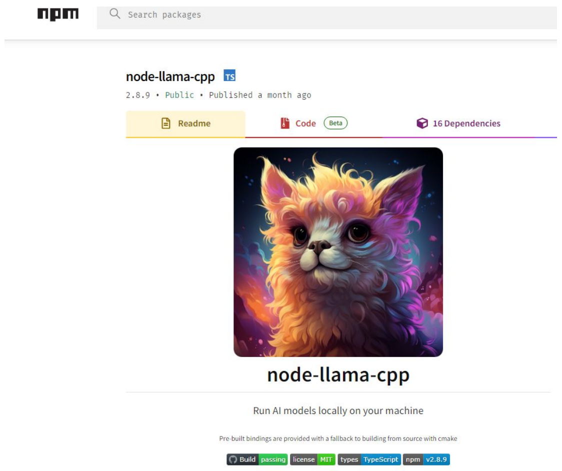
Provided files

Name	Quant method	Bits	Size	Max RAM required	Use case
mistral-7b-instruct-v0.1.Q2_K.gguf	Q2_K	2	3.08 GB	5.58 GB	smallest, significant quality loss - not recommended for most purposes
mistral-7b-instruct-v0.1.Q3_K_S.gguf	Q3_K_S	3	3.16 GB	5.66 GB	very small, high quality loss
mistral-7b-instruct-v0.1.Q3_K_M.gguf	Q3_K_M	3	3.52 GB	6.02 GB	very small, high quality loss
mistral-7b-instruct-v0.1.Q3_K_L.gguf	Q3_K_L	3	3.82 GB	6.32 GB	small, substantial quality loss
mistral-7b-instruct-v0.1.Q4_0.gguf	Q4_0	4	4.11 GB	6.61 GB	legacy; small, very high quality loss - prefer using Q3_K_M
mistral-7b-instruct-v0.1.Q4_K_S.gguf	Q4_K_S	4	4.14 GB	6.64 GB	small, greater quality loss
mistral-7b-instruct-v0.1.Q4_K_M.gguf	Q4_K_M	4	4.37 GB	6.87 GB	medium, balanced quality - recommended
mistral-7b-instruct-v0.1.Q5_0.gguf	Q5_0	5	5.00 GB	7.50 GB	legacy; medium, balanced quality - prefer using Q4_K_M
mistral-7b-instruct-v0.1.Q5_K_S.gguf	Q5_K_S	5	5.00 GB	7.50 GB	large, low quality loss - recommended
mistral-7b-instruct-	Q5_K_M	5	5.13	7.63 GB	large, very low quality loss -

How to I load a model?

- llama.ccp
- ollama
- Hugging Face transformers
- ...

- node-llama-cpp



node-llama-cpp TS

2.8.9 • Public • Published a month ago

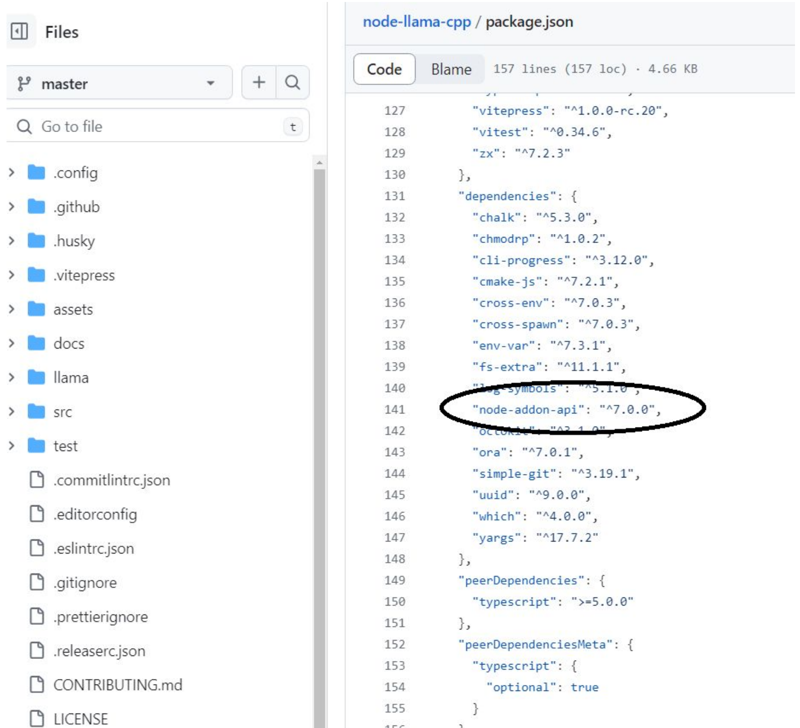
node-llama-cpp

Run AI models locally on your machine

Pre-built bindings are provided with a fallback to building from source with cmake

What’s this under the covers ?

- node-addon-api


Files

master

Go to file

.config
.github
.husky
.vitepress
assets
docs
llama
src
test
.commitlintrc.json
.editorconfig
.eslintrc.json
.gitignore
.prettierignore
.releaserc.json
CONTRIBUTING.md
LICENSE

node-llama-cpp / package.json

Code Blame 157 lines (157 loc) · 4.66 KB

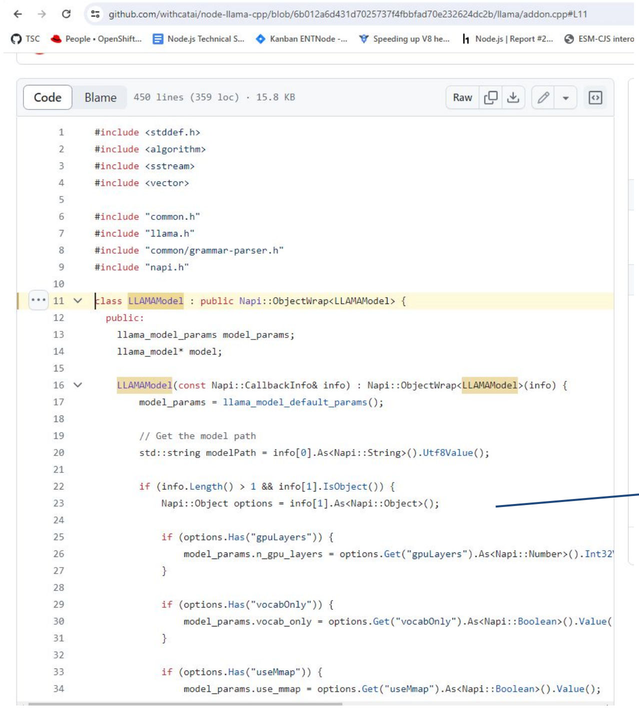
```
    "vitepress": "^1.0.0-rc.20",
    "vitest": "^0.34.6",
    "zx": "^7.2.3"
  },
  "dependencies": {
    "chalk": "^5.3.0",
    "chmodrp": "^1.0.2",
    "cli-progress": "^3.12.0",
    "cmake-js": "^7.2.1",
    "cross-env": "^7.0.3",
    "cross-spawn": "^7.0.3",
    "env-var": "^7.3.1",
    "fs-extra": "^11.1.1",
    "log-symbols": "^5.1.0",
    "node-addon-api": "^7.0.0",
    "octokit": "^3.1.0",
    "ora": "^7.0.1",
    "simple-git": "^3.19.1",
    "uuid": "^9.0.0",
    "which": "^4.0.0",
    "yargs": "^17.7.2"
  },
  "peerDependencies": {
    "typescript": ">=5.0.0"
  },
  "peerDependenciesMeta": {
    "typescript": {
      "optional": true
    }
```

The state of the Node.js core: The Monthly Dev #29

[Building Native addons like its 2023]()

node-addon-api - https://github.com/nodejs/node-addon-api

What’s this under the covers ?



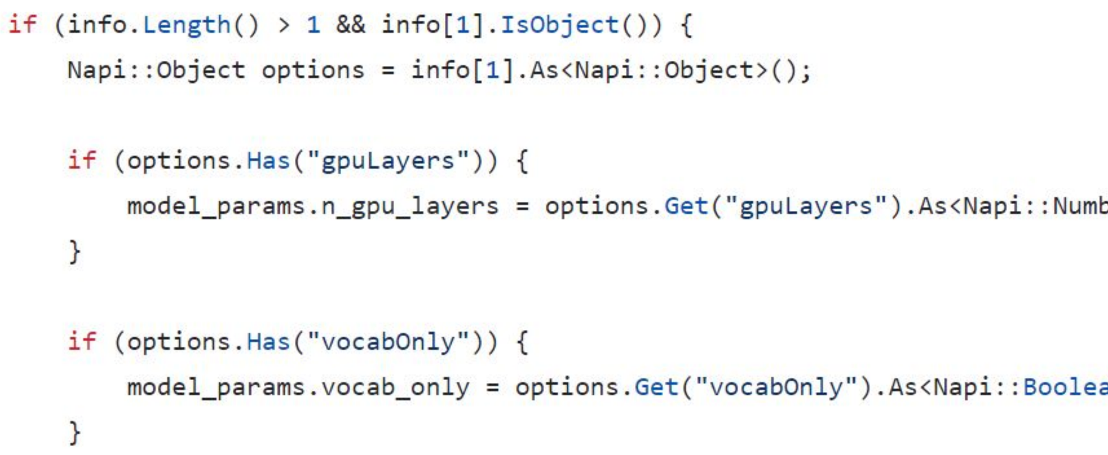
github.com/withcatai/node-llama-cpp/blob/6b012a6d431d7025737f4fbbfad70e232624dc2b/llama/addon.cpp#L11

TSC · People • OpenShift... · Node.js Technical S... · Kanban ENTNode -... · Speeding up V8 he... · Node.js | Report #2... · ESM-CJS intero

Code · Blame · 450 lines (359 loc) · 15.8 KB · Raw

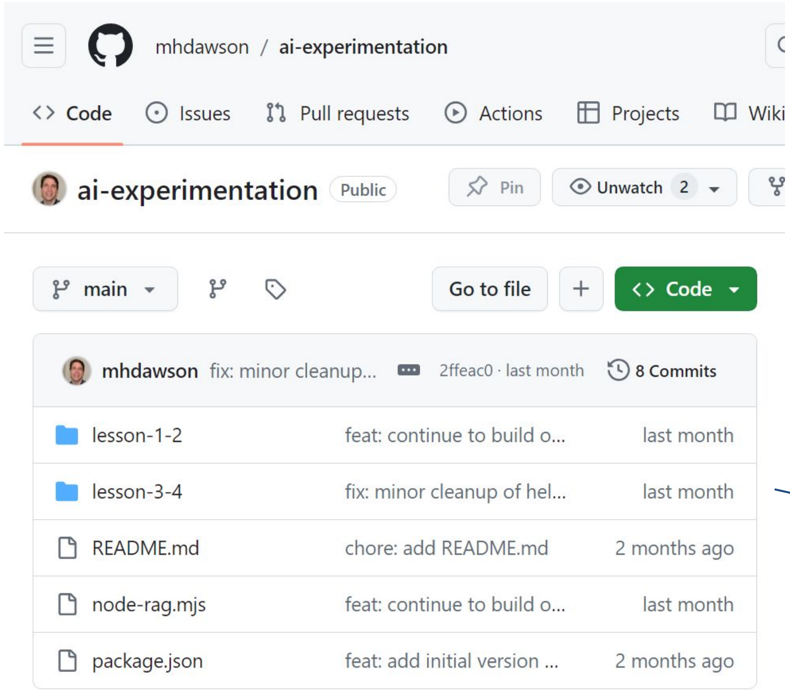
```cpp
#include <stddef.h>
#include <algorithm>
#include <sstream>
#include <vector>

#include "common.h"
#include "llama.h"
#include "common/grammar-parser.h"
#include "napi.h"

class LLAMAModel : public Napi::ObjectWrap<LLAMAModel> {
  public:
    llama_model_params model_params;
    llama_model* model;

    LLAMAModel(const Napi::CallbackInfo& info) : Napi::ObjectWrap<LLAMAModel>(info) {
        model_params = llama_model_default_params();

        // Get the model path
        std::string modelPath = info[0].As<Napi::String>().Utf8Value();

        if (info.Length() > 1 && info[1].IsObject()) {
            Napi::Object options = info[1].As<Napi::Object>();

            if (options.Has("gpuLayers")) {
                model_params.n_gpu_layers = options.Get("gpuLayers").As<Napi::Number>().Int32
            }

            if (options.Has("vocabOnly")) {
                model_params.vocab_only = options.Get("vocabOnly").As<Napi::Boolean>().Value(
            }

            if (options.Has("useMmap")) {
                model_params.use_mmap = options.Get("useMmap").As<Napi::Boolean>().Value();
```

https://github.com/withcatai/node-llama-cpp/blob/6b012a6d431d7025737f4fbbfad70e232624dc2b/llama/addon.cpp



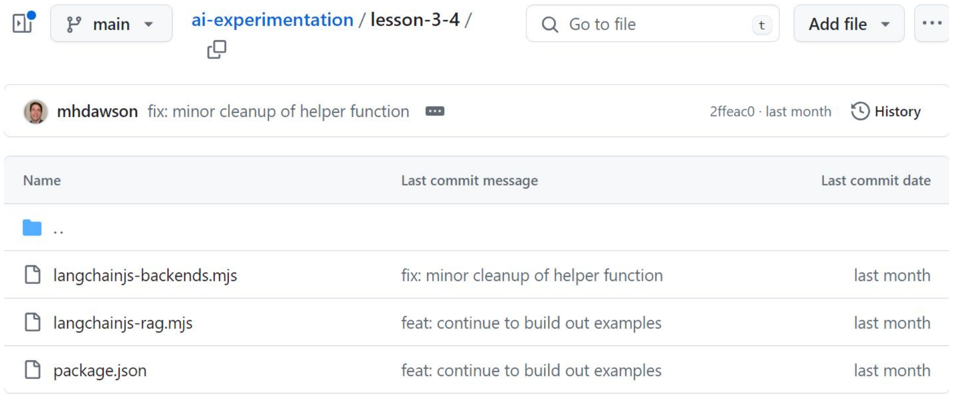
```cpp
if (info.Length() > 1 && info[1].IsObject()) {
    Napi::Object options = info[1].As<Napi::Object>();

    if (options.Has("gpuLayers")) {
        model_params.n_gpu_layers = options.Get("gpuLayers").As<Napi::Numb
    }

    if (options.Has("vocabOnly")) {
        model_params.vocab_only = options.Get("vocabOnly").As<Napi::Boolea
    }
```

Where's the code ?



mhdawson fix: minor cleanup... 2ffeac0 · last month 8 Commits

lesson-1-2	feat: continue to build o...	last month
lesson-3-4	fix: minor cleanup of hel...	last month
README.md	chore: add README.md	2 months ago
node-rag.mjs	feat: continue to build o...	last month
package.json	feat: add initial version ...	2 months ago

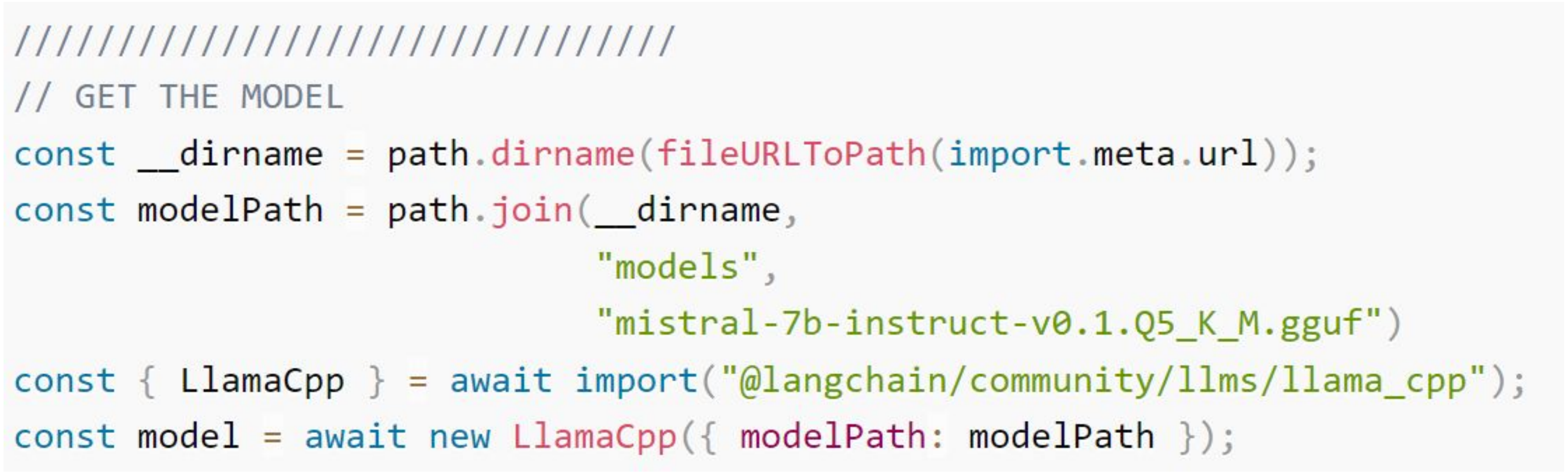


mhdawson fix: minor cleanup of helper function 2ffeac0 · last month History

Name	Last commit message	Last commit date
..		
langchainjs-backends.mjs	fix: minor cleanup of helper function	last month
langchainjs-rag.mjs	feat: continue to build out examples	last month
package.json	feat: continue to build out examples	last month

https://github.com/mhdawson/ai-experimentation

Loading the model in Langchain.js

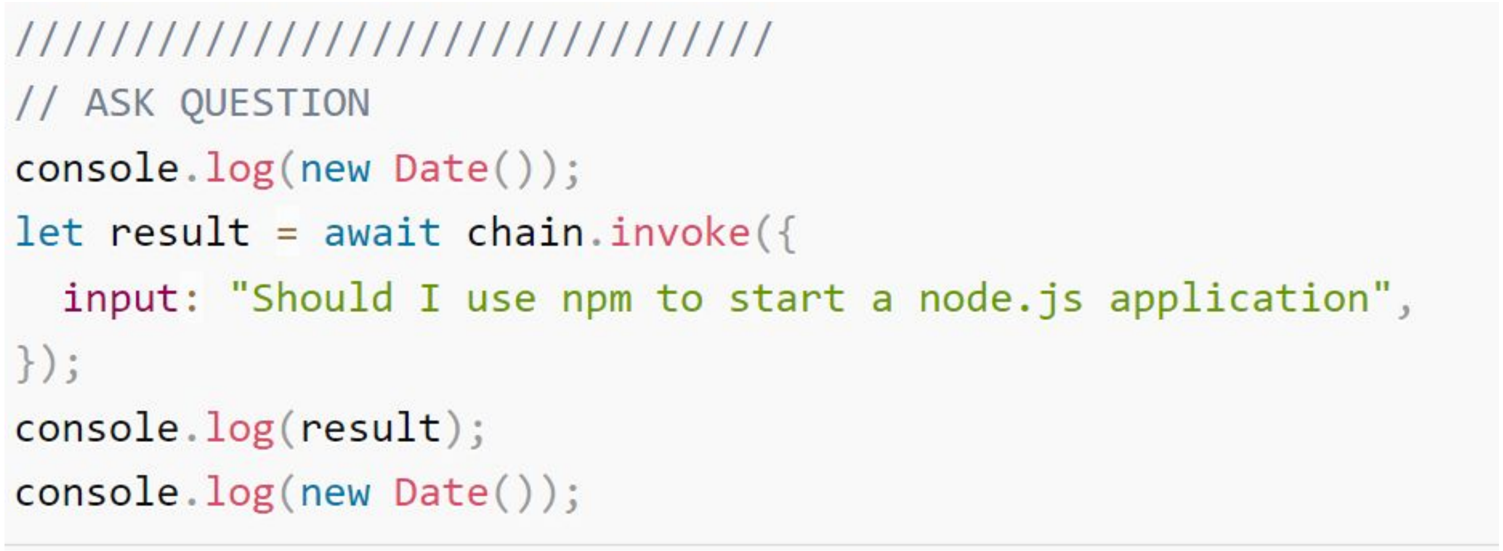


```
////////////////////////////////
// GET THE MODEL
const __dirname = path.dirname(fileURLToPath(import.meta.url));
const modelPath = path.join(__dirname,
                            "models",
                            "mistral-7b-instruct-v0.1.Q5_K_M.gguf")
const { LlamaCpp } = await import("@langchain/community/llms/llama_cpp");
const model = await new LlamaCpp({ modelPath: modelPath });
```

Asking my first question - create a chain

```
////////////////////////////////
// CREATE CHAIN
const prompt =
  ChatPromptTemplate.fromTemplate(`Answer the following question if you don't
know the answer say so:
Question: {input}`);
const chain = prompt.pipe(model);
```

Asking my first question - ask a question



```
////////////////////////////////
// ASK QUESTION
console.log(new Date());
let result = await chain.invoke({
  input: "Should I use npm to start a node.js application",
});
console.log(result);
console.log(new Date());
```

Asking my first question - 25 seconds later….

```
2024-03-11T22:08:23.372Z
Assistant: Yes, you should use npm to start a Node.js application. NPM (Node Package Manager) is the
default package manager for Node.js and it provides a centralized repository of packages that can be
used in your applications. It also allows you to manage dependencies between packages and automate
tasks such as testing and deployment. If you are new to Node.js, I would recommend using npm to get
started with your application development.
2024-03-11T22:08:45.774Z
```

Not the answer we want people to get based on the

nodejs-reference-architecture

https://github.com/nodeshift/nodejs-reference-architecture

JSDrops Growing Success Across Organizations -
https://www.youtube.com/watch?v=GncwXJBwcgQ

Hmm, 25 seconds is a bit long

- Good news, node-llama-cpp supports GPUs
  - enabled by default for MacOS (non intel)
  - easy to enable for Windows

Turning on the GPU windows/NVIDIA

- install the CUDA toolkit (version 12.x or higher).
- install the C/C++ compiler for your platform, including support for CMake and CMake.js.
- npx --no node-llama-cpp download --cuda

Turning on the GPU

25 → 3 Seconds

NVIDIA 4060Ti 16G

Fast but still wrong answer!

- Often want to add additional knowledge
  - Building/Training a model is a lot of work
  - Just want to add a bit of specific info

Fast but still wrong answer!

- Prompt Engineering
  - Prompt includes
    - question
    - context
  - For example in chatbot, context can include chat history

https://en.wikipedia.org/wiki/Prompt_engineering

Retrieval Augmented Generation (RAG)

- Ingest documents
- Extract relevant chunks
- Add chunks to prompt context

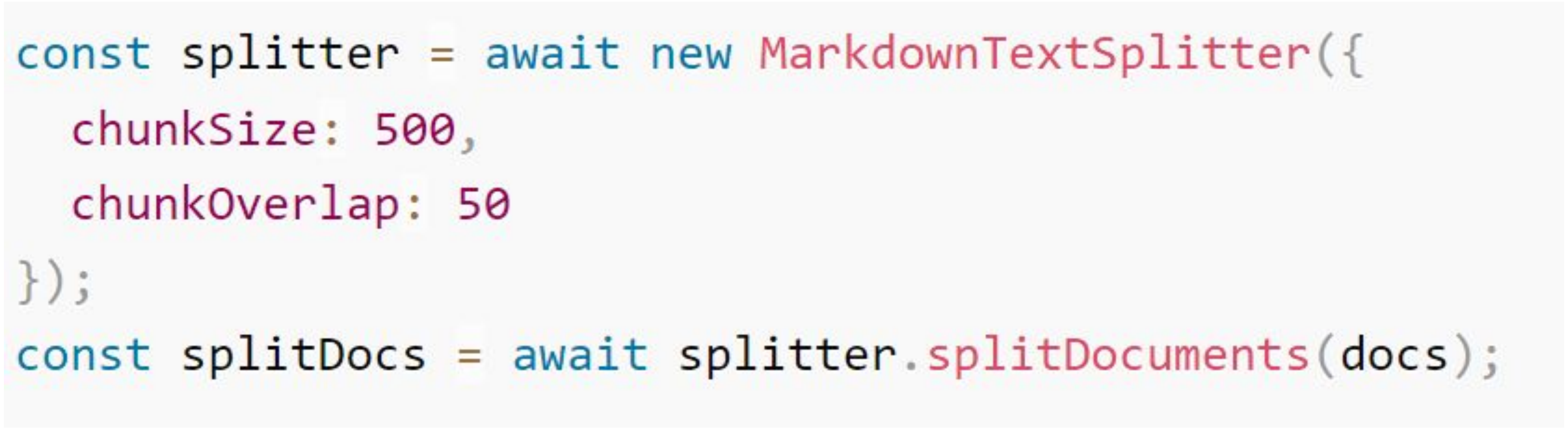
Note! - supported context is limited, for example 2k

Load the documents

nodejs-reference-architecture

```
const docLoader = new DirectoryLoader(
  "./SOURCE_DOCUMENTS",
  {
    ".md": (path) => new TextLoader(path),
  }
);
const docs = await docLoader.load();
```

Split the documents



```
const splitter = await new MarkdownTextSplitter({
  chunkSize: 500,
  chunkOverlap: 50
});
const splitDocs = await splitter.splitDocuments(docs);
```

Store the chunks in a database

```
const vectorStore = await MemoryVectorStore.fromDocuments(
  splitDocs,
  new HuggingFaceTransformersEmbeddings()
);
const retriever = await vectorStore.asRetriever();
console.log("Augmenting data loaded - " + new Date());
```

```
retriever.getRelevantDocuments("Should I use npm to start a node.js
application?");
```

Create Chain

```
const prompt =
  ChatPromptTemplate.fromTemplate(`Answer the following question based only on the provided context, if you
don't know the answer say so:

<context>
{context}
</context>

Question: {input}`);

const documentChain = await createStuffDocumentsChain({
  llm: model,
  prompt,
});

const retrievalChain = await createRetrievalChain({
  combineDocsChain: documentChain,
  retriever,
});
```

Ask Question

```
////////////////////////////////
// ASK QUESTIONS

console.log(new Date());
let result = await retrievalChain.invoke({
  input: "Should I use npm to start a node.js application",
});
console.log(result);
console.log(new Date());
```

A Better Answer

'Assistant: It is generally not necessary to use `npm` to start a Node.js application. If you avoid using it in the container, you will not be exposed to any security vulnerabilities that might exist in that component or its dependencies. However, it is important to build security into your software development process when developing Node.js modules and applications. This includes managing dependencies, managing access and content of public and private data stores such as npm and github, writing defensive code, limiting required execution privileges, supporting logging and monitoring, and externalizing secrets.'

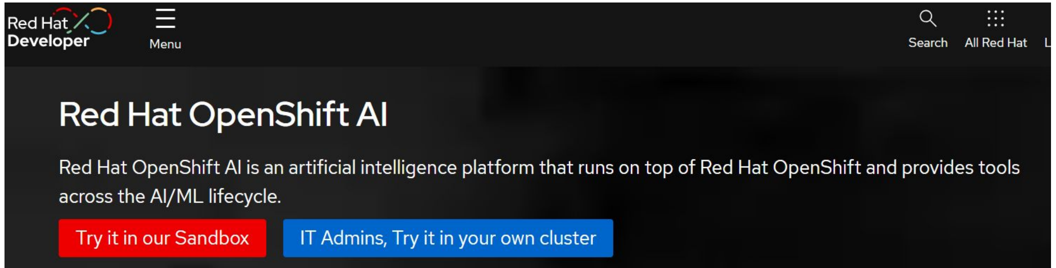
Switching to other model serving back ends

```
async function getModel(type, temperature) {
  console.log("Loading model - " + new Date());

  let model;
  if (type === 'llama-cpp') {
    const __dirname = path.dirname(fileURLToPath(import.meta.url));
    const modelPath = path.join(__dirname, "models", "mistral-7b-instruct-v0.1.Q5_K_M.gguf")
    const { LlamaCpp } = await import("@langchain/community/llms/llama_cpp");
    model = await new LlamaCpp({ modelPath: modelPath,
                                 batchSize: 1024,
                                 temperature: temperature,
                                 gpuLayers: 64 });
  } else if (type === 'openAI') {
    const { ChatOpenAI } = await import("@langchain/openai");
    const key = await import('../key.json', { with: { type: 'json' } });
    model = new ChatOpenAI({
      temperature: temperature,
      openAIApiKey: key.default.apiKey
    });
  } else if (type === 'Openshift.ai') {
```

Switching to other model serving back ends

```
} else if (type === 'openAI') {
  const { ChatOpenAI } = await import("@langchain/openai");
  const key = await import('../key.json', { with: { type: 'json' } });
  model = new ChatOpenAI({
    temperature: temperature,
    openAIApiKey: key.default.apiKey
  });
} else if (type === 'Openshift.ai') {
  const { ChatOpenAI } = await import("@langchain/openai");
  model = new ChatOpenAI(
    { temperature: temperature,
      openAIApiKey: 'EMPTY',
      modelName: 'mistralai/Mistral-7B-Instruct-v0.2' },
    { baseURL: 'http://vllm.llm-hosting.svc.cluster.local:8000/v1' }
  );
```

https://developers.redhat.com/products/red-hat-openshift-ai/overview

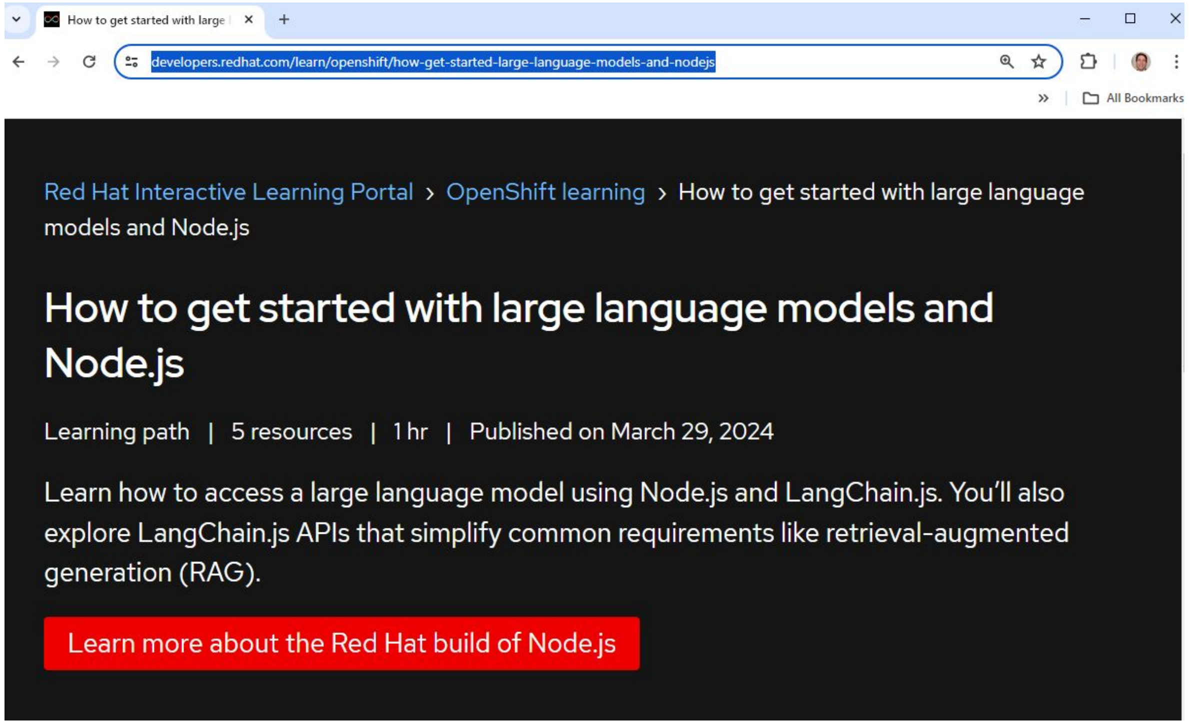


Red Hat OpenShift AI

Red Hat OpenShift AI is an artificial intelligence platform that runs on top of Red Hat OpenShift and provides tools across the AI/ML lifecycle.

To Dive Deeper

https://developers.redhat.com/learn/openshift/how-get-started-large-language-models-and-nodejs


How to get started with large
developers.redhat.com/learn/openshift/how-get-started-large-language-models-and-nodejs
All Bookmarks

Red Hat Interactive Learning Portal › OpenShift learning › How to get started with large language models and Node.js

How to get started with large language models and Node.js

Learning path | 5 resources | 1 hr | Published on March 29, 2024

Learn how to access a large language model using Node.js and LangChain.js. You'll also explore LangChain.js APIs that simplify common requirements like retrieval-augmented generation (RAG).

Some Key Takeaways

- It’s easy to run a local model
- Strengths of Node.js still apply
- Libraries to improve DevX
    - TypeScript/JavaScript often supported

Q/A

Copyright and Trademarks