


# AI for the mere mortal web developer

Ash Ryan Arnwine | Director of Developer Relations, Nylas   
[linkedin.com/in/ashryan](https://www.linkedin.com/in/ashryan) | [sayhi@ashryan.io](mailto:sayhi@ashryan.io)

# “Experimenting with AI in DevRel”

[ashryan.io/experimenting-with-ai-in-devrel](https://ashryan.io/experimenting-with-ai-in-devrel)



**Focus on the mere mortal developer**

# You might be a mere mortal developer if...

- You code but you aren't an LLM expert (like me)
- You want to sprinkle some AI magic into your app (like me)
- You want to learn by doing (like me)

Email triager

Email autoresponder

Email autoresponder

Calendar chat

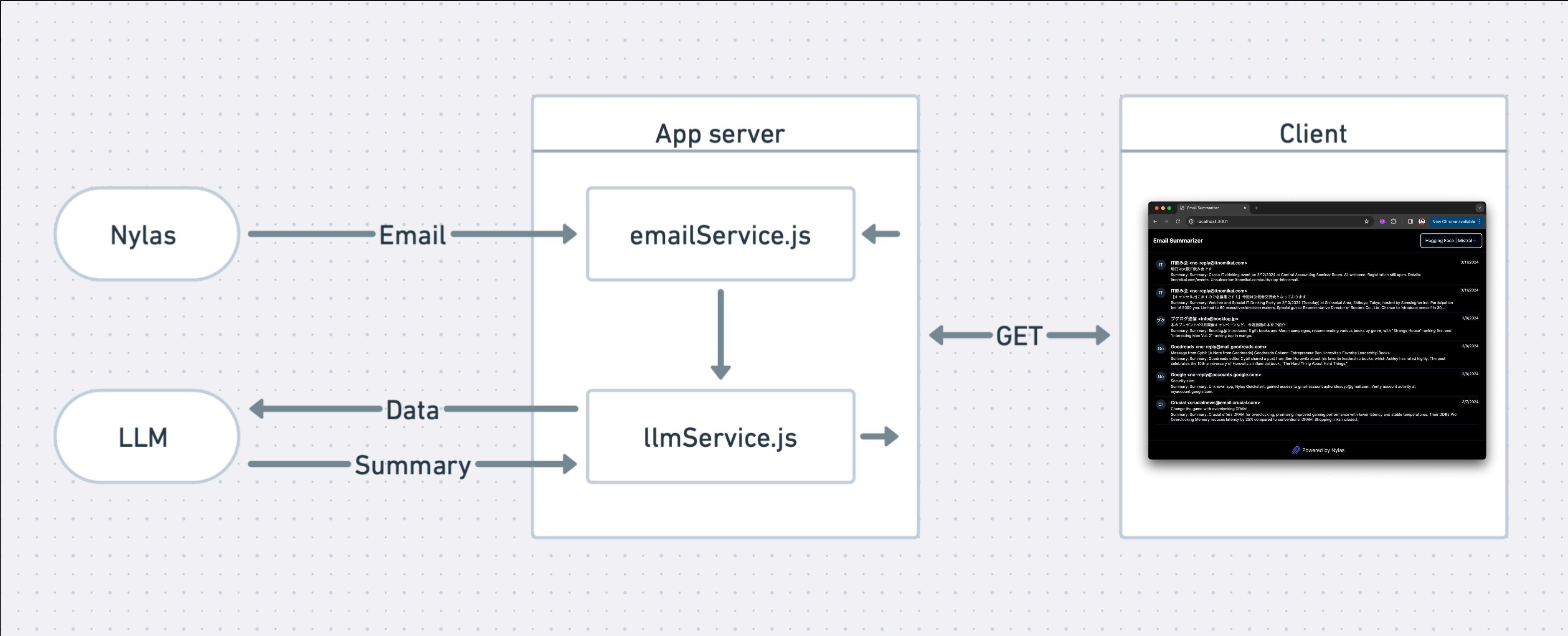
# Email Summarizer

The screenshot shows a web browser window with the URL localhost:3001. The page title is "Email Summarizer". In the top right corner, there is a dropdown menu showing "Hugging Face | Mistral". The main content area displays a list of six summarized email entries, each with a category icon, sender information, subject, and a brief summary. The entries are:

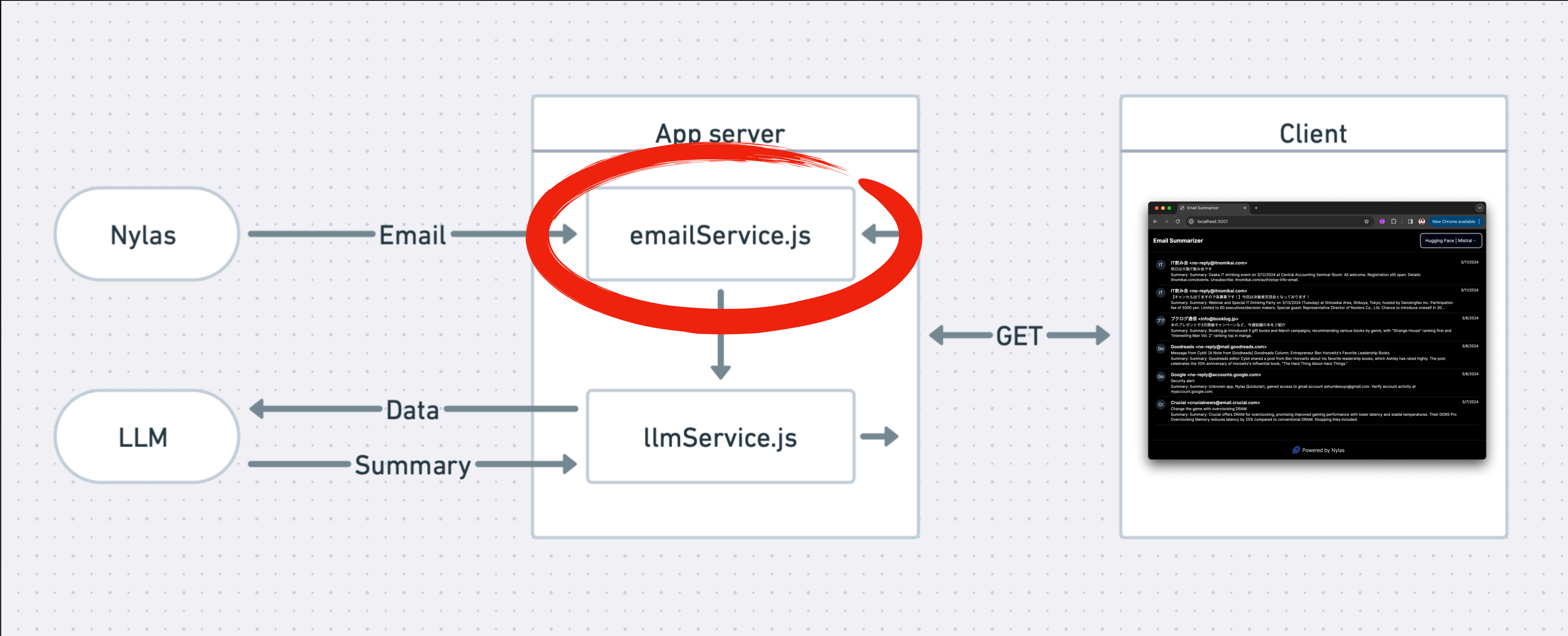
- IT** IT飲み会 <no-reply@itnomikai.com> 3/11/2024  
明日は大阪IT飲み会です  
Summary: Summary: Osaka IT drinking event on 3/12/2024 at Central Accounting Seminar Room. All welcome. Registration still open. Details: itnomikai.com/events. Unsubscribe: itnomikai.com/auth/stop-info-email.
- IT** IT飲み会 <no-reply@itnomikai.com> 3/11/2024  
【キャンセル出てますので急募集です！】今回は決裁者交流会となっております！  
Summary: Summary: Webinar and Special IT Drinking Party on 3/13/2024 (Tuesday) at Shinsekai Area, Shibuya, Tokyo, hosted by Samsingfan Inc. Participation fee of 5000 yen. Limited to 60 executives/decision makers. Special guest: Representative Director of Rooters Co., Ltd. Chance to introduce oneself in 30...
- ブク** ブクログ通信 <info@booklog.jp> 3/8/2024  
本のプレゼントや3月開催キャンペーンなど、今週話題の本をご紹介します  
Summary: Summary: Booklog.jp introduced 5 gift books and March campaigns, recommending various books by genre, with "Strange House" ranking first and "Interesting Man Vol. 2" ranking top in manga.
- Go** Goodreads <no-reply@mail.goodreads.com> 3/8/2024  
Message from Cybil: [A Note from Goodreads] Goodreads Column: Entrepreneur Ben Horowitz's Favorite Leadership Books  
Summary: Summary: Goodreads editor Cybil shared a post from Ben Horowitz about his favorite leadership books, which Ashley has rated highly. The post celebrates the 10th anniversary of Horowitz's influential book, "The Hard Thing About Hard Things."
- Go** Google <no-reply@accounts.google.com> 3/8/2024  
Security alert  
Summary: Summary: Unknown app, Nylas Quickstart, gained access to gmail account ashuridesuyo@gmail.com. Verify account activity at myaccount.google.com.
- Cr** Crucial <crucialnews@email.crucial.com> 3/7/2024  
Change the game with overclocking DRAM  
Summary: Summary: Crucial offers DRAM for overclocking, promising improved gaming performance with lower latency and stable temperatures. Their DDR5 Pro Overclocking Memory reduces latency by 25% compared to conventional DRAM. Shopping links included.

At the bottom of the page, there is a logo for "Powered by Nylas".

# Under the hood



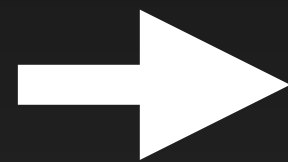
# Under the hood





# emailService.js

## Fetching emails with Nylas



```
const fetchEmailsFromNylas = async (nylas, nylasGrantId, limit) => {  
  
  try {  
    const messages = await nylas.messages.list({  
      identifier: nylasGrantId,  
      queryParams: { limit, in: "inbox" },  
    });  
  
    return messages.data;  
  } catch (error) {  
    throw error;  
  }  
};
```

# emailService.js

## Prepping data for the LLM

```
const prepareEmailForLLMAPI = (email) => {
  const { date, body, subject, from, to } = email;

  const formattedDate = new Date(date * 1000).toLocaleDateString();

  const formattedFrom = from
    .map((contact) => `${contact.name} <${contact.email}>`)
    .join(`, `);

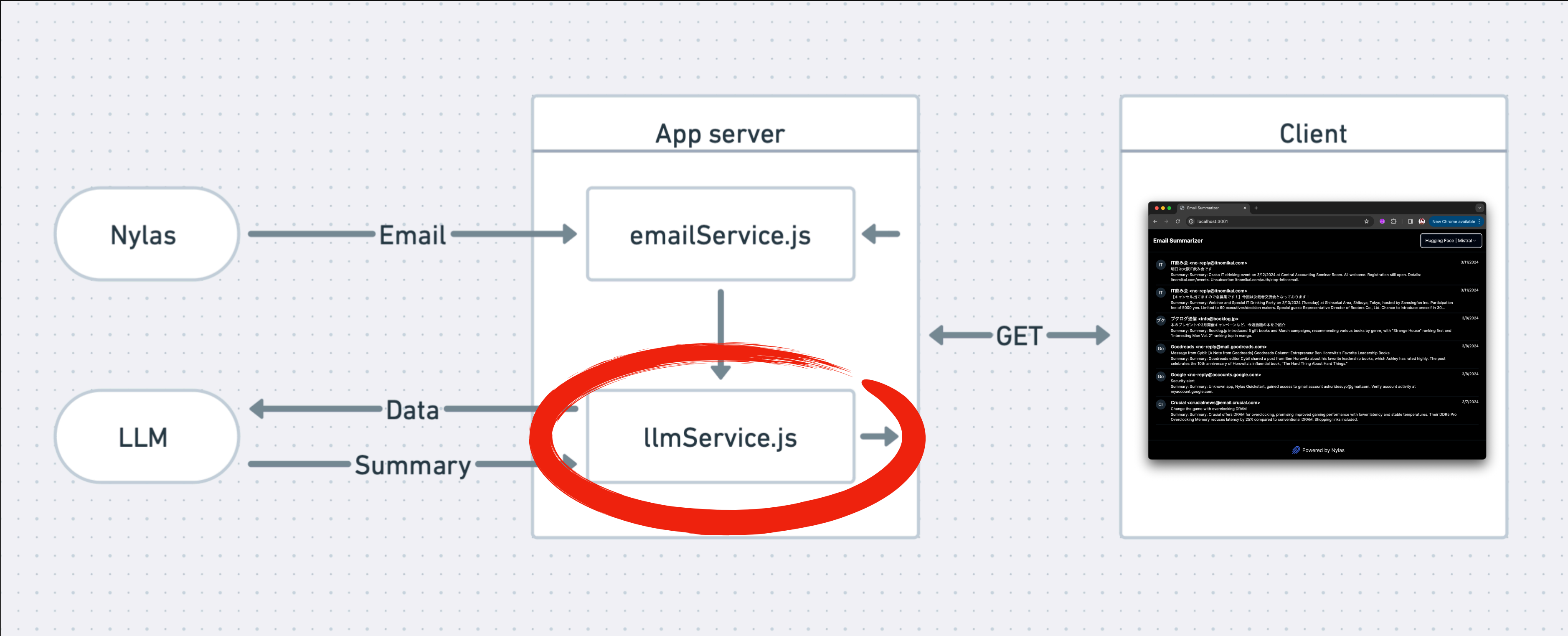
  const formattedTo = to
    .map((contact) => `${contact.name} <${contact.email}>`)
    .join(`, `);

  const formattedInitials = from
    .map((contact) => getInitials(contact.name, contact.email))
    .join("");

  const cleanedBody = stripHtml(body).result;

  return {
    formattedDate,
    formattedFrom,
    formattedInitials,
    formattedTo,
    subject,
    cleanedBody,
  };
};
```

# Under the hood



# llmService.js

Making a generic class calling LLMs

```
class LLMService {  
  async summarize(email) {  
    throw new Error("No LLM selected");  
  }  
  
  generatePrompt(preppedEmail) {  
    return "<PROMPT GOES HERE>";  
  }  
}  
  
export default LLMService;
```

# llmService.js

Show us your prompt already



```
return `
```

```
You are an email summarizer receiving an email body with the HTML tags stripped out.  
You have 3 short sentences to summarize the following message.
```

```
Some important rules:
```

- You must summarize the email in 100 characters or less.
- Do NOT show or refer directly to any links or email addresses.
- Your summary must be entirely in English, regardless of the language of the email.
- Do NOT respond to the prompt or refer to it. You are only allowed to send the summary.

```
The message to summarize:
```

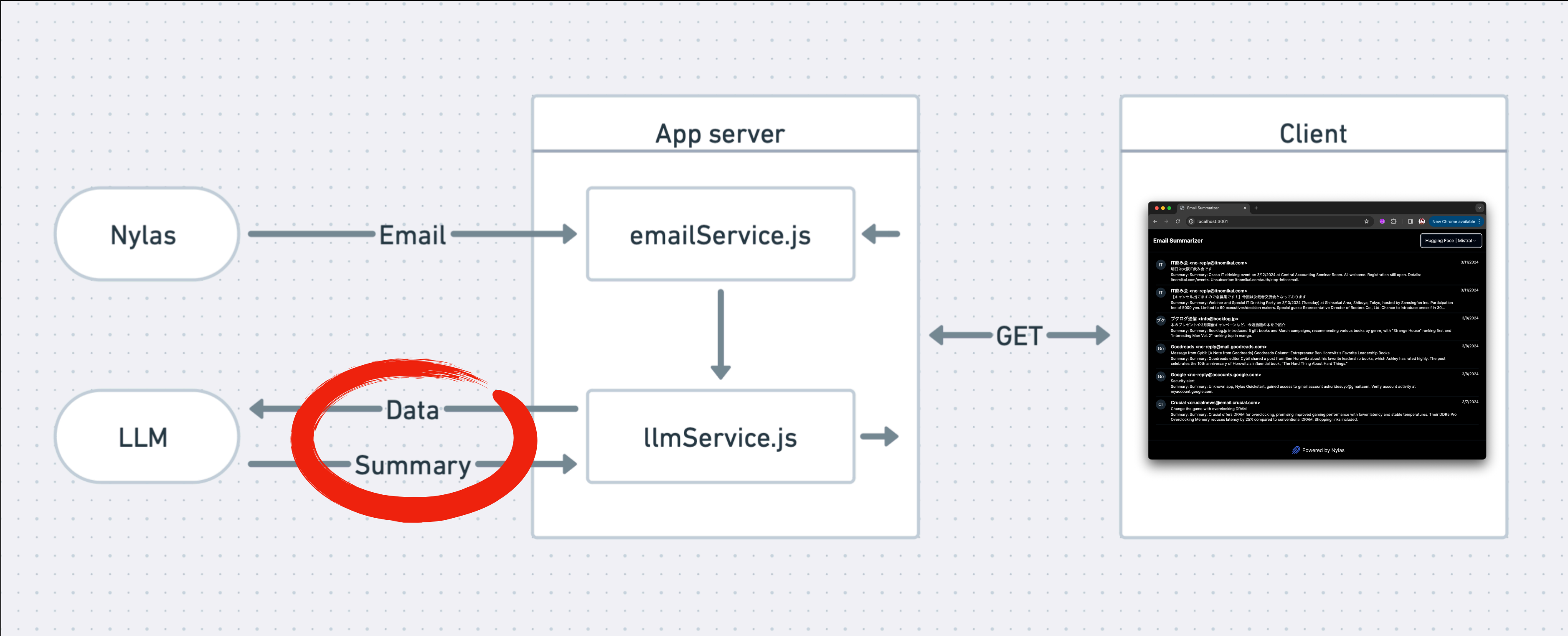
```
${preppedEmail.formattedFrom} sent an email to ${preppedEmail.formattedTo}  
with the subject ${preppedEmail.subject} on ${preppedEmail.formattedDate}.
```

```
The email body is:
```

```
${preppedEmail.cleanedBody}
```

```
`;  
`;
```

# Under the hood



**Let's make some LLM API calls**

# openAIService.js

## You knew it was coming



```
import LLMService from "./llmService.js";
import OpenAI from "openai";

const openAI = new OpenAI({ apiKey: process.env.OPENAI_API_KEY });

class OpenAIService extends LLMService {
  constructor() {
    super();
    this.llm = openAI;
  }

  async summarize(preppedEmail) {
    const prompt = this.generatePrompt(preppedEmail);

    const response = await this.llm.chat.completions.create({
      model: "gpt-3.5-turbo",
      messages: [{ role: "system", content: prompt }],
    });

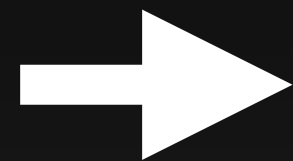
    return response.choices[0].message.content;
  }
}

export default OpenAIService;
```



# openAIService.js

You knew it was coming



```
import LLMService from "./llmService.js";
import OpenAI from "openai";

const openAI = new OpenAI({ apiKey: process.env.OPENAI_API_KEY });

class OpenAIService extends LLMService {
  constructor() {
    super();
    this.llm = openAI;
  }

  async summarize(preppedEmail) {
    const prompt = this.generatePrompt(preppedEmail);

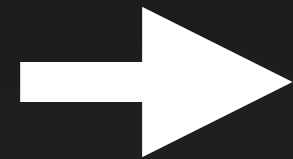
    const response = await this.llm.chat.completions.create({
      model: "gpt-3.5-turbo",
      messages: [{ role: "system", content: prompt }],
    });

    return response.choices[0].message.content;
  }
}

export default OpenAIService;
```

# openAIService.js

You knew it was coming



```
import LLMService from "../llmService.js";
import OpenAI from "openai";

const openAI = new OpenAI({ apiKey: process.env.OPENAI_API_KEY });

class OpenAIService extends LLMService {
  constructor() {
    super();
    this.llm = openAI;
  }

  async summarize(preppedEmail) {
    const prompt = this.generatePrompt(preppedEmail);

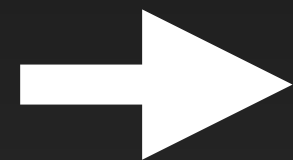
    const response = await this.llm.chat.completions.create({
      model: "gpt-3.5-turbo",
      messages: [{ role: "system", content: prompt }],
    });

    return response.choices[0].message.content;
  }
}

export default OpenAIService;
```

# openAIService.js

You knew it was coming



```
import LLMService from "./llmService.js";
import OpenAI from "openai";

const openAI = new OpenAI({ apiKey: process.env.OPENAI_API_KEY });

class OpenAIService extends LLMService {
  constructor() {
    super();
    this.llm = openAI;
  }

  async summarize(preppedEmail) {
    const prompt = this.generatePrompt(preppedEmail);

    const response = await this.llm.chat.completions.create({
      model: "gpt-3.5-turbo",
      messages: [{ role: "system", content: prompt }],
    });

    return response.choices[0].message.content;
  }
}

export default OpenAIService;
```

Demo

# Code compare

## OpenAI

```
async summarize(preppedEmail) {
  const prompt = this.generatePrompt(preppedEmail);

  const response = await this.llm.chat.completions.create({
    model: "gpt-3.5-turbo",
    messages: [
      {
        role: "system",
        content: prompt,
      },
    ],
  });

  return response.choices[0].message.content;
}
```

## Hugging Face

```
async summarize(preppedEmail) {
  const prompt = this.generatePrompt(preppedEmail);

  const response = await this.llm.textGeneration({
    model: "mistralai/Mistral-7B-Instruct-v0.2",
    inputs: prompt,
  });

  return response.generated_text;
}
```

## Ollama

```
async summarize(preppedEmail) {
  const prompt = this.generatePrompt(preppedEmail);

  const response = await ollama.chat({
    model: "llama2",
    messages: [
      {
        role: "system",
        content: prompt,
      },
    ],
  });

  return response.message.content;
}
```

# Wait, what are those services?

OpenAI

Hugging Face

Ollama

# Wait, what are those services?

OpenAI

[platform.openai.com](https://platform.openai.com)

GPT 4, Whisper, etc

Cloud APIs

Hugging Face

Ollama

# Wait, what are those services?

OpenAI

[platform.openai.com](https://platform.openai.com)

GPT 4, Whisper, etc

Cloud APIs

Hugging Face

[huggingface.co](https://huggingface.co)

Community-centric AI hub

Cloud APIs and local APIs

Ollama



# Wait, what are those services?

OpenAI

[platform.openai.com](https://platform.openai.com)

GPT 4, Whisper, etc

Cloud APIs

Hugging Face

[huggingface.co](https://huggingface.co)

Community-centric AI hub

Cloud APIs and local APIs

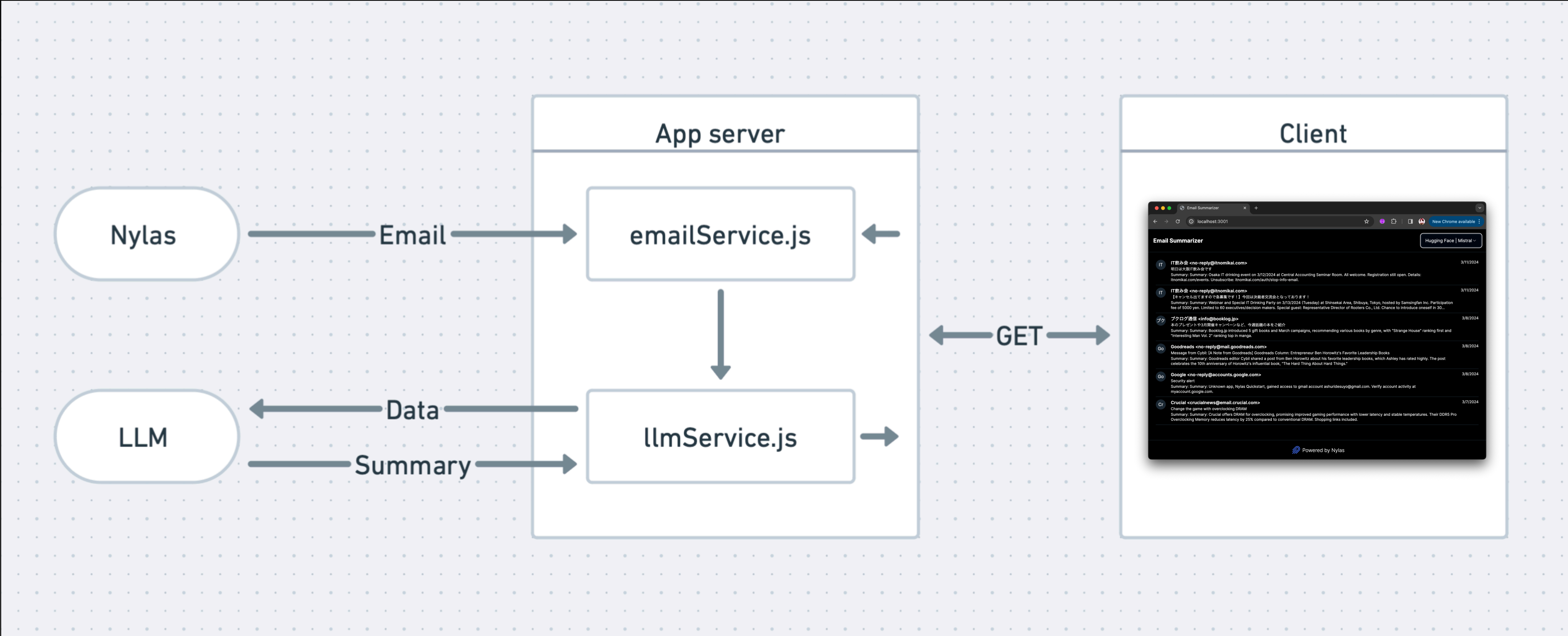
Ollama

[ollama.com](https://ollama.com)

CLI with REST API

Local APIs

# Under the hood



# LLM checklist

For mere mortal developers

- ✓ Define your task
- ✓ Choose your service
- ✓ Choose your model
- ✓ Prep your data
- ✓ Devise your prompt
- ✓ Handle the output

# LLM checklist

For mere mortal developers

✓ Define your task

✓ Choose your service

✓ Choose your model

✓ Prep your data

✓ Devise your prompt

✓ Handle the output

? Tune the LLM to user data

? Store (evolving) user prefs

? Managing cost

**“It’s all in the repo!”**

—The repo: [github.com/nylas-samples/austin-api-summit-2024](https://github.com/nylas-samples/austin-api-summit-2024)

**“OK, but what is this like in prod?”**

—You

# Vibebox

Your inbox + vibes



Vibebox is your nook of happiness in a world brimming with urgent messages.

Whether it's a bill or a note from a pal, Vibebox re-crafts each email into a joyful read, as warm as a cuddle from your favorite furry friend.

**Let's get started!**

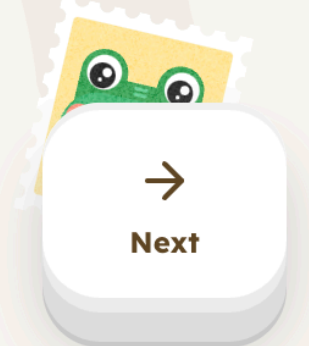
**Connect Your Email**

Google Voice <voice-noreply@google.com>

New text message from 26266




🌈👋🌟 Hello there, lovely friend! 🌸 Guess what? Your flight to New York/Newark is all set and waiting for you at Gate C43! ✈️ Make sure to bring your passport with you—it's your magical ticket to adventure! 🇺🇸🌟 We're super excited to welcome you on board soon! 🌟💖 Just a little reminder, this message flew to your inbox because you wanted to know about text messages. If you prefer a quieter inbox, you can tweak your settings anytime. 📧🌟 Sending you lots of love and sparkles! Keep shining bright, dear one! 🌟💖 Warm hugs and happy vibes, 🧚👉 Your Fairy-like Friend 🌟🌟





# AI for the mere mortal web developer

Ash Ryan Arnwine | Director of Developer Relations, Nylas   
[linkedin.com/in/ashryan](https://www.linkedin.com/in/ashryan) | [sayhi@ashryan.io](mailto:sayhi@ashryan.io)