CueCode

- SEAN BAKER
- ANDREW BAUSAS
- FREDDIE BOATENG
- KOBE FRANSSEN
- BRYAN HAWICKHORST
- JOHN HICKS
- DIYA PATEL
- CHASE WALLACE

Team Red CS410W project

Elevator Pitch

CueCode lets a Web application generate API calls from natural language with minutes of development time. “I booked an appointment for Patricia Davis for Thursday at 2pm” can become an API call to your appointment booking backend with little additional programming effort.

A good API specification and a few key questions are all the model needs to start generating API requests.

This allows rapid development of natural language processing features typical of those created during the Generative AI boom, without having to take humans or business rules out of the loop. CueCode can add AI features to your app without any backend code changes or specialized NLP or large language model (LLM) skills.

CueCode is made by developers, for developers - as seen in CueCode’s easy-to-use client libraries.

Table of Contents

Team Bios

- SEAN BAKER
- ANDREW BAUSAS
- FREDDIE BOATENG
- KOBE FRANSSEN
- BRYAN HAWICKHORST
- JOHN HICKS
- DIYA PATEL
- CHASE WALLACE

The Societal Problem

- User interfaces don’t speak the user’s language
- Turning bulk unstructured data into structured data is difficult
- Humans are kept out of the loop in current AI agent based systems
- To develop human-in-the-loop natural language to API systems, it would take:
  - Specialized skills
  - Extensive backend programming changes
  - One-off development per application
  - => A lot of money

2.1 Problem Statement

To solve the above problems, we must commoditize LLM development for existing Web apps.

However, there are no frameworks/tools that leverage OpenAPI or GraphQL specifications, the two most common ways to describe API capabilities.

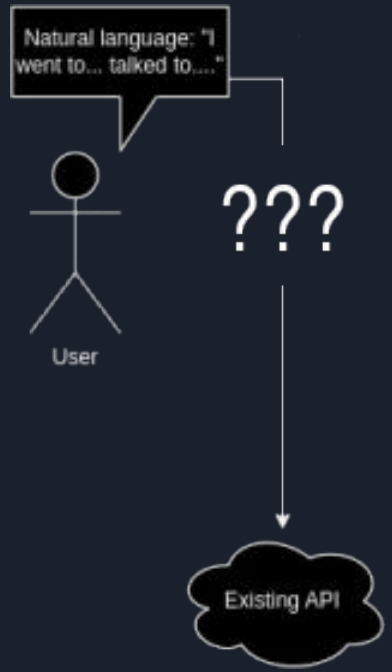
2.2 Problem Characteristics

Problems with current NLP/LLM processing for creating API calls:

- Hand decision-making to the LLM
  - removing human checks
  - business logic
- One-off, defined per application
- Lack a clearly defined concept of entity relationships
- Require awareness of prompt engineering and other more complex AI techniques
- => Heavy development effort

2.3 Current Process Flow

- Encode API structure
  - Build Python classes [9]
- Verify output is in JSON format
  - Tools exist for verifying a JSON format and even that LLM output matches a JSON schema. (LangChain [9], Guidance AI [6])
  - (This alone does not make an NLP-to-API engine.)
- Tell the LLM about the API structure
  - One-shot prompt is common
  - Could not find examples of encoding API information in the vector store.
- Tie it all together with backend programming
- Make your application aware of LLM API call suggestions



3 Solution

We aim to build client libraries for web app developers that interface with CueCode's servers, which will deploy LLMs to convert natural language to structured API calls.

3.1 Solution Statement

What that means:

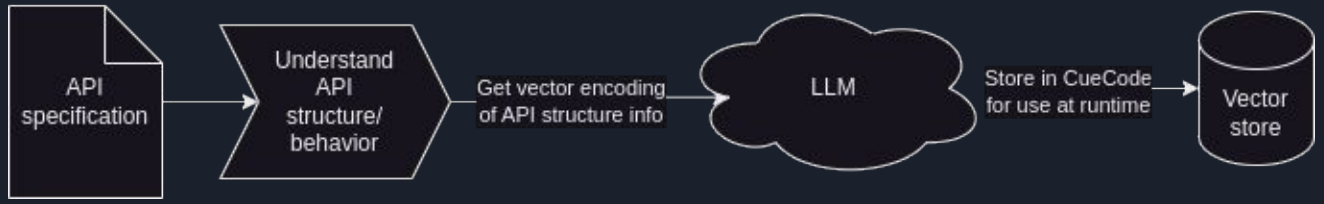
Developers will be able to use existing API specifications, which is CueCode makes understandable by LLMs, to define the structure of their API calls.

For example, if a client service representative were to provide input to an application using CueCode in natural language, “I called Patricia Davis and rescheduled her appointment from August 1st to August 16th.” The application can then use CueCode’s libraries, which have been configured using documentation about the structure of their data, to generate the following JSON:

```
{"request":{"reschedule":{"last": "Davis", "first":"Patricia", "from":{"month":8,
"day":1,"year":2024}, "to":{"month":8, "day":1,"year":2024}}}}
```

Which would then be converted into the appropriate API call to change the appointment date in their database, or prompt the user for additional information.

3.2 Solution Process Flow (training time)



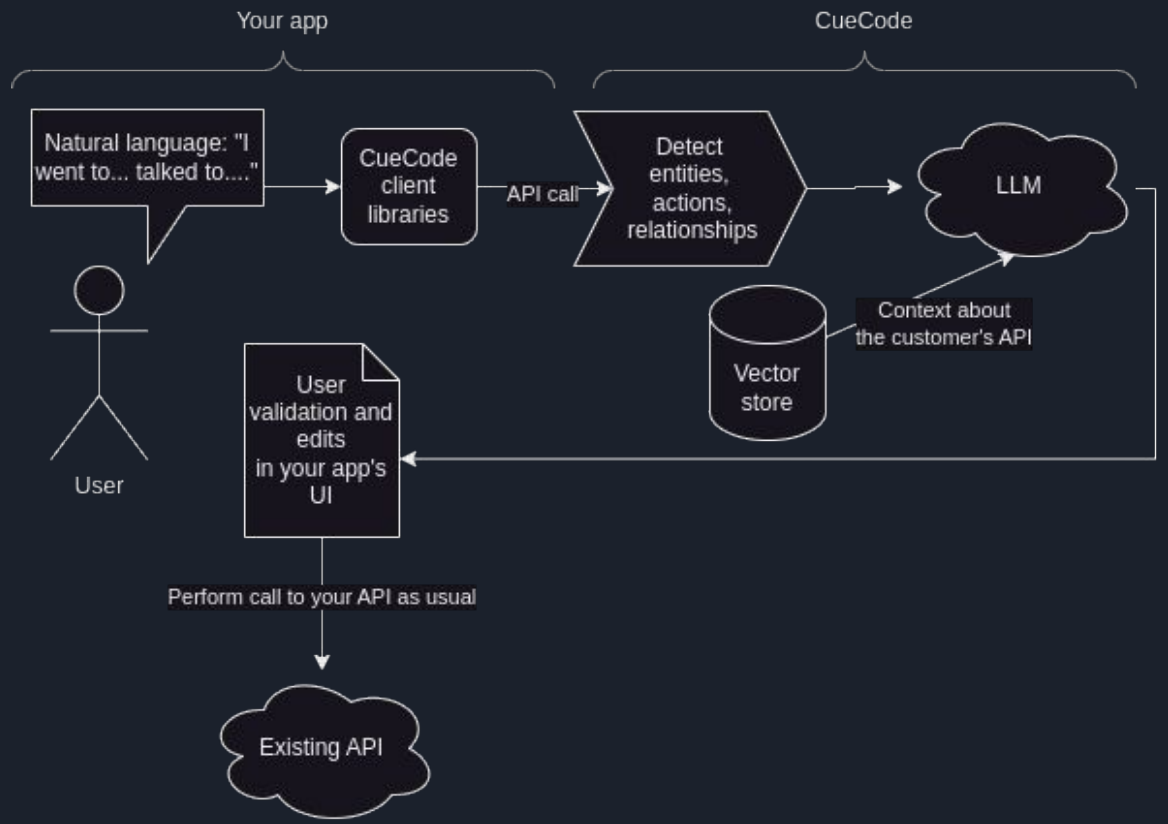
Training time:

- Upload API specification
- Answer a few questions
- CueCode stores the structure and requirements for your API in a vector store to aid the LLM in generating responses at runtime

3.2 Solution Process Flow (runtime)

Use CueCode in your app:

- Integrate text processing via CueCode libraries
- At runtime, let CueCode figure out the structured data contained in the text
- Use CueCode's extracted structured data. e.g.:
  - Show suggestions to the user
  - Perform API calls in a batch job
  - Validate through business rules
  - Whatever your use case requires



3.3 What it Will Do

- Will implement NLP capabilities to enable and understand natural language
- Will offer a user friendly interface (API) that developers can use
- Will enable quick iteration and prototyping by allowing developers to test and refine how their applications respond to the natural language inputs.
- Will provide tools for customizing NLP models to fit specific domains/industries ensuring better performance for unique use cases.
- Will include documentation and support resources to help developers implement and troubleshoot various systems effectively.
- Will reduce the time and financial investment typically required for implementing NLP, making it affordable for smaller teams and startups
- Will use API specifications, enabling context-aware replies that complement the distinct functionality and data structure of each application.
- Will allow for real time analysis and response generation, enhancing user experience through immediate feedback and interactions.

3.4 What it Will Not Do

- Will not replace human judgment when interpreting language in terms of making subjective decisions beyond its programming.
- Will not act as an AI agent
- Will not be perfect, misinterpretations could occur with certain slang, ambiguous phrasing or idioms.
- Will not be able to handle complex conversations.
- Will struggle with dialogues conversations that require deeper understanding.
- Will not have ready to use applications pre installed, developers will need to build their own solutions and install any necessary software/applications they need.
- Will not automatically make API calls on users' behalf; requests must first have human permission before being fulfilled.
- Will not have programming tutorials, developers will need to possess knowledge of programming to utilize CueCode effectively.
- Will not ensure data privacy, users must manage and secure their data to the best of their abilities.

3.5 Competition Matrix

✔- Full Implementation
✔ - Partial Implementation

Feature	CueCode	OpenAI Functions	Google Natural Language API	Spacy.io	LangChain	GenKit
Entity recognition	✔		✔	✔		✔
Plug and Play	✔				✔	✔
LLM suggests action	✔	✔			✔	
Retrieval Augmented Generation	✔	✔			✔	✔
Requires no LLM Experience	✔		✔	✔		

4 Development Tools

Not in scope for Feasibility iteration 1

5 Major Functional Components

Not in scope for Feasibility iteration 1

5.1 Major Functional Components Diagram

Not in scope for Feasibility iteration 1

6 Risks

Not in scope for Feasibility iteration 1

E.g., what if we find we need to store application data in the vector store, not just the API schema, to get CueCode to work at all? That would change the way we market it and the tooling we would need to create for our developer customers.

7 References

[1]

“Against LLM maximalism · Explosion.” Accessed: Sep. 10, 2024. [Online]. Available: https://explosion.ai/blog/explosion.ai

[2]

E. at Zafin, “Bridging the Gap: Exploring use of Natural Language to interact with Complex Systems,” Engineering at Zafin. Accessed: Sep. 10, 2024. [Online]. Available: https://medium.com/engineering-zafin/bridging-the-gap-exploring-using-natural-language-to-interact-with-complex-systems-11c1b056cc19

[3]

Y. Su, A. H. Awadallah, M. Khabsa, P. Pantel, M. Gamon, and M. Encarnacion, “Building Natural Language Interfaces to Web APIs,” in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, Singapore Singapore: ACM, Nov. 2017, pp. 177–186. doi: 10.1145/3132847.3133009.

[4]

“Firebase Genkit.” Accessed: Sep. 14, 2024. [Online]. Available: https://firebase.google.com/docs/genkit

7 References

[5] “Function Calling.” Accessed: Sep. 14, 2024. [Online]. Available: https://platform.openai.com/docs/guides/function-calling

[6] *guidance-ai/guidance*. (Sep. 25, 2024). Jupyter Notebook. guidance-ai. Accessed: Sep. 25, 2024. [Online]. Available: https://github.com/guidance-ai/guidance

[7] “OpenAPI Specification - Version 3.1.0 | Swagger.” Accessed: Sep. 10, 2024. [Online]. Available: https://swagger.io/specification/

[8] *OpenAPITools/openapi-generator*. (Sep. 10, 2024). Java. OpenAPI Tools. Accessed: Sep. 10, 2024. [Online]. Available: https://github.com/OpenAPITools/openapi-generator

7 References

[9]

"Tool/function calling | LangChain." Accessed: Sep. 14, 2024. [Online]. Available: https://python.langchain.com/v0.1/docs/modules/model_io/chat/function_calling/

[10]

"What Is NLP (Natural Language Processing)? | IBM." Accessed: Sep. 10, 2024. [Online]. Available: https://www.ibm.com/topics/natural-language-processing

[11]

"Cloud Natural Language," Google Cloud. Accessed: Sep. 26, 2024. [Online]. Available: https://cloud.google.com/natural-language

8 Appendix

8.1 Real World Product vs Prototype Table

Not in scope for Feasibility iteration 1.

That said, we will likely implement CueCode for OpenAPI specs but not GraphQL specs.