Creating ChatGPT applications with BAML

How can you easily create structured output apps in less than an hour with a Python package BAML

Gintautas Svedas

2024 Oct 22, Hacker Dojo

Table of Contents

What is BAML?

The features of BAML

Explaining the sample function

Building our sample application together

Question time

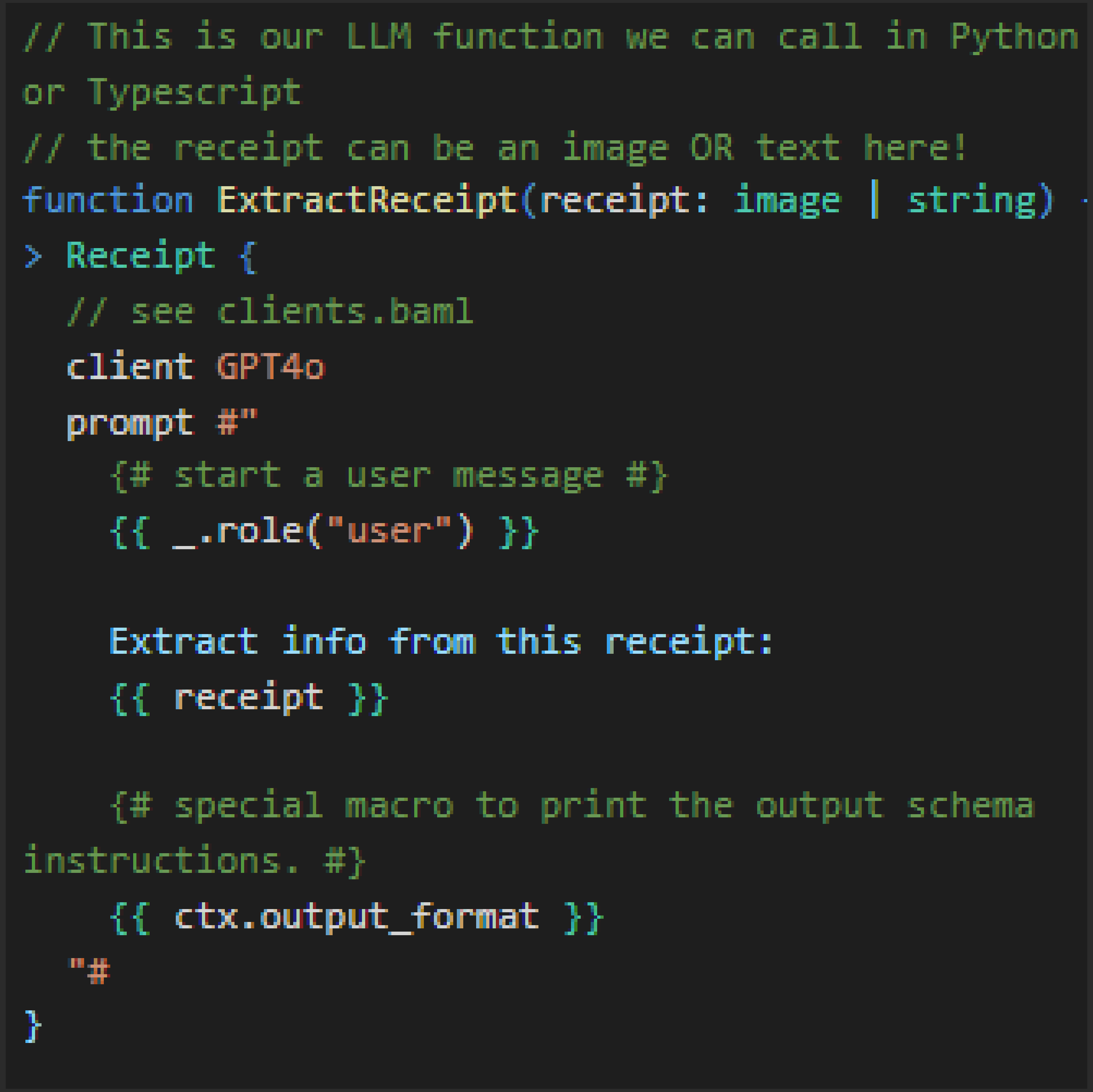
A little about me

- I’m Gintautas, I just came here to the Bay Area a month ago
- I come from Europe, specifically Lithuania
- Currently working as an AI engineer at SAP (German company)
- Love building all techy things
- Hiking, board games, meeting new people

What is BAML?

BAML is a domain-specific language to write and test LLM functions.

An LLM function is a prompt template with some defined input variables, and a specific output type like a class, enum, union, optional string, etc.

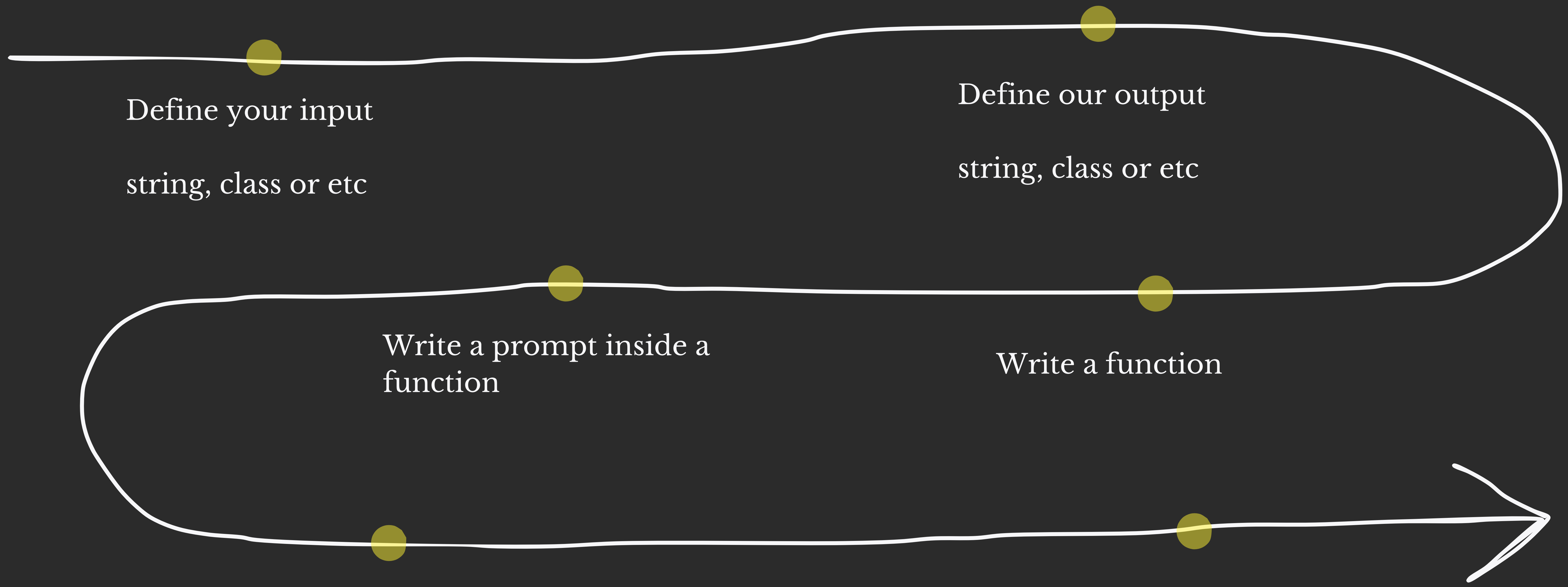


```
// This is our LLM function we can call in Python
or Typescript
// the receipt can be an image OR text here!
function ExtractReceipt(receipt: image | string) -
> Receipt {
  // see clients.baml
  client GPT4o
  prompt #"
    {# start a user message #}
    {{ _.role("user") }}

    Extract info from this receipt:
    {{ receipt }}

    {# special macro to print the output schema
instructions. #}
    {{ ctx.output_format }}
  "#
}
```

How does it work?



Define your input

string, class or etc

Define our output

string, class or etc

Write a function

Write a prompt inside a function

Test the function

Generate Python code from BAML

BAML Features

- Python and Typescript support
- Wide model support
- Realtime Prompt Previews
- Resilience and fallback features
- Observability Platform
- Type validation
- Streaming
- Testing support

Examing a simple function

```
// Declare some data models for my function, with descriptions
class Resume {
  name string
  education Education[] @description("Extract in the same order listed")
  skills string[] @description("Only include programming languages")
}

class Education {
  school string
  degree string
  year int
}

function ExtractResume(resume_text: string) -> Resume {
  // LLM client with params you want (not pictured)
  client GPT4Turbo

  // BAML prompts use Jinja syntax
  prompt #"
    Parse the following resume and return a structured representation of the data in the schema below.

    Resume:
    ---
    {{ resume_text }}
    ---

    {# special Jinja macro to print the output instructions. #}
    {{ ctx.output_format }}

    JSON:
  "#
}
```

Describing each element for the prompt

Function signature

ChatGPT prompt

Structured Output

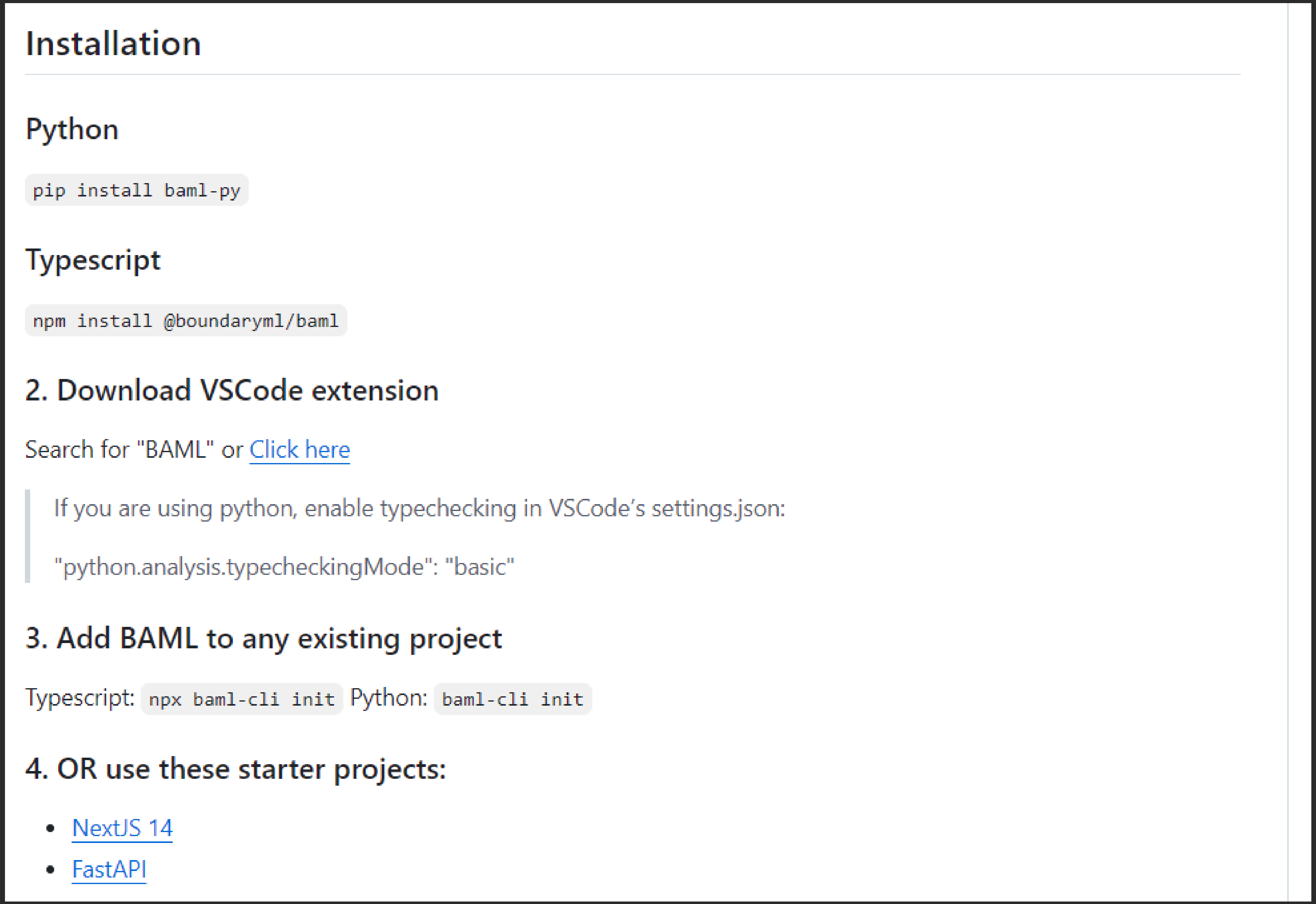
LLM Function

Let’s go to PromptFiddle.com and see the examples

Now, everyone, please open VS Studio or
your favorite editor and follow along :)

I highly encourate to use VS Studio/
Cursor as we will need a special extension
to ease our work.

Search Baml Language VS Code Extension on
Google
or
https://shorturl.at/GBy4p



Installation

Python

```
pip install baml-py
```

Typescript

```
npm install @boundaryml/baml
```

2. Download VSCode extension

Search for "BAML" or Click here

> If you are using python, enable typechecking in VSCode's settings.json:
>
> "python.analysis.typecheckingMode": "basic"

3. Add BAML to any existing project

Typescript: `npx baml-cli init` Python: `baml-cli init`

4. OR use these starter projects:

- NextJS 14
- FastAPI

Question time