OpenAPI Bot: A Chatbot to Help You Understand REST APIs*

Hamza Ed-douibi[1] [0000-0003-4342-4818], Gwendal Daniel[1] [0000-0003-0692-0628], Jordi Cabot[1,2] [0000-0003-2418-2489]

[1] UOC. Barcelona, Spain
{hed-douibi,gdaniel}@uoc.edu
[2] ICREA. Barcelona, Spain
jordi.cabot@icrea.cat

Abstract. REST APIs are an essential building block in many Web applications. The lack of a standard machine-readable format to describe these REST APIs triggered the creation of several specification languages to formally define REST APIs, with the OpenAPI specification currently taking the lead. OpenAPI definitions are consumed by a growing ecosystem of tools aimed at automating tasks such as generating server/client SDKs and API documentations. However, current OpenAPI documentation tools mostly provide simple descriptive Web pages enumerating all the API operations and corresponding parameters, but do not offer interactive capabilities to help navigate the API and ask relevant information. Therefore, learning how to use an API and how its different parts are interrelated still requires a considerable time investment. To overcome this situation we present our OpenAPI Bot, a chatbot able to read an OpenAPI definition for you and answer the questions you may have about it.

1 Introduction

REST APIs are a key component of many modern Web applications. In recent years, the OpenAPI specification has positioned itself as *de facto* choice to describe these APIs. The OpenAPI specification is "a standard, programming language-agnostic interface description for REST APIs"[3].

Several tools leverage OpenAPI definitions to automate API development tasks such as generating Software Development Kits (SDKs) for a number of frameworks and languages (e.g., APIMatic and Swagger Codegen) or generating documentation (e.g., Swagger UI and RedDoc). We are specially interested in this latter group as, in our opinion, understanding how to properly use a new API is a very error-prone and time-consuming task. Unfortunately, current doc tools do not help much here as they focus on generating simple descriptions of individual API components. Developers cannot ask more advanced questions or have any kind of more interactive exploration to find the info they are looking for.

* Work supported by the Spanish government (TIN2016-75944-R project)

[3] https://github.com/OAI/OpenAPI-Specification

Meanwhile, chatbots applications are increasingly adopted in various domains such as e-commerce or customer services as a direct communication channel between companies and end-users. We believe chatbots could also help in the API domain by assisting developers in their API discovery process. Initial experiments in this field have targeted so far Java APIs [4] and Stack Overflow answers [1]. [5] is more similar to our initiative as it derives a bot from an OpenAPI specification but its focus is to facilitate the execution of the API, not to help developers understand the potential of the API itself.

In this paper we present OpenAPI Bot, a chatbot that leverages the OpenAPI specification to help developers understand REST APIs. OpenAPI Bot provides a quick way to get information about the metadata, operations, and data structures of an API, as well as advanced insights which are not directly grasped from the API specification.

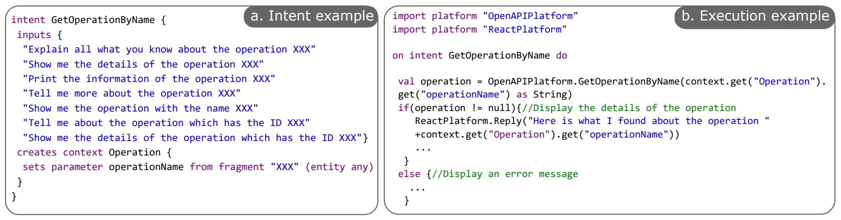
2 Overview

OpenAPI Bot is built with Xatkit [2], a flexible multi-platform (chat)bot development framework. Xatkit comprises three Domain-Specific Languages (DSLs) allowing the definition of different components of a chatbot, namely: *Intent DSL*, which defines the user inputs through training sentences, and context parameter extraction rules; *Execution DSL*, which defines how the bot should respond to the matched intents; and *Platform DSL*, that details the available operations and actions available to the bot (e.g., sending a message, querying a database, etc) depending on the platforms the bot interacts with. Platforms are provided by Xatkit itself. These languages are complemented by an execution engine that takes care of the deployment of the bots by registering the defined intents to a given NLP engine (DialogFlow in our case), and manages their execution.

Figure 1 shows a snippet of the OpenAPI Bot definition[4]. The bot defines a set of intents representing typical questions and navigation queries related to an OpenAPI definition. Figure 1.a shows the intent `GetOperationByName`, which includes training sentences to get an API operation using its name. The intent creates the `Operation` context containing the `operationName` parameter which is extracted from the user input. Our bot uses two Xatkit platforms: the `ReactPlatform`, a platform that receives user inputs and sends messages through a web-based component, and the `OpenAPIPlatform`, a custom platform we created to manipulate OpenAPI definitions. The OpenAPI Bot's execution model binds the specified intents to the platform's actions. Figure 1.b shows a snippet of the execution model containing the rule to execute when the intent `GetOperationByName` is matched. This rule first invokes the `GetOperationByName` action from the `OpenAPIPlatform`, then checks the returned value to display either the requested operation or an error message if it does not exist.

Figure 2 shows an overview of the key components of the `OpenAPI Bot`. The *OpenAPI Bot Definition* presented earlier is given as input to the *OpenAPI Bot Runtime* which is composed of the core *Xatkit Runtime* (that manages the deployment and execution of the bot), as well as the *OpenAPI Runtime* that contains the concrete implementation of the `OpenAPI Platform`'s actions. To do so, it relies on the *OpenAPI Modeling SDK* [3], our model-based framework to manipulate OpenAPI definitions.

[4] Complete sources for the example available at `https://github.com/opendata-for-all/openapi-bot/`


a. Intent example
b. Execution example

```
intent GetOperationByName {
 inputs {
  "Explain all what you know about the operation XXX"
  "Show me the details of the operation XXX"
  "Print the information of the operation XXX"
  "Tell me more about the operation XXX"
  "Show me the operation with the name XXX"
  "Tell me about the operation which has the ID XXX"
  "Show me the details of the operation which has the ID XXX"}
 creates context Operation {
  sets parameter operationName from fragment "XXX" (entity any)
 }
}
```

```
import platform "OpenAPIPlatform"
import platform "ReactPlatform"

on intent GetOperationByName do

 val operation = OpenAPIPlatform.GetOperationByName(context.get("Operation").
 get("operationName") as String)
 if(operation != null){//Display the details of the operation
    ReactPlatform.Reply("Here is what I found about the operation "
    +context.get("Operation").get("operationName"))
    ...
  }
 else {//Display an error message
   ...
  }
```

Fig. 1. A snippet of the definition of `GetOperationByName` intent.

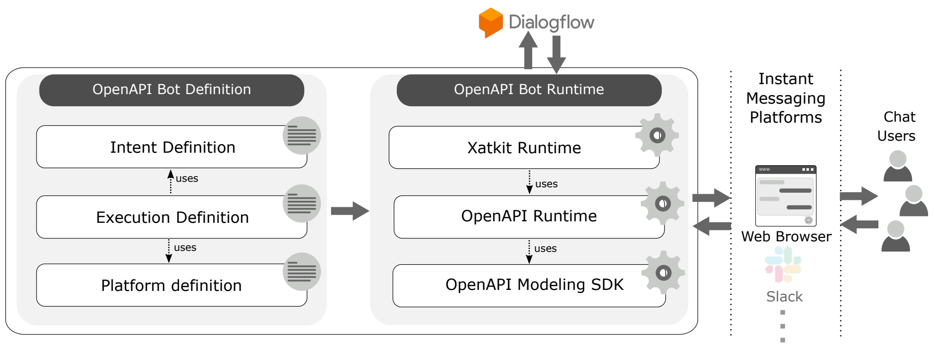
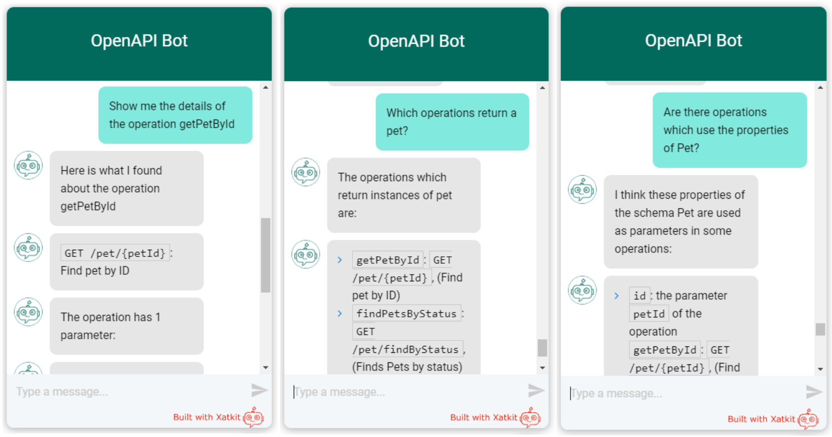


Fig. 2. OpenAPI Bot architecture overview.

3 Example

OpenAPI Bot is up and running at `https://som-research.uoc.edu/tools/openapi-bot/`. The bot is initially minimized. Clicking on the button (bottom-right side) opens a chat widget. To begin with, the bot asks the user to provide the URL of the OpenAPI definition she wants to learn about. After this, the user can start asking questions about the imported API. Figure 3 shows three interaction examples related to the Petstore API[5]. The first screenshot illustrates a simple question to know the details of the operation `getPetById`. Similar questions could be asked for the other parts of the API (e.g., the schema definitions, the metadata information, etc). The second and third screenshots illustrate two advanced questions to find which operations return instances of `Pet`, and which ones use the properties of the schema `Pet`, respectively. Getting this information by directly reading the OpenAPI definition is not trivial. Indeed, the OpenAPI Bot relies on a set of heuristics we implemented in the OpenAPI SDK to discover some advanced insights about OpenAPI definitions which are not obvious at first glance. See additional examples in the website.

[5] `https://petstore.swagger.io/v2/swagger.json`



Show me the details of the operation getPetById

Here is what I found about the operation getPetById

`GET /pet/{petId}` : Find pet by ID

The operation has 1 parameter:

Which operations return a pet?

The operations which return instances of pet are:

- `getPetById` : `GET /pet/{petId}` , (Find pet by ID)
- `findPetsByStatus` : `GET /pet/findByStatus` , (Finds Pets by status)

Are there operations which use the properties of Pet?

I think these properties of the schema Pet are used as parameters in some operations:

- `id` : the parameter `petId` of the operation `getPetById` : `GET /pet/{petId}` , (Find

Fig. 3. Interaction examples of OpenAPI Bot using the Petstore API.

4 Conclusion

In this paper, we presented OpenAPI Bot, a chatbot that leverages the OpenAPI specification (currently, the bot understands OpenAPIv2) to help developers understand REST APIs by asking questions on the API using Natural Language. Besides simple questions, OpenAPI Bot is able to provide some useful information which is not easy to infer from a more lengthy read at the specification. We are working on improving OpenAPI Bot by continuously monitoring how developers use it (e.g. to see what questions they are interested in that the bot is so far unable to answer). Also, we plan to support OpenAPI version 3 and explore how to use the bot as a new end-user interface to also execute calls on the API itself.

References

1. Cai, L., Wang, H., Xu, B., Huang, Q., Xia, X., Lo, D., Xing, Z.: AnswerBot: An Answer Summary Generation Tool Based on Stack Overflow. In: Proc of ESEC/FSE. pp. 1134–1138 (2019)
2. Daniel, G., Cabot, J., Deruelle, L., Derras, M.: Xatkit: A Multimodal Low-Code Chatbot Development Framework. IEEE Access 8, 15332–15346 (2020)
3. Ed-douibi, H., Cánovas Izquierdo, J., Bordeleau, F., Cabot, J.: WAPIml: Towards a Modeling Infrastructure for Web APIs. In: Int. Conf. on Model Driven Engineering Languages and Systems Companion. pp. 748–752 (2019)
4. Tian, Y., Thung, F., Sharma, A., Lo, D.: APIBot: question Answering Bot for API Documentation. In: Int. Conf. on Automated Software Engineering. pp. 153–158 (2017)
5. Vaziri, M., Mandel, L., Shinnar, A., Siméon, J., Hirzel, M.: Generating chat bots from web API specifications. In: Proc. of the Onward! pp. 44–57 (2017)