Architecture Design

ZOMATO RESTAURANT ANALYSIS AND PREDICT THEIR RATINGS

Revision Number -1.1

Last Date of Revision - 20/02/2024

RABIN LAMICHHANE

Document Control

DATE	VERSION	DESCRIPTION	AUTHOR
10/02/2024	1.0	Introduction, Architecture Deployment	RABIN
20/02/2024	1.1	Final Revision	RABIN

TABLE OF CONTENTS

1. INTRODUCTION

1.1 INTRODUCTION

Any software needs the architectural design to represent the design of the software. IEEE defines architectural design as “the process of defining a collection of hardware and software components and their interfaces to establish the framework for the development of a computer system.” The software that is built for computer-based systems can exhibit one of these many architectures.
Each style will describe a system category that consists of

- A set of components (eg: a database, computational modules) that will perform a function required by the system.
- The set of connectors will help in coordination, communication and cooperation between the components.
- Conditions that how components can be integrated to form the system.
- Semantic models help the designer to understand the overall properties of the system.

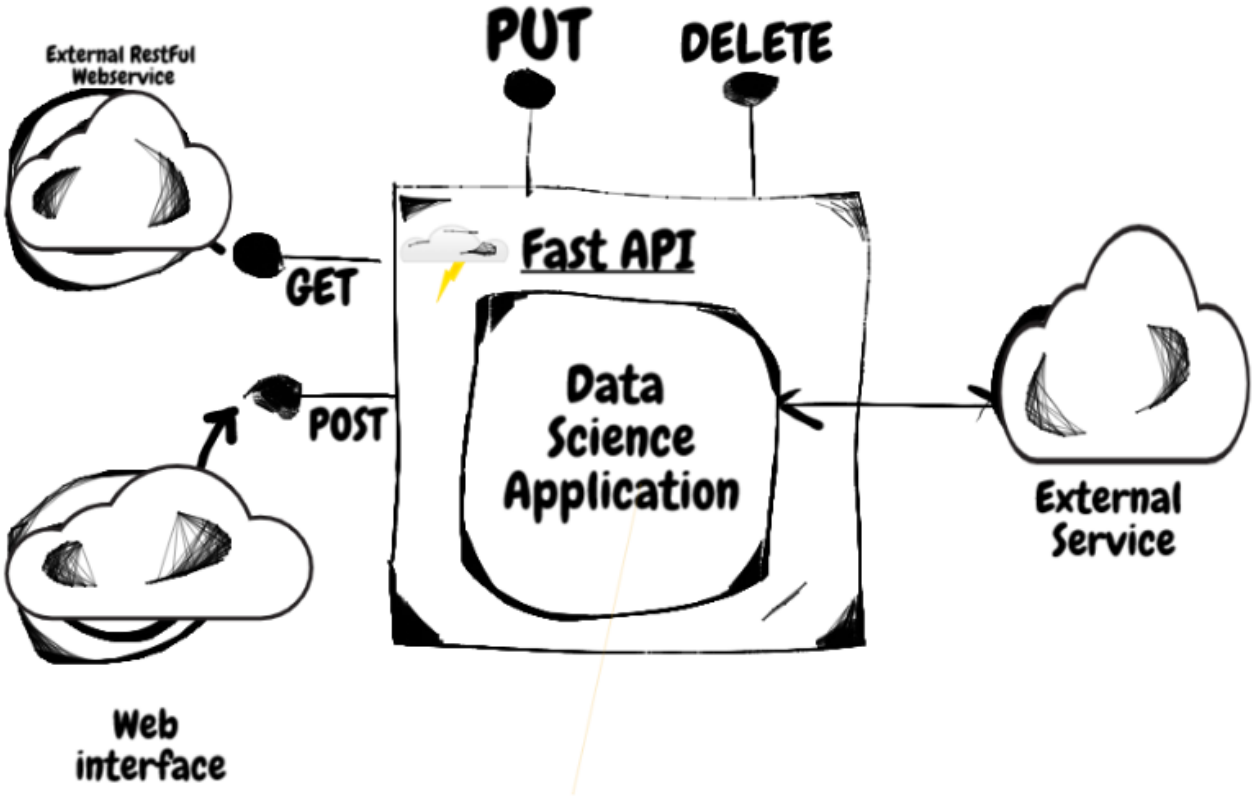
1.2 WHAT IS SCOPE

Architecture Design Document (ADD) is an architectural design process that follows a step-by-step refinement process. The process can be used for designing data structures, required software architecture, source code and ultimately, performance algorithms. Overall, the design principles may be defined during requirement analysis and then refined during architectural design work.

2. ARCHITECTURE

2.1 FASTAPI ARCHITECTURE

FastAPI is a modern web framework developed in Python. It has gained a lot of attraction in recent times. Its development structure is very much similar to Flask, which was the de-facto option for anyone starting with web development using Python. FastAPI is easy to use, its documentation is well written and easy to follow along. FastAPI offers all the standard features of an API building tool but it is not limited to this. It comes with a lot of flexibility such as backward WSGI mounting support which, most of the users, are not aware of.



2.1 COMPONENTS OF FASTAPI

1. **Include Flask, Django, Dash, or any other WSGI**

   FastAPI is an ASGI web framework. What this means is that different requests don't have to wait for the others before them to finish doing their tasks. Different requests can do their task completion in no particular order. On the other hand, the WSGI frameworks process request in a sequential manner. FastAPI allows any type of WSGI application like Flask to be mounted inside a FastAPI application. What this means is that on the root level, you can have the main FastAPI app and further, for different routes, you can have WSGI applications such as Flask which will handle all requests for that specific path.

2. **Mount different FastAPI apps for different routes**

   Following the same rules of mounting different WSGI applications for different routes, you can also mount different FastAPI applications within the FastAPI application. This would mean that every sub-FastAPI application would have its docs, would run independent of other applications, and will handle its path-specific requests.

3. **Automatic docs**

   Interactive API documentation and exploration web user interfaces. As the framework is based on OpenAPI, there are multiple options, 2 included by default.

3. DEPLOYMENT

3.1 FASTAPI DEPLOYMENT

The deployment process lets you clone content from one stage in the pipeline to another, typically from development to test, and from test to production.
During deployment, FAST API copies the content from the current stage, into the target one. The connections between the copied items are kept during the copy process. FAST API also applies the configured deployment rules to the updated content in the target stage. Deploying content may take a while, depending on the number of items being deployed. During this time, you can navigate to other pages in the FAST API portal, but you cannot use the content in the target stage.