Web Applications Architecture

Spyros Alertas

Created On: September 02, 2023
Last Edit: September 3, 2023

Contents

1 Introduction

This document is a brief description of the main parts that compose a web application and some of the different choices we have for each of them.

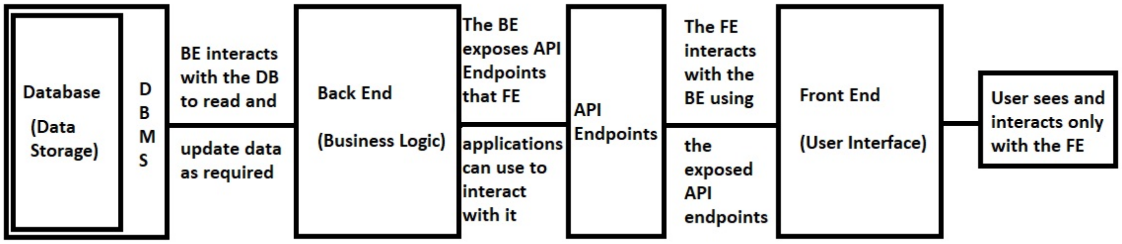
2 Main Parts of a Web Application

The main parts of a web application are:

- **Database:** Responsible for the storage of data.
- **Back End:** Implements the business logic.
- **API:** Defines the way Back End and Front End applications communicate.
- **Front End:** The user interface users see and interact with.

Flow description:

- Step 1. User interacts with the Front End application.
- Step 2. The Front End application sends request to the Back End application via API call.
- Step 3. The Back End application receives request from the Front End.
- Step 4. The Back End will retrieve and update the data in the database as required.
- Step 5. Once Back End has finished will return an appropriate response to the Front End application.
- Step 6. The Front End application will display the needed information to User.


Database
(Data Storage)
D B M S
BE interacts with the DB to read and
update data as required
Back End
(Business Logic)
The BE exposes API Endpoints that FE
applications can use to interact with it
API Endpoints
The FE interacts with the BE using
the exposed API endpoints
Front End
(User Interface)
User sees and interacts only with the FE

3 Database

3.1 Role of Database

The role of the database is to provide ways to the application to access and manipulate data. The database is accessed by Back End applications only. Front End cannot directly interact with the database.

3.2 Database Options

The two most popular categories of databases are: 1. SQL and 2. NoSQL. For more on SQL databases click here and for NoSQL click here.

Some popular SQL Databases:

- Oracle Database
- Microsoft SQL Server
- MySQL
- PostgreSQL
- SQLite

Some popular NoSQL Databases:

- MongoDB
- Apache Cassandra

4 Back End

4.1 Role of Back End

The role of the Back End is to implement the business logic of the application. It will expose API Endpoints which can be used by Front End applications. It is also responsible for the communication with the Database.

4.2 Back End Options

- Java + Sprint Boot
- JavaScript + NodeJS + ExpressJS
- Python + Flask/Django

4.3 Layers in Back End

I will use as example the architecture of a Java + Spring Boot Application.

- **Database Layer:** The actual database (MySQL/MongoDB/Oracle SQL/etc).
- **Persistence Layer:** Responsible for Data Persistence. Consists of **Entities** and **DAOs**.

  Entity: a lightweight persistence domain object. Typically, an entity represents a table in a relational database.

  Data Access Object (DAO): is responsible for two concepts:

One - Encapsulation of the details of the Persistence Layer and

Two - Providing CRUD interface for a single entity.

Note I: Persistence Layer (or else Repository Layer) is the layer that is responsible for data persistence, to access the cache and database.

Note II: CRUD stands for Create-Read-Update-Delete.

- **Business Layer:** Contains all the business logic. It is responsible for validation and authorization. Consists of **Services**.

  Service: -

- **Presentation Layer:** This is the top layer. It is responsible for handling HTTP requests and performing authentication. It is also responsible for handling the request (inputs) and the response.

  Controller: -

4.4 Authentication vs Authorization

Authentication and Authorization are two different things.

- **Authentication:** is the process of verifying who the user is.
- **Authorization:** is the process of checking if user is authorized to perform an action, i.e. retrieve some information, or delete some data.

5 API

5.1 Role of API

The API will define the rules of communication between the Back End and the Front End applications.
Note: API stands for Application Programming Interface.

5.2 API Options

- REST
- SOAP
- WebSocket
- GraphQL

6 Front End

6.1 Role of Front End

The Front End is the User Interface, it's what Users actually see and how they communicate with the Back End application. The Front End only communicates with the Back End applications via API calls.

6.2 Front End Options

- HTML and CSS are common no matter the framework you chose
- TypeScript + Angular (Web Application)
- JavaScript + React (Web Application)
- TypeScript + Ionic (PWA Angular Based)
- JavaScript + Native (PWA React Based)

Note: PWA stands for Progressive Web Application. It allows with a single code base, the application to run as web and android application at the same time. Android Users have the option to install the application like a normal android application however in reality it runs in browser in a standalone window instead of a browser tab.