Abstract

Ajax, JSON, Angular, Bootstrap, RWD, Node.js, and Ticketmaster API

Ajax, JSON, Angular, Bootstrap, RWD, Node.js, and Ticketmaster API

1. Objectives

1. To get experience with creating backend applications using JavaScript/Node.js on the server side with Express framework.
2. To get experience with using Angular, TypeScript, and Bootstrap on the client side and creating responsive front-end.
3. To get experience with using HttpClientModule of Angular for AJAX.
4. To get experience with Ticketmaster APIs, Spotify APIs, Google Maps APIs, Google Geocoding APIs, Ipinfo APIs, Facebook APIs, and Twitter APIs.
5. To get experience with Cloud Platform (GCP, AWS, Azure)

2. System Overview

The system contains three components: 1) browser (frontend), 2) Node.js application (backend) and 3) Ticketmaster servers. I have implemented both the frontend and the backend. Backend will include two major functionalities: serving the frontend static files to the browser and responding to the frontend's AJAX requests by fetching data from Ticketmaster servers. I did not directly call the Ticketmaster APIs from the frontend as it requires disclosing a secret API key to the public. The data flow diagram after an AJAX call is shown below in **Figure 1**.

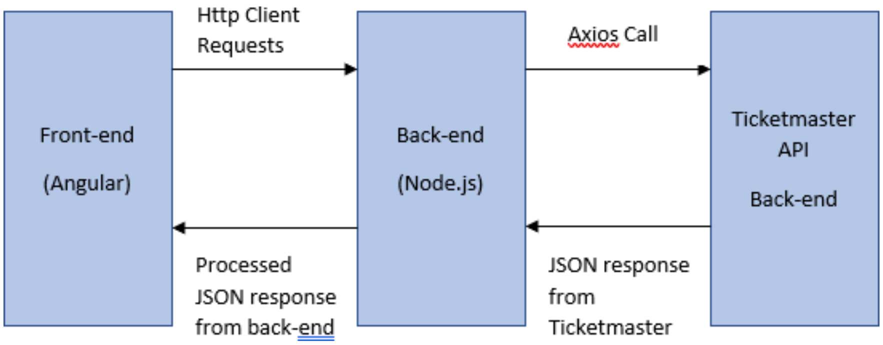

Figure 1: System design overview

NOTE: Setup authorization is required to call Ticketmaster API endpoints and is using request headers with the API key for the app. Refer to the Additional Hints section to see how to add authentication headers.

3. Description

In this exercise, I'll be creating a web application that allows you to search for event information using the Ticketmaster API, and the results will be displayed in a card in tabular format. The application will also allow users to mark events as "Favorites" and see the list of all events marked as favorites. Also, users can share a post on Facebook and a tweet on Twitter about the events.

All implementation details and requirements will be explained in the following sections.

There are 2 front-end routes/pages for this application:

a) Search Route ['/search'] – It is the default route of this application which is used to search for events and see event details
b) Favorites Route ['/favorites'] – It displays the list of favorite events

Implemented in Mobile View as well. Responsive