Developer's Guide - Tour Visualizer

By: Faris Ashriem, Levi McFetridge, Christian Deguzman

1. System Architecture Overview

Tour Visualizer uses a two-part architecture:
Frontend: Angular 20 + CesiumJS
Backend: .NET 8 API
Frontend responsibilities:

- UI (search, events, popups)
- Cesium rendering
- API communication via proxy

Backend responsibilities:

- Communicate with Ticketmaster API
- Return artist and event data to the frontend

2. Local Setup

Backend:
dotnet watch run --urls http://localhost:3000/
Frontend:
ng serve --proxy-config proxy.config.json
Proxy routes /api to backend.

3. Project Structure (Frontend)

src/app/

- pages/main-page/
  - cesium-component/
  - widget-component/
- core/services/
  - events.service.ts

4. Component Responsibilities

4.1 DisplayComponent

- Coordinates Widget and Cesium components
- Handles:
  - onSearch
  - flyToEvent
  - resetting arcs when changing artists
- Passes event objects into Cesium

4.2 WidgetComponent

Handles:

- Search bar
- Debounced input
- Popup results
- Event cards

- Emits search and selectEvent events

Popup auto hides on new search.

4.3 CesiumComponent

Responsibilities:

- Initialize Cesium viewer
- Fly to coordinates
- Draw arcs between events
- Remove old arcs when artist changes
- Display markers/entities

Stores arc entities for removal.

5. Cesium Features

5.1 Custom PolylineTrail Material

Used to animate glowing arcs.
Material type:
PolylineTrail
Custom class sets:

- color
- duration
- startTime

5.2 Drawing Arcs

Steps:

1. Convert lat/lon to Cartesian3
2. Generate arc positions
3. Add Cesium entity with PolylineTrail material

Arcs stored in an array and removed when needed.

6. Backend API

Endpoints
GET /api/events?attractionId=
Returns artist ID, name, images, and event details, including coordinates.

7. Data Flow Summary

1. User types search
2. Widget emits search
3. EventsService calls backend
4. User selects artist
5. DisplayComponent loads events
6. CesiumComponent renders events and arcs
7. User selects event
8. Camera flies to the location

8. Clearing Arcs

Example removal flow:
this.cesium.clearArcs();
this.cesium.drawArcsFor(events);
Entities removed using viewer.entities.remove.

9. Styling Notes
Widget positioning:
```
.widget {
 position: absolute;
 top: 20px;
 left: 20px;
 width: 320px;
 }
```
Globe always full-screen:
```
.globe {
 width: 100%;
 height: 100%;
 }
```