Sequence Diagram: View Available Events

This diagram models the behavior of the system when a user accesses the main event feed, either in list view or map view. It demonstrates the interaction between the user interface, controller logic, recommendation engine, database proxy, and event entity.

The diagram includes two alternative interface modes:

- Viewing events as a scrollable list
- Viewing events on a map interface

Each flow handles the case where events are either available or unavailable.

Scenario 1: Viewing Events in List View

The user initiates the flow by requesting to view the event feed. This request is handled by the `EventController`, which forwards the call to the `Recommendation` component. The recommendation engine calls `getEventFeed(accountUIPreferences)`, using stored user preferences to retrieve appropriate event data. The database proxy receives the request, creates and returns a collection of `Event` objects.

When the response is returned to the front-end, the system proceeds to the `ListGUI`. If the event list is not empty, the GUI calls `renderEvents(events)` and passes the data to `EventTabGUI`, which renders the list entries. If the event list is empty, the `ListGUI` calls `showEmptyListMessage()` to inform the user that no events were found.

Scenario 2: Viewing Events in Map View

The user alternatively chooses to view events on the map. This action begins when the user requests to view the event map. The system first calls `initializeMap(location)` through the `GoogleMapsController`, setting up the interactive map based on the user's location.

Once the map is initialized, `MapGUI` triggers a refresh to retrieve events by calling the same `getRecommendedEvents()` flow via the `EventController`, `Recommendation`, and `DBProxy`, resulting in a list of `Event` objects.

Back in the GUI, if the event list is not empty, the `MapGUI` passes the list to the `GoogleMapsController` using `refreshNearbyEvents()`, which renders the markers on the map. If the list is empty, the `MapGUI` instead calls `showEmptyListMessage()` to display a visual indicator that no events were found near the current location.