Course Project for Developing Scalable Apps

Project Motivation and Overview

Currently the Conference Central application is pretty limited - conferences have just name, description and date when the conference happens. But usually conferences have more than that - there are different sessions, with different speakers, maybe some of them happening in parallel! Your task in this project is to add this functionality. Some of the functionality will have well defined requirements, some of it will more open ended and will require for you think about the best way to design it.

You do not have to do any work on the frontend part of the app to successfully finish this project. All your added functionality will be testable via APIs Explorer.

You will have to submit your app ID, and text for the parts of the project that require explanation.

Task 1: Add Sessions to a Conference

Overview

Sessions can have speakers, start time, duration, type of session (workshop, lecture etc…), location. You will need to define the Session class and the SessionForm class, as well as appropriate Endpoints.

You are free to choose how you want to define speakers, eg just as a string or as a full fledged entity.

Define the following Endpoints methods

- getConferenceSessions(websafeConferenceKey) -- Given a conference, return all sessions
- getConferenceSessionsByType(websafeConferenceKey, typeOfSession) Given a conference, return all sessions of a specified type (eg lecture, keynote, workshop)
- getSessionsBySpeaker(speaker) -- Given a speaker, return all sessions given by this particular speaker, across all conferences
- createSession(SessionForm, websafeConferenceKey) -- open only to the organizer of the conference

Define Session class and SessionForm

In the SessionForm pass in:

- Session name
- highlights
- speaker
- duration
- typeOfSession
- date
- start time (in 24 hour notation so it can be ordered).

Ideally, create the session as a child of the conference.

Explain your design choices

Explain in a couple of paragraphs your design choices for session and speaker implementation.

Task 2: Add Sessions to User Wishlist

Overview

Users should be able to mark some sessions that they are interested in and retrieve their own current wishlist. You are free to design the way this wishlist is stored.

Define the following Endpoints methods

- addSessionToWishlist(SessionKey) -- adds the session to the user's list of sessions they are interested in attending

  You can decide if they can only add conference they have registered to attend or if the wishlist is open to all conferences.

- getSessionsInWishlist() -- query for all the sessions in a conference that the user is interested in

Task 3: Work on indexes and queries

Create indexes

Make sure the indexes support the type of queries required by the new Endpoints methods.

Come up with 2 additional queries

Think about other types of queries that would be useful for this application. Describe the purpose of 2 new queries and write the code that would perform them.

Solve the following query related problem

Let’s say that you don't like workshops and you don't like sessions after 7 pm. How would you handle a query for all non-workshop sessions before 7 pm? What is the problem for implementing this query? What ways to solve it did you think of?

Task 4: Add a Task

Overview

When a new session is added to a conference, check the speaker. If there is more than one session by this speaker at this conference, also add a new Memcache entry that features the speaker and session names. You can choose the Memcache key.

The logic above should be handled using App Engine's Task Queue.

Define the following Endpoints method

- getFeaturedSpeaker()