

AWS Final Project Checkpoint

Uvent | Tim Chen + Tal Davidi

Overview of progress

Over the past weekend, we attended YHack and worked intensively on our final project. This was a good opportunity to have a lengthy amount of time to dedicate towards the project and get a lot done. This checkpoint update will go into a high level overview of what we accomplished, but the Devpost for our project can be found here:

<https://devpost.com/software/uvent> for more details.

Components completed

- Daily web scraping background process (Lambda Function with Cloudwatch) - Each morning, Uvent will automatically update our event database with any new postings from the [UMD calendar](#) we're using as our dataset
- Frontend web interface (Hosted on Elastic Beanstalk) - We built nice a web interface for users to submit events they know about as well as scroll through our database of events.
- Publicly Exposed API (API Gateway and Lambda) - We created lambda functions to interact with our DynamoDB instance and exposed an API for anyone to play around with (and for us to use in the other components)
- Database (DynamoDB) - We decided on a schema and created the table
- Initial work on Alexa Skill (more in remaining work section)

Remaining work

We have started working on the Amazon Alexa Interface for our project. We'll be adding more intents and conversation starter options and just finishing this part in general.

Changes since proposal

We decided to take a step back and analyze the original scope of our project. We concluded that adding in the machine learning aspect of tailoring events to personal preferences would be a bit too ambitious (we discovered more complex issues than we initially anticipated such as persisting sessions and figuring out unique users, the accuracy of the categorization, and the amount of time the other parts took).

Challenges

We spent a good amount of time on the date/time issues. There were many formats of the dates we were working with, from strings to UTC vs non-UTC to Date types. With any date/time based project, there's a lot of converting between different formats, time zones, etc... which quickly gets frustrating. That took some time to get hashed out, but eventually we got it.

CORS also gave us some trouble with sending POST requests to our API. We were able to add a Access-Control-Allow-Origin fix that allowed us to send the requests.

Wifi was a huge issue when we were starting out (as with all hackathons). We had a hard time doing the AWS work because it would time out. We decided to switch over to frontend work, and even that was challenges (the CDNs weren't loading, so we just downloaded the actual files and later swapped it back when the wifi was better).

The Alexa skill itself ended up being more complex than we expected due to the nature of how we stored out data. Since we didn't want to repeatedly query the DB in between calls there needed to be some way to remember what happened in previous calls (which does not automatically happen). For this we had to implement significant logic to handle remembering past choices what the Alexa skill had already said.