**Web Summit Featured Attendees**

By Amit Sinha

**Prerequisite**

* Internet connection required
* You must have Node.js installed on your computer.
* The Node Package Manager (NPM) must be up to date.
* Before going any further, let’s take a look at our project structure:
* Websummit  
  —-> css

—-> fonts

—-> img

—-> js

——–>lib

—-> models

——–> attendees.js

—-> node\_modules

—-> routes

——–> attendeeslist.js

—-> app.js

—-> index.html

—-> package.json

* We’ve already taken care of **package.json** so let’s move onto the **app.js** file.  It is a wrapper for all server configuration and included files.  For example we’ll determine what port to use and what routes to include in this file.

Postman

Error handling

Responsive layout

**How can I run it?**

from nodeJs go to the folder where the application resides. Run:

nodejs app.js

it should start a webserver on 127.0.0.1:8000

There are 3 restapi in JSON Format, you can use postman to validate the restapi:

<http://127.0.0.1:8000/attendeeslist> - It returns all the results

<http://127.0.0.1:8000/attendeeslist>?page=1&size=100 - It returns 100 results

<http://127.0.0.1:8000/attendeeslist/attendees/:ID> – Each Attendee are returned based on ID

**Technologies used**

* MongoDB
* NodeJS
* JQuery Bootstrap CSS

**Quick overview**

* Nodejs is responsible for API, it can get users by page size 25,
* converted the CSV file to JSON file - it's simplier and way more structurized

**Task overview and status**

* The API can be created in whatever language you want. -- Implemented
* The API should be RESTful and the results can be returned in whatever format you think appropriate. -- Implemented
* There should be an endpoint to return a list of all attendees.
  + It should support pagination of results and the default page size should be 25. -- Implemented
  + The call should return the minimal information that you need to render your list of attendees. -- Implemented
* There should be an endpoint to return the full detail for a single attendee keyed by their id. -- Implemented

**Use Case**

* Per page 25 attendees displayed
* List sorted on the base of “importance”, higher importance would be shown first
* Displaying 5 attendees per row.
* Hovering on the overlay div of attendees would show limited information about the attendee
* Clicking on the attendees would pop up the complete information about the attendee by making a new call to the API
* Pagination <previous> and <next> would behave as expected