[Durham](http://www.dur.ac.uk/) / [Computer Science](http://www.dur.ac.uk/computer.science/) / [s.p.bradley](http://www.dur.ac.uk/s.p.bradley) / [teaching](http://www.dur.ac.uk/s.p.bradley/teaching) / [WP](http://www.dur.ac.uk/s.p.bradley/teaching/WP) / assignment2017

**WP: DRAFT Web Programming Assignment DRAFT**

**Dates and weightings**

* This assignment contributes 25% to the overall mark for the module, and 50% of the coursework mark for the module.
* It should be submitted through duo by 14.00 on Wednesday 13th December 2017. Marks should be returned to you by 15th January 2018.

**Scenario: Events**

* People attend events at venues at certain times. You need to write a web app to allow a list of events to be searched and updated
* Provide a nodejs web service and a single page web app to access it
* A separate service provides authentication (part of a microservices architecture)
* Integrate an external web service to provide more event data

**Tasks**

* You need to demonstrate your knowledge and skills under the following headings.
* Complete them one at a time, in whichever order you prefer.
* Credit will be given for working code that meets some or all of the requirements listed below.

**Locally hosted nodejs web service for events (30%)**

* Develop a nodejs web REST service that can run on the local machine
* The base URL for this web-service should be

http://127.0.0.1:8090/events2017

* It is important that you get this right as marking the web service will be automated; you will lose marks if it is not in the correct place
* I will provide some simple tests for the web service
* In the following requirements this URL will be referred to as BASE
* All responses should be in JSON format and labelled as content-type

application/json

* Initially the auth\_token (when required) should be "concertina", this will be expanded upon later on
* Storage of events can be via flat files or a database, your choice

concTisim\_129.899.13.12

**Events API definition**

GET BASE/venues

* List all the venue details
* No parameters
* Example response. Your response must follow this structure exactly, but with your data in it
* {
* "venues":{
* "v\_1":{
* "name":"Grinton Lodge Youth Hostel",
* "postcode":"DL11 6HS",
* "town":"Richmond",
* "url":"http://www.yha.org.uk/hostel/grinton-lodge",
* "icon":"http://www.yha.org.uk/sites/all/themes/yha/images/logos/yha\_header\_logo.png"
* },
* "v\_2":{
* "name":"Sage Gateshead",
* "postcode":"NE8 2JR",
* "town":"Gateshead",
* "url":"http://www.sagegateshead.com/",
* "icon":"http://www.sagegateshead.com/files/images/pageimage/1683.7123dea7/630x397.fitandcrop.jpg"
* }
* }
* }

GET BASE/events/search

* Parameter search url-encoded string to be used to search event title (optional)
* Parameter date url-encoded string representing the date to search for (optional)
* Example response
* {
* "events":[
* {
* "event\_id":"e\_1",
* "title":"Swaledale Squeeze 2018",
* "blurb":"The biggest and best concertina weekend in the world. Held each May in Grinton Lodge YHA, North Yorkshire",
* "date":"2018-05-21T16:00:00Z",
* "url":"http://www.swaledalesqueeze.org.uk",
* "venue":{
* "name":"Grinton Lodge Youth Hostel",
* "postcode":"DL11 6HS",
* "town":"Richmond",
* "url":"http://www.yha.org.uk/hostel/grinton-lodge",
* "icon":"http://www.yha.org.uk/sites/all/themes/yha/images/logos/yha\_header\_logo.png",
* "venue\_id":"v\_1"
* }
* }
* ]
* }

GET BASE/events/get/:event\_id

* Parameter event\_id in URL (required)
* Single event object returned, in same format as elements in search result list
* Example response (if event\_id not defined or incorrect)
* {
* "error": "no such event"
* }

POST BASE/venues/add

* Parameter auth\_token (required)
* Parameter name text value (required)
* Parameter postcode text value (optional)
* Parameter town text value (optional)
* Parameter url text value (optional)
* Parameter icon text value of url (optional)
* Example response (if auth\_token not defined or incorrect)
* {
* "error": "not authorised, wrong token"
* }

* Similar error response if note all required parameters are present
* Response code of 400 should be set in case of error

POST BASE/events/add

* Parameter auth\_token (required)
* Parameter event\_id text value (required)
* Parameter title text value (required)
* Parameter venue\_id text value (required)
* Parameter date must accept ISO8601 (required)
* Parameter url text value of url (optional)
* Parameter blurb text value (optional)

**Single page HTML web app to access events (10%)**

* Should be located at BASE/index.html
* Initially a search form is provided, allowing user to choose keywords and/or dates
* When search results are produced, they should be listed under the search box with title, date and venue name
* When a search result is clicked, full details of the event and venue should be displayed
* All results should be accessed via the above web service with AJAX
* Response from service should be converted to HTML and rendered into the page

**Authentication (10%)**

To be used for admin

* Design your own authentication web service in nodejs
* Provide a POST method which takes username, password and IP address issues an auth\_token for that IP address which lasts for two hours
* Provide a GET method which takes an auth\_token and IP address and returns whether or not the token is valid
* The auth\_token "concertina" should be valid for all times for IP addresses 129.234.xxx.yyy

**Single page HTML web app to administer events (10%)**

* Should be located at BASE/admin.html
* List and add venues
* Add events to a selected venue
* If auth\_token is not defined then forward to a login page
* Save auth\_token as cookie after login

**Responsive design (10%)**

* Ensure that your app page and admin page are accessible and responsive on mobile devices: use a collapsing menu bar for access to event search or admin page
* Use a front end framework to mark up your code, e.g. [bootstrap](http://getbootstrap.com/), [semantic-ui](http://semantic-ui.com/)
* Use a grid layout for your app page

**Use an events web API (10%)**

* Use an existing events web service such as <http://api.eventful.com/docs>
* Could use a web service directory such as <https://www.programmableweb.com/>
* Combine your local events with events recovered from the remote web service
* Choose a particular focus for your events (e.g. concertinas) and try to make sure that you request only events associated with that
* Only include events from the UK

**Cloud deployment (10%)**

* Deploy your app and the associated services on a free cloud platform e.g. Red Hat OpenShift, IBM Bluemix
* Include a link to your app
* <https://developer.ibm.com/answers/questions/417094/error-404-not-exist-and-npm-errors-awake-01-condit.html>

https://salty-atoll-51159.herokuapp.com/

**Video Presentation (10%)**

* Submit a 2 minute (max) video demonstrating your software
* Include demonstration of how to start the program
* All functionality apart from web service (computer-marked) will be assessed by what is demonstrated in the video
* If it is not demonstrated in the video, you will not get a mark for it
* Quality of video presentation will be marked separately from functionality
* Lose 10% of marks for every 10 seconds over 2 minutes

You need to submit all of your source code and a link to the working application. Throughout a portion of the marks will be awarded for good design and coding practice. All code will be analysed for similarity by a plagiarism detection tool. It is appropriate to use and submit libraries as long as these are identified as such in the source code.

[Durham](http://www.dur.ac.uk/) / [Computer Science](http://www.dur.ac.uk/computer.science/) / [s.p.bradley](http://www.dur.ac.uk/s.p.bradley) / [teaching](http://www.dur.ac.uk/s.p.bradley/teaching) / [WP](http://www.dur.ac.uk/s.p.bradley/teaching/WP) / assignment2017

*Steven Bradley 2017-10-17*