Eventlite REST interface – acceptance tests for week 9.

User stories

- "As an event organiser, I want to be able to embed information about my event in my own website. I want to do this by getting live data from Eventlite in JSON format."
- "As a venue owner, I want to be able to have a live list of the *next three events coming up* at my venue on the venue's website."
- "As a developer tasked with integrating Eventlite data into another application, I want all REST responses to be valid JSON, have a media-type of "application/json", and include appropriate links to all resources, so that my code can successfully navigate Eventlite starting at the homepage."

Specification by example

The following examples have been derived from the above user stories to guide Eventlite developers and ensure a consistent REST interface across all versions of Eventlite.

Note: The following JSON examples are to demonstrate the format so there are placeholders used (<>) instead of real data. The order of fields in JSON is not important. Also note that for URIs the presence or absence of a slash (/) at the end is equivalent, so either is acceptable.

Homepage JSON

Running:

```
$ curl -H “Accept: application/json” http://localhost:8080/api
```

Returns:

```
{
  "_links" : {
    "venues" : {
      "href" : "http://localhost:8080/api/venues"
    },
    "events" : {
      "href" : "http://localhost:8080/api/events"
    },
    "profile" : {
      "href" : "http://localhost:8080/api/profile"
    }
  }
}
```

Don’t worry if you have implemented other entities that show up within the homepage JSON, but events and venues **must** be present.

Events list JSON

Running:

```
$ curl -H "Accept: application/json"
http://localhost:8080/api/events
```

Returns:

```
{
  "_embedded" : {
    "events" : [ {
      <an event>
    }, {
      <another event>
    } ]
  },
  "_links" : {
    "self" : {
      "href" : "http://localhost:8080/api/events"
    }
  }
}
```

Event JSON

Running:

```
$ curl -H “Accept: application/json”
http://localhost:8080/api/events/<id>
```

Returns:

```
{
  "date" : "<date>",
  "time" : "<time>",
  "name" : "<name>",
  "_links" : {
    "self" : {
      "href" : "http://localhost:8080/api/events/<id>"
    },
    "event" : {
      "href" : "http://localhost:8080/api/events/<id>"
    },
    "venue" : {
      "href" : "http://localhost:8080/api/events/<id>/venue"
    }
  }
}
```

Venues list JSON

Running:

```
$ curl -H “Accept: application/json”
http://localhost:8080/api/venues
```

Returns:

```
{
  "_embedded" : {
    "venues" : [ {
      <a venue>
    }, {
      <another venue>
    } ]
  },
  "_links" : {
    "self" : {
      "href" : "http://localhost:8080/api/venues"
    },
    "profile" : {
      "href" : "http://localhost:8080/api/profile/venues"
    }
  }
}
```

Venue JSON

Running:

```
$ curl -H "Accept: application/json"
http://localhost:8080/api/venues/<id>
```

Returns:

```
{
  "name" : "<name>",
  "capacity" : <capacity>,
  "_links" : {
    "self" : {
      "href" : "http://localhost:8080/api/venues/<id>"
    },
    "venue" : {
      "href" : "http://localhost:8080/api/venues/<id>"
    },
    "events" : {
      "href" : "http://localhost:8080/api/venues/<id>/events"
    },
    "next3events" : {
      "href" : " http://localhost:8080/api/venues/<id>/next3events"
    }
  }
}
```

The output of "next3events" should be modelled on the JSON for a list of events, and contain the next three events coming up for that venue. If there are fewer than three events coming up then the list will be shorter.