API as2 part2

What were some of the alternative design options considered? Why did you choose the selected option?

1. One of the alternative design options is using xml rather than using jSON object in my return response. For me, I choose jSON object because the data format of jSON is relatively simple, easy to read and write, and the format of json is compressed and easy to parse the language.
2. Another alternative design consideration is that when the request is made, we can either put the data in the request header or simply add the data to the URL. I chose to append the data to the URL because the format of our request parameters is relatively simple, and in this case it is simpler and more convenient to put the data directly in the url. When the data format is more complex, it is more appropriate to put the data in the request header.

What changes did you need to make to your tests (if any) to get them to pass?Why were those changes needed, and do they shed any light on your design?

1. The default port for the rest of the guarantees is 8080, while the default port for sparkJava is 4567. To ensure that the test cases are fired as expected, I changed the default port for sparkJava to 8080. This did not affect my tests.
2. The first version of my URL used '? to convert data, e.g.. /addEvent?aventInfo={eventInfo}. But I changed it to /addEvent/{eventInfo} as this is more applicable to applying sparkJava framework to transfer my data more conveniently. This also doesn't affect my testing and design.

Pick one design principle discussed in class and describe how your design adheres to this principle.

When designing, I follow the following principle:

Operations on resources	
Create	`http post`
Retrieve	`http get`
Update	`http put` (or `post`)
Delete	`http delete`
etc.	

For add / insert (create) operation, I used post method to request the server. For any function that retrieve/ get information, I used get method. For modify / update operation, I used put method. For delete operation, I used delete method to request the server.