Logic implementation

A report by Mark Erlich

Introduction

As discussed in last week meeting one of the most urgent and important features that our client needs are a time measuring system, for measuring the amount of time on each question, measuring time between mouse clicks or maybe keyboard buttons that were pressed and so forth. The exact use of the time measurement need to be clarified more specifically with the client. In this report I will cover (and show in practice) a straightforward way for event handling and taking timestamps using libraries and interfaces in JavaScript.

Taking timestamps

The point in tracking the time is to be as accurate as possible. So, to fulfill this target we can use the "performance" interface that delivers this ability. To support this claim, Google analytics, a service that provides tools for tracking user experience inside a business website, recommends using this API in their guides for time measurements.

So back to performance API, this is a JavaScript interface which provides access to related information to the current web page. An object of this type can be obtained by calling to Window.performance, which is a read only attribute.

The most helpful method of this interface for our project, as I see it, is Performance.now(). This method returns the number of milliseconds elapsed since the page started loading. This is a very important attribute for our project, because this output could be used for the whole time measuring system. When we can gain this kind of output, we can use it to take timestamp when a special event occurs, like a mouse click or a keyboard press (event handling of this kind will be explained in more detail later). Also, the timer is individual to each web page and we could also use this for our project.

So, to summary this part, using a simple and accurate interface in JavaScript we can take timestamps and to build the time measuring system.

jQuery and event handling

So, before I go into detail in how we can do event handling a little explanation on jQuery.

What is jQuery? This is small, fast and feature- rich JavaScript library. It makes HTML document traversal and manipulation, event handling and more very easy – to – use.

Advantages of jQuery:

- Very popular framework and there is a lot of information about her due a extensive use by all the major companies – Google, Microsoft, IBM and so forth.
- As said before easy – to – use.
- Supported by all major web browsers.

Adding jQuery is very easy and could be made by one simple line of code.

```
<script src="http://code.jquery.com/jquery.min.js" type="text/javascript"></script>
```

So Another issue that came up in the meeting is how to capture mouse click in every point on the screen or in other words how to capture the user's experience while he is filling the form. In this matter we will use event handling via jQuery.

jQuery offers a lot of methods to register behaviors of the user when he interacts with the web browser. The two methods that will concern us the most are click for capturing mouse clicks and keyup for capturing keyboard press.

We can capture the events on certain parts of the web page, and in part of them to ignore. If we would like to capture mouse clicks in all parts of the web page we could use the line:

$('html').click(…)

Or if we want to capture a mouse click on a specific button or paragraph we could do it like this:

$('#element id).click(…)

*note: the same applies for a keyboard press only the function is keyup.

So, the input for the click and keyup function is a function that gets an event object. This function, which I will refer as the event handler, will be activated every time when the specified event occurred on the specified element. The event object is guaranteed to be passed to the event handler. This object holds numerous properties that could be helpful. Two of them is pageX and pageY which gives the specific coordinates of the event in the web page, this could be very useful and should be considered.

So, to summarize this part, jQuery is a very useful and easy library. Using the event handling mechanism could be very helpful and could be used to track the user's interaction inside the web page the whole time of the test. The event handling could be integrated with taking timestamps when we could take timestamps in specific mouse clicks or keyboard pressing as the client will require.

Demo

To clarify how those things could help us I will show you a little demo:

This great website(😉) consists of two files: one.html, two.html.

one.html:

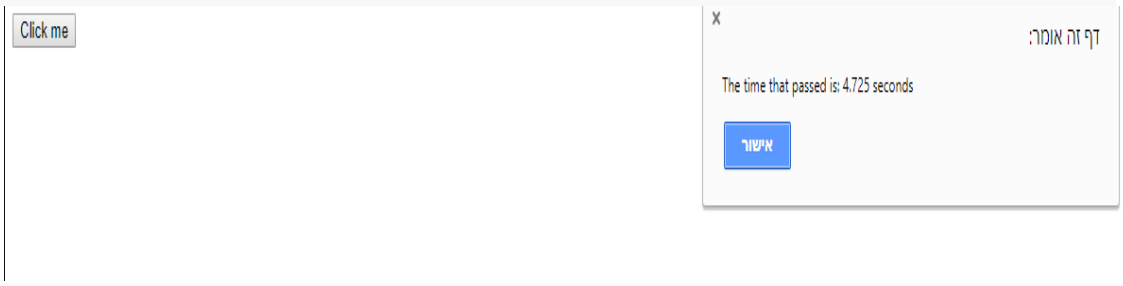
```
1.  <html>
2.  <head></head>
3.  <body>
4.      <input type="button" value="Click me" id="button" onClick="return
    btn_onClick()"/>
5.
6.      <script src="http://code.jquery.com/jquery.min.js" type="text/javascript"></scrip
    t>
7.
8.       <script type="text/javascript">
9.           $(document).ready(function(){
10.              $('html').click(function(event){
11.                  var loadTime = Math.round(performance.now())
12.                  alert('The time that passed is: ' + loadTime/1000 + ' seconds');
13.              });
14.              $('html').keyup(function(event){
15.                  alert("keyboard event: key pressed "+event.keyCode);
16.              });
17.          });
18.      </script>
19.      <script type="text/javascript">
20.      function btn_onClick(){
21.          window.location.href = "two.html";
22.      }
23.      </script>
24.
25. </body>
26. </html>
```

two.html:

```
1.  <html>
2.  <head></head>
3.  <body>
4.      <script src="http://code.jquery.com/jquery.min.js" type="text/javascript"></scrip
    t>
5.      <script type="text/javascript">
6.          $(document).ready(function(){
7.              $('html').click(function(event){
8.                  var loadTime = Math.round(performance.now())
9.                  alert(loadTime);
10.             });
11.         });
12.     </script>
13.
14.
```

As you can see adding jQuery is easy and can be done by one line of code. So I'm using event handlers with jQuery to capture the mouse or the keyboard click, and every time I click the mouse in every point on the screen, a pop-up

jumps with the time since the page first loaded (As presented in the figure below)



When I click the button, and move to second.html the timer resets. This can be very helpful when a user moves to another question in the form.

So as can be seen using jQuery is very easy and short, and using performance API could be very helpful for the time measuring feature.

In addition, using those JavaScript libraries and API's will not affect the fact that we are considering using Angular and TypeScript because the two can be integrated easily.

So, we need to determine what are the specific things that we are measuring, for instance – how many mouse clicks, the time between following mouse clicks or keyboard press and other technical details that needs to be determined with the client thoroughly. But using jQuery and performance API we help us to do it in a very easy way.

References –

https://developer.mozilla.org/en-US/docs/Web/API/Performance Performance API

https://jquery.com/ - what is jQuery?

https://www.w3schools.com/jquery/ advantages of jQuery

https://api.jquery.com/category/events/ - types of events

https://api.jquery.com/click/ - click event

https://api.jquery.com/category/events/event-object/ - The event object