From: rshankar@arra.ooo
To: vinayar@arra.ooo, shalinibr@arra.ooo
Cc: chinthana@arra.ooo
Subject: Please study this code and the comments. I will explain as needed
Date: Tue, 26 Dec 2017 04:09:27 -0700 (*12/26/2017 04:39:27 PM*)

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript Functions</h2>

<p>Match Scheduler</p>

<p id="demo"></p>

<script>

/* var schedulerInput = '{
  "entries": [
    {"U14Boys":"64"},
    {"U14Girls":"32"},
    {"U17Boys":"64"},
    {"U17Girls":"32"},
    {"U19Boys":"64"},
    {"U19Girls":"32"}
  ],
  "Durations": [
    {"Finals":"30"},
    {"SF":"30"},
    {"QF":"20"},
    {"PQF":"20"},
    {"R32":"20"},
    {"R64":"20"},
    {"R128":"20"},
    {"default":"20"}
  ],
  "DatesAndTimes": [
    {"2Jan18":["900", "1300", "1400"]},
    {"3Jan18":["900", "1300", "1400"]}
  ],
  "DatesAndResources": [
    {"2Jan18":"33"},
    {"3Jan18":"29"}
  ],
  "DatesAndEvents": [
    {"2Jan18":[{"U14Boys":"2"}, {"U14Girls":"2"}]},
    {"3Jan18":[{"U14Boys":"2"}, {"U14Girls":"1"}]}
  ]
}';
*/

/*
  Note: DatesAndResources is a map of # of tables available for the
specified date
```

```
  */

function getNext2PowN (e) {
  alert("We have totally " +e + " entries in this category!");
  var x = 0;
  var v = 1;
  while (e > v) {
    v = v * 2;
    if (e == v) {
      alert ("Returning from here!");
      return e;
    }
  }
  return v;
}

function getNumMatches (e) {
  var rname = ["Finals", "SF", "QF", "PQF", "R32", "R64", "R128", "R256",
"R512"];
  var matches = {};
  var n = getNext2PowN (e);
  var diff = n - e;
  var tmp = (n/2) - diff;
  var p = 1;
  var i = 0;
  while (p*2 < (e-1)) {
    matches[rname[i]] = p;
    p = p*2;
    i ++;
  }
  matches[rname[i]] = tmp;
  return matches;
}

function getNumMatchesByEvent (t) {
  var entries = /*schedulerInput[t]; eg. "U14Boys"; */ 21;
  var numMatchesPerRound = getNumMatches (entries);
  for (var rounds in numMatchesPerRound) {
    alert (rounds + " has " + numMatchesPerRound[rounds] + " matches");
  }
  return numMatchesPerRound;
}

function scheduleBuilder (schedulerInputData) {
  /*
    1. Get Num Matches by Event for each event
    2. Form a json with this pseudocode -
        class fullMatchListPerEvent {
          map <RoundName, [matchIdsWithinTheEvent]>;
        }
       With this, you will get an array of matchIds per event
    3. Now, suppose user inputs order as U14Boys.R32, U14GirlsR32,
U14BoysR16, U14GirlsR16 for day-1.  We have to simply concate all the
matchIds for U14Boys.R32, U14GirlsR32, U14BoysR16 and U14GirlsR16 into
one single array.  Then, we have to assign each match in the array to
each tableNumber available for that day (table # starts from 1) and
space it out by # minutes allocated for that round.  If we hit the
break time, we should skip that time and start again at the break-end
time.
    4.  Like this, we should for matchId arrays for each day and simply
allocate them in a loop for each table.
    5. Note that the number of tables may vary for each day
    6. It is better if we take inputs as 'session' rather than 'day'.
Session is a class object containing 'date', 'startTime' and
'endTime'.  With this approach, we can reduce the complexity in '3'
above and also allow for finer control for user in fixing many things.
   */
}

document.getElementById("demo").innerHTML = getNumMatchesByEvent("j");
</script>

</body>
</html>
```