CONTENTS OF THIS FILE
----

* Introduction

* Configuration

Internation connection because using bootstrap CDN link for front end so you can see the ouput as well from CGI.
if php 7.0 installed then just put to the web directory such as /var/www/html and then run http://localhost/boarding

Introduction

Our Files structure is ,

1. index.php Where you can test through CGI interface typing in you browser (http://localhost/boarding).

2. /application/controller/controller.php => Where controller reside

3. /test/PassengerBoardingSorterTest.php => For testing purpose

4. /test/UniteTest.php => Simple class for unite test

I would like to explain each file how it's working

index.php

I am using for frontend simple bootstrap template. On the line 27 I have set some unsorted simple data you can see the blow.

```
$cards   = [
                                        [
                                                  "from" => "Barcelona",
                                                  "to" =>  "New York",
                                                  "instruction" => "It will containe information",
                                                  'time' => '2018-02-02 20:05',
                                                  'transport' => 'Flight' ,
                                                  'transportno' => 'B33',
                                                  'seatno' => 'Y15'
                                        ],

                                        [
                                                  "from"=> "Barcelona",
                                                  "to" => "Gerona",
                                                  "instruction" => "It will containe information",
                                                  'time' => '2018-02-02 20:05',
                                                  'transport' => 'Bus' ,
                                                  'transportno' => 'M31, M32, M33','seatno' => 'Any'
                                        ], // 1

                                        [
                                                  "from" => "Madrid",
                                                  "to" => "Barcelona",
                                                  "instruction" => "It will containe information",
                                                  'time' => '2018-02-02 20:05',
                                                  'transport' => 'Bus' ,
                                                  'transportno' => 'M31, M32, M33',
                                                  'seatno' => 'Any'
                                        ],

                                        ["from" => "New York",
                                        "to" => "Stockholm",
                                        "instruction" => "It will containe information",
                                        'time' => '2018-02-02 20:05', 'transport' => 'Bus' ,
                                        'transportno' => 'M31, M32, M33','seatno' => 'Any'
                                        ], // 0

                                        [
                                                  "from" => "Gerona",
                                                  "to" => "Barcelona",
                                                  "instruction" => "It will containe information",
                                                  'time' => '2018-02-02 20:05',
                                                  'transport' => 'Bus' ,
                                                  'transportno' => 'M31, M32, M33',
                                                  'seatno' => 'Any'
                                        ], // 2
                    ];

                    // Sort the bording list using Static class
                    $sortBoardingList = PassengerBoardingSorter::ArangeBordingList('from', 'to', $cards, 'instruction');

                    // Form a template so you can get template list
                    $template = '<li class="list-group-item">{{&instruction}}Take {{&transport}} {{&from}} To {{&to}}. Sit in seat
{{&seatno}}</li>';

                    // Regression for to getting the value from array
                    $regex = '/{{&(.*?)}}/';

                    $getTemplate = PassengerBoardingSorter::getTemplate($template,$sortBoardingList ,$regex);



           ?>

          <!--- Getting each data to view -->

          <?= $getTemplate ?>
```

I would like to explain each line and it's dependencies that I am using. I have not used any framework.

$sortBoardingList variable setting the static object which is located in application/controller/PassengerBoardingSorter.php

In this class I have wrote static method ArangeBordingList which accept,

arguments 4 (string $index, string $index2, array $boardingLists, string $instruction) it can be many or can be reduced as per our needs and requirement but for a while I am using for arguments. And the value of arguments is,

$index => from where traveler traveling

$index1 => to which place traveler traveling

$boardingLists => not arranged boarding lists for example one array column. These column name can be anything as per your need but which setting value to $sortBoardingList = PassengerBoardingSorter::ArangeBordingList('from', 'to', $cards, 'instruction'); you can see here I have used what I have used and data input but for good understanding from, to column is presented. On column is updated by the script therefore instruction arguments is as well presented

```
        [
                "from" => "Barcelona",  // From where travelling
                "to" =>  "New York",  // To where travelling
                "instruction" => "It will contain information",  // Any instruction for traveller
                'time' => '2018-02-02 20:05', // Time to travel
                'transport' => 'Flight' , // Bus, Train, Boat, Flight Whatever
                'transportno' => 'B33', // Transport not
                'seatno' => 'Y15' // // Set No
        ],
```

$instruction => Some message you can set which setting data and here message will get append when script runs and detect where journey start and ends at that point message will get append for example "Your journey start from here, You journey end here " something like this.

Now what ArangeBordingList static method does. You can simple see the file /application/controller/PassengerBoardingSorter.php here I am sorting the array column using usort function where it's search first column name. Measuring one array index value to another as you can see below

```
// Get access to each index
                        foreach($boardingLists as $boardingList)
                        {

                                usort($boardingLists, function ($a, $b) use ($index, $index2){

                                return ($a[$index2] === $b[$index] ) ? 0 : 1;
                        });
                        }
```

Its returning arranged array column as I have set the data above. Another part

```
$template = '<li class="list-group-item">{{&instruction}}Take {{&transport}} {{&from}} To
{{&to}}. Sit in seat {{&seatno}}</li>';
```

After getting return arranged associative array from the ArangeBordingList method now it's time you can show these data to the client side. You can simple irritate array and who to HTML document has well but I have used OOP form so you can get the data to your HTML document. As you have set all the array column you can see above $template variable contained name column name which we are getting in our fornt end part make esire to manupulate that from array column.

I wrote getTemplate method which search for the column and replace the value to our html elements. As one array column values coming in on li list. Set you array column list in this form {{&[column_name]}} you can change this regression to any other form as well.

4. Testing

I could have use phpunite but thought it would be good If I write everything from scratch. For this particular task, testing is simple for understanding in this travel boarding list

```
$boardingList[$i]['to'] = $boardingList[$i + 1 ]['from']
```

```
$countColumn = count($bordinglist);

                    // Is all good
                    $isAllGood = true;

                    for($i = 0; $i < $countColumn-1 ; $i++)
                    {
                              if($bordinglist[$i][$to] !== $bordinglist[$i + 1][$from])
                              {

                                        // Change variable
                                        $isAllGood = false;

                                        // Break
                                        break;
                              }
                    }
```

as you can see above. You can use from command line as well go to directory /test/ and then type php PassengerBoardingSorterTest.php or else through CGI interface go to

/test/PassengerBoardingSorterTest.php it will give you message if test is success then OK (1 test, 1 assertion) or otherwise it will generate error.

Thank you for reading. At the moment I have to only this much to write if got chance then I could write more and more. If you have any question then please let me know about these things. For any inconveniences I would like to apologiz.

Regards
Bharat Shah