**Technical problems and solution**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SL# | Issue | Problems | Solutions | Note |
| 1. | Layout-align is not working | Where ‘layout-align’ works and where not? | 1. ‘layout-align’ works where we define ‘layout=row/column’  https://material.angularjs.org/latest/layout/alignment |  |
| 2. | Scroll behaving improperly | get Scroll when needed in a portion of whole page? | use proper height , class=”scollable” and ms-scroll |  |
| 3. | Inner async call | maintain sequence in case of two or more async functions? | call second async functions inside callback of first async function |  |
| 4. | Object reference | assign a variable with the value of another variable without reference? | Var2 with new reference  var2 = angular.copy(var1);  var2 with same reference  var2 = var1 |  |
| 5. | Include updated data in previously bound data set | replace an obj without changing reference | angular.copy(source,destination)  //destination is the bound data set  // source is the updated data | Dont use this;  Arr.splice(0,arr.length())  newarr.forEach(arr.push(item)) |
| 6. | Alignment | layout-align | Layout-align depends on defined layout if  layout = row then layout-align = horizontal vertical  layout = column then layout-align = vertical horizontal |  |
| 7. | Directive reload | Data is not refreshed in directive | Directive controller executes one time.  It will be refreshed when its parent state reload | Use ng-repeat |
| 8. | self-executing function | Why we use self-executing function in all js files? | For using self-executing functions variable and functions will get their own scope |  |
| 9. | Controller service separation | Why operations on data are performed in services? | To reduce load on view end.  Reduce junk on controller | Controller handles view events  controller collects data for service  Controller saves data  Controller handles id generation |
| 10. | Accordion remains open | collapsed all content | Use style=”display:none” |  |
| 11. | Keep first item open for accordion | Collapse first open | ng-style="$index!=0 && {display:'none'}" |  |
| 12. | ng-class vs ng-style | Difference between ng-class and ng-style | For set logic in class attribute then use ng-class for set logic in style then use ng-style |  |
| 13. | Validation | validate form in HTML | See html form validation documents |  |
| 14. | Static vs Homogeneous | difference between sub-menu and info menu | Sub-menu:Heterogeneous Routing. fixed/static hard coded different state for fixed items.  Info-menu- Homogeneous routing. Dynamic items. All items are in same state. Should use directive for routing | Now we call sub-menu the link against what we do save operation  Sub-menu can both homogeneous or heterogeneous routing |
| 15. | State vs directive | Difference between  state and directives | State: separate workspace operation area. Information can be shared with param. Separate binding mechanism  directive: scope optional. Customized html component |  |
| 16. | Async callback traceability | How to know which async call callback is arrived if multiple async call is placed | Pass call reference in 6th params of getItem/getItems  Hold passed param as 2nd param of callback |  |
| 17. | For async call couldnt track view data | Data can’t be tracked through debugger still data showed in view  Reason multipoint callBack | Only callback when all data available |  |
| 18. | Data save was not working | Id generation data save throwing error | If no template obj then id gen will through error | Id object hierarchy is created as per template |
| 19. | data-fetch | data-fetch | Always fetch-data with context. Never fetch all data from root | It will crash the system |
| 20. | How to call other service function | When interface manager should be used | If a function has a callback param then interface manager should be used. But if the function has return value then we can not use interface manager. In that case inject the service in controller and directly call function through service. | Never work with return data from other service  always use interface manager |
| 21 | Print json in html | Print json in html | <pre>{{obj | json}}</pre> |  |
| 22 | Json stringify | Json stringify | angular.toJson(data) |  |
| 23. | Reference vs value | Reference and value | In javascript always reference is passed when we assign a obj or array to other variable. We can not copy only value with javascript. But when we use obj=angular.copy(obj) (if destination is not used) then it will copy value in newly created obj then this obj reference will be passed to this variable |  |
| 24 | ng-sortable | rearrange an array item through html | Use ng-sortable directive in ng-repeat parrent div |  |
| 25 | Filter data | Filter data from getItems | Use searchparams as second param of getItems. Key,value, path can be provided in search params | Mostly use PATH parameter |
| 26 | Primary key learning | TECHDISER\_ID | 1.t\_id is unique in whole project.  2.If any relation between obj is needed then only use t\_id for relation.  3.Object t\_id must be store if obj reference is needed.  4.In child obj do not hold parent obj Id. |  |