1. One object{} in JSON or JS, meas a RECORD in a table of database
2. EVERYTHING in Services is LIST, no single object!

EVEN a single object, would be withing a LIST!!

1. Before get a list from Services, MUST clear()!clear, clear, EVERYTIME !

Delete(), delete(), delete() using splice();

BUT, don’t clear() withing a loop , such as angular.forEach()

1. Before put things to Services’s list, MUST check if the thing is NULL;
2. Service is “db”, MUST has a ID.
3. $scope.selectedVideo4SingerList=getSingerService.clearVideo4Singer();

\*\*\*\*DON’T do the following:

$scope.selectedVideo4SingerList=[{}];

\*\*\*\*the later one is right: list of controller MUST KEEPING SYCN WITH list of Services

1. Three ways to call a function:
2. When a page is loading,( NOT call those specified function, get…)

\*\*\*\*\*\*\*\*\*\*\*\*\*in controllers

$http.get('js/songs.json').success(function(data){

$scope.songList=data.songs;

//\*\*\*\*\*\*\*\*\*\*\*list girl songs\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**getSingerService.clearGirlSong();**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*in html \*\*\*\*\*\*\*\*\*\*\*\*

<video controls="controls" name="Video Name" **ng-src="{{getVideoUrl()}}"**></video>

1. When: button click, link click, or ng-options click(dropdown list)
2. When a function whose returning result has changed

HTML:

<video controls="controls" name="Video Name" **ng-src="{{getVideoUrl()}}"**></video>

CONTROLLERS:

**$scope.getVideoUrl=function(){**

return $sce.trustAsResourceUrl("media/"+$scope.selectedVideo4SingerList$scope.getVideoUrl=function(){

return $sce.trustAsResourceUrl("media/"+$scope.selectedVideo4SingerList[1].CONTACTNAME+".mp4");

};

//\*\*\*\*\*\*\*when $scope.selectedVideo4SingerList has changed, resulting in page reloading.

BUT:

$scope.selectedVideo4SingerList=[{}] or $scope.selectedVideo4SingerList=getSingerService.clearVideo4Singer();

WON’T MAKE $scope.selectedVideo4SingerList change to empty if it is within another function, it works like **initializing a list**, not delete items from a list

BUT, if it is shown within $http.getdata==… , $scope.selectedVideo4SingerList=[{}] would work to initialize a list when the page is loading.

7.The following ways to pass parameter from view to controllers

1. Input text box,
2. Input check box
3. ngRepeat or ngOptions , loop through a list in view, can pass an object of this list as a PARAMETER of a function to controller
4. Westside: different view ( boyBand.html, girlBand.html… can use the same controller which do **the same operation**, operating **different data source.**

Using services to differ data sources which come from the same database

Like Westside: DB is: songs.json, videos.json, singers.json,

Using Services to differ : boySongList, girlSongList, JustinSongList, OneDSongList, WestSongList.

They all use the same controller

1. When a page is loaded, the first thing is: taking all data from DB/JSON:$http.getData

The 2nd thing is: using Services to reproduce data fr. DB, export reproduced data.

**For example from Westside**

**$http.get**('js/songs.json').success(function(data){

$scope.songList=data.songs;

//\*\*\*\*\*\*\*\*\*\*\*list girl songs\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**getSingerService.clearGirlSong();**

angular.forEach($scope.songList,function(s){

if(s.DELETED=="Yes"){

$scope.girlSongList=getSingerService.saveGirlSong(s);

}

});

$scope.girlSongList=getSingerService.listGirlSong();

1. Westside project: given id, take items from a same table

Songs.json: songid, songName,

Given you a few songid, asked you to take songName.

***Solution***: **double loop**

Angular.forEach(songIdList, function(id){

Angular.forEach(songList, function(s){

If(s.ID==id){

selectedSongList.save(s);

}

})

})