# Angular-WorkShop-Basics-BestPractices

## - Step-1

**ACTUALLY:** The project structure.

**TODO:**

* Show the project structure, banckend and best practices on files.
* Create Controllers.
* Create Basic HTML views.
* Create Routes.
* Explain the ControllerAs syntaxes.
* Mention the ui-view directive.
* Create the services files.

**STEPS:**

* Project Structure
* Index Serve by Backend.
* IIFE
* App Module - submodules naming
* Single Responsibility Principle
  + One component per file. (400 code lines)
  + Easier unit testing and mocking.
  + Easier to read a file, maintain and avoid collisions in source controls.
  + Divide the JS file in different files.
* Create Controllers.
  + Small functions
    - 75 LOC less is better
    - Easier to read, maintain, test, reuse, avoid bugs, avoid unwanted coupling.
  + Named functions
  + Members up to top
* Create Basic HTML views.
* Create Routes.
* ControllerAs (use this)
  + view (dotted object)
  + controller (vm = this)
  + route
* ui-view directive
  + named
  + ui-route vs ng-route (#ifTime)
* Create the services files.

## - Step-2

**ACTUALLY:** We have the views with html and styles.  
  
**TODO:**

* Validations UI.
* SignUp controller and SignUp service logic.
* Login controller and Login service logic.

**STEPS:**

* Validations
  + ng-class
  + $dirty
  + $prestine
  + $invalid
  + $focus

… look for the other related terms

* ng-show vs ng-if (change in the validations)
* Keep controllers light.
* Move logic to services

## - Step-3

**ACTUALLY:** New dashboard view. The routing access validation.

**TODO:**

* Explain new dashboard route, view and controllers.
* Explain the validation routing.
* Show the $state.go to dashboard.
* Table with edit (ng-repeat) UI.
* Dashboard Controller and Service Logic.
  + $http

**STEPS:**

* $http - best practices

## - Step-4

**Actually: Improve the performance problem on the ng-repeat in the table using one way data binding.**

**TODO:**

* **Show and explain the one way data binding applied. (:: , ng-bind)**
* **Update the dashboard controller.**
* **Show another possible fix to reduce the number of watchers in the view (ng-if)**
* **Show and explain the filter logic.**
* **Add the filter component to the UI.**

**STEPS:**

* digest cycle
* two way data binding
* one way data binding
* "if" fix - to html
* Filters.
* Custom Filters.

## - Step-5

* Directives
  + Isolate
  + &, = , @

--------------------------------------------------------------------------

# IMPORTANT

- $on, $emit, $broadcast

- angular copy and dom manipulation

- services logic

- Custom Filter.

# IF TIME

-ui-route nested views.

-pagination or infinity scroll