**IDE recommended:**

* Web storm – License has to be purchased
* [https://www.jetbrains.com/webstorm/](https://email.accenture.com/owa/redir.aspx?SURL=gk4jDdFO_f5Up1dNacy-rzWGFI9_owQOd7lVGhTQbgc0F9eun8_SCGgAdAB0AHAAcwA6AC8ALwB3AHcAdwAuAGoAZQB0AGIAcgBhAGkAbgBzAC4AYwBvAG0ALwB3AGUAYgBzAHQAbwByAG0ALwA.&URL=https%3a%2f%2fwww.jetbrains.com%2fwebstorm%2f)
* Bracket – Open source
* [http://brackets.io/](https://email.accenture.com/owa/redir.aspx?SURL=PKg4Q68rFK-7eu10mMk6xf95t1dAFautc5MiPAyX4lU0F9eun8_SCGgAdAB0AHAAOgAvAC8AYgByAGEAYwBrAGUAdABzAC4AaQBvAC8A&URL=http%3a%2f%2fbrackets.io%2f)

**Static code review tool:**

* JSHint – JS validator
* [http://jshint.com/docs/](https://email.accenture.com/owa/redir.aspx?SURL=bII_DH472bjIhZK-YXl-kqfUY7NQHDFRViYD2gEAeho0F9eun8_SCGgAdAB0AHAAOgAvAC8AagBzAGgAaQBuAHQALgBjAG8AbQAvAGQAbwBjAHMALwA.&URL=http%3a%2f%2fjshint.com%2fdocs%2f)

**Unit test coverage:**

* Jasmine can be used to for unit test and should be run using test runner called ***Karma***
* [http://jasmine.github.io/edge/introduction.html](https://email.accenture.com/owa/redir.aspx?SURL=04apqwE2bVoyi25c4ypE8tT_oTxwSE8EwtMaJx-8H-U0F9eun8_SCGgAdAB0AHAAOgAvAC8AagBhAHMAbQBpAG4AZQAuAGcAaQB0AGgAdQBiAC4AaQBvAC8AZQBkAGcAZQAvAGkAbgB0AHIAbwBkAHUAYwB0AGkAbwBuAC4AaAB0AG0AbAA.&URL=http%3a%2f%2fjasmine.github.io%2fedge%2fintroduction.html)
* [http://karma-runner.github.io/0.13/index.html](https://email.accenture.com/owa/redir.aspx?SURL=X4aUEg3QOSxQFkPVYgmf_vXZJ0FSk9L47-U3WpnPfCY0F9eun8_SCGgAdAB0AHAAOgAvAC8AawBhAHIAbQBhAC0AcgB1AG4AbgBlAHIALgBnAGkAdABoAHUAYgAuAGkAbwAvADAALgAxADMALwBpAG4AZABlAHgALgBoAHQAbQBsAA..&URL=http%3a%2f%2fkarma-runner.github.io%2f0.13%2findex.html)

**Scaffolding and Generator:**

* Yeoman
* [http://yeoman.io/learning/index.html](https://email.accenture.com/owa/redir.aspx?SURL=aPSvJxOMWNW7tuUu3CjgqpVFm-Ragc2jdoKvnCfcv500F9eun8_SCGgAdAB0AHAAOgAvAC8AeQBlAG8AbQBhAG4ALgBpAG8ALwBsAGUAYQByAG4AaQBuAGcALwBpAG4AZABlAHgALgBoAHQAbQBsAA..&URL=http%3a%2f%2fyeoman.io%2flearning%2findex.html)

**AOWP – Accenture Open Web Platform:**

* [https://aowp.accenture.com/](https://email.accenture.com/owa/redir.aspx?SURL=XHRblO2rrOtlEBf0jWbPl49rK_f3HKYU8yNaDnDrqyU0F9eun8_SCGgAdAB0AHAAcwA6AC8ALwBhAG8AdwBwAC4AYQBjAGMAZQBuAHQAdQByAGUALgBjAG8AbQAvAA..&URL=https%3a%2f%2faowp.accenture.com%2f)

**Reference project:**

* Responsive design
* [https://innersource.accenture.com/aowp/aowp-demo-responsive-design](https://email.accenture.com/owa/redir.aspx?SURL=1jia8j8HrXOXGZz9TXsC71_-Zc6pmAC-clMcrPIMja80F9eun8_SCGgAdAB0AHAAcwA6AC8ALwBpAG4AbgBlAHIAcwBvAHUAcgBjAGUALgBhAGMAYwBlAG4AdAB1AHIAZQAuAGMAbwBtAC8AYQBvAHcAcAAvAGEAbwB3AHAALQBkAGUAbQBvAC0AcgBlAHMAcABvAG4AcwBpAHYAZQAtAGQAZQBzAGkAZwBuAA..&URL=https%3a%2f%2finnersource.accenture.com%2faowp%2faowp-demo-responsive-design)
* Angular
* [https://innersource.accenture.com/aowp/aowp-academy-angular](https://email.accenture.com/owa/redir.aspx?SURL=EbV-caPiPIr3cu9w3fxRrXAJqFSo-racDm3LShS6BnM0F9eun8_SCGgAdAB0AHAAcwA6AC8ALwBpAG4AbgBlAHIAcwBvAHUAcgBjAGUALgBhAGMAYwBlAG4AdAB1AHIAZQAuAGMAbwBtAC8AYQBvAHcAcAAvAGEAbwB3AHAALQBhAGMAYQBkAGUAbQB5AC0AYQBuAGcAdQBsAGEAcgA.&URL=https%3a%2f%2finnersource.accenture.com%2faowp%2faowp-academy-angular)
  1. Constant: Have to use angular constant variables, better to say configuring purpose.(unnecessary constants should not be there)
  2. Lodash/underscore : These are the utility JS lib to use for better performance and get rid of writing down custom functionality with more codes.( <http://stackoverflow.com/questions/24604103/angular-js-best-practice-for-defining-a-utility-library>)
  3. (function(){ \_ownFuntion(); });: This is the self-calling function, for execute any method after the controller completely loaded.
  4. SessionStorage(BOM): This is for managing the session values in browser activating through AngularJS.
  5. jQuery Lib: No need to add the jQuery library for angular DOM navigation. We can use jqLite.
  6. Services should have the generic code, so that it can be able to only call from controller, no need to write down same functionality in separate places.
  7. In every controller there should have configurable variable options. So that all the static variables can be initialized as centralized.
  8. Always should aware about the exception handling whenever we are going to initiate any operation open user input.
  9. For initiating a particular variable in some of the methods we should maintain a method where we should place the operation or logic, otherwise if we need to change any single logic then we need to update every place.

**Code review comments & suggestions:**

**hilti-customer-mgmt.html**

1. jquery should not be used in Angular project library.

2. Remove un wanted js imports

3. Footer <a> link href should not be hard coded. Can be configurable. [Good to have]

4. Remove author name meta tag if not required.

**my-subscription-with-model**

1. Remove jquery code from html file.

2. Rather than having static table html, create a JSON data source and use ng-repeat to form table. The html size will be optimized. [Good to have, not mandatory]

**hilti-home-controller**

1. Optimize the Angular Service invocation with callback implementation. This will make it re-usable, lesser code size.

2. Rather than calling $scope.loadUsers(); at the end of file, write it is self-invoking JavaScript.

3. Some re-factoring is required to abstract common code rather than repeating multiple times. After that the code will be maintainable and code size will be reduced.

**customer-resource / resource-manage**

1. Write a generic implementation for REST communication which will Expose an Factory API. The API will take URL, Params, Headers dynamically and will have single place of execution.

**password-service**

1. Identify the number which could vary as per business logic and create Constants for those.

**useful-services (dev source file name has typo as usefull-services.js)**

1. Utilize the utility libraries like lodash, underscore so it will guarantee no code duplication, efficient way of data retrieval, lesser code size.

**hilti-app**

1. Re-factoring required.

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

**Overall comments in JavaScript code :**

1. Remove commented code.

2. Should have proper indentation.

3. Use utility libraries like Lodash, underscore. The unwanted for loop, if conditional checks will be lowers as those are abstracted in library.

4. Naming convention should be uniform. Some service file names ends with '-server.js' while other not.

5. It is good practice to add 'use strict'. Please refer <http://www.w3schools.com/js/js_strict.asp>

6. $rootScope is parent of all scope and have global scope. Adding data in $rootScope have pros and cons. The problem in populating data in $rootScope is, for every single attribute change it will perform dirty checks and it will impact in performance. Verify and add data only if it is mandatory.

Ref link for using all files at all: <https://github.com/DaftMonk/generator-angular-fullstack/issues/164>

Queries:

When using “yo angular” it includes the sass, sass with bootstrap, how to install less rather than sass  
915010024109662