AHK: [?**storage**]

|  |
| --- |
| HTML: |
| < Omitted >  Note: clearTest() called from HTML page to clear the storage. Note: Storage is bound to the key "test" in this example code. |
| CSS: |
| < Omitted > |
| JAVASCRIPT: |
| var eS = angular.module('exampleStore', ['localStorage']);  eS.controller("MainCtrl",  function MainCtrl($scope, $store) {  $store.bind($scope, 'test', 'Some Default Text');    $scope.**clearTest** = function(){  $store.remove('test');  };  }  );  var ls = angular.module('localStorage',[]);  ls.factory("$store",function($parse){  /\*\* Global Vars \*\*/  var storage = (typeof window.localStorage === 'undefined') ? undefined : window.localStorage,  supported = !(typeof storage == 'undefined' || typeof window.JSON == 'undefined');  var publicMethods = {    **set**: function(key,value){  storage.setItem(key, value);  return value;  },    **get**: function(key){  var item = storage.getItem(key);  return item;  },    **remove**: function(key) {  storage.removeItem(key);  return true;  },  /\*\*Bind - directly bind a localStorage value to a $scope variable \*\*/  **bind**: function ($scope, key, def) {  def = def || '';  if (!publicMethods.get(key)) {  publicMethods.set(key, def);  }  $parse(key).assign($scope, publicMethods.get(key));  $scope.$watch(key, function (val) {  publicMethods.set(key, val);  }, true);  return publicMethods.get(key);  }  };  return publicMethods;  }); |