

Editar del ABM

Alternativa 1. Pasando parámetros entre estados.

Idea general

Usar el mismo controller que agrupe la lógica para diferentes estados y vistas del ABM

Los parámetros simples (strings, enteros, boolean) se especifican en la url del estado y se pueden recuperar inyectado en `$stateParams` en el controller. Los parámetros de tipo objetos se tienen que declarar en el atributo `params` de la vista y luego se recuperan de la misma forma, a través del `$stateParams`.

En el ejemplo podemos ver un caso de cada uno.

Primero declaramos un nuevo state en `route.js`

```
.state('home.inventario.edit', {
  url: '/inventario/nuevo/:index',
  views: {
    'sub-body': {
      templateUrl: 'js/inventario/views/nuevoItem.html',
      controller: 'InventarioCtrl'
    }
  },
  params: {
    itemSelected: null
  }
});
```

NOTA: definimos *params* y declaramos *:index* en la url

Modificamos `listItems.html`

```
<span ui-sref="home.inventario.edit({itemSelected:itm, index:$index})"
      class="glyphicon glyphicon-pencil"></span>
```

NOTA: agregamos el estado al que vamos a navegar cuando en `ui-sref` y los parámetros que le vamos a enviar

Modificamos el controllers.js para indicarle de donde sacar los parámetros.
Acá inyectamos el \$stateParams. Como el controller se usa para mas de una pagina y estado, tengo que preguntar si están llegando los parámetros

```
.controller('InventarioCtrl', ['$scope', 'IntentarioService', '$stateParams',  
function($scope, IntentarioService, $stateParams){  
  
    if($stateParams && $stateParams.itemSelected){ //Si item selected esta definido  
        $scope.item = $stateParams.itemSelected;  
        $scope.index = $stateParams.index;  
        $scope.isEdit = true;  
    }  
}
```

Nota: declaro una nueva variable isEdit para identificar si estoy editando o no. Esto no sería necesario si tenemos otro template, pero complejizamos el mantenimiento (cada cambio sería x2).

Cambio para blanquear el form

```
$scope.nuevo = function() {  
    $scope.isEdit = false;  
    $scope.item = {};  
};
```

Cambio para editar el item

```
$scope.editar = function(it, index){  
    IntentarioService.editItem(it, index);  
    $scope.item = {};  
};
```

Cambio en el template nuevoItem.html

```
<button ng-hide="isEdit" type="submit" ng-click="guardar(item)"  
    class="btn btn-default">Crear nuevo Item</button>  
<button ng-show="isEdit" type="submit" ng-click="editar(item, index)"  
    class="btn btn-default">Editar el Item</button>
```

Basados en la variable, se muestra un botón o el otros. Nunca los dos. Pueden usar ng-if si quieren. A mi me queda más claro con ng-hide y ng-show

Modifico el servicio IntentarioService para agregar un método nuevo

```
editItem = function(item, index){  
  var items = localStorageService.get('items');  
  items.splice(index, 1, item);  
  localStorageService.set('items', items);  
},
```

recuerden de agregarlo a la “interface” para que puedan usarlo

```
return {  
  saveItem: saveItem,  
  editItem: editItem,  
  getItem: getItem,  
  getItems: getItems,  
  borrarItem: borrarItem  
};
```

El ejemplo andando se los paso adjunto al email!