

```
<lightning:layout multipleRows="true">

  <lightning:layoutItem size="12" padding="around-small">

    <div class="slds-grid slds-grid--align-left" >
      <lightning:buttonIcon iconName="utility:chevronleft" variant="bare" onclick="{! c.handleClick }" /> &nbsp;

      <div class="slds-text-heading_small slds-text-color_default"><center><b>Retour</b></center></div>
    </div>

  </lightning:layoutItem>
</lightning:layout>
```

Figure IV.7 – Interface de composant «ListProjet.cmp»

#### \* Contrôleur «ListProjetController.Js»

Le contrôleur «ListProjetController.Js» côté client gère les événements au sein de composant «List projet» comme illustre la figure IV.8.

```
doInit :function(component,event,helper){
  // Apex method definition
  var action = component.get("c.loadData");
  // callback function
  action.setCallback(this,function(response){
    //get state
    var state = response.getState();
    // check if state is 'SUCCESS'
    if(state == 'SUCCESS'){
      var result = response.getReturnValue();
      //set value to "UnfilteredData" attribute
      component.set("v.UnfilteredData",result);
      console.log(result);
      // set value to "data" attribute
      component.set("v.data",result);
    }else{
      console.log('something bad happend! ');
    }
  });
}
```

Figure IV.8 – Interface de Contrôleur «ListProjetController.Js»

#### \* Contrôleur «ListProjetHelper.Js»

La figure IV.9 montre l'une des fonctions utilitaires qui peut être réutilisée dans le contrôleur de composant.