

Compte rendu TNE Back-end /Front-end

Dans ce compte rendu nous allons détailler l'implémentation des fonctionnalités décrites dans le cahier des charges.

- Lors du chargement initial de la page, un menu déroulant vide s'affiche

```
<div class="listeArticle">
  <form action="#">
    <fieldset>
      <label for="articles">Selectionnez un article : </label>
      <select name="articles" id="articles">
        <!-- Articles s'affichent ici -->
      </select>
      <button type="button" id="switch" onclick="switchMode()"
>mode edition</button>
      <div>
        <input type="text" id="newArticleName" placeholder="Nom
de l'article...">
        <button onclick="ajouterArticle()">Ajouter
l'article</button>
      </div>
    </fieldset>
  </form>
</div>
<div id="paragraphes">
  <!-- Paragraphes s'affichent ici -->
</div>
```



The screenshot shows a web form with a label 'Selectionnez un article :' followed by a dropdown menu. Below the dropdown is a text input field with the placeholder 'Nom de l'article...' and a button labeled 'Ajouter l'article'.

- Une première requête AJAX permet de récupérer la liste des articles disponibles de manière à compléter le menu déroulant

```
$(document).ready(function(){

  $.getJSON(DATA, {"action":"getArticles"}, function(oRep){
    if (oRep.feedback != "ok") {
      alert("Erreur, veuillez recharger votre navigateur");
    }
  });
});
```

```
    } else {  
        for(i = 0; i < oRep.articles.Length; i++) {  
            var jOption =  
            $("<option>").val(oRep.articles[i].id).html(oRep.articles[i].nom);  
            $("#articles").append(jOption);  
        }  
    }  
});
```



- Lorsque l'utilisateur sélectionne un des articles dans le menu déroulant, une seconde requête est envoyée pour récupérer l'ensemble des paragraphes associés à cet article, qui s'affichent par ordre croissant d'ordre en dessous du menu déroulant, en dessous du titre de l'article

```
$("#articles").selectmenu({  
    select: function( event, data ) {  
        $("#paragraphes").empty();  
        articleCourrant = data.item.value;  
        getP(articleCourrant);  
        if($('#switch').css('display') === 'none') {  
            $('#switch').show();  
        }  
    }  
});  
  
function getP(idArticle) {  
    // Envoyer une requete au serveur pour récupérer les P.  
    // en base de données  
  
    $.getJSON( DATA,  
        { "action" : "getP",  
          "idArticle": idArticle},  
        function (oRep){  
            if (oRep.feedback!="ok") {  
                alert("Rechargez votre navigateur...");  
            }  
        }  
    );  
}
```

```
    }  
    else {  
        for(i=0;i<oRep.paragraphes.Length;i++) {  
            var jP = $("<p>").html(oRep.paragraphes[i].contenu);  
            var jSpan = $("<div>").addClass("blockList")  
                .prepend($("<span>").addClass("poignee ui-icon  
ui-icon-arrowthick-2-n-s"))  
                .append(jP).prepend($("<span>")  
                .addClass("trash ui-icon  
ui-icon-trash")).hide());  
            $("#paragraphes").append(jSpan);  
  
            // TODO: y associer leurs méta-données  
            jP.data(oRep.paragraphes[i]);  
        }  
    }  
});  
}
```

Nous n'avons pas réussi à implémenter la fonction pour enregistrer l'ordre lorsque nous déplaçons un paragraphe.

- Il doit être possible d'ajouter un nouvel article au menu déroulant, ce qui qui affiche le nouvel article, directement en mode édition pour pouvoir y insérer de nouveaux paragraphes

```
function ajouterArticle() {  
    let articleName = $('#newArticleName').val();  
    if (!articleName) return;  
  
    $.getJSON("data.php",  
        {  
            "action": "addArticle",  
            "nomArticle": articleName
```

```
    },  
    function (oRep) {  
        // Crée et ajoute l'option dans le select  
        let jOption = $("").val(oRep.id).html(articleName);  
        $("#articles").append(jOption);  
  
        //mode edition  
  
        $('.edition').show();  
        $('#switch').html('mode Lecture');  
        mode = 'edition';  
        // Selectionne l'option  
  
        $("#articles-button").attr('aria-activedescendant', "ui-id-"+oRep.id)  
            .attr('aria-labelledby', "ui-id-"+oRep.id);  
  
        $("#articles-button").find(".ui-selectmenu-text").html(articleName);  
  
        $("#paragraphes").empty();  
        articleCourrant = oRep.id;  
        getP(articleCourrant);  
        if($('#switch').css('display') === 'none') {  
            $('#switch').show();  
        }  
    }  
);
```

The screenshot shows a web form with the following elements:

- A label "Selectionnez un article :" followed by a dropdown menu containing "test ordre2".
- A button labeled "mode lecture" to the right of the dropdown.
- Below the dropdown, a text input field containing "test ordre2" and a button labeled "Ajouter l'article".
- At the bottom, a small button with a "+" sign next to an empty text input field.

- Un bouton « passer en mode édition » permet de rendre les paragraphes éditables, et d'afficher un bouton d'ajout de paragraphe. Il laisse place à un bouton « revenir au mode normal », qui permet de repasser en mode lecture et de cache le bouton d'ajout de paragraphes.

```
<button type="button" id="switch" onclick="switchMode()" >mode  
edition</button>
```

```
function switchMode() {
```

```
if(mode === 'lecture') {  
    $('.edition').show();  
    $('#switch').html('mode lecture');  
    mode = 'edition';  
}  
else if(mode === 'edition'){  
    $('.edition').hide();  
    $('#switch').html('mode edition');  
    mode = 'lecture';  
}  
}
```

mode edition

mode lecture

- L'utilisateur pourra éditer un nombre quelconque de paragraphes à la fois, en cliquant sur chacun d'eux
- A chaque fois qu'un paragraphe est cliqué pour être édité, un textarea remplace le paragraphe pour permettre l'édition

```
$(document).on( "click",  
    "#paragraphes p",  
    function () {  
        // clic sur un futur P.  
        if(mode === 'edition') {  
            var contenuP = $(this).html();  
            var metaP = $(this).data();  
            //stockage pour echape  
            paragraphes[$(this).data().id] = $(this);  
            // préparer Le futur textarea  
            var jT = $("<textarea>")  
                .attr('id', $(this).data().id)  
                .val(contenuP)  
                .data(metaP);  
  
            // insertion dans Le DOM  
            $(this).replaceWith(jT);  
        }  
    });
```

The screenshot shows a web interface with two paragraphs, each labeled "Nouveau paragraphe". Below the paragraphs is a button with a plus sign (+) inside a light gray box.

- Lorsque l'utilisateur appuie sur 'ESC', il faut annuler tous les paragraphes en cours

d'édition,

i.e. les faire revenir dans leur état initial d'avant que ne débute l'édition et ce même si des changements étaient déjà intervenus au cours de l'édition. Il faudra donc sauvegarder leurs valeurs au moment de passer en mode édition.

Cette fonction n'est pas fonctionnelle, car lors de l'appui sur la touche echappe l'édition du paragraphe est bien annulée mais le texte ne reviens pas sur la page avant un rechargement de celle-ci.

```
$(document).on("keyup", "body", function(LeContexte){
    if(LeContexte.which==27){
        var textarea = document.getElementsByTagName("textarea");
        for( var i = 0 ; i < textarea.length; i++){
            var currentP = paragraphes[i];
            var contenuT = $(currentP).html();
            var metaT = $(currentP).data();
            var jP = $("<p>").html(contenuT).data(metaT);
            $("#"+$(textarea[i])[0].id).replaceWith(jP);
            console.log($(textarea[i]).prop('id'));
        }
    }
})
```

- Lorsque l'utilisateur appuie sur "ENTREE" à l'intérieur d'un textarea, cela valide la modification du paragraphe concerné pour que son nouveau contenu soit envoyé au serveur, et replace le paragraphe initial en enlevant le textarea.

```
$(document).on( "keydown",
    "#paragraphes textarea",
    function(LeContexe){

        switch (LeContexe.which){
            // Validation d'une saisie
            case 13 : {
                // On prépare Le nouveau P.
                var contenuT = $(this).val();
                var metaT = $(this).data();
                var jP = $("<p>").html(contenuT).data(metaT);

                // On met à jour ses méta-données
                // car Le contenu a changé
                jP.data("contenu", contenuT);

                //On envoie une requete au serveur
            }
        }
    })
```

```
$.getJSON(DATA,
    {
        "action": "updateP",
        "id": metaT.id,
        "contenu": contenuT
    },
    function (oRep) {
        if (oRep.feedback != "ok") {
            alert("Rechargez votre navigateur...");
        }
    });

    // On l'insère
    $(this).replaceWith(jP);
    break;
}

default : {
}

}

});
```

- Un bouton permettra d'ajouter un nouveau paragraphe à l'article en cours. À côté du bouton se trouvera un champ de saisie de texte permettant de préciser le contenu initial du paragraphe. Si ce champ est vide, le contenu initial devra être « Nouveau paragraphe ».
- Lors de l'appui sur le bouton d'ajout, le paragraphe ajouté devra être en mode édition, et le focus du navigateur devra être placé à l'intérieur de ce paragraphe.

```
// Insérer un bouton + avant le div des paragraphes
var jP = $("
```

```
var contenu = $(this).next().val();
if( contenu.replace(/\s+/g, '') === "" ) {
    contenu = "Nouveau paragraphe";
}
var jP = $("

").prepend($("<span>").addClass("poignee
ui-icon ui-icon-arrowthick-2-n-s"))

.append($("<p>").addClass("ui-state-default").html(contenu).data({
    "id": 0,
    "ordre": ordreNouveau,
    "contenu": contenu
})).prepend($("<span>").addClass("trash ui-icon
ui-icon-trash")).hide());

$("#paragraphes").prepend(jP);
$.getJSON("data.php",
    {
        "action": "addP",
        "ordre": ordreNouveau,
        "contenu": contenu,
        "article": articleCourrant
    },
    function (oRep) {
        jP.html(oRep.content);
    }
);

// clic sur un futur P.

var contenuP = $("#paragraphes div:first-child
p").text();
var metaP = $("#paragraphes div:first-child").data();
//stockage pour echape
paragraphes[$("#paragraphes div:first-child").data().id]
= $("#paragraphes div:first-child");
// préparer le futur textarea
var jT = $("<textarea>").attr('id', $("#paragraphes
div:first-child").data().id)
    .val(contenuP)
    .data(metaP);

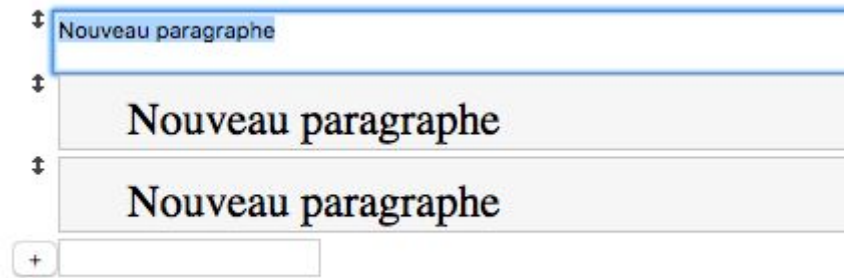
// insertion dans le DOM
$("#paragraphes div:first-child p").replaceWith(jT);

$("#paragraphes textarea").focus().select();


```



```
}  
});
```



- Des interactions de type glisser-déposer permettent de changer l'ordre des paragraphes au sein d'un article. On suggère de les réaliser en utilisant jQuery UI.

```
<div id="paragraphes">  
  <!-- Paragraphes s'affichent ici -->  
</div>
```

```
$("#paragraphes").sortable({  
  handle:".poignee"  
});  
$("#paragraphes").disableSelection();
```

- Lors du survol d'un paragraphe, une icône de croix s'affichera, permettant de supprimer ce paragraphe. On suggère de réaliser cette interaction à l'aide de JQuery UI.

```
$("#div").hover(  
  function() {  
    $( this ).find(".trash").show();  
  }, function() {  
    $( this ).find(".trash").hide();  
  }  
);
```

