UI :

The application has two main forms : the user choose a table from a list, then can modify entries in the selected table

First form : a simple list, with the list of tables for the current user

(authentication is not in the scope of this project)



The user chooses an entry in the list, then go the table form

Two kind of views :
1- batch editing



(<https://js.devexpress.com/Demos/WidgetsGallery/Demo/DataGrid/BatchEditing/Angular/Light/>)

In the column headers, filter and sorting must be activated



2- form editing



(<https://js.devexpress.com/Demos/WidgetsGallery/Demo/DataGrid/FormEditing/Angular/Light/>)

If the HTML editor (<https://js.devexpress.com/Demos/WidgetsGallery/Demo/HtmlEditor/Overview/Angular/Light/>)

Cannot be include in the edit form (it’s not an editor), must set an « edit » button, when user clicks, open a new form with editor only

Perhaps the html which is produced by the devextreme component must be encoded in base64 to be well exchanged with the web API

In the column headers, filter and sorting must be activated

Technical datas :

Response from web API is XML or JSON

(It’s a System.Web.http.ApiController included in an ASP.Net MVC 5 project)

To get the list of tables for the user « Dupond »

<http://n6.tecsoft.fr/paul/api/PreparationApi/getTvtUser?name=Dupond>

To get the description of the table «Menu Board/pizza »

(for the url, the name becomes « Menu%20Board$pizza », « / » is changed to « $»)

<http://n6.tecsoft.fr/paul/api/PreparationApi/getTvtAngTable?name=Menu%20Board$pizza>

to get records from table «Menu Board/pizza »

<http://n6.tecsoft.fr/paul/api/PreparationApi/Orders?name=Menu%20Board$pizza>

to get record id=2 from table «Menu Board/pizza »

<http://n6.tecsoft.fr/paul/api/PreparationApi/OrderById?name=Menu%20Board$pizza&employeeId=2>

to insert a new record to table «Menu Board/pizza »

<http://n6.tecsoft.fr/paul/api/PreparationApi/InsertOrder?name=Menu%20Board$pizza>

to update record id=2 from table «Menu Board/pizza »

<http://n6.tecsoft.fr/paul/api/PreparationApi/UpdateOrder?name=Menu%20Board$pizza/>

to delete record id=2 from table «Menu Board/pizza »

<http://n6.tecsoft.fr/paul/api/PreparationApi/DeleteOrder?name=Menu%20Board$pizza&id=2>

the test class in angular for CRUD operations on Table

export class EmployeeService {

 url = 'http://n6.tecsoft.fr/paul/api/PreparationApi';

 constructor(private http: HttpClient) { }

 getAllEmployee(): Observable<Employee[]> {

 return this.http.get<Employee[]>(this.url + '/Orders?name=Menu%20Board\_pizza');

 }

 getEmployeeById(employeeId: string): Observable<Employee> {

 return this.http.get<Employee>(this.url + '/OrderById?name=Menu%20Board\_pizza&employeeId=' + employeeId);

 }

 createEmployee(employee: Employee): Observable<Employee> {

 const httpOptions = { headers: new HttpHeaders({ 'Content-Type': 'application/json' }) };

 return this.http.post<Employee>(this.url + '/InsertOrder?name=Menu%20Board\_pizza/',

 employee, httpOptions);

 }

 updateEmployee(employee: Employee): Observable<Employee> {

 const httpOptions = { headers: new HttpHeaders({ 'Content-Type': 'application/json' }) };

 return this.http.put<Employee>(this.url + '/UpdateOrder?name=Menu%20Board\_pizza/',

 employee, httpOptions);

 }

 deleteEmployeeById(employeeid: string): Observable<number> {

 const httpOptions = { headers: new HttpHeaders({ 'Content-Type': 'application/json' }) };

 return this.http.delete<number>(this.url + '/DeleteOrder?name=Menu%20Board\_pizza&id=' + employeeid,

 httpOptions);

 }

}

CORS is activated on n6.tecsoft.fr for localhost:59763

(So you can test your code if you set Angular to start on this port)

Below the definition of the web API class

[EnableCors(origins: "http://localhost:59763", headers: "\*", methods: "\*")]

 [Route("api/PreparationApi/{action}", Name = "PreparationApi")]

 public class PreparationApiController : ApiController

structure of TVTAngUser

<ArrayOfTvtAngUser xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<TvtAngUser>

<name>Dupond</name>

<fields>

<TvtAngUserTable>

<name>Menu Board/pizza</name>

<title>The pizza Menu Board</title>

</TvtAngUserTable>

<TvtAngUserTable>

<name>Menu Board/ListeTitre</name>

<title>title list</title>

</TvtAngUserTable>

</fields>

</TvtAngUser>

</ArrayOfTvtAngUser>

Fields for TvtAngUserTable

name : field to get datas from getTvtAngTable

title : for text in first form (table list)

structure of TvtAngTable

<TvtAngTable>

<typeVue>0</typeVue>

<previewDesignID>0</previewDesignID>

<Flags>0</Flags>

<fields>

<TvtAngField>

<name>id</name>

<type>int</type>

<notVisibleTVT>false</notVisibleTVT>

<notVisibleWEB>false</notVisibleWEB>

<heritage>false</heritage>

<columnTitle/>

<columnWidth>0</columnWidth>

<defaultValue/>

<numList>0</numList>

<tableName/>

<tableField/>

</TvtAngField>

<TvtAngField>

….

typeVue : 0 for batch editing, 1 for form editing

previewDesignID : not used

Flags : not used

Fields for TvtAngField

name : field name

int : field type, see below

notVisibleTVT : not used

notVisibleWEB : this field must not be seen in web UI

heritage : not used

columnTitle : optional text for column header

columnWidth : minimal width of column in pixels (perhaps in % if desired), 0 : default

defaultValue : is field value for a record is empty, set this text

numList : not used

tableName : for field that use an entry in a predefined list : name of the table that contains the list (in the example : Menu Board/ListeTitre)

tableField : name of the field to show from table tableName (in the example : long\_name from Menu Board/ListeTitre)

field types :

empty or nil : defaut type : string

int : integer

decimal : decimal

bool : True or False (must be a checkBox in grid <https://js.devexpress.com/Demos/WidgetsGallery/Demo/CheckBox/Overview/Angular/Light/>)

date : format 2019-06-13 or 2019-06-13 12:32 (must be a devextreme calendar in grid <https://js.devexpress.com/Demos/WidgetsGallery/Demo/Calendar/Overview/Angular/Light/>)

table1 : field data is empty or nil or an entry of another table (table name in tableName field, the data to show are in field « tableField »)

rtf : long text, must be edited by an HTML editor (front color, Bold, Italic, differents fonts and size must be supported)
 format of text is HTML, if there is problem on server with XML tag (XML files are generated for android devices which display datas)
 data will be encoded in base64

media : media from TVTools Media library

special keyword «TVTCloud:/ » must be replaced in this example by «https://n6.tecsoft.fr/»
for example : in datas : TVTCloud:/Medias/videos/E.LECLERC\_TROYES/ticket\_rond\_jaune.png will be transformed in <https://n6.tecsoft.fr/Medias/videos/E.LECLERC_TROYES/ticket_rond_jaune.png>

first field is always a counter field (unique ID)

note about table1 type :

from table « Menu Board/pizza »

field « Titre » is taken from field « long\_name » from table « Menu Board/listeTitre »

<TvtAngField>

<name>Titre</name>

<type>table1</type>

<notVisibleTVT>false</notVisibleTVT>

<notVisibleWEB>false</notVisibleWEB>

<heritage>false</heritage>

<columnTitle>Title</columnTitle>

<columnWidth>0</columnWidth>

<defaultValue>0</defaultValue>

<numList>0</numList>

<tableName>Menu Board/ListeTitre</tableName>

<tableField>long\_name</tableField>

</TvtAngField>

So before displaying the field « Titre », you must load the table « Menu Board/listeTitre »

Then you must show it with a combobox.



Datas for table «Menu Board/pizza» can be retreived by

<http://n6.tecsoft.fr/paul/api/PreparationApi/Orders?name=Menu%20Board$pizza>

<ArrayOfEmployee xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<Employee>

<EmpId>1</EmpId>

<fields>

<TVTField>

<val>MARGUERITA</val>

</TVTField>

<TVTField>

<val>

Tomate, mozzarella, tomates fraîches, salade, chorizo, olives

</val>

</TVTField>

…

It’s a list of field « Employee » : one class « Employee » for each record of the table

EmpId is the ID of the record, so it’s the first field of the record.

Then you find a list of «TVTField» class, which contain the datas for the other fields of the record

So if there are 9 fields in a record (in TvtAngTable), there is 8 fields «TVTField» in «Employee» class

to get list of Medias for user « Dupond »

<http://n6.tecsoft.fr/paul/api/PreparationApi/getMedias?name=Dupond>

you get a hierarchical structure of TvtAngMediaItem

<TvtAngMediaItem>

<id>7848</id>

<type>1</type>

<level>0</level>

<name/>

<dirType>1</dirType>

</TvtAngMediaItem>

<TvtAngMediaItem>

<id>8504</id>

<type>1</type>

<level>7848</level>

<name>schording.png</name>

<dirType>0</dirType>

</TvtAngMediaItem>

<TvtAngMediaItem>

<id>8551</id>

<type>1</type>

<level>7848</level>

<name>Charlotte.jpg</name>

<dirType>0</dirType>

</TvtAngMediaItem>

<TvtAngMediaItem>

<id>8552</id>

<type>1</type>

<level>7848</level>

<name>sous1</name>

<dirType>1</dirType>

</TvtAngMediaItem>

<TvtAngMediaItem>

<id>8553</id>

<type>1</type>

<level>8552</level>

<name>Renault quality made en MP4.mp4</name>

<dirType>0</dirType>

</TvtAngMediaItem>

….

In our environnement it is displayed as :



First TvtAngMediaItem is root item

Field of TvtAngMediaItem :

id : database id of item

type : always 1 in this scope

level : 0 root level or id of the upper level

name : media name or directory name

dirType : 0 : it’s a media, 1 : it’s a directory

for example « Renault quality made en MP4.mp4 » has a level id of 8552

which is a directory level, named « sous1 »

« sous1 » has a level id of 7848

Which is the master level

So full url for media « Renault quality made en MP4.mp4 » is

<https://n6.tecsoft.fr/medias/videos/sous1/Renault%20quality%20made%20en%20MP4.mp4>

to get thumbnail of this media :

<https://n6.tecsoft.fr/medias/videos/sous1/temp/Renault%20quality%20made%20en%20MP4_mp4.jpg>

Medias must be sorted by front end solution

Media must be showned as a treeView or with other method

A method for upload a new media must be developed

For example with

<https://js.devexpress.com/Demos/WidgetsGallery/Demo/FileUploader/ChunkUploading/Angular/Light/>

in c#, to get full name of a Media

 public string getFullMediaItemName(int id, string name)

 {

 if (name == null) name = "";

 try

 {

 int i = mil[id].level;

 if (i == 0 || mil[i].level == 0)

 {

 return name;

 }

 else

 {

 if (name != "") name = mil[i].name + "\\" + name;

 else name = mil[i].name;

 return getFullMediaItemName(i, name, ref dirType2);

 }

 }

 catch (Exception ex)

 {

 }

 return "";

 }

In c# to show directory and medias at the root of a mediaItem « directory » which id=ID

 foreach (int i in myApp.mil.Keys)

 {

 if (myApp.mil[i].level == ID && myApp.mil[i].compteur > 0)

 {

 r = new myNamemei(myApp.mil[i]);

 lmd.Add(r);

 }

 }

 if (lmd.Count > 0)

 {

 lmd.Sort();

 for (int i = 0; i < lmd.Count; i++)

 {

 try

 {

 if (lmd[i].mi.dirType == 1)

 {

 TreeListNode node = new TreeListNode() { Content = new FileSystemItem(lmd[i].mi.name, 1, lmd[i].mi.compteur), Image = FileSystemImages.ClosedFolderImage };

 if (treeListNode != null) treeListNode.Nodes.Add(node);

 else

 {

 mainLevels.Add(lmd[i].mi.compteur);

 view.Nodes.Add(node);

 }

 node.IsExpandButtonVisible = HasFiles(lmd[i].mi.compteur) ? DefaultBoolean.True : DefaultBoolean.False;

 }

 }

 catch { }

 }

 for (int i = 0; i < lmd.Count; i++)

 {

 try

 {

 if (lmd[i].mi.dirType == 0)

 {

 TreeListNode node = new TreeListNode();

 node.Content = new FileSystemItem(lmd[i].mi.name, 2, lmd[i].mi.compteur);

 string s = myApp.getImageName(lmd[i].mi.compteur,lmd[i].mi.type, 0);

 if (s != null)

 {

 //node.Image = myApp.getBitmapImage(s);

 node.Image = new BitmapImage(new Uri(s, UriKind.RelativeOrAbsolute));

 }

 if (treeListNode != null) treeListNode.Nodes.Add(node);

 else

 {

 mainLevels.Add(lmd[i].mi.compteur);

 view.Nodes.Add(node);

 }

 node.IsExpandButtonVisible = HasFiles(lmd[i].mi.compteur) ? DefaultBoolean.True : DefaultBoolean.False;

 }

 }

 catch { }

 }

 }