

Etape5 : gestion d'événements

Ajouter deux boutons. Ajouter une nouvelle clause `dojo.require()` pour inclure le code pour le widget `dijit.form.Button`.

```
<script type="text/javascript">
dojo.require("dijit.form.DateTextBox");
dojo.require("dijit.form.ComboBox");
dojo.require("dijit.form.CheckBox");
dojo.require("dijit.form.Button");
</script>
```

Juste au dessus de la ligne: `</body>`

Ajouter:

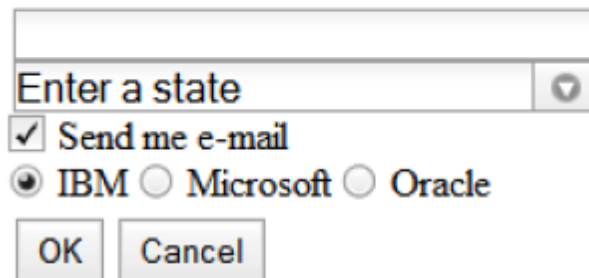
```
<br />
<button dojoType="dijit.form.Button">OK</button>
<button dojoType="dijit.form.Button">Cancel</button>
```

Sauvegardez les changements et rafraichir le web browser.

Vérifiez que les boutons apparaissent.

Application One

Enter a date below:



Comprendre la gestion des événements de Dojo

Pour un élément régulier de bouton, vous ajoutez un gestionnaire d'événement onclick comme suit:

```
<button onclick="showEventStd(this, event);">OK</button>
```

Les «this» et les mots clés "event" indiquent au navigateur de fournir l'élément et l'objet de l'événement en tant que paramètres à la fonction de gestionnaire. La fonction de gestionnaire ressemble à ceci:

```
function showEventStd(theElement, theEvent) {
alert("Event " + theEvent.type + " for element " + theElement.nodeName);
}
```

Dans Dojo, il ya deux différences principales.

1. L'attribut onclick d'un bouton Dojo prend le nom de la méthode de gestionnaire d'événement. Vous n'appellez pas la méthode comme vous le faites pour un bouton DOM.
2. Deuxièmement, la méthode de gestionnaire d'événement reçoit un seul paramètre - l'objet de l'événement.

Le gestionnaire d'événements est enregistré comme suit:

```
<button dojoType="dijit.form.Button" onclick="showEvent">OK</button>
```

La méthode de gestionnaire d'événement ressemble à ceci:

```
function showEvent(theEvent) {  
    alert("Event " + theEvent.type);  
}
```

Assurez-vous que vous comprenez les deux principales différences.

L'objet de l'événement passé à la fonction de gestionnaire est une version personnalisée de l'objet DOM de l'événement.

Certains navigateurs ne suivent pas strictement les spécifications DOM W3C. Par exemple, l'objet DOM événement devrait avoir une propriété appelée target qui pointe vers l'objet Element où l'événement a eu lieu. Mozilla suit cette correctement. IE, cependant, appelle cette propriété srcElement. Faire face à ces différences peut être délicat. Dojo résout le problème en fournissant les propriétés standard manquantes. Par exemple, l'objet Dojo événement aura la propriété target dans tous les navigateurs. Ainsi, la fonction de gestionnaire d'événement suivant fonctionnera pour tous les navigateurs.

```
function showEvent(theEvent) {  
    alert("Event " + theEvent.type + " for element " + theEvent.target.nodeName);  
}
```

Ajout de gestionnaire d'événements.

Créer des gestionnaires d'événements pour les boutons, comme indiqué ci-dessous.

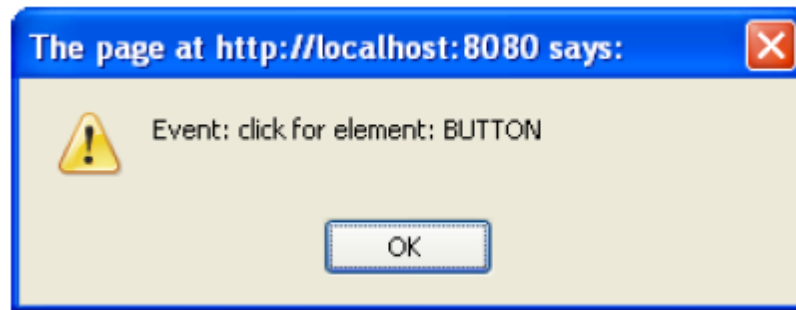
```
<button dojoType="dijit.form.Button" onclick="showEvent">OK</button>  
<button dojoType="dijit.form.Button" onclick="showEvent">Cancel</button>
```

Ecrire la méthode de gestionnaire comme indiqué en gras ci-dessous:

```
<script type="text/javascript">  
dojo.require("dijit.form.DateTextBox");  
dojo.require("dijit.form.ComboBox");  
dojo.require("dijit.form.CheckBox");  
dojo.require("dijit.form.Button");
```

```
function showEvent(theEvent) {  
    alert("Event: " + theEvent.type +  
        " for element: " + theEvent.target.nodeName);  
}  
</script>
```

Sauvegardez et rafraichir le browser. Cliquez sur le bouton OK. La boîte de dialogue d'alerte ressemblera à ceci:



Etape6 : debug avec dojo

Utilisation d'alert () pour le débogage peut être ennuyeux. Dojo a construit un mécanisme de journalisation. Il est assez facile à utiliser.

Premièrement, activer le débogage. Cela se fait en définissant la propriété de configuration IsDebug Dojo à true.

```
<script type="text/javascript" djConfig="parseOnLoad: true, isDebug: true"  
src="dojo-release-1.0.0/dojo/dojo.js">  
</script>
```

Ensuite, remplacer la fonction alert () avec console.debug ().

```
function showEvent(theEvent) {  
    console.debug("Event: " + theEvent.type +  
        " for element: " + theEvent.target.nodeName);  
}
```

Enregistrer les modifications. Actualiser le navigateur.

La console de débogage est visible en bas de la page.

Cliquez sur les boutons et assurez-vous que les messages de débogage apparaissent. Manipuler votre IHM.

Gérer Plus d'événements

La case à cocher et boutons radio tirent également l'événement onclick. Créer la fonction de showEvent comme gestionnaire d'événements, comme indiqué ci-dessous.

```
<input id="ch1" dojoType="dijit.form.CheckBox" checked="checked"
value="Y" onclick="showEvent"/>
<label for="ch1">Send me e-mail</label>
<br />
<input id="rad1" dojoType="dijit.form.RadioButton" checked="checked"
name="vendor" value="IBM" onclick="showEvent"/>
<label for="rad1">IBM</label>
<input id="rad2" dojoType="dijit.form.RadioButton" name="vendor"
value="MS" onclick="showEvent"/>
<label for="rad2">Microsoft</label>
<input id="rad3" dojoType="dijit.form.RadioButton" name="vendor"
value="Oracle" onclick="showEvent"/>
<label for="rad3">Oracle</label>
```

La combo box et la date box, d'autre part, ont un onchange événement. Le gestionnaire d'événements onchange fonctionne différemment. Il ne reçoit pas l'objet de l'événement. Au lieu de cela, il reçoit la nouvelle valeur introduite par l'utilisateur.

Créer la fonction de showNewValue que le gestionnaire d'événements onchange pour la zone de liste déroulante et la date comme indiqué en gras ci-dessous.

```
<input dojoType="dijit.form.DateTextBox" onchange="showNewValue"/>
<br />
<select dojoType="dijit.form.ComboBox" autocomplete="true"
value="Enter a state" onchange="showNewValue">
<option selected="selected">California</option>
<option>Illinois</option>
<option>New York</option>
<option>Nebraska</option>
</select>
```

Développer la fonction showNewValue dessous de la fonction de showEvent.

```
function showNewValue(val) {
console.debug("User entered: " + val);
}
```

Enregistrer les modifications. Actualiser le navigateur. Assurez-vous que les événements de la boîte de date et combo box soient gérés correctement.

The screenshot shows a web form with the following elements:

- A date input field containing "11/8/07".
- A dropdown menu (ComboBox) with "Nebraska" selected.
- A checkbox labeled "Send me e-mail" which is checked.
- Three radio buttons labeled "IBM", "Microsoft", and "Oracle". "IBM" is selected.
- "OK" and "Cancel" buttons.

Below the form is a console window with a "Clear" button and the following log messages:

- User entered: Tue Nov 20 2007 00:00:00 GMT-0500 (Eastern Standard Time)
- User entered: Tue Nov 20 2007 00:00:00 GMT-0500 (Eastern Standard Time)
- User entered: New York
- User entered: Nebraska
- User entered: Thu Nov 08 2007 00:00:00 GMT-0500 (Eastern Standard Time)
- User entered: Thu Nov 08 2007 00:00:00 GMT-0500 (Eastern Standard Time)

Exercice8 : DOJO rich application

Une application Internet riche aura un bureau comme disposition d'écran. À cette fin, Dojo fournit disposition de gestionnaire widgets.

Etape1 : Créer un modèle de page

Créez un nouveau fichier HTML appelé template.html.

Régler le contenu de la manière suivante.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Template</title>
<link rel="stylesheet" type="text/css"
href="dojo-release-1.0.0/dijit/themes/tundra/tundra.css" />
<script type="text/javascript"
djConfig="parseOnLoad: true, isDebug: true"
src="dojo-release-1.0.0/dojo/dojo.js">
</script>
<script type="text/javascript">
dojo.require("dijit.form.Button");
</script>
</head>
<body class="tundra">
</body>
</html>
```

Enregistrez et fermez le fichier.

Le ContentPane Widget

Un ContentPane Widget est le gestionnaire de mise en page le plus simple. C'est juste un groupe d'autres widgets. Il n'impose pas toute disposition particulière. D'une certaine manière, il agit comme un élément <div>. ContentPane a une puissante fonction. Il peut utiliser le contenu d'une URL externe et remplir son contenu interne de manière asynchrone.

Cela vous permet de charger le contenu dans le volet sur demande.

Copiez template.html et le coller comme content_pane.html.

Ouvrir content_pane.html.

Ajouter un dojo.require () charge de déclaration dans le code pour le widget. Ceci est illustré ci-dessous en caractère gras.

```
<script type="text/javascript">
dojo.require("dijit.form.Button");
dojo.require("dijit.layout.ContentPane");
</script>
```

Dans la balise body, ajouter un widget volet contenu, comme indiqué ci-dessous en caractères gras.

```
<body class="tundra">
<div dojoType="dijit.layout.ContentPane"
style="background: red">
<p>Hello World</p>
</div>
</body>
```

Enregistrer les modifications.

Exécutez le fichier HTML sur le serveur pour l'afficher dans un navigateur. Il devrait ressembler à ceci.



Etape2 : Créer un modèle de page

De toute évidence, rien de trop excitant. Le widget se comporte comme un élément <div> régulier.

Maintenant, nous allons externaliser le contenu.

Créez un nouveau fichier HTML appelé content1.html. Définir le contenu du fichier:

```
<h3>This is content 1</h3>
<p>Some content here</p>
```

Enregistrer les modifications.

Retour dans le content_pane.html, changer le balisage widget comme indiqué ci-dessous.

```
<div dojoType="dijit.layout.ContentPane"
style="background: red"
href="content1.html">
Loading ...
</div>
```

Enregistrer les modifications.

Actualiser le navigateur.

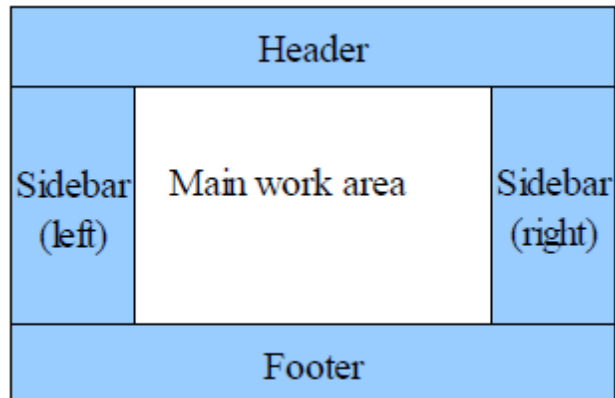


Lorsque la principale Page (content_pane.html) se charge, le widget ContentPane montre "Loading ...". Cela devient finalement remplacé par le contenu de content1.html.

Le LayoutContainer Widget

La plupart des applications de bureau ont cette mise en page:

1. La couche supérieure a menus et d'outils.
2. Il y a des encadrés sur le côté gauche et / ou à droite.
3. Il y a un bar et widgets état au bas.
4. Au milieu, nous avons la zone de travail principale.



Le même schéma se produit dans la plupart des sites Web, où nous avons-tête, pied de page, barre latérale et principale zone de contenu.

La LayoutContainer capture ce modèle. Chaque zone est représentée par un widget agencement, tel qu'un ContentPane. Chaque boîte a une position, qui peut être l'un en haut, bas, gauche, droite et le client. La position "client" se réfère à la zone de travail principale. Il est aussi celle qui croît avec la taille de la fenêtre du navigateur. Tous les autres boîtes ont une largeur fixe (à gauche et à droite) ou la hauteur (pour les haut et en bas).

Copiez template.html et le coller comme layout_container.html.

Ajouter deux dojo.require () les déclarations ci-dessous en caractères gras.

```
<script type="text/javascript">  
dojo.require("dijit.form.Button");  
dojo.require("dijit.layout.ContentPane");  
dojo.require("dijit.layout.LayoutContainer");  
</script>
```

Dans la balise <body>, ajoutez le conteneur de présentation comme suit.

```
<div dojoType="dijit.layout.LayoutContainer">  
  <div dojoType="dijit.layout.ContentPane" layoutAlign="top"  
    style="background:red;height:50px">  
    Header  
  </div>  
  <div dojoType="dijit.layout.ContentPane" layoutAlign="bottom"  
    style="background:blue">  
    Footer  
  </div>
```

```

<div dojoType="dijit.layout.ContentPane" layoutAlign="left"
style="background:green;width: 120px;">
Sidebar
</div>
<div dojoType="dijit.layout.ContentPane" layoutAlign="client">
Client area
</div>
</div>

```

Dans Dojo 1.0, le widget LayoutContainer, le <body> et <html> éléments doivent tous avoir une largeur et une hauteur définie. Sinon, la page brise mal (surtout dans IE). Nous allons résoudre ce problème maintenant.

Dans la balise <head>, définir un style comme suit.

```

<style>
.screen {height: 400px; width: 600px; margin: 0; padding: 0;}
</style>

```

Définissez la classe de l'élément <html>:

```
<html xmlns="http://www.w3.org/1999/xhtml" class="screen">
```

Définissez la classe de l'élément du corps:

```
<body class="screen tundra">
```

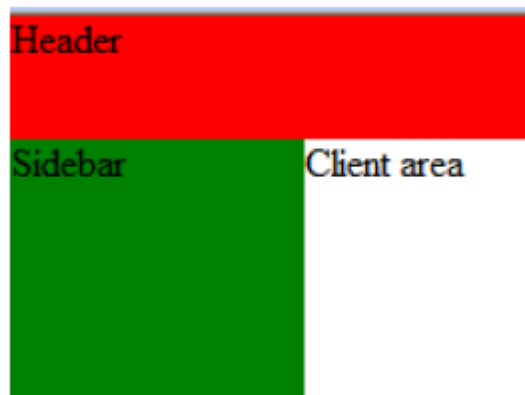
Définissez la classe du widget LayoutContainer:

```
<div dojoType="dijit.layout.LayoutContainer" class="screen">
```

Enregistrer les modifications.

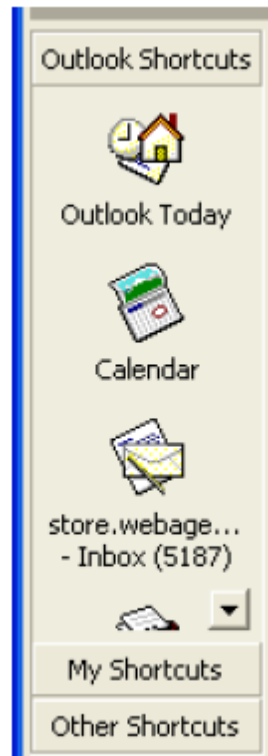
Remarque:, nous avons mis une taille fixe pour l'écran. Vous pouvez également le mettre à une figure de% (comme 75%). Cela rendra la zone cliente plus grande ou ou plus petite comme la fenêtre du navigateur est redimensionnée.

Exécutez le fichier sur le serveur.



Etape3 : Le AccordionContainer Widget

AccordionContainer pout regrouper d'autres widgets en grappes. Un seul groupe est visible à la fois. Ce widget est popularisé par MS Outlook. Le widget apparaît généralement sur le côté gauche de la zone de travail du client.



Chaque tas de widgets est créé en utilisant un widget AccordionPane.

Ajouter une déclaration nouvelle `dojo.require()` ci-dessous les existants.

```
dojo.require("dijit.layout.AccordionContainer");
```

Ce sera également inclure le code pour le widget AccordionPane.

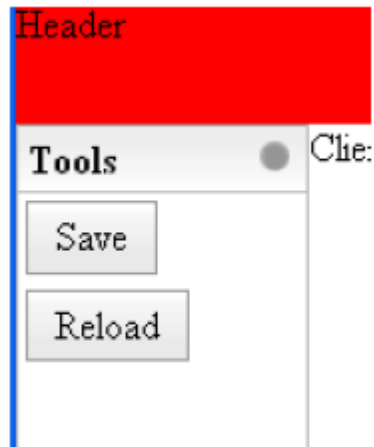
Dans la barre latérale de gauche, ajouter le widget AccordionPane comme indiqué ci-dessous en caractères gras.

```
<div dojoType="dijit.layout.ContentPane" layoutAlign="left"
style="background:green;width: 120px;">
<div dojoType="dijit.layout.AccordionContainer">
<div dojoType="dijit.layout.AccordionPane" selected="true" title="Tools">
<button dojoType="dijit.form.Button">Save</button><br/>
<button dojoType="dijit.form.Button">Reload</button>
</div>
<div dojoType="dijit.layout.AccordionPane" title="Options">
<button dojoType="dijit.form.Button">Configure</button><br/>
<button dojoType="dijit.form.Button">Session</button>
```

```
</div>
</div>
</div>
```

Enregistrer les modifications.

Actualiser le navigateur.



Vous pouvez cliquer sur Options pour voir cette bande de widgets.

Parce que, tous les widgets AccordionPane montrent pas leurs contenus tout le temps, vous pouvez charger leur contenu à la demande. Il suffit de régler l'attribut href à une URL. Lorsque le AccordionPane est d'abord ouvert, le système ira chercher le contenu à la demande. Exemple:

```
<div dojoType="dijit.layout.AccordionPane" title="Options" href="options.html">
</div>
```

Etape4 : Le TabContainer Widget

Afficher le contenu onglets est très facile. Chaque onglet est créé en utilisant un ContentPane. Le titre de l'onglet est réglé en utilisant l'attribut title de la ContentPane.

Copiez template.html que tab.html.

Ouvrir tab.html.

Ajouter deux dojo.require () éléments à inclure TabContainer et ContentPane.

```
<script type="text/javascript">
dojo.require("dijit.form.Button");
dojo.require("dijit.layout.ContentPane");
dojo.require("dijit.layout.TabContainer");
</script>
```

Dans la balise body, ajouter le TabContainer comme suit.

```

<div dojoType="dijit.layout.TabContainer"
style="width:500px;height:100px">
<div dojoType="dijit.layout.ContentPane" title="Tab A">
The content of tab A.
</div>
<div dojoType="dijit.layout.ContentPane" title="Tab B" href="content1.html">
Loading...
</div>
<div dojoType="dijit.layout.ContentPane" title="Tab C"
closable="true" selected="true">
The content of tab C.
</div>
</div>

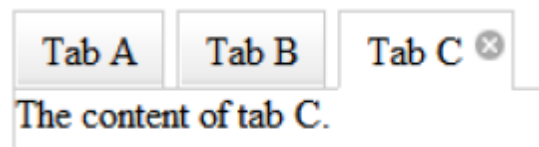
```

Remarque:

1. Le contenu de l'onglet B est récupérée à la demande à partir d'un fichier externe.
2. Tab C peut être supprimé (refermable est définie sur true).
3. Tab C se ouvre par défaut (sélectionné est définie sur true).

Enregistrer les modifications.

Exécutez le fichier dans le serveur



Etape5 : Utilisation de la grille Widget

TurboAjax Inc. a fait don de leur widget, la grille XHTML commerciale au Dojo. Il a été intégré dans Dojo que le widget dojox.Grid. L'existence de "dojox" dans le nom du paquet indique son encore au stade expérimental. Toutefois, le widget, dans sa forme originale (TurboWidget) est un sophistiqué. Nous allons commencer petit et d'exposer lentement les fonctionnalités les plus avancées.

Nous allons créer la grille la plus simple possible, qui ressemble à ceci.

Product Title	Price	Type
Baseball gloves	12.34	Sports
Tennis ball	5.99	Sports
T-shirt	12.45	Clothing
Hat	12.45	Clothing

Copiez template.html et le coller comme grid.html.

Ouvrir grid.html. Ajouter une déclaration nouvelle dojo.require () comme indiqué en gras ci-dessous.

```
dojo.require("dijit.form.Button");  
dojo.require("dojox.grid.Grid");
```

Remarque: Bien, le nom du widget est dojox.Grid, vous devez importer dojox.grid.Grid. Cela peut être tout simplement une incohérence qui sera corrigé à l'avenir.

Importer un fichier spécial de CSS, juste pour le widget de grille. Remarque: Ce fichier CSS est absolument nécessaire. Sans elle, le widget de grille échouera tout simplement de rendre correctement.

```
<link rel="stylesheet" type="text/css"  
href="dojo-release-1.0.0/dijit/themes/tundra/tundra.css" />  
<link rel="stylesheet" type="text/css"  
href="dojo-release-1.0.0/dojox/grid/_grid/tundraGrid.css" />
```

Créer la structure d'affichage

Le widget de la grille peut vous aider à créer des tableaux très complexes. Malheureusement, cela signifie, vous avez besoin de prendre un peu de temps pour apprendre comment il fonctionne, même lorsque vous créez une table très basique.

Une vue est une unité verticale indépendamment de défilement. Dans la plupart des cas, nous avons besoin de faire défiler toute la table comme une unité. Par conséquent, dans la plupart des cas, il y aura une vue unique par table.

Une vue est une collection de lignes. C'est la partie facile. Nous savons tous que la table est une collection de lignes et qu'une ligne a plusieurs colonnes. Cependant, dans une grille, une rangée peut avoir plusieurs sous-lignes et chaque sous-ligne peut avoir un nombre différent de colonnes.

Dans notre cas, nous avons une table très simple. Il y aura un seul sous-ligne par ligne. Ce sous-ligne aura trois colonnes - Nom du produit, le prix et le type.

Product Title (Subrow 1, Column 1)	Price (Subrow 1, Column 2)	Type (Subrow 1, Column 3)
------------------------------------	----------------------------	---------------------------

Dessous de la ligne:

```
dojo.require("dojox.grid.Grid");
```

Ajoutez

```
var subrow1 = [{name: "Product Title"}, {name: "Price"}, {name: "Type"}];
```

Remarque, comment nous ajoutons trois colonnes à la sous-ligne. Chaque objet de la colonne possède une propriété de nom qui définit le titre d'en-tête de la colonne.

Ensuite, ajoutez la ligne suivante à définir la vue.

```
var view = {rows: [ subrow1 ]};
```

Dans notre cas, la vue ne dispose que d'une sous-ligne. L'objet de vue possède une propriété appelée lignes qui prend en tableau des sous-lignes. Remarque: Le nom de la propriété est déroutant. Idéalement, il aurait dû être appelé sous-lignes. Si cela peut aider à tous, vous pouvez également utiliser la propriété de cellules en place de lignes au même effet.

Enfin, définir l'ensemble de la structure de grille. Dans notre cas, nous avons une seule vue.

```
var structure = [ view ];
```

Dans la balise <body>, ajouter la grille comme suit.

```
<body class="tundra">
<div dojoType="dojox.Grid" autoWidth="true" structure="structure"></div>
</body>
```

L'attribut autoWidth est important. Cette règle automatiquement la largeur de la grille à une valeur raisonnable. Sans elle, la grille prend toute la largeur de la page et ressemble vraiment mauvais.

Enregistrer les modifications.

Exécutez le fichier sur le serveur.

Product Title	Price	Type

Assurez-vous que la structure de la grille semble bon.

Données approvisionnement pour la grille

Les données sont fournies sous forme de tableau JavaScript simple. Il se agit d'un ensemble de rangées. Chaque ligne, contient un tableau de données cellulaires.

Ajoutez le code de script suivant:

```
var data = [
["Baseball gloves", 12.34, "Sports"],
["Tennis ball", 5.99, "Sports"],
["T-shirt", 12.45, "Clothing"],
["Hat", 12.45, "Clothing"]
];
```

```
var productModel = new dojox.grid.data.Table(null, data);
```

Cr  er le mod  le avec la grille:

```
<div dojoType="dojox.Grid" autoWidth="true" model="productModel"
structure="structure"></div>
```

Enregistrer les modifications.

Ex  cutez le fichier sur le serveur.

Product Title	Price	Type
Baseball gloves	12.34	Sports
Tennis ball	5.99	Sports
T-shirt	12.45	Clothing
Hat	12.45	Clothing

