

UNIVERSITÉ DE ROUEN

MASTER 1 GIL - 2010/2011

Manuel d'Utilisation - HTML5Charts

Equipe de projet

Pierre COLLIGNON

Ludovic THUEUX

Maxence LUCE

Lyes KIMOUCHE

Radu POPICA

Abdourahmane DJIGO

Client

Florent NICART

Table des matières

1	Une librairie de diagramme en HTML5	2
1.1	Les diagrammes	2
2	Utilisation simple de la librairie	4
2.1	Création d'une source de données interne	4
2.2	Création d'un diagramme	4
2.3	Exemple	4
3	Utilisation avancée de la librairie	5
3.1	Utilisation de sources externes	5
3.2	Utilisation de connecteurs	5
3.3	Stylisation	5
3.3.1	Description du XML de stylisation	5
3.3.2	Utilisation des sources externe et interne de stylisation	6

Chapitre 1

Une librairie de diagramme en HTML5

Le produit *HTML5Charts* est une librairie JavaScript permettant le dessin de diagrammes représentant un ensemble de données cohérent. L'automatisation du traitement des données en vue de l'affichage est entièrement personnalisable par le biais d'options fournies à l'utilisateur de la librairie, ou par la réimplémentation des interfaces prévues à cet usage.

Ce produit a été conçu pour prouver la faisabilité du remplacement progressif de Flash par HTML5.

1.1 Les diagrammes

Quatre types diagrammes sont disponibles :

- L'histogramme
- L'histogramme avec effet 3D
- Le diagramme circulaire
- Le diagramme en ligne

Pour chacun de ces composants, deux représentations sont possibles car les données sont modélisée par un tableau à double entrée.

On peut donc dessiner en fonction des lignes ou en fonction des colonnes de ce tableau.

Les dessins seront différents et le choix entre les deux s'effectue par rapport aux données que l'utilisateur veut mettre en avant.

Un exemple permettant de visualiser ce principe pourrait être celui d'une société désirant faire un bilan du nombre d'acheteurs pour chaque produit qu'elle vend et pour chaque succursale. Elle pourra vouloir soit considérer le nombre de ventes par succursale pour chaque produit, ou elle mettra donc en avant la comparaison entre chaque produit à l'intérieur d'une succursale, ou le nombre de ventes par produit pour chaque succursale, ou elle mettra en avant la comparaison entre chaque succursale pour un type de produit donné.

Cette bibliothèque sera utilisée par deux catégories d'utilisateurs, la personne voulant intégrer un diagramme sur son site et la personne regardant le diagramme sur ce site, nous les appellerons respectivement le développeur et l'internaute.

Pour le développeur, comme dit précédemment, l'objectif est d'intégrer un diagramme sur son site. Pour se faire, il disposera de notre bibliothèque contenant les quatre diagrammes ainsi que d'autres fonctionnalités exprimées plus loin.

Cette personne communiquera ses données sous forme d'un fichier XML – respectant les contraintes d'un fichier XSD fourni, applicable indépendamment du diagramme souhaité. Cette spécification des données pourra être donnée de deux façons différentes : soit l'écrire directement dans la page HTML – appelé en interne –, soit être séparé de la page dans un fichier où son chemin serait spécifié – appelé en externe.

Le style – la couleur des dessins, la position de la légende – pourra être spécifié de la même manière que les données mais indépendamment.

Toutes les combinaisons sont possibles, les données peuvent être fournies en interne et le style en externe et inversement, les deux en interne ou les deux en externe.

Pour l'internaute, le diagramme est déjà intégré sur un site qu'il visite. Il a également un rôle sur ces diagrammes – hormis sur le diagramme en ligne – puisqu'il peut interagir avec. Lorsqu'il passe la souris sur une des données – une barre sur les histogrammes ou une part sur le diagramme circulaire – celle-ci est mise en valeur par un changement de couleur. Ceci lui permettra de mettre en évidence une donnée. Egalement, en plus de cela, une information sera donnée sous forme d'info-bulle donnant le nom de la donnée ainsi que sa valeur exacte. Ce système affiche un rectangle incluant ce texte et suit la souris tant qu'elle est à l'intérieur de la donnée.

Chapitre 2

Utilisation simple de la librairie

La librairie peut être utilisée simplement et immédiatement afin d’afficher les diagrammes en utilisant la source de données interne avec une stylisation par défaut.

2.1 Création d’une source de données interne

La source de données interne est un fichier XML contenu dans une balise html `<pre>`. Rien de particulier n’est requis pour l’utiliser, il faudra cependant veiller à respecter la XSD spécifiée pour le formatage XML.

Une fois les données présentées sous la forme d’un XML, un objet JavaScript `InternalDataSource` devra être utilisé pour lire son contenu.

Cet objet respecte l’interface `IDataSource` qui possède deux méthodes : `loadData(callback)` et `getDataMatrix()`. La méthode `loadData` prend en argument une fonction qui sera appelée lorsque les données auront été chargées, la méthode `getDataMatrix` permettant d’accéder à une vue abstraite des données, utilisable durant tout le reste de la vie du programme.

2.2 Création d’un diagramme

Quatre classes de diagrammes sont utilisables : `HistoDiagram`, `Histo3DDiagram`, `PieDiagram` et `LineDiagram`.

La création d’un diagramme après chargement d’une source de données interne se fait en fournissant dans le constructeur de la classe désirée l’objet DOM correspondant au Canvas dans lequel il doit être dessiné ainsi que le type de traitement désiré (ligne ou colonne). Ensuite, une fois l’objet diagramme créé, nous pouvons utiliser la méthode `setData(dataMatrix)` de l’interface `IDiagram` afin de fournir au diagramme une matrice des données.

La méthode `setData` redessine immédiatement le diagramme.

2.3 Exemple

Nous considérons que la balise `<pre>` contenant le fichier xml a pour identifiant `pre1` et que la balise `<canvas>` contenant la zone d’affichage a pour identifiant `canvas1`.

```
var ids = new InternalDataSource('pre1');
ids.loadData(function() {
// Dès que la source de données est chargée, cette fonction est exécutée
    var diag = new PieDiagram(document.getElementsByTagName('canvas1')[0], 'row');
    diag.setData(ids.getDataMatrix());
});
```

Chapitre 3

Utilisation avancée de la librairie

Nous présentons ici une utilisation plus complexe de la librairie, permettant d'utiliser des fichiers XML sur un serveur externe, d'utiliser autre chose que le format XML, et de changer la forme par défaut du dessin des diagrammes.

3.1 Utilisation de sources externes

Les sources externes permettent de déléguer la fourniture du fichier XML à un tier, qui devra cependant respecter notre XSD. L'utilisation est similaire à celle de la source interne, à la différence près que lors de la création de l'objet `ExternalDataSource` on devra passer l'url du fichier XML au lieu de l'identifiant de la balise `<pre>`.

3.2 Utilisation de connecteurs

Les connecteurs sont un des aspects essentiels de la modularité de la librairie. Ces objets doivent être capable de créer un diagramme pour n'importe quel type de données. On peut par exemple créer un diagramme en remplaçant le format XML par le format JSON. Un connecteur est créé par l'utilisateur de la librairie en fournissant au constructeur de l'objet `ConnectorDataSource` l'implémentation personnalisée des deux fonctions `loadData` et `getDataMatrix`.

Un exemple utilisant le JSON est fournit dans le code source du projet.

3.3 Stylisation

La stylisation permet de personnalisé profondément l'apparence des diagrammes. Afin de rester cohérent avec la fourniture des données, la stylisation est donnée par un fichier XML, soit interne au site web, soit chez un fournisseur externe.

3.3.1 Description du XML de stylisation

La stylisation consiste à fournir un ensemble de couleur différent de l'ensemble par défaut, ainsi que des paramètres de positionnement de la légende.

```
<style>
<colors>
<color>Nom_couleur</color>
...
</colors>
<legend>
<x>abscisse du coin supérieur gauche</x>
<y>ordonnée du coin supérieur gauche</y>
```

```
<w>longueur du rectangle</w>
<h>hauteur du rectangle</h>
</legend>
</style>
```

3.3.2 Utilisation des sources externe et interne de stylisation

L'utilisation des sources de stylisation est très similaire à celle des sources de données : On crée d'abord un objet de type `InternalStyleSource` ou `ExternalStyleSource` en fournissant respectivement un identifiant de balise `<pre>` ou une url de fichier XML. Les deux méthodes disponibles chez ces objets implémentant l'interface `IStyleSource` sont `loadData` et `getStyleSource` qui se comportent exactement comme leur équivalent d'`IDataSource` .