

## ANNEXE 4 Le projet Velib'

Pour JCDecaux, le projet s'est déroulé sous des fortes contraintes de temps : démarrage en janvier pour une livraison au 14 juillet 2007 au plus tard sous peine de pénalités de retard importantes. Ce délai était non négociable pour des raisons politiques.

A ces contraintes de temps s'ajoutaient des exigences fortes de la mairie pour avoir un regard sur l'efficacité du système et obtenir des engagements sur cette efficacité (avec pénalité si le contrat était non respecté).

Par exemple, le temps d'attente maximum pour prendre ou déposer un vélo dans un rayon de 300 m devait être inférieur à 3 minutes. De plus des contraintes de disponibilité de l'ensemble du système mis en place avaient été contractualisées : 200 000 euros de pénalité pour plus d'une heure d'arrêt dans le mois.

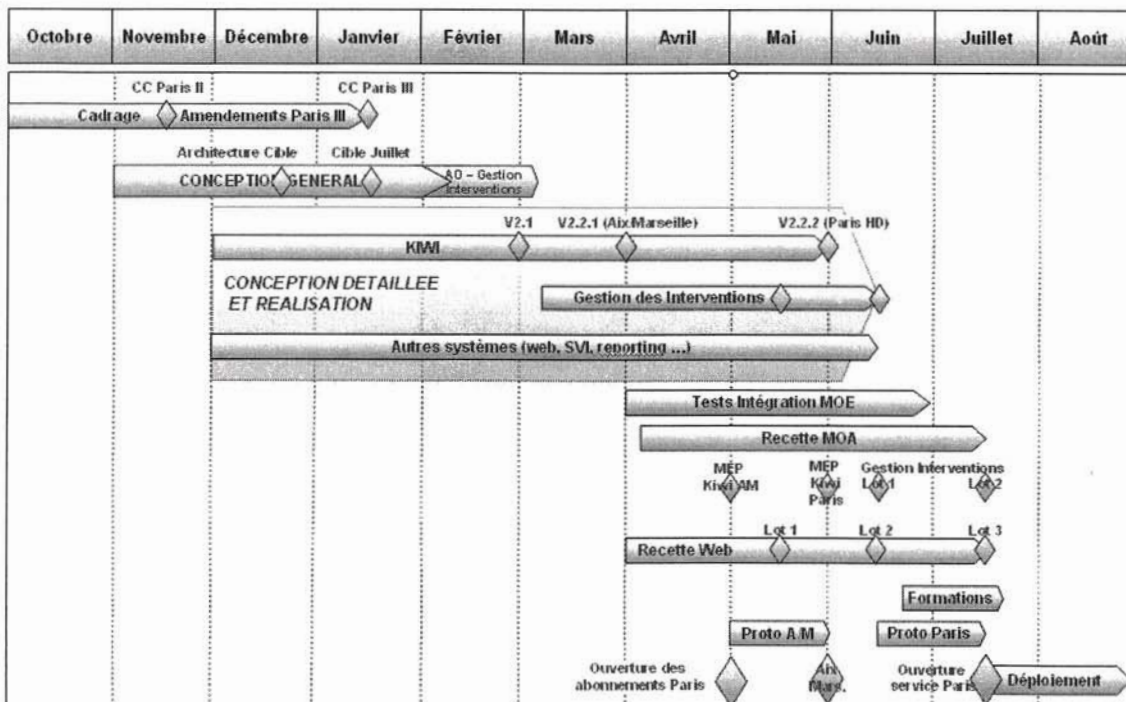
La gestion de projet mise en place a été un mélange « d'approche commando » avec plus de 900 personnes mobilisées et « d'approche fortement structurée » avec un système de planification détaillé par jalons.

Des services prévus ont été mis en attente ou externalisés (par exemple, le rachat d'une solution de *call-center*) en attente de solutions plus adaptées dans un deuxième temps.

Vous trouverez ci-dessous des extraits de compte rendu du comité de pilotage concernant la gestion de projet :

« [...] Il a été demandé au chef de projet d'identifier les points de blocage en termes de ressources ainsi que les projets informatiques devant être décalés afin de respecter le planning. Ces points feront l'objet d'une communication au comité de direction (CODIR).

- Le *reporting* d'utilisation de l'application ne sera pas opérationnel à l'ouverture du système. Des *reporting* partiels seront réalisés et agrégés manuellement.
- Le planning doit être confronté aux contraintes de planning MOA : principalement recette, fonctionnement en mode prototype et formation. À ce stade, la MOA signale qu'une livraison de la version de Paris au 31/05 ne permettra pas de réaliser les recettes et les formations dans les temps impartis. Il faut donc travailler sur un lotissement avec des livraisons intermédiaires de façon à commencer la recette avant fin mai.
- A été soulevé la nécessité de statuer rapidement sur l'externalisation ou non de la production des badges afin de prendre en compte dans le planning les évolutions à intégrer au module *Kiwi* (gestion des bornes) en cas d'externalisation. [...] »



## ANNEXE 5 Velib' et développement durable

### ***Dossier de presse Velib (extraits) : Rémi Pheulpin – Directeur général Recherche, Production et Opérations JCDecaux***

La régulation des vélos se fera avec des véhicules propres GNV. Les déplacements des agents de maintenance s'effectueront avec des vélos à assistance électrique.

Une péniche, disposant de douze escales tout au long de la Seine, récupérera les vélos nécessitant une maintenance lourde. À son bord, un marinier et onze mécaniciens pour remettre à neuf les vélos endommagés.

Le gros entretien, nécessitant de l'eau, sera assuré par des agents se déplaçant avec dix véhicules électriques transportant de l'eau de pluie récupérée sur les toits des sites JCDecaux. Ce procédé permet la suppression de l'utilisation de détergent pour l'entretien. L'eau « pure » possède en effet un pouvoir nettoyant naturel. Le produit antigraffiti est également écologique.

Le vélo est recyclable à 99 %. JCDecaux a d'ailleurs mis en place une filière de recyclage des pneus de vélo usagés, ce qui est une première en France.

### ***Développement durable et SI - Source : empa.ch : L'Empa est un institut de recherche en science des matériaux et en technologie – 2003***

L'absence quasi-totale de prise de conscience des problèmes d'énergie et de pollution dans le domaine des technologies de l'information est stupéfiante. Et cela bien que la part de la technologie de l'information dans la consommation totale de l'énergie et des ressources s'accroisse sans cesse.

L'optimisation des processus à l'aide de la TI est une voie qui selon eux s'ouvre pour réaliser un telle dématérialisation, par exemple avec le routage "intelligent" (« *dynamic vehicle routing* ») en transport routier, ou l'impression digitale avec laquelle l'exemplaire d'un livre n'est imprimé que lorsqu'un acheteur se présente (« *print on demand* »), ou encore en agriculture où la surveillance par satellite permet un dosage précis des engrais en fonction des caractéristiques du sol (« *precision farming* »).

L'Empa a étudié quel serait le potentiel écologique d'un remplacement des médias imprimés par des médias électronique et des vidéoconférences (au lieu de conférences traditionnelles avec les déplacements que cela implique).

Il est nécessaire de tenir compte du fait qu'à elles seules les solutions techniques ne peuvent pas conduire à un développement durable. Le plus grand risque réside dans de qu'on appelle l'effet de rebond par lequel les économies de temps et d'argent rendues possibles sont compensées ou plus que compensées pour la satisfaction d'une demande supplémentaire. Ainsi par exemple le télétravail ne conduit pas forcément à une diminution du trafic routier. Les télé-travailleurs n'utilisent certes plus leur voiture pour se rendre à leur travail, mais s'en servent davantage pour se rendre dans un centre de fitness ou pour d'autres activités de loisirs.

### ***SI au service du développement durable - source wikipedia***

Ce serait simpliste de croire que la problématique du développement durable dans les systèmes d'information se résume au recyclage du matériel informatique. Se mettre au service du développement durable implique de structurer les informations utiles pour la gestion des programmes concernés, et plus particulièrement pour la gestion des données. Étant donné l'ampleur du volume des données manipulées, il est apparu qu'il fallait employer des référentiels de données pour la documentation, permettant aux utilisateurs d'accéder à l'information, de la partager, et de structurer leurs propres dictionnaires de données.

Différentes organisations aux États-Unis ont lancé des programmes sur les métadonnées pour l'environnement et des secteurs connexes.

### ***2006 - JCDecaux remporte le Prix Usine Nouvelle des ingénieurs de l'année pour Cyclocity® (Velo'v)***

JCDecaux SA, numéro un de la communication extérieure en Europe et numéro deux mondial, a reçu hier le prix Usine Nouvelle des ingénieurs de l'année (catégorie Développement Durable) pour Cyclocity®. [...] En récompensant l'équipe de 50 collaborateurs de JCDecaux qui a conçu ce projet en interne, le jury a voulu distinguer une réalisation innovante (8 brevets déposés) qui illustre l'alliance entre l'industrie et la technologie au service de la société.

## ANNEXE 6

### Extrait d'une interview de M. Bertrand Kienz (DSI de JCDecaux) – juillet 2007

#### Quels ont été les choix techniques pour cette plate-forme Web ?

Plus que des choix d'outils, c'était le choix d'un partenaire puisque nous nous sommes reposés sur ses compétences et les technologies avec lesquelles il était le plus efficace. Et c'était d'autant plus important que nous ne disposions que de 4 mois et demi pour parvenir à une solution performante. Sur Paris, c'est Smile, une société avec laquelle nous avons déjà travaillé dans le passé, notamment pour le site Vélo'v à Lyon, qui a remporté l'appel d'offres sur l'ensemble de notre cahier des charges. Et Smile a gagné avec la solution de gestion de contenu *eZ publish* qui présentait de nombreux avantages, notamment en termes de capacité de publication et de paramétrage en fonction des sites et systèmes des villes. Le système ne sera en effet pas seulement implémenté à Paris, mais aussi à Séville, Marseille, Mulhouse, etc. Nous avons donc besoin d'un noyau un peu commun à l'ensemble tout en permettant aux différentes villes d'y mettre leur touche, notamment via une charte graphique qui leur soit propre. L'autre aspect du projet tient à son caractère dynamique, aux fonctions opérationnelles délivrées par le biais de *Web services*. A ce niveau, il était impératif de bien maîtriser le développement et la maintenance car ce sont des services qui étaient amenés à être très sollicités. Par conséquent, compte tenu des exigences de disponibilité, nous assurons nous-même l'hébergement.

*Christophe AUFFRAY, JDN Solutions*

### Extrait d'un entretien avec Bertrand Kientz (DSI de JCDecaux) – décembre 2007

Les quelques mois d'exploitation de Velib' nous ont permis de repérer et de corriger les principaux « péchés de jeunesse ». Il nous reste quelques difficultés à surmonter, notamment celle de la régulation des vélos qui représente un enjeu important. Nous travaillons sur des études statistiques qui nous permettent de proposer à la Mairie de Paris un maillage plus efficace des stations. Notre dispositif de régulation fonctionne à plein mais ne peut en effet résoudre à lui seul ce problème. Nous régulons en moyenne 4 000 vélos par jour pour un nombre journalier de trajets qui dépasse 200 000, ce qui nous permet de respecter notre engagement contractuel : plus de 95 % des accès nécessitent moins de 3 minutes pour déposer ou accéder à un vélo sur une station ou les stations voisines. En octobre 2007, nous avons atteint 97%. Pour encore améliorer la fluidité, nous réfléchissons à un projet de tarification qui permettra une auto-régulation, sur le principe de la modulation des tarifs de péage.

Nous mettons en place un tableau de bord qui va nous permettre de mieux déterminer les axes d'amélioration de l'application. Nous avons ainsi déjà pu constater des problèmes de communications GPRS entre les bornes et notre centre d'exploitation qui réduisent sensiblement la disponibilité de notre système. Nous réfléchissons à l'amélioration de nos bornes qui leur assure un fonctionnement et une communication avec le site central 24/24.

Nous réfléchissons également à l'enrichissement de l'interface de notre module KIWI, qui gère les bornes, pour permettre aux utilisateurs de nous informer directement sur la borne d'un dysfonctionnement constaté, afin de leur éviter de contacter notre centre d'appel.

## ANNEXE 7

### Gestion de la maintenance corrective

#### L'entretien et la maintenance des vélos

Le nettoyage régulier des velib' (nettoyage et contrôle visuel de leur état) est assuré par des agents d'entretien se déplaçant avec dix véhicules électriques transportant de l'eau de pluie.

Pour les petites réparations, les déplacements des 200 agents de maintenance de premier niveau (échange standard de pièces et réparations mineures) s'effectuent avec des vélos à assistance électrique disposant d'une remorque pouvant accueillir quelques velib'. Chaque station dispose d'un stock de pièces de rechange (patins et câbles de frein, pédales, ampoules, roues...). Les agents de maintenance, équipés de PDA synchronisés avec le système d'information, sont renseignés sur l'état des stations de leur secteur géographique (environ douze stations par agent) : la disponibilité des vélib', les interventions à réaliser, ...

Chaque secteur géographique est associé à un atelier de réparation qui assure les opérations de maintenance de deuxième niveau sur les vélib' défectueux et accueille les clients ayant des problèmes avec leur vélib' (échange standard).

Une péniche, disposant de douze escales tout au long de la Seine, traite les vélib' nécessitant une maintenance lourde. À son bord, onze opérateurs mécaniciens sont chargés de remettre à neuf les vélib' endommagés (maintenance de troisième niveau). La péniche dispose d'un stock d'équipements (vélib' neufs ou remis en état) et de pièces de rechange permettant d'assurer toutes les tâches journalières de maintenance des velib' et d'alimentation des stocks des stations Velib' et des ateliers de réparation. À chaque escale, des containers permettent de stocker les pièces de rechange nécessaires aux ateliers qui y sont rattachés. Ces pièces sont ensuite déposées dans les ateliers à l'occasion des tournées de régulation.

Un magasin central, situé à Puteaux, à proximité d'une des escales, est chargé de la réception des livraisons et du stockage des pièces de rechange et des vélib' en attente d'affectation.

#### La régulation des vélib'

Vingt véhicules propres permettent d'assurer la régulation des vélib' et l'approvisionnement des ateliers et des stations en pièces de rechange. Chaque tournée, à partir de données statistiques sur l'état de chaque station, consiste à récupérer des velib' dans des stations surchargées d'un ou plusieurs secteurs pour les déposer dans d'autres qui n'en ont pas assez. Chaque velib' endommagé et non réparable sur place est ramené à l'escale associée à la tournée. Des containers permettent de stocker les vélos endommagés dans l'attente du passage de la péniche et, inversement, de stocker des vélos neufs ou réparés déposés par la péniche.

#### Quelques indicateurs significatifs à intégrer au tableau de bord de la maintenance

- MTBF (*Mean Time Between Failure*) : moyenne des temps entre deux défaillances d'un équipement réparable.
- MTTR (*Mean Time To Repair*) : durée moyenne de réparation de l'équipement. C'est le ratio du temps technique de réparation par le nombre de défaillances.
- Indicateur de disponibilité : ratio du temps disponible sur le temps total de service.
- Indicateur de coût de maintenance : ratio des coûts de maintenance sur le nombre d'équipements.

#### Détermination des pièces critiques

Le responsable de la maintenance a choisi de déterminer le taux de criticité d'une pièce à partir de la formule  $CR = P \times I \times E \times U$  où chaque indicateur peut prendre différentes valeurs :

- P = Incidence d'une panne sur l'équipement (1 : inutilisable, 2 : pb de sécurité, 3 : moindre confort, 4 : sans incidence)
- I = Importance de la pièce (1 : coûteuse sans pièce de remplacement, 2 : coûteuse avec pièces de remplacement, 3 : bon marché sans pièce de remplacement, 4 : bon marché avec pièces de remplacement)
- E = État de la pièce (1 : à réformer, 2 : à rénover, 3 : mauvais état, 4 : bon état, 5 : neuf)
- T = Taux d'utilisation (1 : 100 %, 2 : fort, 3 : moyen, 4 : faible, 5 : très faible)

Exemple :

- pour une béquille latérale, P = 3, I = 4, E = 4, T = 3, soit CR = 144 (pièce non critique)
- pour une chaîne, P = 1, I = 2, E = 2, T = 1, soit CR = 4 (pièce critique)

## ANNEXE 8

### Procédure de traitement d'une demande de travail (maintenance corrective)

#### Maintenance corrective

1. Agent de régulation
  - Si l'agent de régulation constate un défaut lors d'un passage en station, il bloque le vélo défectueux repéré sur sa bornette.
  - Il effectue ensuite une demande de travail et, s'il estime que la réparation ne peut être effectuée sur place, il l'associe à une demande d'enlèvement.
2. Agent de maintenance de premier et de deuxième niveau
  - Si l'agent de maintenance constate un défaut lors d'un passage en station, il lance un ordre de travail correctif (OTC)
    - S'il peut effectuer la réparation sur place, il clôture l'OTC lorsque le travail est terminé
    - sinon il échange le velib' défectueux contre un velib' en état et le ramène dans l'atelier de réparation du secteur
    - s'il peut réaliser la réparation dans l'atelier, il clôture l'OTC lorsque le travail est terminé
    - si la réparation s'avère impossible, il cascade l'OTC vers la péniche, avec demande d'enlèvement
  - Si l'agent intervient à la suite d'un OTC
    - il échange le velib' défectueux contre un velib' en état et le ramène dans l'atelier de réparation du secteur
    - si la réparation peut être réalisée dans l'atelier, l'OTC est clôturée lorsque le travail est terminé
    - si la réparation s'avère impossible, l'OTC est cascadée vers la péniche, avec demande d'enlèvement.

#### Pour toutes les demandes de travail non prises en charge immédiatement

1. Le système génère l'OTC et précise le temps de travail et les pièces nécessaires.
2. Le système propose une planification (date, heure, durée, opérateur mécanicien) tenant compte du délai de retour du velib' dans l'atelier du secteur ou à la péniche et alerte en cas de risque de rupture de stock, indiquant le temps de mise à disposition des pièces nécessaires.
3. Le responsable de maintenance peut, si nécessaire, valider une demande exceptionnelle de réapprovisionnement
4. Après validation du responsable de maintenance, le système génère l'ordre d'enlèvement en précisant le lieu, la date et l'heure de l'escale de la péniche pour la récupération et met à jour le planning de l'opérateur mécanicien concerné.

#### Clôture d'une OT

L'agent qui clôture un OT fournit un compte-rendu du travail et la liste des pièces utilisées. Le système met à jour la configuration du velib' et lance automatiquement les ordres de réapprovisionnement vers le magasin en amont.

## Extrait du schéma relationnel de la gestion des OTC

Velib(numero, codeModele, dateMiseEnService, etat, dateMiseHorsService, motif, ...)

numero : clé primaire

codeModele : clé étrangère en référence à code de MODELE (non décrit ici)

*Les velib' sont disponibles 24/24, 7/7. Un velib' est pris en compte dans les statistiques de sa première mise en service à sa mise hors service, quel qu'en soit le motif. Les motifs principaux sont le vol (motif = "VOL"), les dégradations volontaires (motif="DGR"), les accidents (motif="ACC") et la vétusté (motif="VTS").*

OTC(numero, dateHeureDemande, numVelib, dateHeureDebutPanne, descriptionCause, ...)

numero : clé primaire

numVelib : clé étrangère en référence à numero de VELIB

PriseEnChargeOTC(numero, numOTC, dateHeureDebutReparation, ...)

numero : clé primaire

numOTC : clé étrangère en référence à numero de OTC

ClotureOTC(numero, numOTC, dateHeureRemiseEnService)

numero : clé primaire

numOTC : clé étrangère en référence à numero de OTC

On dispose de :

- la fonction *DATEDIFF(partie\_date, dateDebut, dateFin)* qui renvoie l'intervalle entre deux dates, où *partie\_date* spécifie la partie de date dans laquelle l'intervalle doit être mesuré (*year | quarter | month | week | day | hour | minute | second | millisecond*). Le résultat est un entier signé égal à (dateFin - dateDebut) exprimé en parties de date. Une constante date peut être représentée sous forme d'une chaîne ('2007-03-14 17:41:25') ou d'un entier (20070314174125).
- La fonction *DATEJOUR()* qui retourne la date système.

## ANNEXE 9

### Exemple d'exploitation des marqueurs (site officiel et mashup)

The screenshot displays the Velib' website interface. At the top, the Velib' logo is on the left, and the slogan "la ville est plus belle à vélo" is on the right. Below the logo is the "MAIRIE DE PARIS" logo. A navigation bar contains links for "ACCUEIL", "COMMENT ÇA MARCHE?", "ABONNEMENTS ET TARIFS", "ACTUALITÉS", and "LES STATIONS".

The main content area is divided into several sections:

- Votre recherche:** A search box with instructions: "Sélectionnez une station et cliquez sur Plus de détails pour obtenir ici des informations supplémentaires." It includes a "Nouvelle recherche" button, a "N° de station" input field, and links for "Envoyer à un ami" and "S'abonner".
- Les stations:** A map of Paris with a pop-up window for "Station n° 4018" at "1 RUE SAINT DON". The pop-up shows "Vélos disponibles : 19" and "Points d'attache disponibles : 0". It includes links for "Plus de détails..." and "Ajouter à mes favoris".
- Accéder à mon compte:** A button labeled "Mon compte" with a right-pointing arrow.
- Activer mon compte Velib':** A section with the text: "Vous venez de recevoir votre confirmation d'abonnement? Vous avez 45 jours pour activer votre compte Velib'. Activer mon compte".
- Rechercher:** A button labeled "Rechercher dans le site" with a right-pointing arrow.
- Liens utiles:** A search bar with a dropdown arrow.
- Abonnez-vous:** A section with a "16°C" weather indicator and a "Créer un profil" button. It includes instructions for using the map and a list of options: "Stations par arrondissements", "Stations par adresse", "Sélection directe", "Trajets", and "Bookmarks".

The map shows various Parisian districts and station locations. A pop-up window on the map says: "Bienvenue sur la carte Vélib'. Double cliquez sur la carte pour trouver les stations les plus proches ou utilisez les onglets à gauche".

### Extrait du fichier XML *Vdonnees.xml*

```
<?xml version="1.0" encoding="utf-8" ?>
<markers>

<marker lat="48.877809" lng="2.347025" info="5 rue de bellefond 75009 Paris"
details="Quelques informations supplémentaires" />

<marker lat="48.883043" lng="2.346605" info="19 rue d abbeville 75009
Paris" details="Quelques informations supplémentaires" />

...
...

</markers>
```

**Extrait du fichier XHTML permettant la création de la carte et des marqueurs  
depuis le fichier XML *Vdonnees.xml***

```
1. <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
2.   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
3. <html xmlns="http://www.w3.org/1999/xhtml">
4.   <head>
5.     <meta http-equiv="content-type" content="text/html; charset=utf-8"/>
6.     <title>Utilisation de XML</title>
7.     <script src="http://maps.google.com/maps?file=api&v=2"
8.       type="text/javascript"></script>
9.     <script type="text/javascript">
10.    // CDATA permettant de définir une section ne devant pas être analysée par le processeur XML
11.    //<![CDATA[
12.      function load() {
13.        if (GBrowserIsCompatible()) {
14.          // crée la carte
15.          var carte = new GMap2(document.getElementById("map"));
16.          carte.addControl(new GSmallMapControl());
17.          carte.addControl(new GMapTypeControl());
18.          // centre la carte sur les coordonnées géographiques
19.          carte.setCenter(new GLatLng(48.9, 2.3), 8);
20.          // création d'un marqueur
21.          function creerMarker(point, onglet1, onglet2) {
22.            var marker = new GMarker(point);
23.            // création des fenêtres d'information
24.            var infoTabs = [
25.              new GInfoWindowTab("Infos", onglet1),
26.              new GInfoWindowTab("Détails", onglet2)
27.            ];
28.            GEvent.addListener(marker, "click", function() {
29.              marker.openInfoWindowTabsHtml(infoTabs);
30.            });
31.            return marker;
32.          }
33.          // charge le fichier Xml
34.          GDownloadUrl("Vdonnees.xml", function(data) {
35.            // parse le texte du document Xml et retourne une représentation DOM.
36.            var xml = GXml.parse(data);
37.            var markers = xml.documentElement.getElementsByTagName("marker");
38.
39.            for (var i = 0; i < markers.length; i++) {
40.              var point = new GLatLng(parseFloat(markers[i].getAttribute("lat")),
41.                parseFloat(markers[i].getAttribute("lng")), 9);
42.              var marker = creerMarker(point, markers[i].getAttribute("info"), markers[i].getAttribute("details"));
43.              // ajoute le marqueur à la carte
44.              carte.addOverlay(marker);
45.            }
46.          });
47.        }
48.      //]]>
49.    </script>
50.  </head>
51.  <body onload="load()" onunload="GUnload()">
52.    <div id="carte" style="width: 600px; height: 200px"></div>
53.  </body>
54. </html>
```

**Tournez la page S.V.P.**



## ANNEXE 10

### Extrait du fichier *stations.xml*

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<arrondissements>
<arrondissement id = "9" nom="9e arrondissement " />
<stations>
<station id= "9001" etat="vide" Velos="0" nbbornes="16" erreurs="4" Adresse="5 rue de bellefond 75009 paris"/>
<station id= "9002" etat="vide" Velos="0" nbbornes="14" erreurs="2" Adresse="19 rue d' abbeville 75009 paris"/>
<station id= "9005" etat="disponible" Velos="12" nbbornes="7" erreurs="1" Adresse="95 rue de Dunkerque 75009 Paris "/>
...
</stations>
<arrondissement id = "10" nom = "10e arrondissement"/>
<stations>
<station id= "10001" etat="disponible" Velos="9" nbbornes="11" erreurs="0" Adresse="face au 50 rue rené boulanger 75010
paris"/>
...
</stations>
...
</arrondissements>
```

## ANNEXE 11

### Extrait des interfaces du modèle objet de document XML (DOM)

#### Interface *Document* (extrait)

```

interface Document : Node {
...
    readonly attribute Element          documentElement;
...
    NodeList                            getElementsByTagName(in DOMString tagName);
    // retourne une liste de nœuds de tous les objets Elements ayant un nom de balise donnée
    //(tagName ou * pour toutes), laquelle est ordonnée selon l'ordre de lecture prédéfini de
    // l'arbre du document.
};
    
```

#### Interface *NodeList*

```

interface NodeList {
    Node                                item(in unsigned long index);
    // retourne l'item spécifié par le paramètre index, ou null si la valeur du paramètre index
    // n'est pas valide
    readonly attribute unsigned long    length;
    // nombre de noeuds dans la liste
};
    
```

#### Interface *Node* (extrait)

L'interface *Node* est le type de donnée principal du DOM. Il représente un noeud unique de l'arbre du document. Il contient les méthodes et attributs utilisables par tous les types de nœuds.

```

interface Node {
    // Type de noeud : constantes de la propriété nodeType :
    const unsigned short  ELEMENT_NODE    = 1;
    // le noeud est un élément
    const unsigned short  ATTRIBUTE_NODE  = 2;
    // Le noeud est un Attr.
    const unsigned short  TEXT_NODE       = 3;
    //Le noeud est un Text.
    readonly attribute DOMString          nodeName;
    //Nom du noeud selon la propriété nodeType
    attribute DOMString                 nodeValue;
    //Valeur du noeud selon la propriété nodeType
    readonly attribute unsigned short     nodeType;
    // code du type de nœud
    readonly attribute Node               parentNode;
    // parent de ce noeud
    readonly attribute NodeList           childNodes;
    // liste de noeuds constituée des enfants du noeud
    readonly attribute Node               firstChild;
    // premier enfant de ce nœud ou Null si un tel noeud n'existe pas.
    readonly attribute Node               nextSibling;
    // noeud qui suit immédiatement le noeud courant ou null
    readonly attribute NamedNodeMap      attributes;
    //un NameNodeMap si c'est un élément, null sinon
    boolean                              hasChildNodes() ;
    // renvoie true si le noeud a un enfant, false sinon
...
};
    
```

### Interface *NameNodeMap* (extrait)

Les objets implémentant l'interface *NameNodeMap* sont utilisés pour représenter des collections de nœuds non ordonnées directement adressables par un nom

```
interface NameNodeMap {  
  Node getNamedItem(in DOMString name);  
  // renvoie un nœud à partir de son nom  
  Node item(in unsigned long index);  
  readonly attribute unsigned long length;  
  ...  
};
```

### Interface *Élément* (extrait)

```
interface élément : Node {  
  readonly attribute DOMString tagName;  
  // nom de l'élément  
  DOMString getAttribute(in DOMString attributname);  
  // valeur d'attribut à partir du nom de l'attribut.  
  Attribut getAttributeNode(in DOMString attributname);  
  // Permet d'obtenir un nœud de type Attr à partir de son nom.  
  NodeList getElementsByTagName(in DOMString name);  
  // renvoie tous les descendants avec le nœud name  
  ...  
};
```

### Interface *Attr* (extrait)

L'interface *Attr* représente un attribut d'un objet de type *Element*.

```
interface Attr: Node {  
  readonly attribute DOMString name;  
  // nom de l'attribut courant  
  readonly attribute boolean specified;  
  // Prend la valeur true si une valeur a été donnée à l'attribut courant dans le document //original ; false  
  // sinon.  
  attribute DOMString value;  
  // valeur de l'attribut  
};
```