

Mini Projet TIS4 XML 2020

Date	Version
7 mai 2020	1.0

Sommaire

- Objectif du mini-projet
- Objectifs techniques
 - Modélisation XML
 - Requêtes XPath
 - Programme Java
 - Web Service
- Évaluation
 - Livrables
 - Critères de notation
- References

Objectif du mini-projet

L'objectif de ce mini-projet est de vous permettre de réviser les concepts et technologies vus pendant l'enseignement de Technologies XML et Web Service et d'en mettre une partie en pratique.

Les années précédentes le mini-projet se déroulait pendant une séance d'examen par groupe de deux.

Cette année, le travail s'effectuera en groupe. Pour simplifier l'organisation nous conserverons les mêmes groupes que ceux formés pour le projet Système d'Information de Santé.

Après une brève introduction, les objectifs techniques de ce mini-projet sont décrits. Le paragraphe "Évaluation" fait le tour des modalités, merci de le lire *très attentivement*. Le sujet comporte également des liens pour vous aider tout au long de la réalisation.

La date de rendu des livrables est fixée au 31 mai.

L'énoncé peut sembler long, mais c'est écrit gros...

Plus sérieusement : tous les exercices ont été testés et vous demanderont un temps de travail raisonnable évalué approximativement au temps que vous aurez demandé la révision et le passage de l'examen classique. Merci de bien vous répartir le travail (voir la déclaration d'intégrité).

Merci également de bien communiquer entre vous tout au long du projet de façon à ce que chacun ait bien compris le travail des autres.

Si vous rencontrez des difficultés, n'hésitez pas à me contacter, nous pourrions organiser une séance sur Discord pour répondre à vos questions.

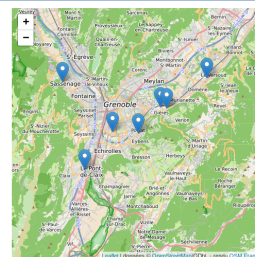
DÉTECTION D'UN CLUSTER DE 7 FOYERS SUR 50 ADRESSES ENREGISTRÉES

Adresses

ID	Vue	Commune	Longitude	Latitude
38189_1800_00019	19, Rue Christophe Turt	38180 Grenoble	5.725045	45.160725
38209_0090_00009	9, Place Georges Brassens	38120 Ponsat	5.758808	45.159129
38474_0050_00041	41, Rue François Dato	38360 Sassenage	5.658854	45.207821
38179_0566_00001	1, Allée du Vieux Ruisseau	38810 Gières	5.789296	45.184889
38111_0760_00006_Im	6, Rue du Côleau	38640 Clen	5.686079	45.125351
38179_3051_00002	2, Allée des Charmilles	38610 Gières	5.797048	45.182861
38038_0105_00079	79, Rue de Châteaufort	38420 Vercieu	5.850737	45.211789

Adresses des foyers du cluster

Carte Interactive



Introduction

Le contexte est celui de la surveillance active de l'épidémie de covid-19 après le déconfinement.

Il s'agit de surveiller la répartition géographique des patients diagnostiqués positifs au SARS-CoV2 et de pouvoir lister ou visualiser leurs localisations afin de détecter au plus tôt la formation de clusters.



Chaque groupe doit utiliser un fichier de données différent.

- Cliquez [ici pour accéder aux fichiers de données](#) (merci de bien télécharger et utiliser le fichier comportant le nom de votre groupe)

Le principe général de ce mini-projet est d'utiliser le fichier pour créer des traitements à l'aide de technologies XML et un Web Service associé. Les différents traitements proposés permettent de lister les données et de les visualiser avec des interrogations simples.



Il existe de nombreuses possibilités et technologies pour créer une application ayant les mêmes fonctionnalités et objectifs techniques que ceux proposés dans ce sujet. Cependant comme l'objectif du mini-projet est avant tout revoir les notions vues ensemble, il s'agit d'utiliser ici seulement les technologies XML balayées pendant l'enseignement (de XML aux Web Services en passant par XPath, les transformations XSL et les XML Schema).

L'utilisation d'autres technologies est donc à proscrire dans le cadre de ce projet (mais n'hésitez pas à faire des propositions alternatives après le 31 mai ! Par exemple pour proposer une version ReST du Web Services permettant de produire l'image de la page 1 directement dans un navigateur web).

Les données d'entrée sont constituées d'un fichier XML unique comportant une liste d'adresses et la géolocalisation de ces adresses (les adresses ont été tirées au hasard parmi 411 444 adresses du département de l'Isère, fournies par la base de données des adresses françaises valides, cf [\[1\]](#)).

Les informations ne sont pas complexes et la lecture rapide du document XML devrait vous suffire pour en comprendre le sens. N'hésitez pas à me contacter si cela n'était pas le cas.


Objectifs techniques

Les objectifs techniques sont répartis en 5 groupes :

- Modélisation XML Schema
- Requêtes XPath
- Transformations XSLT
- Programme Java
- Web Service

Chaque groupe d'objectifs peut se faire indépendamment de autres groupes.

Chaque objectif technique ci-dessous :

- est identifié par un acronyme (par exemple **S1**)
- comporte un élément décrivant le nom du livrable technique associé. Par exemple  `web/xsd/depistage.xsd` indique quel le livrable correspondant à l'objectif doit se nommer `web/xsd/depistage.xsd`.

Les noms des livrables sont donnés relativement à la racine de votre projet Netbeans.

Les objectifs identifiés par un **+** sont des exercices *bonus*. Ils seront évalués en plus des autres objectifs et ne sont pas obligatoires. Vous êtes cependant encouragés à tenter leurs réalisations : ils vous permettront d'approfondir vos connaissances et rendront le projet plus intéressant du point de vue fonctionnel !



Les livrables techniques ne respectant pas le nom indiqué ne seront pas évalués.

Modélisation XML

Il y a trois objectifs dans cette partie modélisation XML.

S1  `web/xsd/depistage.xsd`


Créer un modèle XML Schema des données utilisées. Tous les types doivent être nommés et le XML Schema doit créer un nouveau namespace :

`http://polytech-grenoble.fr/TIS/4/XML/Projet`


S2  `web/xsd/depistage-id.xsd`

Idem que **S1**, mais le document XML Schema définit en plus :

- une clé pour l'élément `id` des foyers,
- l'unicité de cette clé (chaque valeur de `id` est unique dans tous les éléments `foyer`),
- une référence obligatoire à un `id` existant pour tous les éléments `localisation`.

S2  `web/xml/data-X-instance.xml` (où `X` est votre nom de groupe)

Créer un document XML valide par rapport au Schema XML.

 Aucune modification ne doit être apportée au corps du document XML lui-même. La seule modification autorisée ici est l'ajout d'attributs à la racine `depistages` du document.


Requêtes XPath

Il faut fournir des requêtes XPath afin d'atteindre les objectifs suivants.


Les fichiers à livrer ici sont au format texte simple avec une seule ligne

R1  `web/xpath/xpath-R1.txt`


Calculer le nombre d'éléments `foyer` dans la liste

R2  `web/xpath/xpath-R2.txt`


Sélectionner tous les noeuds `foyer`

R3  web/xpath/xpath-R3.txt

Sélectionner l'élément XML `localisation` correspondant à un `id` donné


R4  web/xpath/xpath-R4.txt

Sélectionner le sous-élément `commune` (code postal et nom de la commune) d'un élément `foyer` correspondant à un `id` donné

R5  web/xpath/xpath-R5.txt

Sélectionner le sous-élément `longitude` d'un foyer correspondant à une voie et une commune donnée.

Exemple : sélectionner `longitude` pour le foyer dont `voie` vaut `18, Chemin de la Réglane` et `commune` vaut `38200 Vienne`

R6+  web/xpath/xpath-R6.txt

Créer une requête de test renvoyant `true` si et seulement si le nombre d'éléments `foyer` dans le document est bien identique au nombre d'éléments `localisation` dans le document

Transformations XSLT

Il faut fournir des transformations XSL permettant d'atteindre les objectifs suivants.

T1



web/xsl/liste.xsl

Transformation permettant de créer un document html listant l'ensemble des adresses présentes dans le document XML sous forme d'un tableau où chaque ligne comprend une adresse complète avec géolocalisation (id, voie, commune, longitude et latitude)

Exemple

LISTE DES ADRESSES				
Id	Voie	Commune	Longitude	Latitude
38191_0500_00645	645, Route du Signal	38750 Huez	6.068739	45.094463
38185_1600_00019	19, Rue Christophe Turc	38100 Grenoble	5.725545	45.160725
38393_0210_00011	11, Impasse du Guillot	38260 Saint-Hilaire-de-la-Côte	5.325216	45.391741
38063_0018_00135	135, Route de Bramafan	38690 Burcin	5.434416	45.427611
38563_0420_00008	8, Rue Dode	38500 Voiron	5.593272	45.365545
38246_0050_00025	25, Chemin du Cachard	38110 Montagnieu	5.433031	45.527496
38309_0090_00009	9, Place Georges Brassens	38320 Poisat	5.759808	45.159129
	115, Impasse			

(la présentation du tableau en couleurs n'est pas obligatoire)



Pour afficher la bonne localisation, vous devrez sans doute utiliser `xsl:variable` pour réutiliser la valeur de l'élément `id` dans votre requête XPath.

T2



web/xsl/liste-map.xsl

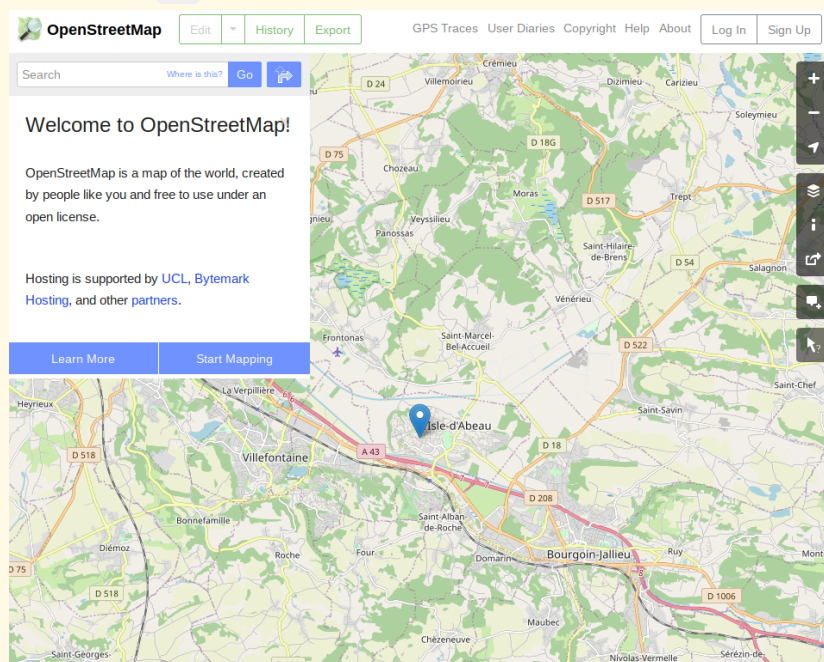
Idem **T1** mais où les adresses sont triées par code postal et où chaque **id** du tableau comporte un lien vers la carte Open Street Map comportant un marqueur à la localisation correspondante.

Exemple

LISTE DES ADRESSES

Id	Voie	Commune	Longitude	Latitude
38193_0050_00035	35, Rue des Cordiers	38080 L'Isle-d'Abeau	5.215365	45.619361
38352_0017_00018	18, Chemin du Bois	38080 Saint-Alban-de-Roche	5.223425	45.590523
38415_0800_00735	735, Rue de la Rivoire	38080 Saint-Marcel-Bel-Accueil	5.24431	45.650198
38339_0001_00017	17, Rue Gérard Vernay	38090 Roche	5.159418	45.588786
38185_1600_00019	19, Rue Christophe Turc	38100 Grenoble	5.725545	45.160725
38246_0050_00025	25, Chemin du Cachard	38110 Montagnieu	5.433031	45.527496

En cliquant sur le premier **id** du tableau, la carte Open Street Map correspondante s'affiche :



Pour ajouter un lien vers une carte Open Street Map simple (avec un seul marqueur), il suffit d'invoquer l'opération par défaut de l'API du web service d'Open Street Map avec les bons paramètres.

Exemple

Vous pouvez [cliquer sur ce lien](#) pour activer le service et visualiser un marqueur au point de la longitude **5.72550** et de latitude **45.18465** (où on ne pensait pas forcément avoir tellement envie de s'y retrouver il y a quelques semaines en arrière 😊).

Ce lien peut être généré en html en écrivant :

```
<a href="https://www.openstreetmap.org/?mlon=5.75250&mlat=45.18465">
```

où `mlon` est la valeur de la longitude du marqueur et `mlat` la valeur de sa latitude.



En XSLT il faudra remplacer le caractère `&` par son caractère de substitution.

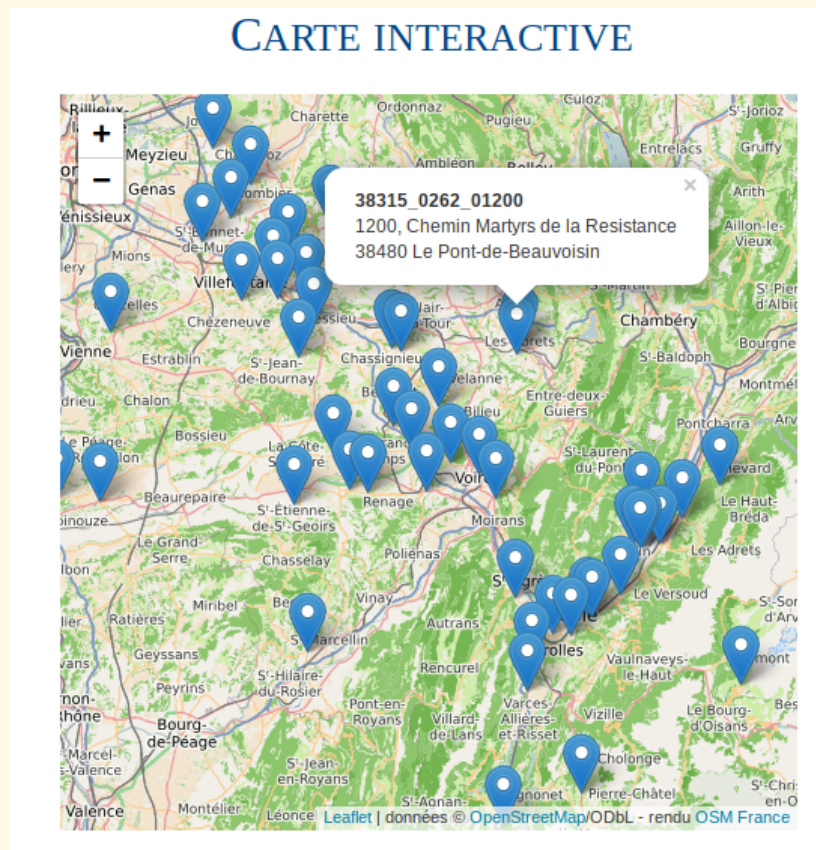
T3+



web/xs1/carte.xsl

Créer un document html affichant l'ensemble des adresses présentes dans le document XML sous forme d'une carte interactive avec un marqueur par adresse

Exemple



Options supplémentaires:

- le clic sur un marqueur affiche une info bulle avec l'id et l'adresse complète,
- la carte est centrée à la position moyenne des localisations.



Voir [2] pour un tuto pas à pas pour insérer une carte avec une liste de marqueurs dans une page html.




T3+ utilisant du `javascript`, il se peut qu'il faille passer par un serveur web afin de pouvoir tester vos pages html (dans certains navigateurs, le javascript ne s'exécute que si la page est lue par le protocole `http`).

Pour cela, sous Netbeans, il suffit de mettre les pages html dans le répertoire `web` de votre projet Netbeans de type "Web Application" et de cliquer sur "Run" : le serveur Glassfish sera lancé et vous pourrez accéder aux pages du site web local afin de charger les pages par le protocole `http`.

Programme Java

Le programme Java permet de mettre au point les algorithmes et la programmation du service web.

P1  src/java/cluster/CalculerCluster.java

Il s'agit d'un programme principal (`main`) qui calcule le nombre de foyers autour d'un foyer donné et affiche les résultats sur la sortie standard.

Le programme principal doit lancer des traitements permettant de parcourir l'ensemble des données et calculer le nombre et la liste (id et adresse) des foyers dont la localisation se situe dans un rayon de 10 km autour d'une adresse donnée désignée par son id.

Exemple de résultat d'exécution

```
Vérification des risques de clusters
```

```
Entrez l'id du foyer à vérifier :
```

```
38185_1600_00019
```

```
L'adresse de "38185_1600_00019" est :
```

```
19, Rue Christophe Turc
```

```
38100 Grenoble
```

```
Sa localisation est :
```

```
(longitude : 5.725545, latitude : 45.160725)
```

```
Calcul des foyers dans un rayon de 10Km...
```

```
Ce foyer est au centre d'un cluster de 7 foyers au total.
```

```
Liste des 6 foyers du cluster autour de "38185_1600_00019"
```

```
---
```

```
38309_0090_00009
```

```
9, Place Georges Brassens
```

```
38320 Poizat
```

```
---
```

```
38524_0003_00035
```

```
35, Chemin de Bardonnanche
```

```
38760 Varcès-Allières-et-Risset
```

```
---
```

```
38474_0050_00041
```

```
41, Rue Francois Gerin
```

```
38360 Sassenage
```

```
---
```

```
38179_0566_00001
```

```
1, Allée du Vieux Ruisseau
```

```
38610 Gières
```

```
---
```

```
38111_0760_00006_bis
```

```
6, Rue du Côteau
```

```
38640 Claix
```

```
---
```

```
38179_0051_00002
```

```
2, Allée des Charmilles
```

```
38610 Gières
```



Consultez [\[3\]](#) et [\[4\]](#) pour le calcul de la distance à vol d'oiseau entre deux points du globe terrestre.

Web Service


Le dernier objectif technique de ce mini-projet permet de créer un Web Service SOAP comportant plusieurs opérations. Le nom du package de votre Web Service doit être :

```
fr.polytechgrenoble.depistage
```


La classe du Web Service doit s'appeler `DepistageClusters` et le livrable doit être :

```
src/java/fr/polytechgrenoble/depistage/DepistageClusters.java
```


Les opérations suivantes sont demandées :

01  `+nbAdresses():int`


L'opération `nbAdresses()` qui renvoie le nombre d'adresses du document ressource (fichier XML).

02  `+localisationId(id:String):String`

L'opération `localisationId(..)` affiche la longitude et la latitude du foyer dont l' `id` est passé en paramètres.

03  `+afficherId(id:String):String`

L'opération `afficherId(..)` affiche la voie, la commune et la localisation du foyer dont l' `id` est passé en paramètres.

04  `+afficherCluster(id:String):String`

L'opération `afficherCluster(..)` est identique à **P1** mais est accessible directement en tant que Web Service. Elle calcule le nombre de foyers autour d'un foyer dont l' `id` est passé en paramètre et renvoie une chaîne de caractère contenant la liste de ces foyers.



Un exemple complet et fonctionnel de Web Service SOAP dont vous pouvez vous inspirer est présenté pas à pas dans le TP5, cf [\[5\]](#).



Le document XML utilisé comme ressource du Web Service doit obligatoirement être le document contraint et valide par rapport au XML Schema (et non le document brut fourni initialement).

Évaluation

Livrables

La date limite pour le rendu de l'ensemble des livrables est fixée au **31 mai**.

Voici la liste des livrables attendues :

Formulaire d'intégrité



Declaration integrite intellectuelle XML (pdf)

Merci de lire puis de remplir la déclaration d'intégrité intellectuelle et de la déposer sur Moodle.

- [Cliquez ici pour récupérer le formulaire](#)
- [Cliquez ici pour accéder à la page de dépôt de votre déclaration](#)

Projet Netbeans



Projet.zip

Le projet Netbeans contenant le code des différents objectifs techniques et permettant de lancer le programme Java et les opérations du Web Service.

Le projet doit être déposé sous forme d'une archive zip (merci de nettoyer au préalable le projet avec la target "clean").

- [Cliquez ici pour accéder à la page de dépôt de votre archive](#)

Présentation



lien vers le partage

Une présentation (de 10 à 15 pages) avec des commentaires (merci de privilégier, si possible, un enregistrement audio ou vidéo d'une durée maximum de 15 minutes). N'hésitez pas à ajouter une vidéo courte de démonstration du résultat obtenu avec le Web Service.

Pour éviter les problèmes de taille et de format de document de présentation, ce livrable sera déposé sous forme d'un lien de partage où je pourrais récupérer directement votre ou vos fichiers de présentation.

La présentation devra contenir :

- des éléments de gestion de projet (2 à 3 pages au plus étant donné la taille de ce mini-projet) décrivant *impérativement* votre mode d'organisation, le nombre d'heures prévues/consommées par objectif ou groupes d'objectifs, la répartition des rôles et un bilan rapide des difficultés rencontrées et des solutions trouvées
- une description et explication technique des différents résultats obtenus (étapes, technologies, algorithmes,...)

- [Cliquez ici pour indiquer votre lien de partage](#)

Critères de notation

Vous serez évalués sur :

- le nombre de fonctionnalités et leurs degrés d'achèvements (y compris les objectifs bonus),
- la qualité du code et des commentaires,
- l'utilisation adéquate des technologies vues en cours,
- la présence de l'ensemble des éléments demandés et la clarté de la présentation.

References

- [1] Liste des adresses valides officielles de France. <https://adresse.data.gouv.fr/donnees-nationales> ↩
- [2] Tutoriel en ligne pour inserer une carte Open Street Map dans une page html et visualiser des marqueurs de position. <https://nouvelle-techno.fr/actualites/pas-a-pas-inserer-une-carte-openstreetmap-sur-votre-site> ↩
- [3] Code Java (mais également dans d'autres langages) permettant de calculer la distance à vol d'oiseau entre deux points du globe terrestre. <https://www.geeksforgeeks.org/program-distance-two-points-earth/> ↩
- [4] Explication sur la formule de Haversine utilisée dans le calcul de la distance à vol d'oiseau entre deux points du globe terrestre. https://fr.wikipedia.org/wiki/Formule_de_haversine ↩
- [5] TP5 du module d'enseignement (l'exercice 5 décrit la création d'un Web Service SOAP pas à pas). <http://xmltis4.free.fr/tutorials/xml/05/index.html> ↩