 S. Hérauvillle	Langage Web 2	M1 GIL
	TP n°1 : Création d'un schéma XML avec XSD	Janvier 2017
		page 1 / 4

I - Objectifs

- Création d'un schéma XML pour l'écriture d'une transaction SEPA, et génération du flux XML correspondant.
- Ce flux sera ensuite exploité par une application java
- Projet géré dans un IDE avec une chaîne de compilation maven.

II - Démarrage

- Démarrez Eclipse et vérifiez les software présents :
 - Vérifiez la version du jdk java → version 8
 - Vérifiez que les outils XML / XSL / XSD sont disponibles dans Eclipse
Help → Installation details (Les installer si nécessaire)
 - Utilisez la perspective XML
- Créez un nouveau projet :
 - File → new → project → Maven → Maven project
 - Sélectionnez un archetype maven : maven-archetype-archetype
 - Saisir les paramètres : groupID : fr.univ
 artifactID : sepa
- Création du fichier XSD :
 - Dans l'explorateur de projet
src → main → resources.
 - Click droit sur le dossier resources → new → file : tp1.sepa.01.xsd

III - Création du schéma : DrctDbtTxInf

Question III.1 - Création du fichier XSD

Jusqu'à présent, Eclipse ne propose aucune aide à la rédaction (pas de complétion automatique). Ajoutez les lignes suivantes dans votre fichier XSD pour commencer :

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

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://univ.fr/sepa" xmlns="http://univ.fr/sepa">

</xs:schema>
```

Enregistrez les modifications. Vous constaterez qu'à partir de maintenant, Eclipse fait des propositions de complétion.

Décrivez les éléments de la ligne <xs:schema... .

Question III.2 - Schéma XSD

Rédigez un schéma XSD décrivant un flux XML représentant une transaction SEPA entre établissements bancaires. La présentation simplifiée de ce type de document (norme ISO 20022) est fournie en annexe sur UniversiTice, et servira de document de référence dans la suite de votre travail. Commencez par prendre connaissance de ce document.

Dans un premier temps, vous utiliserez seulement le **niveau transaction**, dont un résumé incomplet est résumé ci-dessous, pour concevoir votre schéma XSD :

Niv	Structure des champs	Commentaire
1	<u>Direct Debit Transaction Information</u>	Détail d'une transaction
1.1	Payment Information	Commentaire identifiant la transaction
1.2	Instructed Amount	Montant de la transaction
1.3	Direct Debit Transaction	Identification de la transaction
1.3.1	Mandate Identifier	Identifiant du mandat
1.3.2	Date of signature	Date d'émission
1.4	Debtor Agent	Banque émetrice du mandat
1.4.1	BIC	Identifiant bancaire au format BIC
1.4.2	OR Identifiant	Identifiant bancaire non normalisé
1.5	Debtor	Débiteur
1.5.1	Name	Nom du débiteur
1.6	Debtor Account	Compte du débiteur
1.6.1	IBAN	Numéro IBAN
1.6.2	OR Private Identification	Identification non normalisée
1.7	Remittance Information	Commentaire

Remarques :


- L'ordre des éléments de niveau 1.x n'est pas imposé
- Si un élément optionnel est défini, alors tous les éléments qui le compose doivent être définis.
- Utiliser l'espace de nom défini en annexe III.1

Question III.3 - Génération XML

Validez votre XSD puis générez le fichier XML :

- * Pour valider : click droit sur le fichier XSD, puis Validate
- * Pour générer le xml : click droit sur le fichier XSD, puis Generate

Générez le fichier `tp1.out1.xml` , et vérifiez que son contenu correspond à la structure attendue.

 S. Hérauvill	Langage Web 2	M1 GIL
	TP n°1 : Création d'un schéma XML avec XSD	Janvier 2017
		page 3 / 4

Question III.4 - Définition des éléments

Si ce n'est déjà fait, définissez les éléments décrits au chapitre III.6 de l'annexe à l'extérieur de l'élément racine et incluez les à partir d'une référence.

Générez un fichier `tp1.out2.xml` et vérifiez le résultat obtenu.

Question III.5 - Vérification des formats

Max35Text	Le nombre de caractères doit être compris entre 1 et 35 (limites comprises) Les caractères autorisés doivent respecter les indications fournies en annexe II.4
Montant	Respecter le format décrit en annexe II.5
IBAN	Respectez le format décrit en annexe II.6

Question III.6 - Vérification des occurrences

Vérifiez que les occurrences spécifiées en annexe III.5 sont bien respectées

Générez le fichier `tp1.out4.xml` et validez son contenu.

IV - Validation d'un flux XML d'après un schéma XSD

Dans cette seconde partie, nous allons écrire un programme en Java qui consomme un flux XML en entier mais ne fait rien. Les objectifs sont les suivants :

- Tester la validation d'un flux XML,
- Un premier contact avec l'API SAX.

Ainsi, le programme sera simplement constitué d'un test JUnit. Vous pourrez vous inspirer du code disponible à cette adresse <http://www.edankert.com/validate.html> pour compléter les classes suivantes.


Question IV.1 - Interface de gestion des erreurs.

Les API java pour SAX et DOM utilisent toutes deux la même interface pour leur gestionnaire d'erreurs. Ajoutez dans la section `src/main/java` une classe `SimpleErrorHandler` implémentant l'interface `org.xml.sax.ErrorHandler`. Vous ajouterez un attribut `bool errorOccured=false` que vous positionnerez à vrai dans chacune des méthodes d'interception d'erreur et vous ajouterez un accesseur `bool hasError()` pour le lire.

Question IV.2 - Classe de validation

Ajoutez dans la section `src/test/java` une classe `validateXML` contenant la méthode suivante : `"void should_validate_with_DOM()"` annotée avec `Test` qui valide un document XML en utilisant l'API DOM.

Placez une assertion sur la valeur de `hasError` de votre gestionnaire d'erreur à la fin de la méthode `void should_validate_with_DOM()`.

 <i>S. Hérauville</i>	Langage Web 2	M1 GIL
	TP n°1 : Création d'un schéma XML avec XSD	Janvier 2017
		page 4 / 4

Question IV.3 - Testez votre programme

- Celui-ci doit charger votre fichier XML et terminer sans erreur (par exemple `tp1.out4.xml`),
- Copiez le fichier `tp1.out4.xml` en `tp1.bad.xml`.

Introduisez dans ce fichier, des modifications qui ne soient pas conformes au schéma XSD.

Vérifiez qu'une erreur est maintenant déclenchée dans ce cas.

Question IV.4 - Modification de la classe `validateXML`

De manière similaire, ajoutez dans la classe `validateXML` une méthode

`"void should_validate_with_SAX()"` annotée avec `Test` validant un fichier XML en utilisant cette fois l'API `SAX`.

Question IV.5 - Test

Testez cette nouvelle méthode avec le même protocole que la précédente (§2.3).