

RAPPORT DE PROJET
SUR
“RÉFÉRENTIEL ÉTAMINE”

Remis à
UNIVERSITÉ DE VILLETANEUSE

Dans le cadre du module de gestion de projet pour l’obtention du

Master 1
INFORMATIQUE

PAR

Quentin AMELOT
Paul BAUDOUIN
Fayize KAIMOU
Damien LARMINÉ
Jérémie NIZOU
Gisio TABERA

UNIVERSITÉ PARIS 13
99 Avenue Jean Baptiste Clément, 93430 Villetaneuse
2014-2015

**RAPPORT DE PROJET
SUR
“RÉFÉRENTIEL ÉTAMINE”**

**Remis à
UNIVERSITÉ DE VILLETANEUSE**

Dans le cadre du module de gestion de projet pour l’obtention du

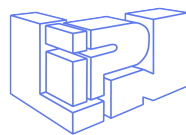
**Master 1
INFORMATIQUE**

PAR

**Quentin AMELOT
Damien LARMINÉ
Fayize KAIMOU
Gisio TABERA
Jéréemie NIZOU**

**Chef de projet
Responsable Développement
Développement JAVA
Développement PHP
Responsable Base de données**

**SOUS LE TUTORAT DE
M. Michael FORTIER**



**99 Avenue Jean Baptiste Clément, 93430 Villetaneuse
2014-2015**

Remerciements

Nous tenons à remercier **M. Michael FORTIER** pour son encadrement et son expertise tout au long du semestre, nous permettant de mener à bien ce projet.

Nous tenons également à remercier **Mr. Lionel POURNIN** ainsi que **Mr. Nadi TOMEH**, qui, malgré leur travail d'évaluation, ont pris le temps de nous conseiller. Leur implication fut une des clés du bon déroulement de ce projet.

Quentin AMELOT
Damien LARMINÉ
Gisio TABERA
Fayize KAIMOU
Jérémie NIZOU

ABSTRACT

Le but de ce projet était de mettre en place un référentiel de pays, c'est à dire rendre disponible les différentes informations concernant les pays du monde. Ce référentiel devait être implémenté sous la forme d'un Webservice JAVA. Afin de rendre l'information encore plus accessible, un client Php ainsi qu'un client JAVA ont été implémentés.

Le Webservice en lui même se présente sous la forme d'un JAR à exécuter. Un tomcat intégré permet le déploiement du service. Il est paramétrable grâce à plusieurs fichiers de configuration.

Le Client JAVA se présente sous la forme d'un code source à intégrer comportant plusieurs méthodes de test. Il dispose de commentaires et d'une Javadoc complète afin de faciliter l'intégration.

Le Client Php se présente sous la forme d'un site web comportant différentes pages et dont la principale fonction, sous l'onglet "Utiliser Etamine", permet d'interagir avec le Webservice.

Enfin, une application de gestion permet l'interaction avec la base de données du Webservice. Elle dispose d'une fonction de login avec mot de passe encrypté. Une fois authentifié, l'utilisateur peut ajouter, supprimer ou modifier des pays de la base de données suivant son niveau d'authentification.

Table des matières

1	Introduction	8
2	Spécification des Systèmes Dynamiques	9
2.1	Technologies utilisées	9
2.1.1	Maven	9
2.1.2	Spring	9
2.1.3	SpringWS	10
2.1.4	Soap et SoapUI	10
2.1.5	JAXB	11
2.1.6	Tomcat	11
2.1.7	JDBC	12
2.1.8	JUnit	12
3	Analyse des Besoins	13
3.1	Base de données	13
3.2	WebService	13
4	Plan du Système	14
4.1	Interactions	14
5	Tests et Assurance Qualité	16
5.1	Méthodes de test et résultats	16
5.1.1	Tests du client Java	16
5.1.2	Tests du WebService	18
5.1.3	Tests de l'Application de Gestion	20
6	Planification du Projet	23
6.1	Diagramme de Gantt	23

7	Implémentation	24
7.1	Web Service	24
7.2	Client Java	24
7.3	Client PHP	25
7.4	Application de Gestion	25
8	Conclusion et Futures Améliorations	26
8.1	Conclusion	26
8.2	Futures améliorations	26
	Références	26
	Annexe A	28
A.1		
	Cahier des Charges	28
A.2		
	Documentation Utilisateur	44

Table des figures

4.1	Schéma WebService	15
-----	-----------------------------	----

Chapitre 1

Introduction

Actuellement, il est impossible de trouver rapidement et facilement des informations utiles sur des pays.

Il est obligatoire de faire de longues recherches et les informations récupérées doivent être traitées avant d'être utilisables.

L'objectif de ce projet est de développer une solution durable permettant d'en finir avec ces recherches épuisantes et frustrantes en fournissant aux utilisateurs des informations exhaustives, faciles à trouver et surtout directement utilisables.

Le projet WebService Pays pour Étamine a pour but le développement d'une solution web permettant la mutualisation et l'accès aux données sur les pays du monde entier.

La principale application développée durant ce projet est un WebService permettant de communiquer avec un référentiel des pays contenant des informations utiles comme le code ISO, le libellé ou le taux de change.

Les utilisateurs pourront accéder à ces informations par le biais d'un client communiquant avec le WebService et utiliser directement ces informations dans d'autres applications.

Une application de gestion est aussi développée pour permettre aux propriétaires de la base de données de pouvoir éditer facilement la base de données et ainsi de l'actualiser. Des scripts pourront ainsi facilement être créés pour automatiser l'actualisation des données.

Chapitre 2

Spécification des Systèmes Dynamiques

2.1 Technologies utilisées

2.1.1 Maven

Maven est un outil de gestion et d'automatisation de projets développé par la Fondation Apache Software.

C'est un outil développé pour Java, il permet de créer une application à partir de ses sources et d'informations telles que ses dépendances et modules externes.

Maven optimise la création et assure le bon déroulement de la fabrication de l'application.

Il utilise le paradigme du Project Object Model qui décrit le projet dans un fichier contenant les dépendances, les modules externes, l'ordre de production, le numéro de version, etc...

Nous avons utilisé cet outil dans le cadre de ce projet car il nous a permis d'optimiser la production des applications développées.

Nous avons généré des sources et centralisé les dépendances et librairies nécessaires grâce à Maven ce qui a permis d'avoir une vision claire du projet et de l'état d'avancement.

2.1.2 Spring

Spring est un framework libre et open-source de développement Java sous licence Apache.

Spring est considéré comme un conteneur léger qui crée et met en relation des objets.

Il se base sur trois concepts clés qui sont l'inversion de contrôle, qui donne le contrôle de l'exécution au framework plutôt qu'à l'application elle-même, la programmation orientée aspect, qui sépare le cœur de l'application des aspects techniques, et une couche d'abstraction.

Nous avons utilisé Spring car c'est un framework très puissant et facilement accessible.

Il nous a permis de facilement définir l'infrastructure de nos applications et de gérer tous les objets présents à la création et à l'exécution de ces applications.

2.1.3 SpringWS

Spring Web Services est un produit issu du framework Spring permettant la création de Web services.

Ce framework facilite la création et l'utilisation de Web services créés en utilisant le protocole SOAP (voir sous-section SOAP et SoapUI).

Comme il est basé sur Spring, Spring-WS crée des web services très flexibles et permet même la mise en place de concepts inspirés de Spring comme l'injection de dépendances.

Nous avons utilisé Spring-WS pour sa capacité à créer des web services à la fois flexibles et complets mais aussi pour sa facilité d'intégration aux composants Spring déjà présents dans les différentes applications.

Nous avons ainsi pu réutiliser les différentes configurations et expériences liées à Spring que nous avons déjà pour accélérer la production et la mise en place du web service pays.

2.1.4 Soap et SoapUI

SOAP était l'acronyme de Simple Object Access Protocol, un protocole de Remote Procedure Call (RPC) qui permet d'appeler des méthodes à distances.

SOAP permet la transmission de messages entre objets distants à l'aide du protocole HTTP ou SMTP. Il utilise des métadonnées et est composé de deux parties, une enveloppe contenant des informations sur le message et un modèle de données qui

contient les données à transmettre.

Nous avons utilisé SOAP car c'est un protocole qui est indépendant de la plateforme et du langage utilisé, ce qui a facilité le développement du web service.

SoapUI est une application open source qui permet le test de web services utilisant le protocole SOAP.

Il permet d'inspecter un web service, de simuler une exécution et de réaliser des tests fonctionnels.

Nous avons utilisé SoapUI afin de tester notre web service et de localiser les éventuels problèmes de conception ou d'exécution.

2.1.5 JAXB

JAXB est l'acronyme de Java Architecture for XML Binding, une interface de programmation Java qui permet la création de classes Java à partir d'un fichier XSD et inversement.

Grâce à un mapping entre les types XML et Java, il assure la création des classes, de leurs constructeurs ainsi que leurs mutateurs et accesseurs.

Nous avons utilisé JAXB car il facilite l'utilisation de XML avec Java grâce à son système de schémas.

Le framework Spring se configurant à l'aide de fichiers XML, l'utilisation de JAXB a été encore plus importante afin d'être le plus efficient possible.

2.1.6 Tomcat

Tomcat est un conteneur web (serveur) libre appartenant à la Fondation Apache. En plus d'être un serveur, Tomcat gère les servlets ainsi que les Java Server Pages (JSP).

Il a été développé en Java et est donc indépendant de la plate-forme pour son exécution.

Nous avons utilisé Tomcat car c'est aujourd'hui le serveur HTTP multiplate-forme le plus intéressant.

De plus Spring intègre directement Tomcat, ce qui a facilité la création et le développement des applications.

2.1.7 JDBC

JDBC est l'acronyme de Java DataBase Connectivity, une interface de programmation Java permettant la communication entre une application Java et une base de données.

Cette API fournit des méthodes pour rechercher et modifier une base de données.

Nous avons utilisé JDBC pour la communication entre notre référentiel de pays et notre web service.

Sa facilité d'utilisation nous a permis de rapidement mettre en place une liaison sûre entre notre base de données et notre web service.

2.1.8 JUnit

JUnit est un framework de test unitaire pour Java.

JUnit utilise un système de TestCase qui contiennent des méthodes de test pour tester une classe et des TestSuite qui permettent d'exécuter certains TestCase prédéfinis. Il permet ainsi de tester certaines parties précises des application pour détecter d'éventuelles erreurs conceptuelles ou de programmation.

Nous avons utilisé JUnit pour tester les différentes classes créées dans l'ensemble des applications développées.

Cela nous a permis d'éviter de trouver des erreurs dans le code vers la fin du développement.

Chapitre 3

Analyse des Besoins

3.1 Base de données

La base de données est composée de nombreuses informations sur les pays du monde entier.

Réunir toutes ces informations est un travail extrêmement compliqué et qui nécessite de chercher sur de nombreux sites différents car les informations ne sont pas centralisées.

Notre client, Mr Fortier, possède cependant des scripts permettant de récupérer certaines de ces informations mais pour que cela soit efficace, il était nécessaire de faire un travail d'édition sur la base de données déjà existante.

Nous avons dû pour cela refaire entièrement la base de données pour permettre l'automatisation de la récupération des données grâce aux scripts.

3.2 Webservice

Le web service est la pierre angulaire de ce projet.

C'est l'application centrale qui va être la plus utilisée.

Il était donc nécessaire de la programmer de la manière la plus propre et la plus efficace possible afin d'éviter des lenteurs à l'exécution ou des erreurs de traitement.

Chapitre 4

Plan du Système

4.1 Interactions

Le projet est composé d'un web service, d'un client PHP, d'un client Java, d'une base de données et d'une application de gestion.

Les clients Java et PHP envoient des requêtes SOAP afin de demander des informations sur des pays.

Le web service reçoit les requêtes et les traite en envoyant une requête sur la base de données à l'aide de JDBC.

La base de données envoie une réponse et le web service utilise SOAP pour envoyer cette réponse aux clients.

L'application de gestion permet à un administrateur de se connecter directement à la base de données pour ajouter, supprimer ou modifier un pays manuellement dans la base de données.

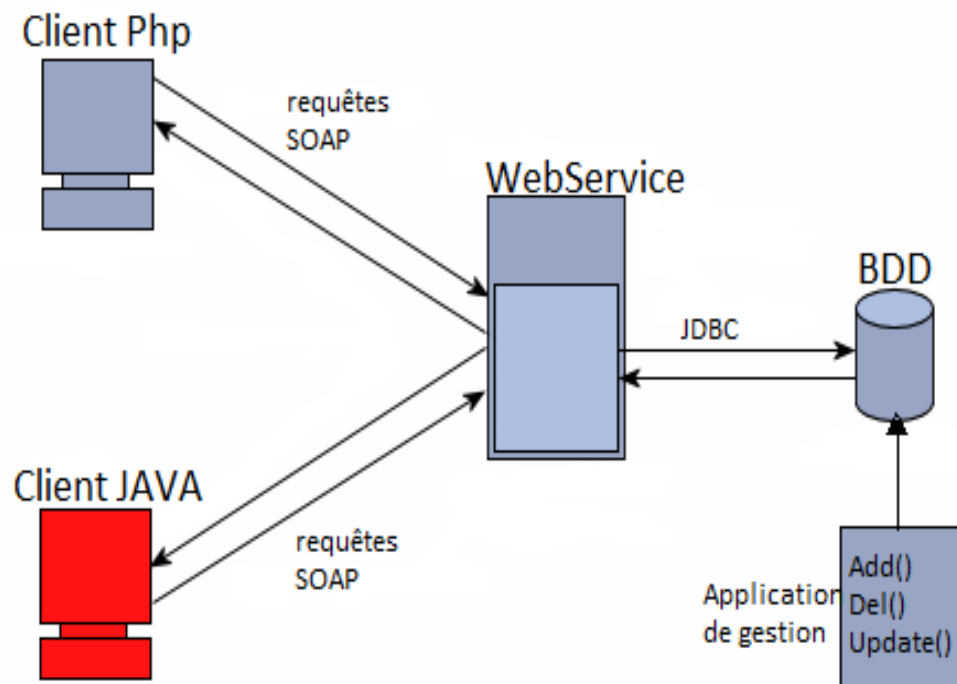


FIGURE 4.1 – Schéma WebService

Chapitre 5

Tests et Assurance Qualité

Le projet étant destiné à un public ouvert il était essentiel que les fonctions soient testées, au moins sur les clients et le WebService. Les tests vérifient les principales fonctionnalités des applications.

5.1 Méthodes de test et résultats

5.1.1 Tests du client Java

Les méthodes de test du client JAVA servent à tester les différentes réponses du WebService aux différentes requêtes.

```
1  /**
2   * Test of getAllPaysNameRequest method, of class PaysClient.
3   */
4   @Test
5   public void testGetAllPaysNameRequest() {
6       System.out.println("getAllPaysNameRequest");
7       ApplicationContext ctx = SpringApplication.run(PaysConfiguration.
8           class);
9
10      PaysClient paysClient = ctx.getBean(PaysClient.class);
11
12      GetAllPaysNameResponse response = paysClient.getAllPaysNameRequest()
13          ;
14      paysClient.printResponse(response);
15  }
16
17  /**
18   * Test of getAllPaysRequest method, of class PaysClient.
19   */
20   @Test
21   public void testGetAllPaysRequest() {
22       System.out.println("getAllPaysRequest");
23       ApplicationContext ctx = SpringApplication.run(PaysConfiguration.
24           class);
25
26       PaysClient paysClient = ctx.getBean(PaysClient.class);
```



```

24
25         GetAllPaysResponse response = paysClient.getAllPaysRequest();
26         paysClient.printResponse(response);
27     }
28
29     /**
30      * Test of getAllPaysDangerRequest method, of class PaysClient.
31      */
32     @Test
33     public void testGetAllPaysDangerRequest() {
34         System.out.println("getAllPaysRequest");
35         ApplicationContext ctx = SpringApplication.run(PaysConfiguration.
36             class);
37
38         PaysClient paysClient = ctx.getBean(PaysClient.class);
39
40         GetAllPaysDangerResponse response = paysClient.
41             getAllPaysDangerRequest(true);
42         paysClient.printResponse(response);
43     }
44
45     /**
46      * Test of getPaysRequest method, of class PaysClient.
47      */
48     @Test
49     public void testGetPaysRequest() {
50         System.out.println("getPaysRequest");
51         ApplicationContext ctx = SpringApplication.run(PaysConfiguration.
52             class);
53
54         PaysClient paysClient = ctx.getBean(PaysClient.class);
55
56         GetPaysResponse response = paysClient.getPaysRequest("france");
57         paysClient.printResponse(response);
58         response = paysClient.getPaysRequest("albanie");
59         paysClient.printResponse(response);
60     }
61
62     /**
63      * Test of getListPaysRequest method, of class PaysClient.
64      */
65     @Test
66     public void testGetListPaysRequest() {
67         System.out.println("getListPaysRequest");
68         ApplicationContext ctx = SpringApplication.run(PaysConfiguration.
69             class);
70
71         PaysClient paysClient = ctx.getBean(PaysClient.class);
72
73         HashMap<String, String> arg = new HashMap();
74         arg.put("libelle", "a");
75         arg.put("danger", "false");
76         arg.put("monnaie_code", "e");
77         arg.put("taux_change", "0.5");

```

```

74         arg.put("taux_change_arg", "+");
75         arg.put("monnaie_perdiem", "100");
76         arg.put("monnaie_perdiem_arg", "+");
77
78         GetListPaysResponse response = paysClient.getListPaysRequest(arg);
79         paysClient.printResponse(response);
80     }

```

project/code/PaysClientTest.java

5.1.2 Tests du Webservice

Les méthodes de test du Webservice sont principalement liées à la DAO. Elles testent le lien entre les requetes et la Base de Données.

```

1  /**
2   * Test of findAllPays method, of class IPaysMetier.
3   */
4  @Test
5  public void testFindAllPays() throws Exception {
6      System.out.println("findAllPays");
7      ListableBeanFactory bf;
8      bf = new XmlBeanFactory(new ClassPathResource("applicationContext.xml"))
9          ;
10     IPaysMetier instance = (IPaysMetier)bf.getBean("paysMetier");
11
12     List<Pays> result = instance.findAllPays();
13     for (Pays pays : result) {
14
15         System.out.println("libelle : " + pays.getLibelleEn());
16     }
17
18     /**
19     * Test of findPays method, of class IPaysMetier.
20     */
21     @Test
22     public void testFindPays() throws Exception {
23         System.out.println("Test findPays");
24         ListableBeanFactory bf;
25         bf = new XmlBeanFactory(new ClassPathResource("applicationContext.xml"))
26             ;
27         IPaysMetier instance = (IPaysMetier)bf.getBean("paysMetier");
28
29         String libelle = "bolivie";
30         int id = 105;
31         try {
32             Pays result = instance.findPays(id);
33             System.out.println("libelle : " + result.getLibelleFr() + result.
34                 getID());
35         }
36         catch (PaysNotFoundException e){
37             e.printStackTrace();
38         }
39     }

```

```

36     }
37 }
38
39
40 /**
41  * Test of findAllPays method, of class IPaysMetier.
42  */
43 @Test
44 public void testFindAllPaysDanger() throws Exception {
45     System.out.println("findAllPaysDanger");
46     ListableBeanFactory bf;
47     bf = new XmlBeanFactory(new ClassPathResource("applicationContext.xml"));
48     ;
49     IPaysMetier instance = (IPaysMetier)bf.getBean("paysMetier");
50
51     List<Pays> result = instance.findAllPaysDanger(true);
52     for (Pays pays : result) {
53
54         System.out.println("libelle : " + pays.getLibelleFr());
55     }
56 }
57
58 /**
59  * Test of findAllPays method, of class IPaysMetier.
60  */
61 @Test
62 public void testFindAllPaysList() throws Exception {
63     System.out.println("findAllPaysList");
64     ListableBeanFactory bf;
65     bf = new XmlBeanFactory(new ClassPathResource("applicationContext.xml"));
66     ;
67     IPaysMetier instance = (IPaysMetier)bf.getBean("paysMetier");
68
69     Map<String, String> arg = new HashMap();
70     // arg.put(" libelle ", "i");
71     /// arg.put(" danger", "0");
72     // arg.put(" monnaie_perdiem", "180");
73     // arg.put(" monnaie_perdiem_arg", "-");
74     // arg.put(" indicatif", "i");
75     // arg.put(" taux_change", "1");
76     // arg.put(" taux_change_arg", "+");
77     arg.put("monnaie_code", "e");
78     List<Pays> result = instance.findListPays(arg);
79     for (Pays pays : result) {
80
81         System.out.println("libelle : " + pays.getLibelleFr());
82     }
83     System.out.println(result.size());
84 }

```

project/code/TestJDBCWS.java

5.1.3 Tests de l'Application de Gestion

Les méthodes de l'Application de Gestion servent à tester les interactions avec la base de donnée comme l'ajout, la suppression ou la modification de pays.

```

1  /**
2   * Test of findAllPays method, of class IPaysMetier.
3   */
4  @Test
5  public void testFindAllPays() throws Exception {
6      System.out.println("findAllPays");
7      ListableBeanFactory bf;
8      bf = new XmlBeanFactory(new ClassPathResource("applicationContext.xml"));
9      ;
10     IPaysMetier instance = (IPaysMetier)bf.getBean("paysMetier");
11
12     List<Pays> result = instance.findAllPays();
13     for (Pays pays : result) {
14
15         System.out.println("libelle : " + pays.getIndicatif());
16     }
17 }
18
19 /**
20 * Test of findPays method, of class IPaysMetier.
21 */
22 @Test
23 public void testFindPays() throws Exception {
24     System.out.println("Test findPays");
25     ListableBeanFactory bf;
26     bf = new XmlBeanFactory(new ClassPathResource("applicationContext.xml"));
27     ;
28     IPaysMetier instance = (IPaysMetier)bf.getBean("paysMetier");
29
30     String libelle = "france";
31     try{
32         Pays result = instance.findPays(libelle);
33         System.out.println("libelle : " +result.getLibelleFr());
34     }
35     catch (PaysNotFoundException e){
36         e.printStackTrace();
37     }
38 }
39
40 @Test
41 public void testFindUser() throws Exception {
42     System.out.println("Test findUser");
43     ListableBeanFactory bf;
44     bf = new XmlBeanFactory(new ClassPathResource("applicationContext.xml"));
45     ;
46     IPaysMetier instance = (IPaysMetier)bf.getBean("paysMetier");
47
48     String user = "test";

```

```

47     User result = instance.findUser(user);
48     System.out.println("Login : " + result.getUser());
49     System.out.println("Password : " + result.getPassword());
50 }
51 /**
52  * Test of findAllPays method, of class IPaysMetier.
53  */
54 @Test
55 public void testFindAllPaysDanger() throws Exception {
56     System.out.println("findAllPaysDanger");
57     ListableBeanFactory bf;
58     bf = new XmlBeanFactory(new ClassPathResource("applicationContext.xml"));
59     ;
60     IPaysMetier instance = (IPaysMetier)bf.getBean("paysMetier");
61
62     List<Pays> result = instance.findAllPaysDanger(true);
63     for (Pays pays : result) {
64
65         System.out.println("libelle : " + pays.getLibelleFr());
66     }
67 }
68 /**
69  * Test of findAllPays method, of class IPaysMetier.
70  */
71 @Test
72 public void testFindAllPaysList() throws Exception {
73     System.out.println("findAllPaysList");
74     ListableBeanFactory bf;
75     bf = new XmlBeanFactory(new ClassPathResource("applicationContext.xml"));
76     ;
77     IPaysMetier instance = (IPaysMetier)bf.getBean("paysMetier");
78
79     Map<String, String> arg = new HashMap();
80     arg.put("libelle", "faaaaa%");
81     arg.put("danger", "false");
82     List<Pays> result = instance.findListPays(arg);
83     for (Pays pays : result) {
84
85         System.out.println("libelle : " + pays.getLibelleFr());
86     }
87 }
88 @Test
89 public void testCreatePays() throws Exception {
90     System.out.println("createPays");
91     ListableBeanFactory bf;
92     bf = new XmlBeanFactory(new ClassPathResource("applicationContext.xml"));
93     ;
94     IPaysMetier instance = (IPaysMetier)bf.getBean("paysMetier");
95
96     Map<String, String> arg = new HashMap();
97     arg.put("ID", "243");
98     arg.put("libelle_fr", "atest");

```

```

98     arg.put("indicatif", "aat2");
99     arg.put("monnaie_perdiem", "14");
100    arg.put("taux_change", "1");
101    instance.createPays(arg);
102 }
103
104     @Test
105     public void testDeletePays() throws Exception {
106         System.out.println("deletePays");
107         ListableBeanFactory bf;
108         bf = new XmlBeanFactory(new ClassPathResource("applicationContext.xml"));
109         ;
110         IPaysMetier instance = (IPaysMetier)bf.getBean("paysMetier");
111
112         instance.deletePays("atest6");
113     }
114
115     @Test
116     public void testUpdatePays() throws Exception {
117         System.out.println("updatePays");
118         ListableBeanFactory bf;
119         bf = new XmlBeanFactory(new ClassPathResource("applicationContext.xml"));
120         ;
121         IPaysMetier instance = (IPaysMetier)bf.getBean("paysMetier");
122
123         Map<String, String> arg = new HashMap();
124         // arg.put("libelle", "atest6super");
125         arg.put("indicatif", "attse");
126         arg.put("monnaie_perdiem", "239");
127         arg.put("nationalite", "atestiens");
128         arg.put("monnaie_code", "ATTdT");
129         arg.put("libelle_en", "aitef");
130         arg.put("libelle_fr", "atest6");
131         arg.put("drapeau", "random_drapeau.png");
132         instance.updatePays("atest66", arg);

```

project/code/TestJDBC.java

Note : Les tests doivent s'effectuer manuellement et permettent à l'utilisateur de comprendre le fonctionnement du code

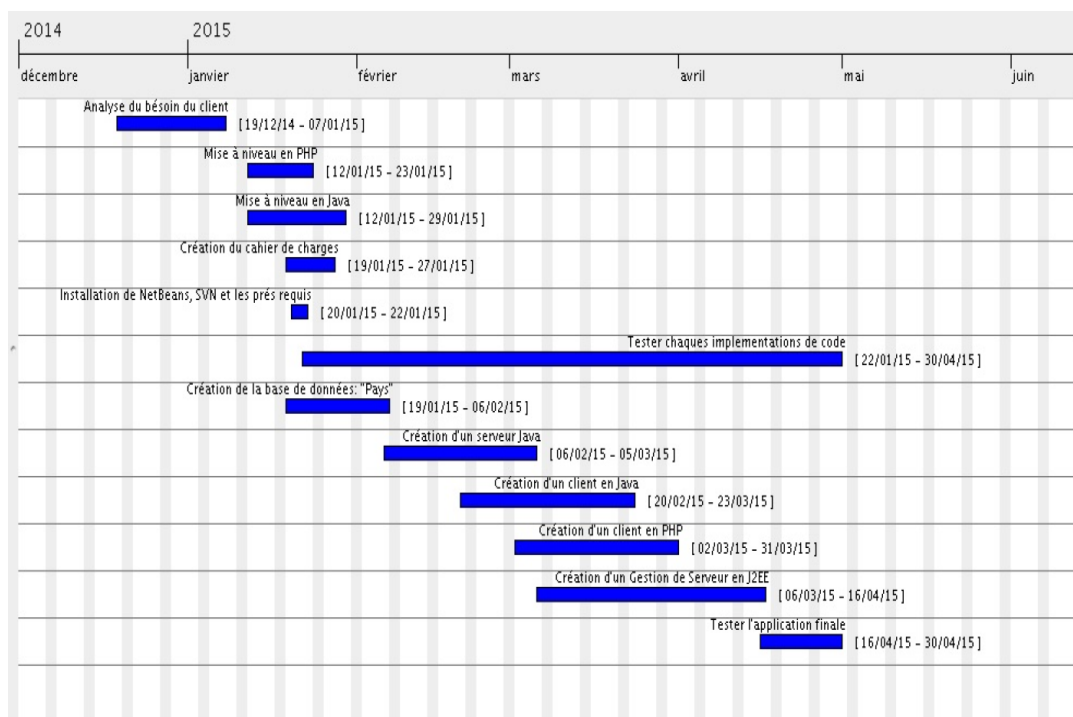
Chapitre 6

Planification du Projet

6.1 Diagramme de Gantt

Au début du projet, nous avons établi un diagramme de Gantt afin de bien organiser le développement des applications du projet.

Nous avons suivi précisément ce diagramme de Gantt et nous avons ainsi séparé le temps du projet de la manière la plus efficiente possible.



Chapitre 7

Implémentation

7.1 Web Service

Le Web Service se lance par le biais de la classe Application.

Une fois lancé, le serveur se configure grâce aux éléments de la classe WebService-ConfigPays.

Il attend ensuite une requête d'un client, et c'est la classe PaysEndPoint qui traitera cette requête.

La classe PaysEndPoint crée un objet PaysRepository et en fonction du type de requête reçu appelle une méthode de cet objet.

L'objet Pays Repository charge une fabrique de beans, des objets gérés par Spring et Spring-WS.

Cette fabrique est utilisée par la classe IPaysMetier qui crée un objet de type Pays-Dao et récupère un bean.

Elle applique ensuite une méthode de l'objet PaysDao au bean récupéré.

La méthode de l'objet PaysDao utilise JDBC pour récupérer les informations nécessaires dans la base de données.

Une fois les informations nécessaires récupérées, la classe PaysEndpoint renvoie une réponse au client.

7.2 Client Java

Le client se lance à partir de la classe Application.

Une fois lancé, il commence à configurer les paramètres nécessaires à la communication avec le web service situé dans la classe PaysConfiguration.

Cette classe charge le fichier XML de configuration géré par Spring et Spring-WS et crée un objet PaysClient puis et le renvoie à la classe Application.

L'objet PaysClient crée un objet Config qui configurera les paramètres de cet objet PaysClient.

La classe application utilise alors l'objet PaysClient récupéré précédemment pour envoyer une requête au web service.

7.3 Client PHP

Le client PHP se présente sous la forme d'un site internet. L'implémentation des fonctions, plus particulièrement celles relatives à SOAP, est détaillée dans la documentation Administrateur.

Il constitue l'équivalent du Client Java en php, avec toutefois une interface graphique soignée et un formulaire intuitif pour effectuer les requêtes.

7.4 Application de Gestion

Une fois l'application de gestion lancée, il est nécessaire de se connecter grâce à un nom d'utilisateur et un mot de passe.

La connexion se fait quand l'application crée un objet MainHandler qui va récupérer un objet paysMetier grâce au fichier de configuration XML géré par Spring et Spring-WS.

L'objet paysMetier va créer un objet paysDao et lui appliquer une méthode de recherche d'utilisateur.

L'objet paysDao va utiliser JDBC pour chercher dans une base de données spécifiques si le nom d'utilisateur et le mot de passe sont corrects, si c'est le cas, la connexion est validée. Une fois la connexion effectuée, l'utilisateur peut cliquer sur ajouter, supprimer ou modifier selon son niveau d'autorisation.

Quand un ajout, une modification ou une suppression est validée l'application récupère un objet paysMetier grâce au fichier de configuration XML géré par Spring et Spring-WS.

Cet objet va créer un objet paysDao et lui appliquer la méthode demandée.

L'objet paysDao va utiliser JDBC pour effectuer la méthode demandée et ajouter, modifier ou supprimer un pays dans la base de données.

Chapitre 8

Conclusion et Futures Améliorations

8.1 Conclusion

Ce projet intéressant et captivant nous aura permis de découvrir et d'apprendre les bases de nombreux frameworks demandés dans le monde du travail pour leur efficacité et leur souplesse. Le module de gestion de projet aura porté notre attention sur des éléments indispensables au bon développement d'un projet comme la documentation ou les contacts réguliers au client pour éviter l'effet tunnel.

Ce projet jouant un rôle important dans l'obtention de notre première année de Master, il était nécessaire de choisir le bon projet. A l'unanimité le groupe pense que ce fut le bon choix étant donné qu'il allie nouvelles technologies, programme de deuxième année et une problématique de nature professionnelle.

8.2 Futures améliorations

Une partie intégrante de chaque projet constitue sa sécurisation quant aux attaques pirates. Trop peu abordée dans ce projet, elle pourrait faire l'objet d'une amélioration future, notamment grâce au module SpringSecurity.

Plus de fonctions pourraient également être ajoutées, comme la possibilité de créer des utilisateurs pour l'application de gestion directement en son sein.

Une interface graphique plus épurée et moderne pour l'application de gestion sera également à prévoir.

Enfin, quelques données supplémentaires pourraient être rajoutées à la base de données.

References

- [1] <https://spring.io/docs>
- [2] <https://spring.io/guides>
- [3] <http://maven.apache.org>
- [4] <https://trac.lipn.univ-paris13.fr/projects/pays/>
- [5] <https://lipn.univ-paris13.fr/~fortier/Wiki/index.php?n=Accueil.Accueil>

Annexe A

A.1

Cahier des Charges

PROJET ETAMINE : CAHIER DES CHARGES

Quentin Amelot

Contents

Contents	2
1 Présentation du Projet	3
1.1 Contexte	3
1.2 Objet	3
1.3 Organisation	4
1.4 Environnement	4
1.5 Outils complémentaires	5
2 Objectifs	5
2.1 Points clés sur les besoins fonctionnels	5
2.2 Limites du projet	6
3 Spécifications	6
3.1 Base de données	6
3.2 Web Service	9
3.3 Clients Web Service	10
3.4 Application de gestion	11
4 Livrables	13
4.1 Documents à remettre	13
4.2 Programmes à remettre	14
5 Calendrier	14
5.1 Date de fin de projet	14
5.2 Échéances intermédiaires	15

1 Présentation du Projet

1.1 Contexte

Le projet de développement Etamine est une application multi-structures et multi-tutelles, développée au laboratoire, qui répond à divers besoins des enseignants chercheurs et personnels administratifs. Elle se compose de dizaines de modules répartis dans différents thèmes :

- administration : gestion des membres, annuaire, trombinoscope. . . ;
- outils d'aide à la recherche : gestion de dépôts SVN, GIT, publications, contrats. . . ;
- messagerie : messagerie web, gestion des filtres (antispam, antivirus). . . ;
-

Certains modules peuvent utiliser des informations partagées et/ou hébergées en dehors du laboratoire, accédé de façon anonyme ou authentifiée. Dans le cadre d'Etamine, l'utilisation des services web est commune.

1.2 Objet

Il s'agit de créer un Web Service qui interagira avec une base de données afin de renvoyer à des clients des informations sur des pays contenus dans la base de données. La base de données contiendra

diverses informations concernant les pays. Il faudra aussi créer un client JAVA et un client PHP afin de communiquer avec le Web Service. Ces clients ne requerront pas d'authentification pour utiliser le Web Service. Le client PHP pourra être intégré dans un navigateur et le client JAVA permettra de s'assurer une intégration dans des applications. Nous créerons aussi une application de gestion afin d'effectuer des modifications sur la base de données. Ce client possédera un système d'authentification afin que seuls les utilisateurs autorisés puissent modifier la base.

1.3 Organisation

Six personnes doivent réaliser ce projet :

- Le chef de projet : Quentin AMELOT
- Le responsable recherche et développement : Damien LARMINÉ
- Le responsable d'assurance qualité (QA) : Paul BAUDOUIN
- Le responsable de la documentation : Jérémie NIZOU
- Le développeur PHP : Gisio TABERA
- Le développeur JAVA : Fayize KAIMOU

Chaque personne possède un rôle bien précis dans la réalisation de ce projet, chaque personne est donc un maillon important de la chaîne.

1.4 Environnement

L'application sera réalisée en Java SE 1.7 à l'aide de l'IDE NetBeans ainsi que la plateforme SVN, car l'application devra s'installer sur n'importe quel type de machines. Nous aurons besoin notamment des librairies Hibernate (à confirmer) pour la synchronisation avec la base de données. Le choix du Framework pour l'interaction avec la base de données se fera plus tard dans le développement du projet. Nous utiliserons JUnit ainsi que Conordion pour les

différents tests unitaires et fonctionnels. En ce qui concerne la base de données et le développement PHP, nous utiliserons phpMyAdmin et son équivalent Mac. Nous utiliserons également l'architecture REST (Representational State Transfer), c'est un style d'architecture permettant de construire des applications pour Web-Service, Web, intranet. Ce n'est pas une technologie à part entière, mais un ensemble de conventions et de bonnes pratiques à respecter. Il utilise les spécifications originelles du protocole HTTP. Nous utiliserons REST pour ne pas réinventer une surcouche comme le font XML-RPC ou SOAP.

1.5 Outils complémentaires

Nous utiliserons l'application web TRAC. Son objectif est de simplifier le suivi et le traitement efficace des problèmes de logiciel, améliorations et les progrès d'ensemble. Cela nous permettra de faciliter la gestion du projet. Un wiki sera mis à disposition afin de répondre aux questions les plus évidentes. Cependant, pour des demandes plus poussées, il sera nécessaire de prendre contact avec l'équipe de développement.

2 Objectifs

2.1 Points clés sur les besoins fonctionnels

Plusieurs points clés sont nécessaires pour bien répondre aux attentes. Premièrement, il s'agit de rendre l'application exécutable sur un réseau local. Il faut qu'il soit facile d'installation et d'utilisation sur n'importe quel type de machines, c'est-à-dire sur un maximum de systèmes d'exploitation, type Linux, Windows ou Mac. Un guide d'utilisation ainsi qu'un guide d'installation seront fournis.

2.2 Limites du projet

Le projet n'a pas de limites : il est censé être diffusé au grand public. Un budget n'est pas nécessaire, car nous n'utiliserons que des logiciels libres et l'utilisation de licences n'est pas obligatoire pour réaliser correctement notre projet. Pour que le projet ait une plus grande envergure, la solution serait de le rendre disponible en ligne, donc accessible pour n'importe qui, peu importe l'endroit. Le manuel d'utilisation pourra être traduit ou directement rédigé en anglais pour une meilleure compréhension par le grand public.

3 Spécifications

3.1 Base de données

La base de données doit servir de référentiel pour les pays. Le référentiel des pays sera accessible à distance, pour chaque pays sera disponible :

- le libellé
- le code ISO
- la devise
- les coordonnées
- le risque
- la nationalité
- le nom du pays en français
- le nom du pays en anglais
- la valeur per diem

- le taux de change

Ce référentiel sera donc accessible via le biais d'un Web Service.

La structure de la table :

Nom	Type
ID	int(11)
indicatif	varchar(10)
nationalité	varcahr(100)
nom_EN	varchar(100)
nom_FR	varchar(100)
monnaie_code	varchar(5)
monnaie_perdiem	decimal(10,4)
taux_change	decimal(10,7)
coords1	text
coords2	text
coords3	text
coords4	text
coords5	text

3. SPÉCIFICATIONS

Le dictionnaire de données :

Champ	Descriptif	Remarque
ID	identifiant	Clé primaire de la base de données
indicatif	code ISO	Code sur 2 ou 3 lettres du pays
nationalité	nationalité du pays	Ex : France=Française, etc.
nom_EN	nom du pays en anglais	
nom_FR	nom du pays en français	
monnaie_code	code de la monnaie	Ex : Euro=EUR, Royaume-Uni=GBP
monnaie_perdiem	montant de la monnaie per diem	L'argent attribué par jour pour vivre dans ce pays
taux_change	taux de change	Par rapport à l'Euro
danger	symbole du danger du pays	0 si pas dangereux, 1 si dangereux Risque établi par le gouvernement
coords1		Coordonnées du pays, utilisé pour la carte interactive
coords2		
coords3		
coords4		
coords5		

Exemple des données contenues :

ID	indicatif	nationalité	nom_EN	nom_FR	monnaie_code	monnaie_perdiem	taux_change	danger	coords1	coords2	coords3	coords4	coords5
1	ad	andorranne	adorra	andorre	EUR	180.0000	1.0000000	0					
2	ae	emiratie	united arab emirates	emirats arabes unis	AED	754.0000	0.2240000	0	514,146, 515,146, 515,146, 515,147, 515,147, 515,1...				
3	af	afghane	afghanistan	afghanistan	AFN	180.0000	0.0139000	1	536,134, 538,132, 538,131, 536,130, 536,130, 535,1...				
4	ag	antiguayenne	antigua and barbuda	antigua-et-barbuda	USD	180.0000	0.8200000	0					
5	ai	anguillaise	anguilla	anguilla	USD	180.0000	0.8200000	0					
6	al	albanaise	albania	albanie	ALL	809.0000	0.0071900	0	446,113, 445,112, 444,111, 444,111, 444,111, 444,1...				
7	am	armenienne	armenia	armenie	EUR	180.0000	1.0000000	1	504,114, 504,114, 504,114, 503,113, 503,113, 503,1...				
8	an	insulaire des antilles neerlandaises	netherlands antilles	antilles neerlandaises	ANG	0.0000	0.0000000	0					
9	ao			angola	EUR	180.0000	1.0000000	1	428,214, 429,214, 430,214, 437,214, 438,214, 439,2...	427,212, 429,211, 429,211, 430,211, 429,212, 429,2...			
10	aq			antarctique		0.0000	0.0000000	0	778,386, 783,386, 783,386, 800,388, 800,400, 867,4...	238,362, 239,361, 240,361, 241,361, 241,360, 236,3...	278,380, 279,379, 281,378, 286,378, 287,377, 288,3...		
11	ar			argentine	USD	180.0000	0.8200000	0	248,319, 248,319, 248,319, 249,319, 253,322, 254,3...	249,317, 241,316, 240,316, 240,313, 240,313, 238,3...	248,319, 248,318, 249,318, 249,318, 248,319, 248,3...		
12	as			samoa	WST	180.0000	0.3490000	0					

3.2 Web Service

Le serveur permettra la récupération des données depuis la base de données. Elles seront transmises aux clients. Ce serveur est une application JAVA ne comportant pas d'interface graphique (GUI), mais regroupant différentes méthodes pour accéder aux données et les renvoyer. Parmi ces méthodes, nous trouverons :

- envoyerDonnées()
- recevoirRequete()
- pays()
- paysParNom()
- paysParRisque()
- paysPar[...]()
- ...

3. SPÉCIFICATIONS

Ceci est une liste non exhaustive permettant de saisir les principales interactions possibles avec ce serveur.

3.3 Clients Web Service

Les clients PHP et JAVA seront conçus avec les mêmes fonctionnalités de base. Ils exécuteront des requêtes prédéfinies choisies par l'utilisateur. Les utilisateurs des clients du Web Service n'ont pas besoin d'être authentifiés pour accéder aux données. Les captures d'écran suivantes donnent une idée des fonctionnalités des clients. L'utilisateur effectue une requête via un formulaire :

The screenshot shows a web application interface with a dark header containing three tabs: "Accueil", "Liste des pays", and "Carte". The main content area is divided into two sections. On the left, under the heading "Choisir Pays?", there is a dropdown menu currently showing "France". Below the dropdown is a scrollable list of countries: Italie, Lettonie, Liechtenstein, Lituanie, Luxembourg, Macédoine, Malte, Moldavie, Monaco, Monténégro, Norvège, Pays-Bas, Pologne, Portugal, and République Tchèque. On the right, under the heading "Informations à obtenir:", there are two columns of checkboxes. The first column includes "Indicatif", "Libelle", "Coordonné", and "Monnai". The second column includes "Monnaie Perdiem", "Taux de change", "Risque", and "Langue".

L'utilisateur consulte la liste des pays, disponible à tout moment :

The screenshot shows the "Liste des pays" section of the web application. The header tabs are "Accueil", "Liste des pays", and "Carte". The main content area features a scrollable list of countries under the heading "Liste des pays". The visible countries in the list are: Antigua et Barbuda, Bahamas, Bermudes, Canada, Iles Cayman, Cuba, République Dominicaine, Dominique, Etats-Unis, Haïti, Jamaïque, Mexique, and Argentine.

L'utilisateur consulte une carte interactive des pays :



En passant sur un pays avec la souris, le pays sera colorié et un type d'information choisi sera affiché, par exemple la devise ou le nom du pays.

3.4 Application de gestion

L'application de gestion offrira quant à elle la possibilité à l'administrateur de modifier sa base de données facilement, avec un certain nombre d'actions de maintenances prédéfinies :

- Ajouter un pays
- Modifier un pays
- Supprimer un pays
- Consulter la base de données

Ces actions ne seront disponibles qu'après s'être authentifié sur une page : Une fois l'utilisateur connecté, il pourra interagir avec la base de données via une interface avec des boutons pour choisir les actions (ajouter un pays, modifier un pays, etc.) puis via des formulaires pour préciser les données à ajouter ou à changer.

3. SPÉCIFICATIONS

[Connecter](#) [Inscription](#)



Étamine

Authentification

UNIVERSITÉ **PARIS** 13 UN

login et mot de passe

login:

Mot de passe:

[SE CONNECTER](#) [EFFACER](#)

ETAMINE

[Accueil](#)

[Modifier](#)

[Ajouter](#)

[Supprimer](#)

[Créer](#)

[Aide](#)



ETAMINE

Modification des données

Accueil

Modifier

Ajouter

Supprimer

Créer

Aide

Entrez les éléments à modifier

Pays:Antiqua et Barbuda

Nationalité:

Code ISO:

Devise:

Taux de change:

Nom du pays en Français:

Nom du pays en Anglais:

Code monnaie:

Coords1:

Coords2:

Coords3:

Coords4:

Coords5:

Enregistrer

Annuler

4 Livrables

4.1 Documents à remettre

- Cahier des charges : Document contenant toutes les spécifications du projet ainsi que son organisation et le calendrier qui le régit
- Documentation d'installation : Document contenant les protocoles utilisés pour installer tous les programmes fournis
- Document utilisateur : Document contenant l'utilisation des programmes fournis ainsi que l'explication du code pour faciliter la réintégration

- Rapport de projet : Document expliquant le travail fourni, et le fonctionnement du groupe durant la période de développement

4.2 Programmes à remettre

- La base de données : C'est le référentiel des pays contenant toutes les informations relatives aux pays
- Le serveur de Web Service en JAVA C'est l'application (le web service) qui permet la communication avec la base de données
- Le client PHP Un client destiné essentiellement à l'intégration dans des sites web.
- Le client JAVA Ce client est destiné à l'intégration dans des applications. Les sources des deux clients seront disponibles, afin de faciliter la réintégration dans de futures applications. Le code devra être commenté et clair.
- Le Client de gestion Destiné à la gestion, il sera privé et réservé à l'administration de la base de données. Il comportera une interface graphique simple permettant à des utilisateurs non familiers avec SQL de modifier la base de données.

5 Calendrier

5.1 Date de fin de projet

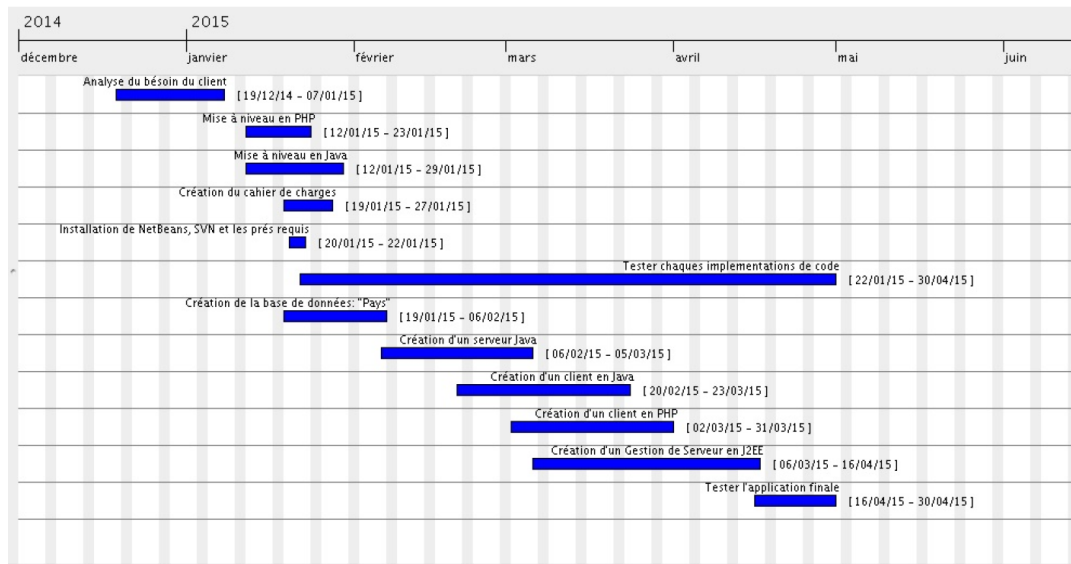
Les dates de soutenances ne nous ont pas encore été communiquées, il semblerait que le projet soit évalué fin avril. Il est

impératif que le projet soit terminé la semaine du 20 avril 2014 au plus tard, pour permettre au client de bien prendre en main l'application. Afin qu'il puisse le jour de la soutenance suivre correctement notre présentation.

5.2 Échéances intermédiaires

Chaque séance permettra à l'équipe de faire le point sur les différentes avancées de chaque rôle dans la réalisation du projet. Un contact constant par mail sera maintenu avec le client pour éviter tout phénomène de tunnel, et de pouvoir, éventuellement, réorienter le projet.

5.3 Diagramme de GANTT



A.2

Documentation Utilisateur

MANUEL D'UTILISATION DU WEB SERVICE PAYS D'ÉTAMINE

Team Etamine

Contents

Contents	2
1 Webservice	2
1.1 Demarrage	2
1.2 Modification du port d'écoute	3
2 Client JAVA	3
3 Client PHP	3
4 Application de Gestion	12
4.1 Demarrage	12
4.2 Connexion	12
4.3 Ajouter un pays	12
4.4 Supprimer un pays	12

1 Webservice

1.1 Demarrage

Pour démarrer le web service pays pour Étamine, il est nécessaire de lancer la classe `Application.java`.

1.2 Modification du port d'écoute

Pour modifier le port d'écoute du web service, il suffit de modifier la valeur "" dans le fichier `application.properties`.

2 Client JAVA

Ce client est prévu pour des développeurs Java.
Il est destiné à être intégré à une autre application.
Il n'y donc pas d'utilisation directe possible excepté par le biais des classes de test.

3 Client PHP

Guide d'utilisation du client PHP Etamine

Lorsqu'on accède à la page index de Etamine, on se retrouve sur la page suivante :

ETAMINE

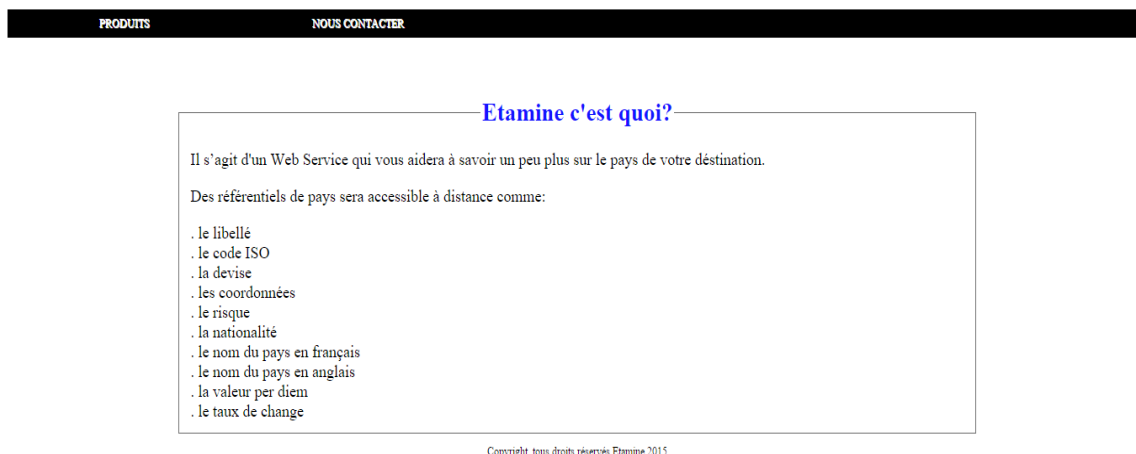


Figure clientphp1 : La page index de Etamine

Sur notre barre de Menu, on trouve deux options. L'un se nomme **PRODUITS** et l'autre **NOUS CONTACTER**.

Dans « PRODUITS », On trouvera trois sous-menus qui sont :

1. **C'est quoi Etamine ?** : C'est la page d'index du site, qui fait une brève description du web service Etamine et les données que l'on pourra extraire lorsqu'on utilise le web service.
2. **Comment utiliser Etamine ?** : C'est la page où l'on peut trouver les explication de l'utilisation de Étamine.
3. **Utiliser Etamine** : C'est la page PHP qui permettra d'accéder et d'utiliser le web service proprement dit.

Le menu « NOUS CONTACTER », il est constitué de 2 sous-menus :

1. **Email** : qui est une formulaire où l'on peut envoyer un message dans le cas d'un bug survenu lors d'une utilisation, ou bien de suggérer des améliorations possible.
2. **Adresse** : Affichera une page qui est directement reliée à l'adresse postale de l'Institut Galilée.

L'utilisation du web service Étamine se fait par la sélection du sous-menu Utiliser Etamine, qui redirigera vers la page suivante :

The screenshot shows the ETAMINE web service interface. At the top, there is a navigation bar with 'PRODUITS' and 'NOUS CONTACTER'. Below this, a section titled 'Sélectionnez votre choix' contains several radio button options: ID, Indicatif, Nom du pays en Français, Nom du pays en Anglais, Nationalité, Taux de change, Code Monnaie, Monnaie Per Diem, Risque, Afficher le risque du pays, and Drapeau. A dropdown menu for 'Choix du pays' is set to 'Andorre', with a 'Chercher' button next to it. Below the dropdown is a link: 'Cliquez ici pour être redirigé sur une autre page de recherche.' At the bottom, there is a section 'Pour afficher tous les Pays' with radio buttons for 'Avec risque' and 'Sans risque', and an 'afficher' button. A copyright notice 'Copyright, tous droits réservés Etamine 2015' is at the very bottom.

Figure client php 2 : la page d'utilisation de Etamine

Elle est composée de trois parties, la première « **Sélectionner votre choix** » fournira comme réponses des informations sur un pays sélectionné par rapport aux choix cochés par l'utilisateur (ID, indicatif, nom du pays en Français, nom du pays en Anglais, nationalité, taux de change, le code monnaie, la monnaie per diem, le risque de danger dans le pays et son drapeau).

Exemple d'utilisation :

This screenshot is identical to the one above but includes red annotations. A red box labeled 'Etape 3' encloses the 'Sélectionnez votre choix' section. Another red box labeled 'Etape 4' encloses the 'Choix du pays' dropdown menu and the 'Chercher' button.

Figure client php 3 : utilisation des paramètres de recherche pour afficher les informations d'un pays

On va sélectionner les informations dont on aura besoin (cf. Etape 3 et 4). L'**étape 3** permet de choisir les valeurs à rechercher. Tandis que l'**étape 4** permet de choisir les pays dont on recherche les informations.

Après avoir fait le choix des paramètres et du pays, il suffit de cliquer sur « Chercher », comme indiqué sur la figure ci-dessous :

ETAMINE

PRODUITS
NOUS CONTACTER

Sélectionnez votre choix

☐ ID
 ☐ Indicateur
 ☐ Nom du pays en Français
 ☐ Nom du pays en Anglais
 ☐ Nationalité

☐ Taux de change
 ☐ Code Monnaie
 ☐ Monnaie Par Dieu
 ☐ Risque
 ☐ Afficher le risque du pays

☐ Drapeau

Choix du pays: Andorre Chercher

Andorre

Andorre

Emirats arabes unis

Emirats arabes unis

Afghanistan

Afghanistan

Antigua-et-Barbuda

Antigua-et-Barbuda

Anguilla

Anguilla

Albanie

Albanie

Arménie

Arménie

Antilles néerlandaises

Antilles néerlandaises

Angola

Angola

Antarctique

Antarctique

Argentine

Argentine

Samoa américaines

Samoa américaines

Autriche

Autriche

Australie

Australie

Aruba

Aruba

Azerbaïdjan

Azerbaïdjan

Bosnie-Herzégovine

Bosnie-Herzégovine

Barbade

Barbade

Bangladesh

Bangladesh

Belgique

Belgique

Figure client php 3a : validation de la recherche selon les paramètres cochés

La réponse sera obtenue sous cette forme :

ETAMINE

PRODUITS
NOUS CONTACTER

☐ ID
 ☐ Indicateur
 ☐ Nom du pays en Français
 ☐ Nom du pays en Anglais
 ☐ Nationalité

☐ Taux de change
 ☐ Code Monnaie
 ☐ Monnaie Par Dieu
 ☐ Risque
 ☐ Afficher le risque du pays

☐ Drapeau

Choix du pays: Madagascar Chercher

Résultat de la recherche

ID	134
Indicatif	mg
Nom du pays en français	Madagascar
Nom du pays en anglais	Madagascar

✕

Figure client php 3b : Réponse affichée selon les paramètres choisis

La deuxième partie, « **Pour afficher tous le Pays** » affichera la liste des pays selon le choix du paramètre risque. Deux choix se présentent, avoir la liste des pays qui sont potentiellement à risque ou bien la liste des pays qui ne le sont pas. Après avoir choisi, cliquer sur « **afficher** »

Exemple d'exécution :

ETAMINE

PRODUITS

NOUS CONTACTER

Selectionnez votre choix

☐ ID

☐ Indicatif

☐ Nom du pays en Français

☐ Nom du pays en Anglais

☐ Nationalité

☐ Taux de change

☐ Code Monnaie

☐ Monnaie Per Diem

☐ Risque

☐ Drapeau

☐ Afficher le risque du pays

Choix du pays:

Pour afficher tous les Pays

☐ Avec risque ☐ Sans risque

Figure client php 4 : choix de la liste de pays par rapport au risque

Après avoir choisi les pays à risques, on obtiendra la réponse suivante:

Liste des pays

☐ ID

☐ In

☐ e en Anglais

☐ Nationalité

☐ Taux de change

☐ Co

☐ Risque du pays

☐ Drapeau

Afghanistan

Arménie

Angola

Azerbaïdjan

Pour afficher tous les Pays

☐ Avec risque ☐ Sans risque

Pour fermer

Figure client php 4a : liste des pays avec présence de risque de danger

La troisième partie, « **cliquer ici pour être redirigé vers une autre page de recherche** » va diriger l'utilisateur vers une page dédiée à des recherches plus générales.

ETAMINE

PRODUITS

NOUS CONTACTER

Selectionnez votre choix

☐ ID

☐ Indicateur

☐ Nom du pays en Français

☐ Nom du pays en Anglais

☐ Nationalité

☐ Taux de change

☐ Code Monnaie

☐ Monnaie Per Diem

☐ Risque

☐ Afficher le risque du pays

☐ Drapeau

Choix du pays:

Andorre

 Chercher

Cliquer ici pour être redirigé sur une autre page de recherche

Pour afficher tous les Pays

* Avec risque

☐ Sans risque

afficher

Copyright, tous droits réservés Etamine 2015

Figure client php 5 : liens pour aller à la page de recherche détaillée

Après avoir cliqué sur ce bouton, la page suivante s'affichera :

ETAMINE

PRODUITS

NOUS CONTACTER

Recherche d'informations

Rechercher un pays

Nom en français *

Nom en anglais *

Chercher

Code monnaie:

(Ex: A, en, ...)

Chercher

Monnaie Per Diem:

Sup *

Inf *

Egale *

Chercher

Taux de change:

Sup *

Inf *

Egale *

Chercher

Risque:

Avec risque *

Sans risque *

Chercher

Figure client php 5a : la page de recherche d'informations

Plusieurs champs sont présents dans cette page pour permettre les recherches.

Rechercher un pays : prendra en paramètres le ou les caractères saisis puis affichera la liste des pays commençant par ces caractères. Et cette liste de pays dépendra aussi du choix du nom du pays, qui est soit en français, soit en anglais.

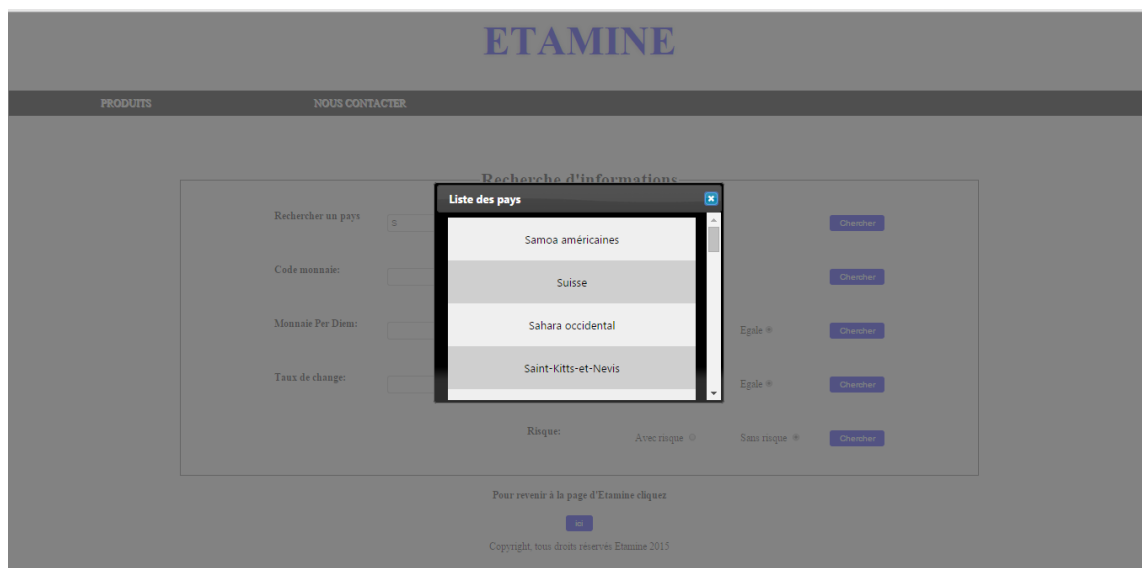


Figure client php 5a1 : Résultat d'une recherche d'un pays en français commençant par « S »

Code Monnaie : Permet d'afficher le pays avec la code monnaie correspondante aux caractères saisis en paramètre dans le champ de recherche.

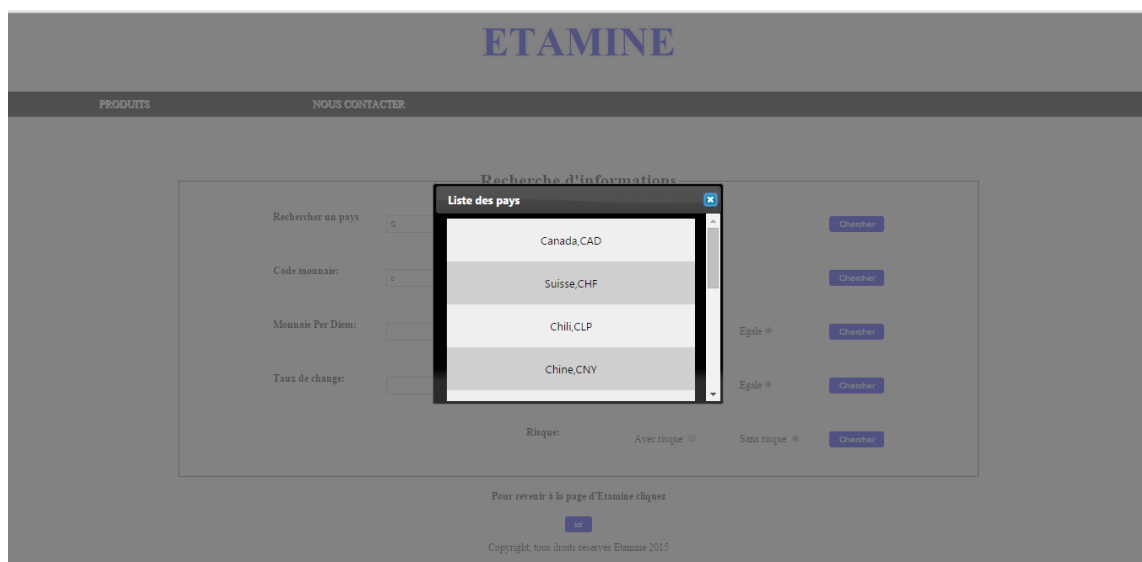
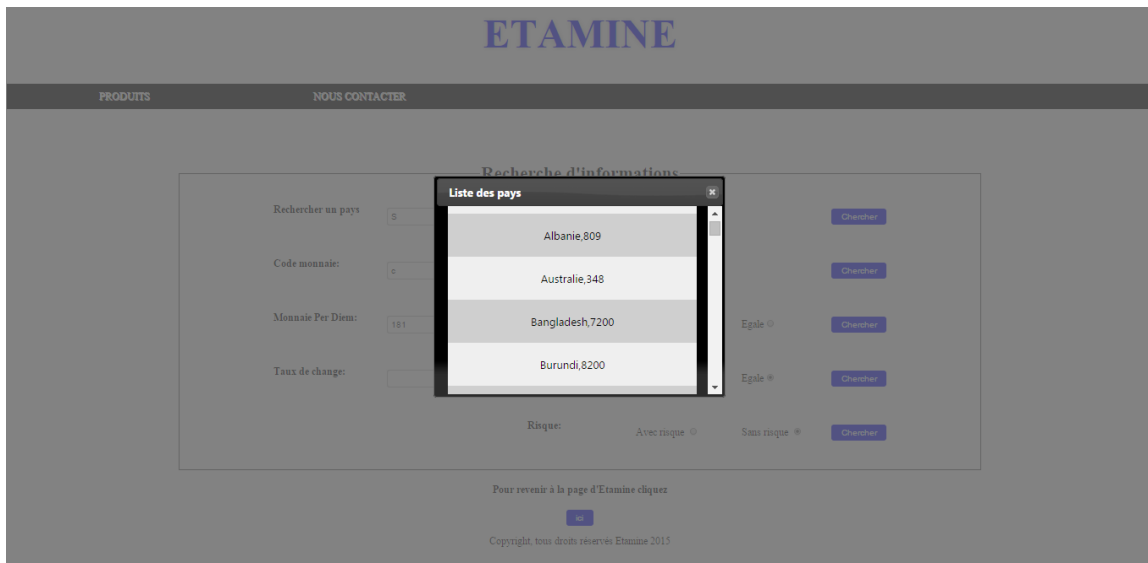


Figure client php 5a2 : Résultat d'une recherche des pays ayant un code monnaie commençant par « C »

Monnaie per diem : Permet de rechercher et d'afficher une liste de pays avec le barème des frais de mission en cours qui est supérieur, inférieur ou égal a celui du nombre saisi.



The screenshot shows the ETAMINE web application interface. The header includes the logo "ETAMINE" and navigation links "PRODUITS" and "NOUS CONTACTER". The main content area is titled "Recherche d'informations". A search form is displayed with the following fields and values:

- Rechercher un pays: 8
- Code monnaie: c
- Monnaie Per Diem: 181
- Taux de change: (empty)

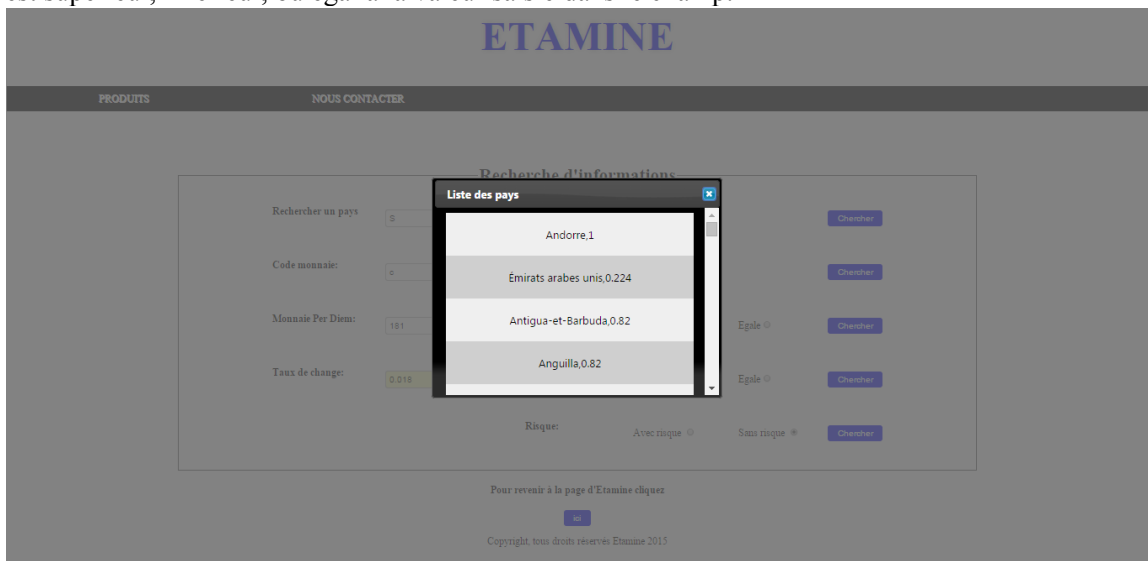
A modal window titled "Liste des pays" is open, displaying the following results:

Pays	Monnaie Per Diem
Albanie	809
Australie	348
Bangladesh	7200
Burundi	8200

Below the modal, there are radio buttons for "Risque:" with options "Avec risque" and "Sans risque". At the bottom, there is a link "Pour revenir à la page d'Etamine cliquez" and a copyright notice "Copyright, tous droits réservés Etamine 2015".

Figure client php 5a3 : Résultat d'une recherche d'un pays ayant une monnaie per diem supérieure à « 181 »

Taux de change : Comme la monnaie per diem, affichera une liste des pays dont le taux de change est supérieur, inférieur, ou égal à la valeur saisie dans le champ.



The screenshot shows the ETAMINE web application interface. The header includes the logo "ETAMINE" and navigation links "PRODUITS" and "NOUS CONTACTER". The main content area is titled "Recherche d'informations". A search form is displayed with the following fields and values:

- Rechercher un pays: 8
- Code monnaie: c
- Monnaie Per Diem: 181
- Taux de change: 0.018

A modal window titled "Liste des pays" is open, displaying the following results:

Pays	Taux de change
Andorre	1
Émirats arabes unis	0.224
Antigua-et-Barbuda	0.82
Anguilla	0.82

Below the modal, there are radio buttons for "Risque:" with options "Avec risque" and "Sans risque". At the bottom, there is a link "Pour revenir à la page d'Etamine cliquez" and a copyright notice "Copyright, tous droits réservés Etamine 2015".

Figure clientphp 5a4 : Résultat d'une recherche de pays ayant un taux de change supérieur à 0,018

Pour le **risque**, le résultat est identique au résultat de la figure client php 4

Pour contacter les développeurs du web service « Etamine » afin de signaler des bugs ou d'éventuelles suggestions, un formulaire pour envoyer un mail est présent dans le menu NOUS CONTACTER, sous-menu EMAIL.

ETAMINE

PRODUITS	NOUS CONTACTER
<div>Choisissez votre interlocuteur</div> <div>contacter: <input type="text" value="votre choix ?"/></div> <div>-----</div> <div>Votre message</div> <div>Objet : <input type="text"/></div> <div><div></div></div> <div>-----</div> <div>Pour vous répondre</div> <div>Votre nom : <input type="text"/></div> <div>Votre mail : <input type="text"/></div> <div><div>Envoyer</div><div>Effacer</div><div>Retour au page menu</div></div>	

Copyright, tous droits réservés Etamine 2015

Figure client php 6 : Formulaire pour l'envoi des messages

Ayant comme destinataire du mail :

Mr Michael FORTIER : Initiateur du projet « web service Etamine » et encadrant du projet.

Mr Quentin AMELOT : Chef de projet.

Mr Damien LARMINÉ : Responsable recherche et développement

Mr Jérémie NIZOU : Responsable de la documentation et de la base de données

Mr Fayize KAIMOU : Développeur JAVA

Mr Gisio TABERA : Développeur PHP

=====//=====//=====

4 Application de Gestion

4.1 Demarrage

Pour démarrer l'application de gestion, il faut lancer la classe `Application.java`.

4.2 Connexion

Afin de se connecter à l'application de gestion, il est nécessaire de rentrer un nom d'utilisateur et un mot de passe valide avant de cliquer sur le bouton login.

4.3 Ajouter un pays

Pour ajouter un pays, il faut cliquer sur le bouton Ajouter un pays. Il faut ensuite remplir les champs concernant le pays à ajouter. Il faut enfin cliquer sur le bouton ajouter pour ajouter le pays.

4.4 Supprimer un pays

Pour supprimer un pays, il faut cliquer sur le bouton Supprimer un pays. Il faut ensuite choisir le pays à supprimer.