

# Version de travail



## **INTERFACE HORANET / WEB CT**

## **SPECIFICATION D'ÉCHANGE**

**Réf : SE-WEB-CT/SmartTransac**  
**Version 1.0 – Révision M**

## Table des versions

Version	Révision	Date	Auteur	Contrôle	Désignation
1.0	A	2/10/2002	D. Dufour	JP Huon	Création du document
1.0	B	21/12/2004	D.Dufour	JP Huon	Extention des WebServices
1.0	C	21/01/2005	D.Dufour	JP Huon	Modification requête GetFamilyAccounts, ajout d'un label au type de compte ticket.
1.0	D	16/02/2005	D.Dufour	JP Huon	Ajout du WebService CreditAccount2
1.0	E	29/03/2005	D.Dufour	JP Huon	Modification du fichier XML joint au WebService CreditAccount2
1.0	F	12/05/2005	D.Dufour	JP Huon	Ajout des requêtes AddCanteenRegistrationWithoutCSN et GetChildTransactionsWithoutCSN
1.0	G	24/01/2006	D.Dufour	JP Huon	Mise à jour du document après mise en production du système.
1.0	H	14/03/2007	F.Robergeau	D.Dufour	Ajout des WebMethods GetAccountDetails et GetInvoiceDetails. Ajout de l'attribut child-csn
1.0	I	15/03/2007	F.Robergeau	D.Dufour	Modification des modèles XSD des WebMethods GetAccountDetails et GetInvoiceDetails.
1.0	J	26/03/2007	F.Robergeau	D.Dufour	Ajout du principe de fonctionnement de chaque méthode.
1.0	K	06/08/2007	F.Robergeau		Précision de la méthode CreditAccount2.
1.0	L	30/08/2007	D.Dufour		- Modification d'une erreur dans le XSD de GetFamilyAccounts (remplacement de child-csn par child-id) - Ajout d'exemples pour chaque type de paiement
1.0	M	30/08/207	D.Dufour		- Ajout du schéma XSD et d'un exemple XML pour GetChildTransactions2 - Ajout du schéma XSD pour CreditAccount2 - Modification du nombre d'occurrence sur les schémas de GetAccountDetails et GetInvoiceDetails

## Table de diffusion

Version	Révision	Date	Entreprise – Noms - Remarques
1.0	A	14/12/2004	Nicolas Farges - Pour première lecture
1.0	B	21/12/2004	Nicolas Farges - Pour première lecture
1.0	C	21/01/2005	Philippe Usclade
1.0	D	17/02/2005	Philippe Usclade
1.0	E	29/03/2005	Philippe Usclade, Guillaume Bort
1.0	F	12/05/2005	Philippe Usclade, Benoit Orihuela
1.0	G	25/01/2006	Philippe Usclade, Benoit Orihuela
1.0	H	14/3/2007	Philippe Usclade, Benoit Orihuela
1.0	I	15/3/2007	Philippe Usclade, Benoit Orihuela
1.0	J	15/3/2007	Philippe Usclade, Benoit Orihuela
1.0	K	06/8/2007	Philippe Usclade, Benoit Orihuela

# SOMMAIRE

<b>1. INTRODUCTION .....</b>	<b>1</b>
1.1. OBJECTIF DU DOCUMENT.....	1
<b>2. INSTALLATION.....</b>	<b>2</b>
2.1. INSTALLATION.....	2
2.2. PARAMETRAGE DU WEB.CONFIG.....	2
2.3. PROCEDURE DE TESTS .....	2
<b>3. LISTE DES SERVICES WEB.....</b>	<b>4</b>
3.1. ADDREGISTRATION .....	4
3.1.1. Principe.....	4
3.1.2. SOAP.....	4
3.1.3. HTTP POST .....	5
3.1.4. EXCEPTIONS.....	5
3.2. GETCHILDTRANSACTIONS2 .....	5
3.2.1. Principe.....	5
3.2.2. SOAP.....	5
3.2.3. HTTP POST .....	6
3.2.4. EXCEPTIONS.....	6
3.2.5. Schema XSD du fichier joint.....	6
3.2.6. Exemple XML du fichier joint.....	7
3.3. GETFAMILYACCOUNTS .....	7
3.3.1. Principe.....	7
3.3.2. SOAP.....	7
3.3.3. HTTP POST .....	8
3.3.4. EXCEPTIONS.....	8
3.3.5. Schema XSD du fichier joint.....	8
3.3.6. Exemple XML du fichier joint.....	9
3.4. CREDITACCOUNT2 .....	10
3.4.1. Principe.....	10
3.4.2. SOAP.....	10
3.4.3. HTTP POST .....	10
3.4.4. EXCEPTIONS.....	11
3.4.5. Schema XSD du fichier joint.....	11
3.4.6. Exemple XML du fichier joint.....	12
3.5. GETACCOUNTDETAILS .....	13
3.5.1. Principe.....	13
3.5.2. SOAP.....	13
3.5.3. HTTP POST .....	13
3.5.4. Schema XSD du fichier joint.....	14
3.5.5. Exemple XML du fichier joint.....	14
3.5.6. Exemple de rapport .....	15
3.6. GETINVOICEDETAILS .....	15
3.6.1. Principe.....	15
3.6.2. SOAP.....	15
3.6.3. HTTP POST .....	16
3.6.4. Schema XSD du fichier joint.....	16

3.6.5. Exemple XML du fichier joint.....	16
3.6.6. Exemple de rapport .....	17

Version de travail

# 1. Introduction

## 1.1. Objectif du document

- Ce document liste les services Web appelables par l'application WebCT ainsi que les exceptions pouvant être retournées.

Après une période de test, 6 WebServices ont été retenus pour la mise en production :

- ✓ **AddRegistration (interface CVQ2)** : permet d'ajouter une téléprocédure au système. Celle-ci servira de référence pour rechercher les informations des individus qu'elle contient. La téléprocédure est le pivot du système.
- ✓ **GetChildTransactions2 (interface CVQ2)** : Permet d'obtenir les 'badgeages' d'un individu sur le système par son identifiant de téléprocédure.
- ✓ **GetFamilyAccounts (interface CVQ)** : Permet d'obtenir l'ensemble des informations financières (facture, comptes prépayés, tickets) des individus d'un foyer.
- ✓ **CreditAccount2 (interface CVQ)** : Permet de revaloriser les comptes ou de payer les factures d'un foyer.
- ✓ **GetAccountDetails (interface CVQ3)** : Permet de récupérer les différentes réévaluations d'un compte.
- ✓ **GetInvoiceDetails (interface CVQ3)** : Permet de récupérer les lignes de facture.

## 2. Installation

### 2.1. Installation

Certains éléments sont obligatoires pour l'installation du service web :

- Présence du Framework 1
- Présence de Microsoft WSE 2.0 SP3
- Présence de MSXML 4.0
- Présence de CentralTransac version 1.7.2664.24067 mini
- Si réinstallation, désinstaller la version.
- Installer Horanet\_WS-WebCT.msi
- Copier le fichier WebInstallation.config en Web.config puis paramétrer le.

### 2.2. Paramétrage du web.config

- Plusieurs clés sont obligatoires pour l'appel au service web.
  - ✓ Appel aux bases de données :
    - **Clé InvokeTableSmartIntegral (valeur 0 ou 1)**  
Définir si l'on invoque les tables spécifiques Smart Integral ou les tables CT (0=CT, 1=SmartIntegral)
    - **Clé SqlConnection**  
Définir la connection de la base où se trouve les tables CT.
    - **Clé SqlConnectionSmartIntegral**  
Définir la connection à la base Smart Integral si la base est indépendante de la base CT
  - ✓ Activation des traces:
    - **Clé SmartWsLog (valeur True ou False)**  
VALUE="True" pour activer les traces sinon value="False"

### 2.3. Procédure de tests

- Ouvrir le port correspondant (ex : 85)

- Vérification de l'espace de nom pour les services web (ex : <http://www.horanet.com/CVQ/> )
- Vérification après exécution de la méthode AddRegistration de la présence du fichier Xml envoyé dans la table ct\_file\_from\_web.
- Démarrage de CentralTransac
- Vérification que le champ ct\_xsl\_parsed soit à 1 dans la table ct\_file\_from\_web.
  - Si = 0 pas de traitement effectué
  - Si = 1 traitement exécuté
  - Si = 2 problème d'exécution

Version de travail



## 3. Liste des Services Web

### 3.1. AddRegistration

#### 3.1.1. Principe

La fonction 'AddRegistration' permet d'ajouter une télé-procédure au système. Celle-ci servira de référence pour rechercher les informations des individus qu'elle contient. La téléprocédure est le pivot du système.

On insert dans la table CT\_FILE\_FROM\_WEB la télé-procédure, puis le service NT CentralTransac scrute cette table afin d'insérer dans KEVEN si une nouvelle télé-procédure est ajoutée.

Selon l'endroit où se situe les tables CT, l'enregistrement d'une famille, usager, ... se fait soit dans les tables spécifiques CT ou dans les tables de SmartIntégral.

#### 3.1.2. SOAP

Le texte suivant est un exemple de demande et de réponse SOAP. Les **espaces réservés** affichés doivent être remplacés par des valeurs réelles.

```
POST /WS-WebCT/CVQ2.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://www.horanet.com/CVQ2/AddRegistration"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <AddRegistration xmlns="http://www.horanet.com/CVQ2/">
      <ZipCode>string</ZipCode>
      <ActivityID>string</ActivityID>
      <ProcClass>string</ProcClass>
      <ProcID>string</ProcID>
      <FamilyID>string</FamilyID>
      <School>string</School>
      <ChildID>string</ChildID>
      <ChildCard>string</ChildCard>
    </AddRegistration>
  </soap:Body>
</soap:Envelope>
```

### 3.1.3. HTTP POST

Le texte suivant est un exemple de demande et de réponse HTTP POST. Les **espaces réservés** affichés doivent être remplacés par des valeurs réelles.

```
POST / WS-WebCT /CVQ2.asmx/AddRegistration HTTP/1.1
Host: localhost
Content-Type: application/x-www-form-urlencoded
Content-Length: length

ZipCode=string&ActivityID=string&ProcClass=string&ProcID=string&FamilyID=string&School=string&ChildID=string&ChildCard=string
```

### 3.1.4. EXCEPTIONS

Ci-dessous, la liste des exceptions pouvant être retournées :

"This Town " + ZipCode + " is not well formatted"
"The ActivityID field is empty"
"The ProcClass field is empty"
"The FamilyID field is empty"
"This ChildCard " + ChildCard + " is not well formatted"
"This Town " + ZipCode + " is unknown"
"This Activity " + ActivityID + " is unknown is this town " + ZipCode

## 3.2. GetChildTransactions2

### 3.2.1. Principe

La fonction 'GetChildTransactions' permet d'obtenir les 'badgées' d'un individu sur le système par son identifiant de télé-procédure et une période donnée.

Par l'identifiant de la télé-procédure, on retrouve le numéro externe de l'utilisateur dans la table CT\_FILE\_FROM\_WEB.

Puis à l'aide de cette identifiant et si on se trouve dans le cas de tables spécifiques CT on appelle la procédure stockée 'CVQ\_GET\_XML\_CHILD\_EVENTS' sinon on utilise la procédure 'CVQ\_XML\_TRANSACTIONS' de SmartIntegral.

### 3.2.2. SOAP

Le texte suivant est un exemple de demande et de réponse SOAP. Les **espaces réservés** affichés doivent être remplacés par des valeurs réelles.

```
POST /WS-WebCT/CVQ2.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://www.horanet.com/CVQ2/GetChildTransactions2"
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetChildTransactions2 xmlns="http://www.horanet.com/CVQ2/">
      <ZipCode>string</ZipCode>
      <ProcID>string</ProcID>
      <start_search>dateTime</start_search>
      <end_search>dateTime</end_search>
    </GetChildTransactions2>
  </soap:Body>
</soap:Envelope>
```

### 3.2.3. HTTP POST

Le texte suivant est un exemple de demande et de réponse HTTP POST. Les **espaces réservés** affichés doivent être remplacés par des valeurs réelles.

```
POST /WS-WebCT/CVQ2.asmx/GetChildTransactions2 HTTP/1.1
Host: localhost
Content-Type: application/x-www-form-urlencoded
Content-Length: length
```

### 3.2.4. EXCEPTIONS

Ci-dessous, la liste des exceptions pouvant être retournées :

"This Town " + ZipCode + " is not well formated"
"The ActivityID field is empty"
"The FamilyID field is empty"
"This ChildCard " + ChildCard + " is not well formated"
"This child " + ChildCard + "(" + ZipCode + ") is unknown or not in this family : " + FamilyID
"This activity " + ActivityID + "(" + ZipCode + ") is unknown"

### 3.2.5. Schema XSD du fichier joint

```
<?xml version="1.0" encoding="utf-8" ?>
<xs:schema attributeFormDefault="unqualified"
elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="events">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="unbounded" name="event">
          <xs:complexType>
            <xs:attribute name="event-id" type="xs:integer" />
            <xs:attribute name="event-date" type="xs:dateTime" />
            <xs:attribute name="event-label" type="xs:string"/>
            <xs:attribute name="event-label2" type="xs:string"/>
            <xs:attribute name="event-label3" type="xs:string"/>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

```

        <xs:attribute name="event-label4" type="xs:string"/>
    </xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>

```

### 3.2.6. Exemple XML du fichier joint

```

<events>
  <event event-id="19646851" event-date="2007-07-02T08:38:51" event-label="BambiSmart" event-label2="RMI"
event-label3="Repas" event-label4="Meal"/>
  <event event-id="19647171" event-date="2007-07-03T08:39:52" event-label="BambiSmart" event-label2="RMI"
event-label3="Repas" event-label4="Meal"/>
</events>

```

## 3.3. GetFamilyAccounts

### 3.3.1. Principe

La fonction 'GetFamilyAccounts' permet d'obtenir l'ensemble des informations financières (facture, comptes prépayés, tickets) des individus d'un foyer.

Grace à l'identifiant de la famille, on retrouve ses comptes associés ainsi que ces factures et contrats.

Si l'on se trouve dans le cas de tables spécifiques CT on appelle la procédure stockée 'CVQ\_GET\_XML\_FAMILY\_ACCOUNTS' sinon on utilise les procédures stockées 'CVQ\_XML\_COMPTES', 'CVQ\_XML\_FACTURES' et 'CVQ\_XML\_CONTRATS' de SmartIntegral.

### 3.3.2. SOAP

Le texte suivant est un exemple de demande et de réponse SOAP. Les **espaces réservés** affichés doivent être remplacés par des valeurs réelles.

```

POST /WS-WebCT/CVQ.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://horanet.com/CVQ/GetFamilyAccounts"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetFamilyAccounts xmlns="http://horanet.com/CVQ/">
      <ZipCode>string</ZipCode>
      <FamilyID>string</FamilyID>
      <start_search>dateTime</start_search>
      <end_search>dateTime</end_search>
    </GetFamilyAccounts>
  </soap:Body>
</soap:Envelope>

```

```

    </GetFamilyAccounts>
  </soap:Body>
</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetFamilyAccountsResponse xmlns="http://horanet.com/CVQ/" />
  </soap:Body>
</soap:Envelope>

```

### 3.3.3. HTTP POST

Le texte suivant est un exemple de demande et de réponse HTTP POST. Les **espaces réservés** affichés doivent être remplacés par des valeurs réelles.

```

POST /WS-WebCT/CVQ.asmx/GetFamilyAccounts HTTP/1.1
Host: localhost
Content-Type: application/x-www-form-urlencoded
Content-Length: length

ZipCode=string&FamilyID=string&start_search=string&end_search=string
HTTP/1.1 200 OK

```

### 3.3.4. EXCEPTIONS

Ci-dessous, la liste des exceptions pouvant être retournées :

"This Town " + ZipCode + " is not well formated"
--

"This family " + FamilyID + "(" + ZipCode + ") is unknown"
--

### 3.3.5. Schema XSD du fichier joint

```

<?xml version="1.0" encoding="utf-16"?>
<xs:schema xmlns:msdata="urn:schemas-microsoft-com:xml-msdata" id="CVQAccounts.xml"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="family">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="unbounded" name="accounts">
          <xs:complexType>
            <xs:sequence>
              <xs:element minOccurs="0" maxOccurs="unbounded" name="account">
                <xs:complexType>
                  <xs:attribute name="account-id" type="xs:string"/>
                  <xs:attribute name="account-value" type="xs:string"/>
                  <xs:attribute name="account-date" type="xs:string"/>
                  <xs:attribute name="account-label" type="xs:string"/>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element minOccurs="0" maxOccurs="unbounded" name="bills">

```

```

<xs:complexType>
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="bill">
      <xs:complexType>
        <xs:attribute name="bill-id" type="xs:string"/>
        <xs:attribute name="bill-value" type="xs:string"/>
        <xs:attribute name="bill-label" type="xs:string"/>
        <xs:attribute name="bill-date" type="xs:string"/>
        <xs:attribute name="bill-date-expiration" type="xs:string"/>
        <xs:attribute name="bill-date-payment" type="xs:string"/>
        <xs:attribute name="bill-payd" type="xs:int"/>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element minOccurs="0" maxOccurs="unbounded" name="contracts">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="child">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="unbounded" name="contract">
              <xs:complexType>
                <xs:attribute name="contract-id" type="xs:string"/>
                <xs:attribute name="contract-label" type="xs:string"/>
                <xs:attribute name="contract-value" type="xs:string"/>
                <xs:attribute name="contract-date" type="xs:string"/>
                <xs:attribute name="buy-price" type="xs:string"/>
                <xs:attribute name="min-buy" type="xs:string"/>
                <xs:attribute name="max-buy" type="xs:string"/>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
          <xs:attribute name="child-card" type="xs:string"/>
          <xs:attribute name="child-id" type="xs:string"/>
          <xs:attribute name="town-id" type="xs:string"/>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:sequence>
<xs:attribute name="id" type="xs:string"/>
</xs:complexType>
</xs:element>
</xs:schema>

```

### 3.3.6. Exemple XML du fichier joint

```

<family id="thierry.dupuis">
  <accounts>
    <account account-id="95999-1-123456" account-value="21265" account-date="2005-01-21T11:14:10.060"/>
  </accounts>
  <bills>
    <bill bill-id="95999-2-654321" bill-value="21478" bill-label="21478" bill-date="2004-12-28T15:35:34.147" bill-date-expiration="2004-12-30T15:35:34.163" bill-date-payment="2004-12-29T15:35:34.163" bill-payd="0"/>
    <bill bill-id="95999-2-654322" bill-value="1296" bill-label="21478" bill-date="2004-12-28T15:35:34.163" bill-date-expiration="2005-01-28T15:35:34.163" bill-date-payment="2005-01-11T15:35:34.163" bill-payd="1"/>
  </bills>
  <contracts>
    <child child-card="000001" child-id="000001" town-id="95999">
      <contract contract-id="95999-3-1234" contract-label="RESTAURANT SCOLAIRE" contract-value="300" contract-date="2004-12-28T15:35:34.163" buy-price="100" min-buy="1" max-buy="10"/>
      <contract contract-id="95999-3-1235" contract-label="RESTAURANT SCOLAIRE" contract-value="800" contract-date="2004-12-28T15:35:34.163" buy-price="100" min-buy="1" max-buy="10"/>
    </child>
  </contracts>
</family>

```

### 3.4. CreditAccount2

#### 3.4.1. Principe

La fonction 'CreditAccount2' permet d'ajouter un paiement sur plusieurs comptes joints. Par exemple, le rechargement de plusieurs comptes tickets n'ayant entraîné qu'une seule transaction bancaire.

Si l'on se trouve dans le cas de tables spécifiques CT on met à jour la table 'CT\_ACCOUNTS' et la table 'CT\_PAYMENT2' sinon on ajoute un événement dans KEVEN pour la mise à jour dans SmartIntégral.

#### 3.4.2. SOAP

Le texte suivant est un exemple de demande et de réponse SOAP. Les **espaces réservés** affichés doivent être remplacés par des valeurs réelles.

```
POST /WS-WebCT/CVQ.aspx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://horanet.com/CVQ/CreditAccount2"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <CreditAccount2 xmlns="http://horanet.com/CVQ/">
      <ZipCode>string</ZipCode>
      <FamilyID>string</FamilyID>
    </CreditAccount2>
  </soap:Body>
</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <CreditAccount2Response xmlns="http://horanet.com/CVQ/" />
  </soap:Body>
</soap:Envelope>
```

#### 3.4.3. HTTP POST

Le texte suivant est un exemple de demande et de réponse HTTP POST. Les **espaces réservés** affichés doivent être remplacés par des valeurs réelles.

```
POST /WS-WebCT/CVQ.aspx/CreditAccount2 HTTP/1.1
Host: localhost
```

```
Content-Type: application/x-www-form-urlencoded
Content-Length: length

ZipCode=string&FamilyID=string
HTTP/1.1 200 OK
```

#### 3.4.4. EXCEPTIONS

Ci-dessous, la liste des exceptions pouvant être retournées :

"This Town " + ZipCode + " is not well formatted"
"This family " + FamilyID + "(" + ZipCode + ") is unknown"
"The payment for family " + FamilyID + "(" + ZipCode + ") has no attached file"

#### 3.4.5. Schema XSD du fichier joint

```
<?xml version="1.0" encoding="utf-8" ?>
<xs:schema attributeFormDefault="unqualified"
elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="bank-transaction">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="1" maxOccurs="1" name="payment">
          <xs:complexType>
            <xs:attribute name="payment-date" type="xs:dateTime" />
            <xs:attribute name="payment-amount" type="xs:integer" />
            <xs:attribute name="payment-ack" type="xs:string" />
            <xs:attribute name="cvq-ack" type="xs:string" />
          </xs:complexType>
        </xs:element>
        <xs:element minOccurs="1" maxOccurs="1" name="family">
          <xs:complexType>
            <xs:attribute name="id" type="xs:string" />
            <xs:attribute name="zip" type="xs:string" />
          </xs:complexType>
        </xs:element>
        <xs:element minOccurs="1" maxOccurs="1" name="accounts">
          <xs:complexType>
            <xs:sequence>
              <xs:element minOccurs="1" maxOccurs="unbounded"
name="account">
                <xs:complexType>
                  <xs:attribute name="account-id" type="xs:string"
use="required" />
                  <xs:attribute name="child-id" type="xs:string"
use="optional" />
                  <xs:attribute name="qantity" type="xs:integer"
use="optional" />
                  <xs:attribute name="price" type="xs:integer"
use="optional" />
                  <xs:attribute name="amount" type="xs:integer"
use="required" />
                  <xs:attribute name="account-old-value" type="xs:integer"
use="optional" />
                </xs:complexType>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:schema>
```



```

        <xs:attribute name="account-old-value-date"
type="xs:dateTime" use="optional" />
        <xs:attribute name="account-new-value" type="xs:integer"
use="optional" />
    </xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
<xs:attribute name="version" type="xs:string" use="optional" />
</xs:complexType>
</xs:element>
</xs:schema>

```

### 3.4.6. Exemple XML du fichier joint

#### Cas de la réévaluation de contrat

```

<?xml version="1.0" encoding="UTF-8"?>
<bank-transaction version="1.0">
  <payment payment-date="2007-08-17T06:07:24.090" payment-amount="3894" payment-ack="70817180659058"
cvq-ack="P_13815_1187366785698"/>
  <family id="13815" zip="95130"/>
  <accounts>
    <account account-id="95130-3-10833760" child-id="13812" qantity="11" price="354" amount="3894"/>
  </accounts>
</bank-transaction>

```

#### Cas de la réévaluation de compte

```

<?xml version="1.0" encoding="UTF-8"?>
<bank-transaction version="1.0">
  <payment payment-date="2007-08-29T11:05:03.058" payment-amount="21712" payment-ack="70829110452487"
cvq-ack="P_5660_1188378242540"/>
  <family id="5660" zip="95620"/>
  <accounts>
    <account account-id="95620-1-1001153369" account-old-value="-21712" account-old-value-date="2007-08-
29T10:07:50.223" account-new-value="0" qantity="0" price="0" amount="21712"/>
  </accounts>
</bank-transaction>

```

#### Cas du paiement de facture

```

<?xml version="1.0" encoding="UTF-8"?>
<bank-transaction version="1.0">
  <payment payment-date="2004-12-28T15:35:34.163" payment-amount="22519" payment-ack="12345678" cvq-
ack="87654321"/>
  <family id="123" zip="95999"/>
  <accounts>
    <account account-id="95999-2-123456" amount="22519"/>
  </accounts>
</bank-transaction>

```

### 3.5. GetAccountDetails

#### 3.5.1. Principe

La fonction 'GetAccountDetails' permet de récupérer les différentes réévaluations d'un compte pour une période donnée.

On récupère l'information grâce à une procédure stockée 'CVQ\_XML\_REGLEMENTS' de la base SmartIntégral.

#### 3.5.2. SOAP

Le texte suivant est un exemple de demande et de réponse SOAP. Les **espaces réservés** affichés doivent être remplacés par des valeurs réelles.

```
POST /WS-WebCT/CVQ3.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://horanet.com/CVQ3/GetAccountDetails"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetAccountDetails xmlns="http://horanet.com/CVQ3/">
      <AccountID>string</AccountID>
      <start_search>dateTime</start_search>
      <end_search>dateTime</end_search>
    </GetAccountDetails>
  </soap:Body>
</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetAccountDetailsResponse xmlns="http://horanet.com/CVQ3/" />
  </soap:Body>
</soap:Envelope>
```

#### 3.5.3. HTTP POST

Le texte suivant est un exemple de demande et de réponse HTTP POST. Les **espaces réservés** affichés doivent être remplacés par des valeurs réelles.

```
POST /WS-WebCT/CVQ3.asmx/GetAccountDetails HTTP/1.1
Host: localhost
Content-Type: application/x-www-form-urlencoded
Content-Length: length

AccountID=string&start_search=string&end_search=string
```

HTTP/1.1 200 OK

### 3.5.4. Schema XSD du fichier joint

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="accountDetails">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="unbounded" name="accountDetail">
          <xs:complexType>
            <xs:sequence>
              <xs:element minOccurs="1" name="date" type="xs:datetime"/>
              <xs:element minOccurs="1" name="holder-name" type="xs:string"/>
              <xs:element minOccurs="1" name="holder-surname" type="xs:string"/>
              <xs:element minOccurs="1" name="value" type="xs:int"/>
              <xs:element minOccurs="1" name="payment-type" type="xs:string"/>
              <xs:element minOccurs="1" name="payment-id" type="xs:string"/>
              <xs:element minOccurs="1" name="payment-ack" type="xs:string"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
      <xs:attribute name="account-id" type="xs:string"/>
    </xs:complexType>
  </xs:element>
  <xs:element>
    <xs:complexType>
      <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:element ref="accountDetails"/>
      </xs:choice>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

### 3.5.5. Exemple XML du fichier joint

```
<accountDetails account-id="13663687">
  <accountDetail>
    <date>2004-12-28T15:35:34.163</date>
    <holder-name>roger</holder-name>
    <holder-surname>TOULOUSE</holder-surname>
    <value>20345</value>
    <payment-type>Chèque</payment-type>
    <payment-id>0101566442</payment-id>
    <payment-ack>0101566442543156</payment-ack>
  </accountDetail>
  <accountDetail>
    <date>2004-12-22T14:58:32.189</date>
    <holder-name>roger</holder-name>
    <holder-surname>TOULOUSE</holder-surname>
    <value>10428</value>
    <payment-type>Automate CB</payment-type>
    <payment-id>0101566442-XXXXXXX-543156</payment-id>
    <payment-ack>0101566442543156</payment-ack>
  </accountDetail>
</accountDetails>
```

### 3.5.6. Exemple de rapport

**SAS VERT MARINE OCÉANIDE**  
Rue du Gaingalet  
BP 31  
85201 FONTENAY LE COMTE CEDEX

Compte

**Liste des règlements**  
Période du 01/09/2003 au 18/03/2004

Chèque

Date	Compte	Titulaire	Débiteur (tiré)	Banque	Montant
10/11/2003 12:08:20	95649	CHABOT VERONIQUE	CHABOT VERONIQUE	Autres Banques	120,00
			1	Chèque	120,00 €
			1	Total	120,00 €

Page 1 de 1

## 3.6. GetInvoiceDetails

### 3.6.1. Principe

La fonction 'GetInvoiceDetails' permet de récupérer les lignes de facture à partir de l'identifiant de la facture passée en paramètre.

On récupère l'information grâce à une procédure stockée 'CVQ\_XML\_FACTURE\_DETAIL' de la base SmartIntégral.

### 3.6.2. SOAP

Le texte suivant est un exemple de demande et de réponse SOAP. Les **espaces réservés** affichés doivent être remplacés par des valeurs réelles.

```
POST /WS-WebCT/CVQ3.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://horanet.com/CVQ3/GetInvoiceDetails"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetInvoiceDetails xmlns="http://horanet.com/CVQ3/">
      <InvoiceId>string</InvoiceId>
    </GetInvoiceDetails>
  </soap:Body>
</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

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

```
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetInvoiceDetailsResponse xmlns="http://horanet.com/CVQ3/" />
  </soap:Body>
</soap:Envelope>
```

### 3.6.3. HTTP POST

Le texte suivant est un exemple de demande et de réponse HTTP POST. Les **espaces réservés** affichés doivent être remplacés par des valeurs réelles.

```
POST /WS-WebCT/CVQ3.asmx/GetInvoiceDetails HTTP/1.1
Host: localhost
Content-Type: application/x-www-form-urlencoded
Content-Length: length
```

```
InvoiceId=string
HTTP/1.1 200 OK
```

### 3.6.4. Schema XSD du fichier joint

```
<?xml version="1.0" standalone="yes"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="invoiceDetails">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="invoiceDetail" minOccurs="0" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="child-name" type="xs:string" minOccurs="1"/>
              <xs:element name="child-surname" type="xs:string" minOccurs="1"/>
              <xs:element name="label" type="xs:string" minOccurs="1"/>
              <xs:element name="unit-price" type="xs:int" minOccurs="1"/>
              <xs:element name="quantity" type="xs:int" minOccurs="1"/>
              <xs:element name="value" type="xs:int" minOccurs="1"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
      <xs:attribute name="invoice-id" type="xs:string"/>
      <xs:attribute name="invoice-value" type="xs:int"/>
    </xs:complexType>
  </xs:element>
  <xs:element>
    <xs:complexType>
      <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:element ref="invoiceDetails"/>
      </xs:choice>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

### 3.6.5. Exemple XML du fichier joint

```
<invoiceDetails invoice-id="4" invoice-value="10000">
  <invoiceDetail>
    <child-name>Lolita</child-name>
    <child-surname>TOULOUSE</child-surname>
    <label>carence</label>
    <unit-price>2300</unit-price>
```

```

<quantity>2</quantity>
<value>4600</value>
</invoiceDetail>
<invoiceDetail>
  <child-name>Pierre</child-name>
  <child-surname>TOULOUSE</child-surname>
  <unit-price>450</unit-price>
  <quantity>12</quantity>
  <value>5400</value>
</invoiceDetail>
</invoiceDetails>

```

### 3.6.6. Exemple de rapport

**R Aperçu avant impression**

100 % 1 Fermer

**SAS VERT MARINE OCÉANIDE**  
 Rue du Gaingalet  
 BP 31  
 85201 FONTENAY LE COMTE CEDEX  
 Tél. 0251000521 Fax 0251000010

**Facture**

N° de compte 1591684  
 N° de facture 184  
 Date de facture 06 août 2004  
 Date d'échéance 31 août 2004  
 Période du 06/08/2004  
 au 30/09/2004 23:59:59

**BLANC MARINE**  
 1 BIS RUE DU PONT  
 94340 JOINVILLE

Usager	Produit	Détail	Prix brut	Qté	Prix total
<b>BLANC MARINE</b>					
	ENTREE TARIF GROUPE	ENTREE TARIF GROUPE	2,30	7	16,10
					16,10

Page 1