

# **Sips Office Server 2.04** **Guide du développeur**

**Version 2.00 – Octobre 2010**



## **REACH YOUR TARGETS >>**

### **Contact**

By phone: +33 (0)811 107 033  
By fax: +33 (0)811 107 033  
By email: [sips@atosorigin.com](mailto:sips@atosorigin.com)

# Sommaire

<b>1. INTRODUCTION.....</b>	<b>2</b>
<b>2. ARCHITECTURE GENERALE.....</b>	<b>3</b>
<b>3. PRINCIPE DE FONCTIONNEMENT.....</b>	<b>4</b>
<b>4. COMMENT INVOQUER LES SERVICES DE L'API SIPS OFFICE SERVER.....</b>	<b>6</b>
4.1 ETAPES DE MISE EN ŒUVRE D'UNE REQUETE DE SERVICE OU D'UNE REQUETE DE COMMANDE .....	6
4.2 LE CONTENU DES REQUETES .....	6
4.3 EXEMPLE DE REQUETE GETSTATUS.....	8
4.4 DEVELOPPER VOS REQUETES DE SERVICE ET VOS REQUETES DE COMMANDE .....	9
<b>5. RESOLUTION DES PROBLEMES RENCONTRES .....</b>	<b>10</b>

## 1. INTRODUCTION

Ce document présente le principe de fonctionnement de l'API Sips Office Server, et explique comment lui envoyer une requête. Pour illustrer le fonctionnement du serveur, des exemples de requêtes sont fournis. Pour mettre en œuvre un service des serveurs Sips via l'API Sips Office Server, il faut se référer au *GUIDE DU COMPOSANT* en question qui détaille les données de la requête, leur utilisation, le paramétrage, etc...

### A qui s'adresse ce manuel ?

Ce manuel s'adresse aux développeurs chargés d'intégrer un des composants de l'API Sips Office Server. Ceci nécessite donc des connaissances dans un langage de programmation. Des exemples de code sont fournis en C, ASP, Java, PHP et PERL.

**Note** : Ce document ne décrit pas comment vous interfacier avec votre système d'information ou votre base de données. Dans les exemples fournis, les variables sont déjà renseignées, vous devrez programmer la lecture et la mise à jour des données de votre système d'information.

### Conventions d'écriture

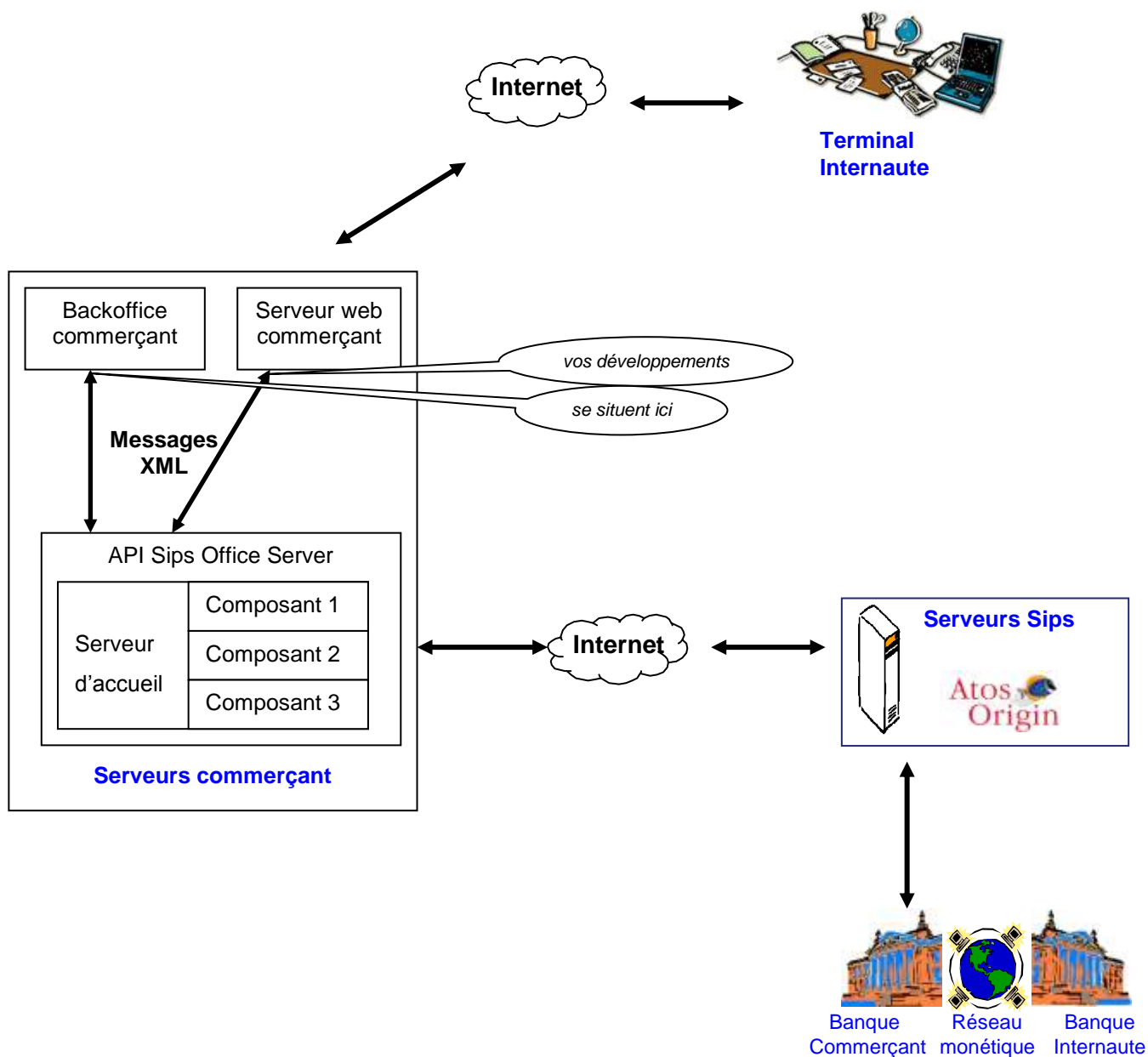
Dans tout le document, les conventions d'écriture suivantes seront utilisées :

- Les renvois à d'autres documentations seront notés en majuscules et en italique.

ex : *LE GUIDE D'INSTALLATION*

## 2. ARCHITECTURE GENERALE

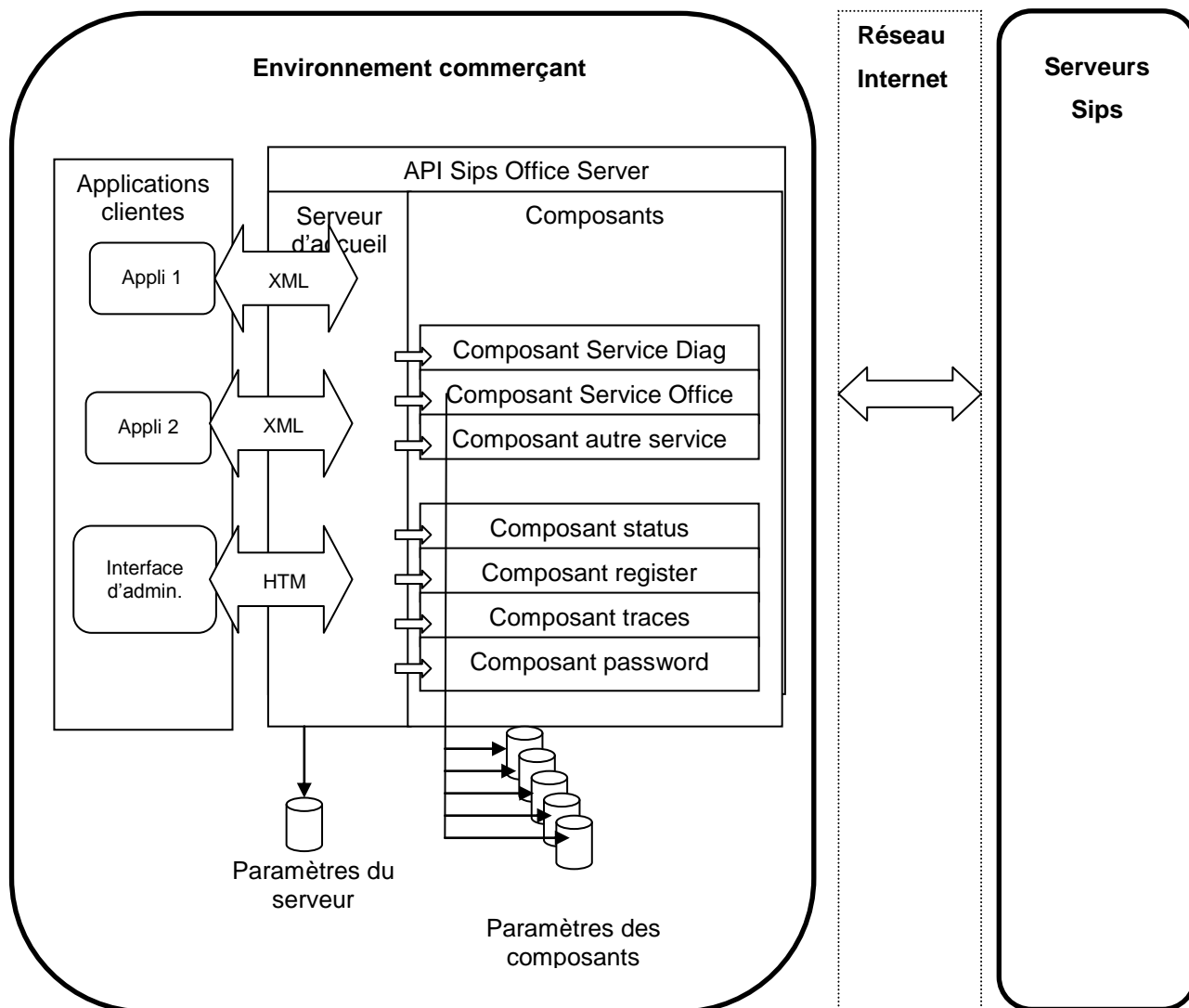
Le schéma ci-dessous présente l'architecture générale et les différents intervenants d'une transaction de paiement sur Internet.



### 3. PRINCIPE DE FONCTIONNEMENT

L'API Sips Office Server est implémenté sous la forme d'un serveur autonome qui reçoit des requêtes de service ou des requêtes de commande (ou de pilotage) en provenance des applications clientes et les transmet à des composants spécifiques pour chaque type de requête.

**Architecture de l'API Sips Office Server :**



## L'API Sips Office Server :

Le serveur joue le double rôle d'accueil des requêtes clientes et d'hébergement des composants qui traitent celles-ci.

Par ailleurs il est doté :

- d'outils de surveillance qui détectent les éventuelles indisponibilités de service,
- d'outils de trace,
- d'outils de contrôle d'accès.

## Le serveur d'accueil :

Le serveur d'accueil est destiné à recevoir en permanence toutes les requêtes clientes. Les requêtes proviennent soit de vos applications clientes, soit de l'interface HTML d'administration.

Ses fonctions sont :

- vérifier la provenance des requêtes,
- valider le contenu (syntaxe),
- déterminer le service demandé,
- transmettre la demande au composant qui saura réellement traiter la requête,
- retourner la réponse finale à l'application cliente.

Les requêtes sont basées sur le format XML. Elles utilisent un « XML-like » proche de la norme XML. Pour simplifier l'écriture des messages, il n'y a pas d'entête XML à fournir dans la trame. Ce format permet de créer des requêtes complexes mais structurées et donc lisibles.

## Les composants :

Les composants de l'API Sips Office Server sont en charge du traitement fonctionnel des requêtes émises par le client. Ils sont appelés par le serveur d'accueil qui aura au préalable effectué les vérifications de syntaxe et de contenu.

2 types de composant :

- Les composants de *service* : ils se connectent vers les serveurs Sips pour accéder à un service distant et se charger d'analyser et retourner la réponse. Ils sont aussi chargés de sécuriser les messages échangés sur Internet avec les serveurs Sips.
- Les composants de *commande* : ils opèrent un traitement local, par exemple vérifier l'état du serveur, consulter les traces, etc... Ils sont au nombre de trois : « status », « register » et « traces ».

## Les applications clientes :

C'est ici que se situent vos développements. Vous devez concevoir les applications clientes chargées d'envoyer les requêtes au serveur d'API sous forme de messages XML et d'analyser la réponse retournée dans un format XML. L'envoi des requêtes est basé sur le protocole TCP/IP suivant le principe classique de question-réponse unique en mode sans état (il n'a pas de maintien de session). Chaque envoi de requête est précédé de l'établissement d'une connexion au serveur. L'initialisation d'une communication se fait exclusivement à l'initiative de l'application cliente via une connexion socket. Vos applications clientes peuvent être installées sur une machine différente de la machine hébergeant l'API Sips Office Server, ou bien sur la même machine.

## 4. COMMENT INVOQUER LES SERVICES DE L'API SIPS OFFICE SERVER

### 4.1 ETAPES DE MISE EN ŒUVRE D'UNE REQUETE DE SERVICE OU D'UNE REQUETE DE COMMANDE

Les services de l'API Sips Office Server sont implémentables sur vos plateformes selon un schéma toujours identique :

1. **Collecte des informations** : vous devez ici récupérer de votre système d'information (base de données, serveur web, etc...) les informations nécessaires à l'envoi de la requête.
2. **Constitution du message** : il s'agit de créer un message de type XML selon les spécifications données dans le *GUIDE DU COMPOSANT* correspondant s'il s'agit d'une requête de service, ou dans le *GUIDE D'ADMINISTRATION* s'il s'agit d'une requête de pilotage.
3. **Envoi de la requête à l'API Sips Office Server** : cette étape consiste à se connecter à l'adresse IP de la machine hébergeant l'API Sips Office Server via les fonctions de connexions réseau fournies par votre langage de programmation et à envoyer le message XML précédemment construit. Des exemples de code dans différents langages sont fournis avec le kit d'installation, et vous montrent comment se connecter au serveur, envoyer la requête et recevoir la réponse. Attention les requêtes de commande et de service sont envoyées respectivement sur le port de commande et le port de service. Les ports du serveur ont été configurés lors de l'installation, ces données sont stockées dans le fichier « config.xml » du serveur (cf. *GUIDE D'INSTALLATION*).
4. **Analyse de la réponse** : la réponse retournée par le serveur d'API est aussi au format XML. Vous devez analyser son contenu en utilisant un parseur XML, et mettre à jour votre système d'informations.

**Note** : l'API Sips Office Server n'est pas livré avec des parseurs XML. Des parseurs sont disponibles en libre téléchargement pour différents langages sur Internet.

### 4.2 LE CONTENU DES REQUETES

XML est le langage qui permet de décrire les données utiles au traitement d'une requête. Cette norme présente de nombreux avantages :

- Structure lisible
- Traitement simple par un parseur
- Utilisable dans des environnements hétérogènes
- Evolutif

Pour construire le message XML de la question, il suffit de connaître le nom du composant appelé, la fonction du composant et les paramètres éventuels de la fonction présentés sous la forme motclé="valeur".

Le message XML doit être écrit sur une seule ligne, et terminé par un retour chariot qui signalera au serveur d'accueil la fin de la requête.

**Note** : Dans les exemples suivants, les messages XML sont présentés sur plusieurs lignes pour une meilleure lisibilité. Les messages réels sont retournés sur une même ligne.

### Requête de service

Pour une requête de service (tag 'service component'), le message se présente sous cette forme :

```
<service component= "identifiant_du_composant" name= "opération">
  <opération
    paramètre1="valeur1"
    paramètre2="valeur2"
    ...
  />
</service>
```

Le message commence par le tag **service**, se termine par « **</service>** » pour marquer la fin du message. Ce tag se compose des attributs **component**, renseigné avec l'identifiant du composant, et **name** qui contient le nom de l'opération réalisée.

Le tag **operation** s'imbrique dans le tag **service** et nécessite une liste de paramètres présentés sous la forme **paramètreX="valeurX"**.

### Requête de commande

Pour une requête de commande (tag 'command component'), le message se présente sous cette forme :

```
<command component= "identifiant_du_composant" name= "opération">
  <opération
    paramètre1="valeur1"
    paramètre2="valeur2"
    ...
  />
</command>
```

### Réponse

A chaque requête, un message XML est retourné en réponse. Les réponses des requêtes de service et de commande sont structurées de la manière suivante :

```
<response component= "identifiant_du_composant" name= "opération"
  paramètre1="valeur1"
  paramètre2="valeur2"
  ...
</response>
```



### 4.3 EXEMPLE DE REQUETE GETSTATUS

Nous ne présenterons dans ce document qu'un exemple de requête « getStatus », car le composant « status » est toujours présent dans un kit API Sips Office Server. Pour les autres trames de requêtes de commande, vous pouvez vous référer au *GUIDE D'ADMINISTRATION*, et pour les requêtes de service, vous devrez vous référer au *GUIDE DU COMPOSANT* en question.

Pour l'exemple choisi, nous détaillons chacune des étapes décrites au paragraphe 4.1.

#### 1. Collecte des informations :

La requête « getStatus » décrit l'état du serveur. Elle n'attend aucun paramètre, il n'y a donc aucune information à récupérer de votre système d'information.

#### 2. Constitution du message :

La requête « getStatus » est une requête de commande, elle nécessite donc l'appel d'un composant de type « command » (tag 'command').

Le message XML de la requête se présente de la manière suivante :

```
<command component="status" name="getStatus"></command>
```

#### 3. Envoi de la requête à l'API Sips Office Server :

Vous devez programmer ici dans votre langage de programmation la connexion via une *socket* vers l'API Sips Office Server. Vous devez connaître pour ceci l'adresse IP de la machine hébergeant l'API Sips Office Server et le numéro du port de commande. Le numéro du port de commande est inscrit dans le fichier « config.xml ».

Les sources d'exemples fournis dans le kit montrent comment se connecter dans les langages suivants : Java, C, ASP, PERL et PHP.

#### 4. Analyse de la réponse :

Voici un exemple de réponse XML obtenue pour une requête getStatus :

```
<response component="status" name="getStatus">
  <getStatus
    startDate="2003/08/27"
    startTime="10:51:01"
    serverDate="2003/08/27"
    serverTime="15:29:36"
    serviceCount="9"
    commandCount="230"
    errorCount="0"
    fatalCount="0" >
    <lastRequest
      Date="2003/08/27"
      Time="13:58:24"
      Name="office / credit">
    </lastRequest>
    <lastCommand
      Date="2003/08/27"
      Time="15:29:30"
      Name="status / getStatus" >
    </lastCommand>
  </getStatus>
</response>
```

Il s'agit maintenant d'analyser la réponse reçue grâce à votre parseur XML. Le format de la réponse ne respectant pas tout à fait la norme, vous devez désactiver le contrôle de l'entête XML. Après avoir analysé et récupéré les données de la réponse, vous pouvez mettre à jour votre système d'information.

#### **4.4 DEVELOPPER VOS REQUETES DE SERVICE ET VOS REQUETES DE COMMANDE**

Si vous souhaitez développer une requête de service (ou une autre requête de commande), il vous suffit de modifier l'exemple précédent de la requête « getStatus », indiquer le numéro de port de service (ou de commande. cf. fichier config.xml), et écrire la trame XML telle que décrite dans le *GUIDE DU COMPOSANT* (ou le *GUIDE D'ADMINISTRATION*).

Des exemples de requête getStatus sont fournis avec l'API Sips Office Server. Ils se trouvent dans le sous-répertoire exemple du répertoire principal de l'API Sips Office Server. Les exemples sont fournis en langages ASP, C pour Solaris, Linux et Windows, JAVA, PERL, PHP.

## 5. RESOLUTION DES PROBLEMES RENCONTRES

Vous pouvez rencontrer au cours de vos développements des problèmes à plusieurs niveaux. Il peut s'agir de :

- Problèmes de connexion entre votre application cliente et l'API Sips Office Server. Les points à vérifier sont :
  - l'adresse IP de l'API Sips Office Server,
  - les access-list de l'API Sips Office Server (consulter le *GUIDE D'ADMINISTRATION*),
  - l'API Sips Office Server est-il bien démarré ?
  - un firewall sépare-t-il votre application cliente de l'API Sips Office Server ?
- Problèmes de connexion entre l'API Sips Office Server et le serveur Sips. Consulter le *GUIDE DU COMPOSANT*.
- Problèmes de syntaxe des messages XML. Consulter le *GUIDE DU COMPOSANT* ou le *GUIDE D'ADMINISTRATION*.