

Netcool/OMNIbus
Version 8.1

*Guide de référence de l'interface HTTP
ObjectServer*



Netcool/OMNIbus
Version 8.1

*Guide de référence de l'interface HTTP
ObjectServer*



Important

Avant d'utiliser le présent document et le produit associé, prenez connaissance des informations générales figurant à la section «Remarques», à la page 59.

Première édition - Juillet 2014

Réf. US : SC27-6268-00

LE PRESENT DOCUMENT EST LIVRE EN L'ETAT SANS AUCUNE GARANTIE EXPLICITE OU IMPLICITE. IBM DECLINE NOTAMMENT TOUTE RESPONSABILITE RELATIVE A CES INFORMATIONS EN CAS DE CONTREFAÇON AINSI QU'EN CAS DE DEFAUT D'APTITUDE A L'EXECUTION D'UN TRAVAIL DONNE.

Ce document est mis à jour périodiquement. Chaque nouvelle édition inclut les mises à jour. Les informations qui y sont fournies sont susceptibles d'être modifiées avant que les produits décrits ne deviennent eux-mêmes disponibles. En outre, il peut contenir des informations ou des références concernant certains produits, logiciels ou services non annoncés dans ce pays. Cela ne signifie cependant pas qu'ils y seront annoncés.

Pour plus de détails, pour toute demande d'ordre technique, ou pour obtenir des exemplaires de documents IBM, référez-vous aux documents d'annonce disponibles dans votre pays, ou adressez-vous à votre partenaire commercial.

Vous pouvez également consulter les serveurs Internet suivants :

- <http://www.fr.ibm.com> (serveur IBM en France)
- <http://www.ibm.com/ca/fr> (serveur IBM au Canada)
- <http://www.ibm.com> (serveur IBM aux Etats-Unis)

*Compagnie IBM France
Direction Qualité
17, avenue de l'Europe
92275 Bois-Colombes Cedex*

Cette édition s'applique à la version 8, édition 1 d'IBM Tivoli Netcool/OMNibus (numéro de produit 5724-S44) ainsi qu'à toutes les éditions et modifications suivantes, sauf indication contraire dans les nouvelles éditions.

© Copyright IBM Corporation 1994, 2014.

Table des matières

Avis aux lecteurs canadiens v

A propos de ce document vii

Public visé	vii
Publications	vii
Accessibilité	ix
Formation technique Tivoli	ix
Informations de support	x
Conventions utilisées dans la présente publication.	x
Lists des abréviations.	xiv

Chapitre 1. Présentation de l'interface HTTP ObjectServer 1

Activation des interfaces HTTP et OSLC dans le serveur ObjectServer.	1
Propriétés ObjectServer contrôlant l'interface HTTP et l'interface OSLC	2
Codage de transfert segmenté	6
Configuration avancée de l'interface HTTP du serveur ObjectServer.	6
Activation et configuration du registre de fournisseurs de services IBM JazzSM	8
Table registry.oslc.	9

Chapitre 2. URI d'interface HTTP. 11

Services de collection de tables	11
Services de collection de tables : requête GET	11
Services de collection de tables : réponse GET	12
Services de collection de tables : requête POST	13
Services de collection de tables : réponse POST	14
Services de collection de tables : requête PATCH	15
Services de collection de tables : réponse PATCH	15
Services de collection de tables : requête DELETE	17
Services de collection de tables : réponse DELETE	17
Services d'élément de ligne	18
Requête GET d'élément de ligne	22
Réponse GET d'élément de ligne	22
Requête PATCH d'élément de ligne	23
Réponse PATCH d'élément de ligne	24
Requête DELETE d'élément de ligne	25
Réponse DELETE d'élément de ligne	25
Fabrique de commandes SQL	26
Requête POST de fabrique de commandes SQL	27
Réponse POST de fabrique de commandes SQL	27
Services d'informations système	28
Requête de collection GET	29
Réponse de collection GET	29
Requête d'élément GET	30
Réponse d'élément GET	31

Chapitre 3. Comportements communs 33

Support HTTP et HTTPS	33
Codes de réponse HTTP	33

Paramètres de requête	34
Mécanismes d'authentification	35
Charge de message JSON de réussite	36
Charge de message d'erreur JSON	36
Chiffrement de message	37
Types MIME Accept	37
Types MIME Content	37
Mise en cache des réponses	37

Annexe A. Exemples de charges de message JSON 39

Exemple d'ensemble de lignes JSON : GET	39
Exemple d'ensemble de lignes JSON : PATCH	40
Exemple d'ensemble de lignes JSON : POST	41
Exemple de fabrique de commandes SQL : POST	42
Exemple de message de réussite JSON	42
Exemple de message d'erreur JSON	42

Annexe B. Exemples de demandes et réponses HTTP 43

Exemple de requête POST de collection de tables	43
Exemple de réponse POST de collection de tables	44
Exemple de requête GET de collection de tables	44
Exemple de réponse GET de collection de tables	44
Exemple de requête PATCH de collection de tables	46
Exemple de réponse PATCH de collection de tables	46
Exemple de requête DELETE de collection de tables	47
Exemple de réponse DELETE de collection de tables	47
Exemple de requête GET d'élément de ligne via RowSerial	47
Exemple de réponse GET d'élément de ligne via RowSerial	47
Exemple de requête GET d'élément de ligne via KeyField	49
Exemple de réponse GET d'élément de ligne via une zone clé	49
Exemple de requête PATCH d'élément de ligne	51
Exemple de réponse PATCH d'élément de ligne	52
Exemple de requête DELETE d'élément de ligne	52
Exemple de réponse DELETE d'élément de ligne	53
Exemple de requête POST de fabrique de commandes SQL	53
Exemple de réponse POST de fabrique de commandes SQL	53
Exemple de requête GET d'informations système	53
Exemple de réponse GET d'informations système.	54
Exemple de requête GET d'élément d'informations système.	54
Exemple de réponse GET d'élément d'informations système.	54
Fichier de configuration JSON avec paramètres de type MIME et en-têtes HTTP	55

Annexe C. Lists des abréviations . . . 57

Remarques 59

Marques 62

Avis aux lecteurs canadiens

Le présent document a été traduit en France. Voici les principales différences et particularités dont vous devez tenir compte.

Illustrations

Les illustrations sont fournies à titre d'exemple. Certaines peuvent contenir des données propres à la France.

Terminologie

La terminologie des titres IBM peut différer d'un pays à l'autre. Reportez-vous au tableau ci-dessous, au besoin.

IBM France	IBM Canada
ingénieur commercial	représentant
agence commerciale	succursale
ingénieur technico-commercial	informaticien
inspecteur	technicien du matériel

Claviers

Les lettres sont disposées différemment : le clavier français est de type AZERTY, et le clavier français-canadien de type QWERTY.








OS/2 et Windows - Paramètres canadiens

Au Canada, on utilise :

- les pages de codes 850 (multilingue) et 863 (français-canadien),
- le code pays 002,
- le code clavier CF.

Nomenclature

Les touches présentées dans le tableau d'équivalence suivant sont libellées différemment selon qu'il s'agit du clavier de la France, du clavier du Canada ou du clavier des États-Unis. Reportez-vous à ce tableau pour faire correspondre les touches françaises figurant dans le présent document aux touches de votre clavier.

France	Canada	Etats-Unis
 (Pos1)		Home
Fin	Fin	End
 (PgAr)		PgUp
 (PgAv)		PgDn
Inser	Inser	Ins
Suppr	Suppr	Del
Echap	Echap	Esc
Attn	Intrp	Break
Impr écran	ImpEc	PrtSc
Verr num	Num	Num Lock
Arrêt défil	Défil	Scroll Lock
 (Verr maj)	FixMaj	Caps Lock
AltGr	AltCar	Alt (à droite)

Brevets

Il est possible qu'IBM détienne des brevets ou qu'elle ait déposé des demandes de brevets portant sur certains sujets abordés dans ce document. Le fait qu'IBM vous fournisse le présent document ne signifie pas qu'elle vous accorde un permis d'utilisation de ces brevets. Vous pouvez envoyer, par écrit, vos demandes de renseignements relatives aux permis d'utilisation au directeur général des relations commerciales d'IBM, 3600 Steeles Avenue East, Markham, Ontario, L3R 9Z7.

Assistance téléphonique

Si vous avez besoin d'assistance ou si vous voulez commander du matériel, des logiciels et des publications IBM, contactez IBM direct au 1 800 465-1234.

A propos de ce document

Tivoli Netcool/OMNIBus est un système de gestion du niveau de service (SLM) qui offre une surveillance centralisée et en temps réel des réseaux complexes et des domaines informatiques.

Le *Guide de référence de l'interface HTTP ObjectServer d'IBM Tivoli Netcool/OMNIBus* décrit les URI et les comportements communs de l'interface de programme d'application (API) appelée interface HTTP ObjectServer. Il décrit également comment activer cette API et fournit des exemples de charges de message JSON et de demandes et réponses HTTP.

Public visé

La présente publication s'adresse aux administrateurs chargés de configurer Tivoli Netcool/OMNIBus.

Publications

Cette section répertorie les publications de la bibliothèque Tivoli Netcool/OMNIBus et les documents associés. Elle indique également comment accéder aux publications Tivoli en ligne et comment organiser les publications Tivoli.

Votre bibliothèque Tivoli Netcool/OMNIBus

Les documents suivants sont disponibles dans la bibliothèque Tivoli Netcool/OMNIBus :

- *Guide d'installation et de déploiement d'IBM Tivoli Netcool/OMNIBus, SC11-7236*
Comprend des procédures d'installation et de mise à niveau de Tivoli Netcool/OMNIBus et décrit la manière de configurer la sécurité et les communications des composants. La publication contient également des exemples d'architectures Tivoli Netcool/OMNIBus et explique comment les mettre en oeuvre.
- *Guide d'administration d'IBM Tivoli Netcool/OMNIBus, SC11-7237*
Explique comment effectuer les tâches d'administration à l'aide de l'interface graphique d'administration de Tivoli Netcool/OMNIBus, des outils de ligne de commande et des commandes de processus. La publication contient également des descriptions et des exemples de syntaxe SQL du serveur ObjectServer et des automatisations.
- *Guide d'administration et d'utilisation de l'interface graphique Web d'IBM Tivoli Netcool/OMNIBus, SC11-7238*
Décrit comment exécuter les tâches d'administration et de visualisation d'événements à l'aide de l'interface graphique Web de Tivoli Netcool/OMNIBus.
- *IBM Tivoli Netcool/OMNIBus User's Guide, SC11-7243*
Présente les outils du bureau et décrit les tâches de l'utilisateur liées à la gestion d'événements à l'aide de ces outils.
- *Guide des sondes et des passerelles d'IBM Tivoli Netcool/OMNIBus, SC11-7239*

Contient des informations de présentation et de référence sur les sondes et les passerelles, notamment la syntaxe des fichiers de règles de sonde et des commandes de passerelle.

- *IBM Tivoli Monitoring for Tivoli Netcool/OMNIbus Agent User's Guide*, SC11-7303

Explique comment installer l'agent de moniteur d'état de Tivoli Netcool/OMNIbus et contient des informations de référence sur l'agent.

- *Guide de référence d'IBM Tivoli Netcool/OMNIbus Event Integration Facility*, SC11-7241

Décrit comment développer des adaptateurs d'événements adaptés à votre environnement réseau et aux besoins spécifiques de votre entreprise. Cette publication décrit également comment filtrer des événements à la source.

- *Guide des messages d'erreur d'IBM Tivoli Netcool/OMNIbus*, SC11-7242

Décrit les messages système dans Tivoli Netcool/OMNIbus et les réponses à apporter à ces messages.

- *Guide d'utilisation de l'API d'administration de l'interface graphique Web (WAAPI) d'IBM Tivoli Netcool/OMNIbus*, SC11-7244

Montre comment administrer l'interface graphique Web Tivoli Netcool/OMNIbus à l'aide de l'interface de programmation d'application XML nommée WAAPI

- *Guide de référence de l'interface HTTP ObjectServer d'IBM Tivoli Netcool/OMNIbus*, SC11-7345
Décrit les URI et les comportements communs de l'interface de programme d'application (API) appelée interface HTTP ObjectServer. Décrit comment activer cette API et fournit des exemples de charges de message JSON et de demandes et réponses HTTP.

- *Guide de référence de l'interface OSLC ObjectServer d'IBM Tivoli Netcool/OMNIbus*, SC11-7345
Décrit les services, les ressources et les comportements communs de l'interface de programme d'application (API) OSLC (Open Services for Lifecycle Collaboration), appelée interface OSLC ObjectServer. Décrit comment activer cette API et fournit des exemples de définitions de fournisseur de services, de charges de message RDF/XML et des demandes et réponses HTTP.

Si vous utilisez d'autres produits IBM pour étendre les fonctionnalités de Tivoli Netcool/OMNIbus, tels que DB2, IBM Tivoli Monitoring ou Tivoli Common Reporting, consultez le centre de documentation du produit concerné, afin d'obtenir les publications appropriées.

Accès à la terminologie en ligne

Le site Web de terminologie IBM regroupe la terminologie des bibliothèques de logiciels IBM en un seul emplacement, pour des raisons pratiques. Ce site est accessible à l'adresse suivante :

<http://www.ibm.com/software/globalization/terminology>

Accès en ligne aux publications

IBM met en ligne les publications pour tous les produits Tivoli, dès leur parution ou leur mise à jour, sur le site Web du centre de documentation Tivoli à l'adresse :

<http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/index.jsp>

Remarque : Si vous imprimez des documents PDF sur un papier qui n'est pas au format Letter, définissez l'option de la fenêtre **Fichier > Imprimer** qui permet à Adobe Reader d'imprimer des pages au format Letter sur le format de papier que vous utilisez.

Commande de publications

Vous pouvez commander de nombreuses publications Tivoli en ligne sur le site Web suivant :

<http://www.ibm.com/e-business/linkweb/publications/servlet/pbi.wss>

Vous pouvez également passer votre commande par téléphone en composant l'un des numéros suivants :

- Aux Etats-Unis : 800-879-2755
- Au Canada : 800-426-4968

Pour les autres pays, contactez le représentant de compte logiciel pour commander des publications Tivoli. Pour connaître le numéro de téléphone de votre représentant local, procédez comme suit :

1. Accédez au site Web suivant :
<http://www.ibm.com/e-business/linkweb/publications/servlet/pbi.wss>
2. Sélectionnez votre pays dans la liste et cliquez sur **Go**. La page d'accueil du centre de documentation IBM s'affiche pour votre pays.
3. Sur le côté gauche de la page, cliquez sur **A propos de ce site** pour afficher la page d'informations qui comporte le numéro de téléphone de votre représentant local.

Accessibilité

Les fonctions d'accessibilité permettent aux utilisateurs présentant un handicap, par exemple les personnes à mobilité réduite ou à déficience visuelle, d'utiliser des logiciels.

Avec ce produit, vous pouvez utiliser les technologies d'assistance pour parcourir l'interface à l'aide de messages sonores. Vous pouvez également utiliser le clavier à la place de la souris pour faire fonctionner certaines fonctionnalités de l'interface graphique.

Formation technique Tivoli

Pour des informations sur la formation technique Tivoli, consultez le site Web IBM Tivoli Education :

<http://www.ibm.com/software/tivoli/education>

Informations de support

Si vous rencontrez un problème avec le logiciel IBM, vous pouvez le résoudre rapidement. IBM vous propose les solutions suivantes pour obtenir le support dont vous avez besoin :

En ligne

Accédez au site service de support logiciel IBM à l'adresse <http://www.ibm.com/software/support/probsub.html>, puis suivez les instructions.

IBM Support Assistant

IBM Support Assistant (ISA) est un plan de travail de maintenabilité logicielle gratuit qui vous aide à résoudre les problèmes liés aux applications logicielles IBM. ISA permet d'accéder rapidement aux informations de support et aux outils de maintenabilité pour identifier les problèmes. Pour installer le logiciel ISA, rendez-vous à l'adresse <http://www.ibm.com/software/support/isa>.

Documentation

Si vous avez une suggestion pour améliorer le contenu ou l'organisation de ce guide, envoyez-la à l'équipe de Tivoli Netcool/OMNIbus Information Development, à l'adresse :

<mailto://L3MMDOCS@uk.ibm.com>

Conventions utilisées dans la présente publication

Cette publication utilise plusieurs conventions pour les dispositions particulières et les actions spéciales ainsi que pour les commandes et les chemins d'accès liés au système d'exploitation.

Variables et chemins d'accès liés au système d'exploitation

Cette publication utilise la convention UNIX pour la définition des variables d'environnement et la notation des répertoires.

Lorsque vous utilisez la ligne de commande Windows, remplacez *\$variable* par *%variable%* pour les variables d'environnement. De la même façon, remplacez chaque barre oblique (/) par une barre oblique inversée (\) dans les chemins de répertoire. Par exemple, sur les systèmes UNIX, la variable d'environnement \$NCHOME désigne le chemin du répertoire de base de Netcool. Sur les systèmes Windows, la variable d'environnement %NCHOME% désigne le chemin d'accès au répertoire de base de Netcool. Les noms de variables d'environnement ne sont pas toujours identiques dans les environnements Windows et UNIX. Par exemple, dans les environnements Windows, %TEMP% est l'équivalent de \$TMPDIR dans les environnements UNIX.

Si vous utilisez l'interpréteur de commandes shell sur un système Windows, vous pouvez utiliser les conventions UNIX.

Emplacement d'origine de Netcool

L'emplacement d'origine de Netcool est le répertoire de base dans lequel Tivoli Netcool/OMNIbus est installé. L'emplacement d'origine de Netcool est défini par la variable d'environnement NCHOME. La valeur de NCHOME est la suivante :

- **UNIX** **Linux** \$NCHOME a comme valeur par défaut /opt/IBM/tivoli/netcool
- **Windows** %NCHOME% a comme valeur par défaut C:\IBM\tivoli\Netcool

Si un chemin de répertoire ou de commande commence par la variable *NCHOME*, cette information s'applique a tous les systèmes d'exploitation pris en charge.

Les autres produits qui utilisent la variable d'environnement *NCHOME*, par exemple IBM Tivoli Network Manager IP Edition, peuvent être installés à l'emplacement d'origine de Netcool. Chaque produit installe ses composants et fichiers spécifiques dans un sous-répertoire de produits dédié dans l'emplacement d'origine de Netcool. Les fichiers qui sont communs à tous les produits sont installés dans des sous-répertoires partagés à l'emplacement d'origine de Netcool.

Noms de répertoire spécifiques au système d'exploitation

Lorsque les fichiers Tivoli Netcool/OMNIbus sont identifiés comme se trouvant dans un répertoire *arch* sous *NCHOME*, *arch* est une variable qui représente le répertoire de votre système d'exploitation, comme indiqué dans le tableau suivant.

Tableau 1. Noms de répertoire pour la variable *arch*

Nom de répertoire représenté par <i>arch</i>	Système d'exploitation
aix5	Systèmes AIX
hpux11hpa	Systèmes HP-UX Itanium
linux2x86	Systèmes Red Hat Linux et SUSE
linux2s390	Linux for System z
solaris2	Systèmes Solaris
win32	Systèmes Windows

Emplacement OMNIHOME

Les sondes et les passerelles ainsi que les versions antérieures de Tivoli Netcool/OMNIbus utilisent la variable d'environnement *OMNIHOME* dans de nombreux fichiers de configuration. Définissez la valeur de *OMNIHOME* comme suit :

- **UNIX** **Linux** Définissez \$OMNIHOME à \$NCHOME/omnibus
- **Windows** Définissez %OMNIHOME% à %NCHOME%\omnibus

Répertoire de base de l'interface graphique Web

REP_INSTALL_WEBGUI

Fait référence au répertoire où l'interface graphique Web est installée. Ce répertoire est connu comme le répertoire de base de l'interface graphique Web. Les valeurs par défaut sont les suivantes :

UNIX **Linux** /opt/IBM/tivoli/netcool/omnibus_webgui

Windows C:\IBM\tivoli\netcool\omnibus_webgui

Le répertoire de base de l'interface graphique Web est distinct des répertoires de base de Jazz for Service Management.

Répertoire de base de Jazz for Service Management

REP_INSTALL_WAS

Fait référence à l'emplacement où WebSphere Application Server est installé. Cet emplacement peut être spécifié lors de l'installation. Les valeurs par défaut sont les suivantes :

UNIX **Linux** /opt/IBM/WebSphere/AppServer
Windows C:\Program Files\IBM\WebSphere\AppServer

REP_INSTALL_JazzSM

Fait référence à l'emplacement où Jazz for Service Management est installé. Cet emplacement peut être spécifié lors de l'installation. Les valeurs par défaut sont les suivantes :

UNIX **Linux** /opt/IBM/JazzSM
Windows C:\Program Files\IBM\JazzSM

REP_INSTALL_JazzSM

Fait référence à l'emplacement du profil du serveur d'applications qui est utilisé pour Jazz for Service Management. Cet emplacement est dans le sous-répertoire /profile du répertoire de base de Jazz for Service Management :

UNIX **Linux** *REP_INSTALL_JazzSM/profile*
Windows *REP_INSTALL_JazzSM\profile*

REP_INSTALL_DASH

Fait référence à l'emplacement où Concentrateur des services d'application du tableau de bord est installé. Cet emplacement peut être spécifié lors de l'installation. Les valeurs par défaut sont les suivantes :

UNIX **Linux** /opt/IBM/JazzSM/ui
Windows C:\Program Files\IBM\JazzSM\ui

Pour d'autres installations de Jazz for Service Management, voir le centre de documentation de Jazz for Service Management à l'adresse <http://www-01.ibm.com/support/knowledgecenter/SSEKCU/welcome>.

Conventions des graphiques de repérage

La documentation Tivoli Netcool/OMNIBus contient des graphiques de repérage pour indiquer que des parties d'une rubrique ou d'une instruction s'appliquent seulement sous certaines conditions. Le tableau suivant décrit la signification de chaque graphique de repérage.

Tableau 2. Graphiques de repérage de Tivoli Netcool/OMNIBus

Graphique	Description
Web GUI	Le texte ou l'instruction s'applique seulement au composant interface graphique Web.
UNIX	Le texte ou l'instruction s'applique seulement aux systèmes d'exploitation UNIX, incluant AIX, HP-UX et Solaris. Les graphiques de repérage suivants sont utilisés pour AIX, HP-UX et Solaris : AIX HP-UX Solaris
Linux	Le texte ou l'instruction s'applique seulement aux systèmes d'exploitation Linux.

Tableau 2. Graphiques de repérage de Tivoli Netcool/OMNIBus (suite)

Graphique	Description
Windows	Le texte ou l'instruction s'applique seulement aux systèmes d'exploitation Windows.
32-bit	Le texte ou l'instruction s'applique seulement aux systèmes d'exploitation 32 bits.
64-bit	Le texte ou l'instruction s'applique seulement aux systèmes d'exploitation 64 bits.
FIPS 140-2	Le texte ou l'instruction s'applique seulement à l'utilisation ou à la configuration du chiffrement FIPS 140-2.
Default	Le texte ou l'instruction décrit le comportement par défaut ou s'applique seulement à des configurations par défaut.
Fix Pack 1	<p>Le texte ou l'instruction s'applique seulement au numéro du groupe de correctifs indiqué par le graphique. Les fonctions ou les améliorations décrites dans le texte sont disponibles seulement après installation du groupe de correctifs indiqué.</p> <p>Remarque : Les groupes de correctifs pour le composant serveur et pour le composant interface graphique Web sont publiés séparément. Les groupes de correctifs pour l'interface graphique Web sont indiqués comme suit :</p> <p>Web GUI Fix Pack 1</p>
Administrator	Le texte ou l'instruction s'applique seulement aux administrateurs de l'interface graphique Web, c'est-à-dire aux utilisateurs auxquels les rôles ncw_user et ncw_admin ont été affectés.
C	Le texte ou l'instruction s'applique seulement au langage de programmation C.
Java	Le texte ou l'instruction s'applique seulement au langage de programmation Java™.

Conventions typographiques

Cette publication utilise les conventions typographiques suivantes :

Gras

- Commandes en minuscules et commandes à casse mixte pour mieux les distinguer dans le texte
- Contrôles d'interface (cases à cocher, boutons de commande, boutons radio, flèches d'incrément, zones, dossiers, icônes, zones de liste, éléments de zone de liste, listes à plusieurs colonnes, conteneurs, options de menu, noms de menu, onglets, feuilles de propriétés), libellés (tels que **Conseil :** et **Remarques relatives au système d'exploitation :**)
- Mots clés et paramètres dans le texte

Italique

- Citations (exemples : titres de publications, disquettes et CD)
- Mots définis dans le texte (exemple : une ligne spécialisée est appelée ligne *point-à-point*)
- Mise en évidence de mots et de lettres (exemple de mot en tant que mot : "Utilisez le mot *que* pour introduire une clause restrictive." ; exemple de lettre en tant que lettre : "L'adresse de numéro d'unité logique doit commencer par la lettre *L*.")

- Nouveaux termes rencontrés dans un texte (sauf dans une liste de définitions) : une *vue* est un cadre situé dans un espace de travail qui contient des données
- Variables et valeurs que vous devez fournir : ... où *nom* représente....

Espacement fixe

- Exemples et extraits de code
- Noms de fichiers, mots clés de programmation et autres éléments qu'il est difficile de distinguer dans du texte
- Texte de message et invites adressées à l'utilisateur
- Texte que l'utilisateur doit entrer
- Valeurs d'arguments ou d'options de commande

Lists des abréviations

La documentation d'API de l'interface HTTP ObjectServer et de l'interface OSLC ObjectServer utilise les abréviations et termes suivants.

HTTP Hyper Text Transfer Protocol. HTTP version 1.1 est défini dans RFC2616. Sauf indication contraire, le terme HTTP dans le présent document désigne à la fois HTTP et HTTPS.

HTTPS

Hyper Text Transfer Protocol Secure, défini dans RFC2818.

JazzSM

Jazz for Service Management, disponible à l'adresse <https://www.ibm.com/developerworks/mydeveloperworks/groups/service/html/communityview?communityUuid=69ec672c-dd6b-443d-add8-bb9a9a490eba>.

JSON JavaScript Object Notation, défini dans ECMA-262.

MIME Multipurpose Internet Mail Extensions. Les types de support MIME sont définis dans *IANA MIME Media Types*.

OSLC Open Services for Lifecycle Collaboration, défini dans <http://open-services.net>.

REST Representational State Transfer, décrit initialement de façon informelle dans le document *Architectural Styles and the Design of Network-based Software Architectures*.

URI Uniform Resource Identifier, défini dans RFC3986.

XML eXtensible Markup Language, défini par W3C.

Chapitre 1. Présentation de l'interface HTTP ObjectServer

L'interface HTTP est une interface de programme d'application (API) légère, hébergée par l'ObjectServer. L'interface HTTP permet d'accéder aux données de table sur le serveur ObjectServer avec un format d'URI structuré qui utilise le protocole HTTP. Les demandes POST, PATCH, GET et DELETE sont prises en charge. Les demandes sont prises en charge sur les identificateurs URI de table ou de ligne. L'accès à l'URI est authentifié par un utilisateur ObjectServer connu, à l'aide de l'authentification HTTP de base. L'interface peut être sécurisée via une connexion HTTPS. Vous pouvez activer l'interface en définissant ses propriétés dans l'ObjectServer.

L'ObjectServer héberge une autre API appelée interface OSLC. Cette API est un fournisseur de serveur d'événements qui présente les événements dans une vue de données liées aux ressources ainsi que le journal associé et le détail des ressources. Pour plus d'informations sur l'interface OSLC, voir le manuel *Guide de référence de l'interface OSLC ObjectServer d'IBM Tivoli Netcool/OMNIBus*.

URI de base

L'URI de base de l'interface HTTP est le suivant.

`http://hôte:port/objectserver/restapi/`

Tâches associées:

«Activation des interfaces HTTP et OSLC dans le serveur ObjectServer»

Les interfaces HTTP et OSLC ObjectServer sont désactivées par défaut, car elles doivent être configurées pour un paramétrage sécurisé.

Activation des interfaces HTTP et OSLC dans le serveur ObjectServer

Les interfaces HTTP et OSLC ObjectServer sont désactivées par défaut, car elles doivent être configurées pour un paramétrage sécurisé.

Avant de commencer

Déterminez quels serveurs ObjectServer dans votre environnement doivent être accessibles via HTTP ou HTTPS. Certains serveurs ObjectServer dans un environnement ne nécessitent pas l'octroi de l'accès aux données ObjectServer via un mécanisme HTTP.

Pourquoi et quand exécuter cette tâche

Etant donné que l'hébergement des interfaces HTTP et OSLC dans l'ObjectServer nécessite un serveur HTTP intégré, l'ObjectServer peut distribuer des fichiers aux clients HTTP. Bien que l'ObjectServer puisse traiter des pages, il n'est pas optimisé pour cela, contrairement à un serveur Web Apache. Par conséquent, n'utilisez pas l'ObjectServer pour héberger autre chose que des pages HTML ou JavaScript rudimentaires.

Procédure

1. Pour activer les interfaces, définissez la propriété **NRestOS.Enable** sur TRUE.

2. Pour configurer le serveur HTTP intégré de sorte que les interfaces soient actives sur un port HTTP, spécifiez le port d'écoute pour le type de connexion. Par exemple, pour que les interfaces écoutent le port 8080, définissez les propriétés comme suit :


```
NHttpd.EnableHTTP : TRUE
NHttpd.ListeningPort : 8080
```
3. Si vous souhaitez que les interfaces soient actives sur le port HTTPS 9090, définissez les propriétés comme indiqué dans l'exemple suivant. Étant donné qu'un port HTTPS est chiffré SSL, un fichier certificat contenant un certificat approprié doit être créé et protégé par un mot de passe.


```
NHttpd.SSLEnable : TRUE
NHttpd.SSLListeningPort : 9090
NHttpd.SSLCertificate : "libellé_certificat"
NHttpd.SSLCertificatePwd : "mot_de_passe"
```
4. Pour permettre la gestion de fichiers par l'ObjectServer, définissez la propriété **NHttpd.EnableFileServing**. La racine des pages gérées est définie par la propriété **NHttpd.DocumentRoot**.
5. Pour générer la liste des références de ressources membres dans le contenu RDF/XML des réponses de la fonction de requête d'événement, de journal et de détail dans les deux instances de ressource **Collection** et **ResponseInfo**, définissez **NRestOS.OSLCRDFMsgFormat** sur «MIGRATION». Pour plus d'informations sur ce paramètre et pour savoir pourquoi vous devez le définir, consultez la section *Mises à jour de l'interface HTTP et de l'interface OSLC* dans les notes sur l'édition.

Concepts associés:

Chapitre 1, «Présentation de l'interface HTTP ObjectServer», à la page 1
 L'interface HTTP est une interface de programme d'application (API) légère, hébergée par l'ObjectServer. L'interface HTTP permet d'accéder aux données de table sur le serveur ObjectServer avec un format d'URI structuré qui utilise le protocole HTTP. Les demandes POST, PATCH, GET et DELETE sont prises en charge. Les demandes sont prises en charge sur les identificateurs URI de table ou de ligne. L'accès à l'URI est authentifié par un utilisateur ObjectServer connu, à l'aide de l'authentification HTTP de base. L'interface peut être sécurisée via une connexion HTTPS. Vous pouvez activer l'interface en définissant ses propriétés dans l'ObjectServer.

Référence associée:

«Codage de transfert segmenté», à la page 6
 Pour mieux contrôler les grands ensembles de résultats, les charges de réponse JSON et RDF/XML peuvent être renvoyés sous forme de flux au client en tant que segments à l'aide du codage de transfert segmenté.

Propriétés ObjectServer contrôlant l'interface HTTP et l'interface OSLC

Propriétés ObjectServer contrôlant les interfaces HTTP et OSLC.

Le tableau suivant répertorie les propriétés ObjectServer qui contrôlent l'interface HTTP et l'interface OSLC.

Tableau 3. Propriétés et options de ligne de commande contrôlant l'interface HTTP et l'interface OSLC

Propriété	Option de ligne de commande	Description
NRestOS.Enable TRUE FALSE	-nrestosenable TRUE FALSE	Active l'interface HTTP et l'interface OSLC pour le serveur ObjectServer. La valeur par défaut est FALSE, c'est-à-dire que les interfaces sont désactivées.
NRestOS.OSLCResource ConfigFile chaîne	-nrestososlrescfg chaîne	Chemin d'accès au fichier de configuration OSLC. Ce fichier JSON définit la façon dont les colonnes du schéma ObjectServer sont mappées avec des propriétés du domaine des événements. Le chemin d'accès par défaut est \$OMNIHOME/etc/restos/resourcecfg.json.

Le tableau suivant répertorie les propriétés ObjectServer qui contrôlent le serveur HTTP intégré.

Tableau 4. Propriétés et options de ligne de commande contrôlant le serveur HTTP intégré

Propriété	Option de ligne de commande	Description
NHttpd.AccessLog chaîne	-nhttpd_accesslog chaîne	Indique le nom et l'emplacement du fichier journal dans lequel le serveur consigne toutes les demandes qu'il traite. Le fichier par défaut est \$OMNIHOME/log/NCOMS_http_access.log.
NHttpd.Authentication Domain chaîne	-nhttpd_authdomain chaîne	Indique le domaine d'authentification utilisé lors de la demande des détails d'authentification sur la connexion HTTP ou HTTPS. La valeur par défaut est omnibus.

Tableau 4. Propriétés et options de ligne de commande contrôlant le serveur HTTP intégré (suite)

Propriété	Option de ligne de commande	Description
NHttpd.ConfigFile chaîne	-nhttpd_configfile chaîne	Indique le chemin d'accès au fichier de configuration JSON. La valeur par défaut est \$OMNIHOME/etc/libnhttpd.json qui active les paramètres mimeType et les en-têtes HTTP dans les fichiers de réponse HTTP. Pour plus d'informations sur le fichier de configuration libnhttpd.json, voir «Configuration avancée de l'interface HTTP du serveur ObjectServer», à la page 6.
NHttpd.DocumentRoot chaîne	-nhttpd_docroot chaîne	Indique la racine de document du service Web imbriqué. La valeur par défaut est \$OMNIHOME/etc/restos/docroot.
NHttpd.EnableFileServing TRUE FALSE	-nhttpd_enablefs TRUE FALSE	Cette propriété permet d'activer la prise en charge de fichier par défaut par l'ObjectServer. Cela permet à l'ObjectServer de faire office de serveur HTTP simple prenant en charge les fichiers du système de fichiers local. La valeur par défaut est FALSE.
NHttpd.ExpireTimeout non signé	-nhttpd_exptimeout non signé	Indique la durée d'inactivité, en secondes, d'une connexion HTTP 1.1 avant qu'elle soit supprimée. La valeur par défaut est 15.
NHttpd.ListeningHostname chaîne	-nhttpd_hostname chaîne	Indique le nom d'hôte ou l'adresse IP d'écoute pouvant être utilisée comme partie nom d'hôte d'un URI de l'interface HTTP ou HTTP de l'ObjectServer. La valeur par défaut est localhost.

Tableau 4. Propriétés et options de ligne de commande contrôlant le serveur HTTP intégré (suite)

Propriété	Option de ligne de commande	Description
NHttpd.SSLListeningPort <i>entier</i>	-nhttpd_sslport <i>entier</i>	Indique le port sur lequel le serveur ObjectServer écoute les demandes HTTPS. La valeur par défaut est 0.
NHttpd.SSLCertificate <i>entier</i>	-nhttpd_sslcert <i>entier</i>	Indique le nom du certificat SSL du serveur. La valeur par défaut est ''.
NHttpd.SSLCertificatePwd <i>chaîne</i>	-nhttpd_sslcertpwd <i>chaîne</i>	Remarque : Cette propriété est obsolète depuis Tivoli Netcool/OMNIBus V8.1. La définition de cette propriété est sans effet.
NHttpd.SSLEnable TRUE FALSE	-nhttpd_sslenable TRUE FALSE	Permet l'utilisation du support SSL. La valeur par défaut est FALSE.
NRestOS.OSLCRDFMsgFormat <i>chaîne</i>	nrestososlcrmf <i>chaîne</i>	Définissez cette propriété sur la chaîne MIGRATION pour redévelopper les utilitaires basés sur l'interface OSLC de l'ObjectServer afin que la liste de références des ressources membres soit générée dans une instance de ressource Collection plutôt que dans une instance de ressource ResponseInfo dans le contenu RDF/XML de la fonction de requête d'événement, de détail et de journal. La valeur MIGRATION signifie que la liste de références des ressources membres est générée à la fois dans une instance de ressource Collection et dans une instance de ressource ResponseInfo . Redéveloppez vos utilitaires OSLC pour générer la liste de références des ressources membres uniquement dans l'instance de ressource Collection . Une fois supprimé le code qui génère la liste dans l'instance de ressource ResponseInfo , réinitialisez cette propriété.

Pour plus d'informations sur les propriétés et les options de ligne de commande de l'ObjectServer, voir le *Guide d'administration d'IBM Tivoli Netcool/OMNIBus*.

Codage de transfert segmenté

Pour mieux contrôler les grands ensembles de résultats, les charges de réponse JSON et RDF/XML peuvent être renvoyés sous forme de flux au client en tant que segments à l'aide du codage de transfert segmenté.

Vous pouvez activer ou désactiver la segmentation dans une section chunking du fichier JSON de configuration serveur `$OMNIHOME/etc/restos/servercfg.json`. Vous pouvez aussi définir la taille des segments et le nombre de lignes à partir duquel un ensemble de résultats doit être considéré comme suffisamment grand pour déclencher la segmentation. Les valeurs par défaut sont les suivantes :

```
{
  "chunking" : {
    "enabled" : true,
    "rowSetSize" : 50,
    "size" : 52428
  }
}
```

Les valeurs de segmentation disponibles sont les suivantes :

- **enabled** : Activer ou désactiver la fonction de segmentation. La valeur par défaut est True.
- **rowSetSize** : Le nombre de lignes de l'ensemble de résultats à partir duquel la segmentation doit être déclenchée. La valeur par défaut est 50.
- **size** : La taille de segment à utiliser. La valeur par défaut est 50000 en base 2.

Tâches associées:

«Activation des interfaces HTTP et OSLC dans le serveur ObjectServer», à la page 1

Les interfaces HTTP et OSLC ObjectServer sont désactivées par défaut, car elles doivent être configurées pour un paramétrage sécurisé.

Configuration avancée de l'interface HTTP du serveur ObjectServer

Utilisez le fichier de configuration `libnhttpd.json` pour fournir des données de configuration supplémentaires à la bibliothèque de sockets HTTP imbriqués, pour définir des éléments statiques définis par l'utilisateur et pour mapper des types MIME à des extensions de fichier.

Les propriétés de la section `httpPort` du fichier de configuration JSON de bibliothèque `libnhttpd $OMNIHOME/etc/libnhttpd.json` définissent le nombre d'utilisateurs qui peuvent être connectés en même temps.

Le fichier de configuration `libnhttpd.json` est édité pour définir des paramètres de type MIME et des en-têtes HTTP dans des réponses HTTP qui sont renvoyées par l'interface HTTP et par l'interface OSLC. Pour activer les paramètres de type MIME et les en-têtes HTTP, activez la propriété **NHttpd.ConfigFile**.

```
{
  "_comment" : "Ce fichier fournit des données de configuration supplémentaires
à la bibliothèque de sockets HTTP imbriquée (libnhttpd).",
  "httpPort" : {
    "_comment" : "Cette section définit des paramètres opérationnels
supplémentaires pour tous les ports HTTP qui sont actifs.",
    "maxConnections" : 500
  }
}
```

```

    },
    "httpRequest" : {
      "_comment" : "Cette section définit des paramètres opérationnels
relatifs aux
demandes HTTP reçues des clients.",
      "maxContentLength" : 1048576
    },
    },
    "httpResponse" : {
      "_comment" : "Cette section définit un ensemble d'éléments statiques
définis par l'utilisateur qui
doivent être renvoyés
dans une réponse HTTP, telle que des en-têtes HTTP.",
      "corsHeaders" : [
        {
          "name" : "Access-Control-Allow-Headers",
          "value" : "Location"
        },
        {
          "name" : "Access-Control-Expose-Headers",
          "value" : "Location"
        }
      ],
      "httpHeaders" : [
      ]
    },
    },
    "mimeTypes" : {
      "_comment" : "Cette section mappe des types MIME sur des extensions de fichier.
Elle est utilisée par
libnhttpd pour déterminer le type MIME d'un fichier
qui doit être géré depuis son URI de gestion de fichier.",
      "application/json" : [
        "json"
      ],
      "application/rdf+xml" : [
        "rdf"
      ],
      "application/xslt+xml" : [
        "xsl", "xslt"
      ],
      "image/jpeg" : [
        "jpg", "jpeg"
      ],
      "image/gif" : [
        "gif"
      ],
      "image/png" : [
        "png"
      ],
      "text/css" : [
        "css"
      ],
      "text/javascript" : [
        "js"
      ],
      "text/HTML" : [
        "htm", "html"
      ],
      "text/plain" : [
        "txt", "log"
      ],
      "text/xml" : [
        "xml"
      ]
    ]
  }
}

```

Les éléments suivants sont les valeurs pour la connexion et les données :

maxConnections

Définit le nombre maximal de connexions autorisées à un même moment. La valeur par défaut est 500.

maxContentLength

Définit la quantité maximale de données qui peut être envoyée au serveur dans une demande HTTP. Si cette valeur est dépassée, la demande se termine par une erreur HTTP 413 (entité de la demande trop grande). La valeur par défaut est 1048576.

Les sections qui activent les paramètres de type MIME et les en-têtes HTTP sont les suivantes :

httpResponse

Définit les en-têtes HTTP se trouvant dans les réponses HTTP qui sont renvoyées par l'interface HTTP et l'interface OSLC. Cette section comporte les sous-sections suivantes :

corsHeaders

Remplace les en-têtes HTTP CORS (Cross-Origin Resource Sharing). Par défaut, les en-têtes par défaut sont remplacés pour indiquer que l'en-tête HTTP **Location** est autorisé et exposé. Ce paramètre est requis pour les messages HTTP 201 Create responses.

httpHeaders

Pour les en-têtes HTTP définis par l'utilisateur. Ces en-têtes sont ajoutés à toutes les réponses HTTP. Utilisez cette section pour ajoutez des valeurs statiques pour les clients. Un modèle d'en-tête est fourni dans l'exemple.

mimeTypes

Cette section affecte une extension de fichier, par exemple .html, à un type MIME. Lorsque la gestion de fichiers est activée, ces définitions sont utilisées pour déterminer le type MIME pour le fichier. Elles définissent également l'en-tête HTTP Content-Type afin que les navigateurs puissent gérer le fichier correctement. Le fichier `$OMNIHOME/etc/libnhttpd/json` contient un jeu par défaut de paramètres de type MIME auquel vous pouvez ajouter des extensions de fichier.

Activation et configuration du registre de fournisseurs de services IBM JazzSM

Si votre environnement utilise Jazz for Service Management (JazzSM), vous pouvez configurer l'ObjectServer pour l'inscrire dans le registre des fournisseurs de services JazzSM. L'ObjectServer est inscrit en tant que fournisseur de services OSLC d'événements. Les inscriptions dans les registres JazzSM sont configurées et gérées par la table du registre de fournisseurs de services OSLC, `registry.oslcsp`.

Pourquoi et quand exécuter cette tâche

L'accès à cette table n'est accordé qu'à l'utilisateur racine et aux administrateurs qui disposent du rôle OSLCAdmin. Les inscriptions ne peuvent pas être mises à jour. Les enregistrements d'inscription peuvent uniquement être insérés et supprimés, mais non supprimés.

Procédure

- Pour créer une inscription, insérez une entrée d'inscription dans la table `registry.oslcsp`. L'exemple suivant montre un modèle de commande SQL INSERT

pour le registre de fournisseurs de services JazzSM qui s'exécute sur l'hôte jazzsm.company.com, sur le port 9080, avec les données d'identification par défaut :

```
INSERT INTO registry.oslcsp ( Name, RegistryURI,  
RegistryUsername, RegistryPassword )  
VALUES ( 'MyRegistration',  
'http://jazzsm.company.com:9080/oslc/pr',  
'system', 'manager' );
```

Une fois l'insertion terminée, l'ObjectServer tente d'inscrire l'interface OSLC du serveur ObjectServer local dans le registre de fournisseurs de services JazzSM défini. Si l'inscription aboutit, l'URI d'inscription qui a été créé est stocké dans la zone RegistrationURI. La zone Registered est définie sur 1. Si l'inscription échoue, la zone Registered est définie sur 0.

- Pour supprimer une inscription d'un registre de fournisseurs de services JazzSM, supprimez l'entrée d'inscription de la table. Par exemple, pour supprimer l'inscription de l'exemple précédent, utilisez la commande SQL DELETE illustrée dans l'exemple suivant :

```
DELETE FROM registry.oslcsp WHERE Name='MyRegistration';
```

Si l'enregistrement d'inscription contient un URI d'inscription enregistré dans le registre de fournisseurs de services JazzSM défini, l'ObjectServer supprime cet enregistrement lorsque vous supprimez la ligne de la table.

Que faire ensuite

Si une inscription échoue, consultez le fichier journal de l'ObjectServer.

Table registry.oslc

Cette table est utilisée pour configurer et gérer des inscriptions de fournisseurs de services OSLC dans les registres de services IBM® JazzSM.

Tableau 5. Table d'inscription de fournisseurs de services OSLC registry.oslcsp.

Colonne	Type	Description
Name	VARCHAR(64)	Nom défini par l'utilisateur de l'entrée de table d'inscription.
RegistryURI	VARCHAR(1024)	Enregistrement des services du fournisseur de services OSLC du service de registre. RegistryURI est la clé principale de la table.
RegistryUsername	VARCHAR(64)	Utilisateur permettant de s'authentifier sur le registre de fournisseurs de services JazzSM.
RegistryPassword	VARCHAR(64)	Mot de passe permettant de s'authentifier sur le registre de fournisseurs de services JazzSM.

Tableau 5. Table d'inscription de fournisseurs de services OSLC registry.oslcsp. (suite)

Colonne	Type	Description
Registered	entier	Indique si l'entrée possède un enregistrement d'inscription auprès du service de registre JazzSM. Les valeurs possibles sont les suivantes : <ul style="list-style-type: none"> • 0 : l'entrée n'a pas d'inscription. • 1 : l'entrée a une inscription.
RegistrationURI	VARCHAR(1024)	URI de l'enregistrement d'inscription dans le registre de fournisseurs de services JazzSM pour cet ObjectServer.
LastRegistered	time	Date et heure de la dernière inscription réussie dans le registre de fournisseurs de services JazzSM.

Chapitre 2. URI d'interface HTTP

L'interface HTTP ObjectServer inclut des URI qui permettent d'accéder aux données de table et aux lignes des tables, d'exécuter des commandes SQL via HTTP et d'accéder aux informations système.

Services de collection de tables

Utilisez l'URI des services de collection de tables pour accéder à une table du magasin de données de l'ObjectServer, telle que la table `system` ou `user`. L'URI des services de collection de tables est l'URI de niveau supérieur.

Le format de cet URI est le suivant.

```
http://hôte:port/objectserver/restapi/base_de_données/table
```

Où `base_de_données` est le nom de la base de données ObjectServer et `table` est le nom de la table dans cette base de données.

L'URI des services de collection de tables prend en charge les méthodes HTTP suivantes : GET, POST, PATCH et DELETE.

Les lignes de la table référencée peuvent être extraites, mises à jour ou supprimées par une requête unique. Une ligne peut également être insérée dans la table par une requête POST. Une seule ligne peut être insérée. L'insertion de plusieurs lignes n'est pas possible.

Les exemples d'URI de collection de tables sont les suivants :

- `http://localhost/objectserver/restapi/alerts/status`
- `http://localhost/objectserver/restapi/catalog/tables`
- `http://localhost/objectserver/restapi/alerts/conversions`

Services de collection de tables : requête GET

Éléments d'une requête HTTP GET sur une collection de tables pour extraire des lignes d'une table ObjectServer.

Tableau 6. Services de collection de tables : requête GET

Élément	Description
Méthode HTTP	GET

Tableau 6. Services de collection de tables : requête GET (suite)

Élément	Description
Paramètres de requête	<p>filter Définit les conditions qu'une ligne dans une table doit satisfaire. Ce paramètre est équivalent à la clause WHERE d'une instruction SQL SELECT.</p> <p>collist Définit les colonnes de la table dans les résultats de la réponse HTTP. Ce paramètre est équivalent au composant de liste de colonne (column list) d'une instruction SQL SELECT.</p> <p>orderby Définit l'ordre de tri de l'ensemble de résultats. Ce paramètre est équivalent à la clause ORDER BY d'une instruction SQL SELECT.</p>
En-têtes de requête	<p>Authorization Obligatoire</p> <p>Host Obligatoire</p>
Accepter	application/json
Corps de la requête	Non applicable

Référence associée:

«Mécanismes d'authentification», à la page 35

Toute connexion à l'interface HTTP ObjectServer nécessite un ensemble de données d'identification utilisateur Tivoli Netcool/OMNIBus pour l'authentification. Seule l'authentification HTTP de base est prise en charge.

«Exemple de requête GET de collection de tables», à la page 44

Sélectionnez toutes les lignes de la table alerts.status.

Services de collection de tables : réponse GET

Éléments d'une réponse HTTP GET à une collection de tables dans un ObjectServer.

Tableau 7. Services de collection de tables : réponse GET

Élément	Description
En-têtes de réponse	<p>Server Nom du moteur HTTPd.</p> <p>Date Date ou heure de la réponse.</p> <p>Connection Etat de la connexion. Les états possibles sont Close ou Keep-Alive.</p>
Content-Type	application/json
Codes de réponse HTTP normale	200 (Créé) : le corps de la réponse contient un message de réussite JSON.

Tableau 7. Services de collection de tables : réponse GET (suite)

Élément	Description
Codes de réponse HTTP erronée	<p>500 (Erreur de serveur interne) : le serveur n'a pas pu exécuter la requête en raison d'un problème interne inattendu. Le corps de la réponse contient le message d'erreur JSON et d'autres informations.</p> <p>Les autres codes réponse d'erreur HTTP courants sont 401 (Non autorisé), 403 (Interdit) et 406 (Inacceptable).</p>

Référence associée:

«Codes de réponse HTTP», à la page 33

Ensemble commun de codes de réponse HTTP pour une méthode HTTP provenant d'une interface HTTP ObjectServer.

«Charge de message d'erreur JSON», à la page 36

L'interface HTTP ObjectServer peut renvoyer une charge de message d'erreur JSON dans un code réponse d'échec, tel que 500. Ce message d'erreur fournit des informations sur l'échec du code retour ObjectServer interne, lié à la demande.

«Exemple de réponse GET de collection de tables», à la page 44

Services de collection de tables : requête POST

Tableau 8. Services de collection de tables : requête POST

Élément	Description
Méthode HTTP	POST
Paramètres de requête	N/D
En-têtes de requête	<p>Authorization Obligatoire</p> <p>Host Obligatoire</p>
Accept	application/json
Content-Type	application/json
Corps de la requête	Ensemble de lignes JSON contenant une ligne à insérer. L'insertion de plusieurs lignes n'est pas prise en charge.

Référence associée:

«Exemple d'ensemble de lignes JSON : POST», à la page 41

Cet exemple d'ensemble de lignes JSON définit une ligne à insérer dans la table alerts.status de l'ObjectServer.

«Exemple de requête POST de collection de tables», à la page 43

Insertion d'une ligne dans la table alerts.status.

Services de collection de tables : réponse POST

Éléments d'une réponse HTTP POST pour la création d'une ligne dans une collection de tables dans l'ObjectServer.

Tableau 9. Services de collection de tables : réponse POST

Élément	Description
En-têtes de réponse	Server Nom du moteur HTTPd. Date Date ou heure de la réponse. Connection Etat de la connexion. Les états possibles sont Close ou Keep-Alive. Location URI de la ressource créée.
Content-Type	application/json
Codes de réponse HTTP normale	201 (Créé) : l'URI de la ligne insérée est contenue dans l'en-tête HTTP Location de la réponse. Le corps de la réponse contient un message de réussite JSON.
Codes de réponse HTTP erronée	400 (Requête incorrecte) : la définition de jeu de lignes JSON de la ligne à insérer n'est pas valide. La ligne n'est pas valide car elle contient des colonnes manquantes, inconnues ou incorrectes ou parce que des valeurs de colonne insuffisantes ou incorrectes ont été indiquées. 409 (Conflit) : collision de clé principale. La ligne existe déjà dans la table. 500 (Erreur de serveur interne) : le serveur n'a pas pu exécuter la requête en raison d'un problème interne inattendu. Le corps de la réponse contient le message d'erreur JSON et d'autres informations. Les autres codes réponse d'erreur HTTP courants sont 401 (Non autorisé), 403 (Interdit)406 (Inacceptable) et 415 (Type de support non pris en charge).

Référence associée:

«Codes de réponse HTTP», à la page 33

Ensemble commun de codes de réponse HTTP pour une méthode HTTP provenant d'une interface HTTP ObjectServer.

«Charge de message JSON de réussite», à la page 36

L'interface HTTP ObjectServer renvoie un message de réussite JSON dans la charge. Le code réponse HTTP informe le demandeur du succès éventuel de la requête et le message JSON fournit des informations supplémentaires. Ces informations incluent le nombre de lignes d'une collection de tables qui ont été affectées par la requête.

«Charge de message d'erreur JSON», à la page 36

L'interface HTTP ObjectServer peut renvoyer une charge de message d'erreur JSON dans un code réponse d'échec, tel que 500. Ce message d'erreur fournit des informations sur l'échec du code retour ObjectServer interne, lié à la demande.

«Exemple de réponse POST de collection de tables», à la page 44

Services de collection de tables : requête PATCH

Éléments d'une requête HTTP PATCH sur une collection de tables pour mettre à jour des lignes dans une table ObjectServer.

Tableau 10. Éléments HTTP PATCH

Élément	Description
Méthode HTTP	PATCH
Paramètres de requête	filter Définit les conditions qu'une ligne dans une table doit satisfaire. Ce paramètre est équivalent à la clause WHERE d'une instruction SQL SELECT.
En-têtes de requête	Authorization Obligatoire Host Obligatoire
Accept	application/json
Corps de la requête	Ensemble de lignes JSON contenant une ligne unique. La ligne définit les colonnes à mettre à jour et sur quelles valeurs les colonnes doivent être mises à jour.

Référence associée:

«Codes de réponse HTTP», à la page 33

Ensemble commun de codes de réponse HTTP pour une méthode HTTP provenant d'une interface HTTP ObjectServer.

«Exemple d'ensemble de lignes JSON : PATCH», à la page 40

Cet exemple met à jour les colonnes Location, LastOccurrence, Acknowledged, OwnerUID et OwnerGID des lignes concordantes dans la table alerts.status.

«Exemple de requête PATCH de collection de tables», à la page 46

Mettez à jour les colonnes Location, LastOccurrence, Acknowledged, OwnerUID et OwnerGID de toutes les lignes de la table alerts.status.

Services de collection de tables : réponse PATCH

Éléments de la réponse HTTP PATCH pour la mise à jour de lignes dans une collection de tables dans l'ObjectServer.

Tableau 11. Services de collection de tables : réponse PATCH

Élément	Description
En-têtes de réponse	<p>Server Nom du moteur HTTPd.</p> <p>Date Date ou heure de la réponse.</p> <p>Connection Etat de la connexion. Les états possibles sont Close ou Keep-Alive.</p>
Content-Type	application/json
Codes de réponse HTTP normale	200 (OK) : le corps de la réponse contient un message de réussite JSON.
Codes de réponse HTTP erronée	<p>400 (Requête incorrecte) : la définition de jeu de lignes JSON de la ligne à insérer n'est pas valide. La ligne n'est pas valide car elle contient des colonnes manquantes, inconnues ou incorrectes ou parce que des valeurs de colonne insuffisantes ou incorrectes ont été indiquées.</p> <p>500 (Erreur de serveur interne) : le serveur n'a pas pu exécuter la requête en raison d'un problème interne inattendu. Le corps de la réponse contient le message d'erreur JSON et d'autres informations.</p> <p>Les autres codes réponse d'erreur HTTP courants sont 401 (Non autorisé), 403 (Interdit)406 (Inacceptable) et 415 (Type de support non pris en charge).</p>

Référence associée:

«Codes de réponse HTTP», à la page 33

Ensemble commun de codes de réponse HTTP pour une méthode HTTP provenant d'une interface HTTP ObjectServer.

«Charge de message JSON de réussite», à la page 36

L'interface HTTP ObjectServer renvoie un message de réussite JSON dans la charge. Le code réponse HTTP informe le demandeur du succès éventuel de la requête et le message JSON fournit des informations supplémentaires. Ces informations incluent le nombre de lignes d'une collection de tables qui ont été affectées par la requête.

«Charge de message d'erreur JSON», à la page 36

L'interface HTTP ObjectServer peut renvoyer une charge de message d'erreur JSON dans un code réponse d'échec, tel que 500. Ce message d'erreur fournit des informations sur l'échec du code retour ObjectServer interne, lié à la demande.

«Exemple de réponse PATCH de collection de tables», à la page 46

Services de collection de tables : requête DELETE

Éléments d'une requête HTTP DELETE sur une collection de tables pour supprimer des lignes dans une table ObjectServer.

Tableau 12. Services de collection de tables : requête DELETE

Élément	Description
Méthode HTTP	
Paramètres de requête	filter Définit les conditions qu'une ligne dans une table doit satisfaire. Ce paramètre est équivalent à la clause WHERE d'une instruction SQL SELECT.
En-têtes de requête	Authorization Obligatoire Host Obligatoire
Accept	application/json
Corps de la requête	N/D

Référence associée:

«Mécanismes d'authentification», à la page 35

Toute connexion à l'interface HTTP ObjectServer nécessite un ensemble de données d'identification utilisateur Tivoli Netcool/OMNIBus pour l'authentification. Seule l'authentification HTTP de base est prise en charge.

«Exemple de requête DELETE de collection de tables», à la page 47

Supprimez toutes les lignes de la table alerts.status.

Services de collection de tables : réponse DELETE

Éléments de la réponse HTTP DELETE pour la suppression de lignes dans une collection de tables dans l'ObjectServer.

Tableau 13. Services de collection de tables : réponse DELETE

Élément	Description
En-têtes de réponse	Server Nom du moteur HTTPd. Date Date ou heure de la réponse. Connection Etat de la connexion. Les états possibles sont Close ou Keep-Alive.
Type de contenu	application/json
Codes de réponse HTTP normale	200 (Créé) : le corps de la réponse contient un message de réussite JSON.

Tableau 13. Services de collection de tables : réponse DELETE (suite)

Élément	Description
Codes de réponse HTTP erronée	<p>400 (Requête incorrecte) : la définition de jeu de lignes JSON de la ligne à insérer n'est pas valide. La ligne n'est pas valide car elle contient des colonnes manquantes, inconnues ou incorrectes ou parce que des valeurs de colonne insuffisantes ou incorrectes ont été indiquées.</p> <p>500 (Erreur de serveur interne) : le serveur n'a pas pu exécuter la requête en raison d'un problème interne inattendu. Le corps de la réponse contient le message d'erreur JSON et d'autres informations.</p> <p>Les autres codes réponse d'erreur HTTP courants sont 401 (Non autorisé), 403 (Interdit) 406 (Inacceptable) et 415 (Type de support non pris en charge).</p>

Référence associée:

«Codes de réponse HTTP», à la page 33

Ensemble commun de codes de réponse HTTP pour une méthode HTTP provenant d'une interface HTTP ObjectServer.

«Charge de message JSON de réussite», à la page 36

L'interface HTTP ObjectServer renvoie un message de réussite JSON dans la charge. Le code réponse HTTP informe le demandeur du succès éventuel de la requête et le message JSON fournit des informations supplémentaires. Ces informations incluent le nombre de lignes d'une collection de tables qui ont été affectées par la requête.

«Charge de message d'erreur JSON», à la page 36

L'interface HTTP ObjectServer peut renvoyer une charge de message d'erreur JSON dans un code réponse d'échec, tel que 500. Ce message d'erreur fournit des informations sur l'échec du code retour ObjectServer interne, lié à la demande.

«Exemple de réponse DELETE de collection de tables», à la page 47

Services d'élément de ligne

Utilisez cet URI pour accéder et faire référence à des lignes spécifiques dans les tables ObjectServer.

Les méthodes suivantes sont prises en charge.

RowSerial

Chaque ligne de chaque table de l'ObjectServer contient une valeur entière qui identifie de manière unique cette ligne dans la table où elle se trouve. Lorsque vous utilisez une requête HTTP GET pour extraire des lignes vers l'URI de services de collection de tables, la valeur RowSerial de chaque ligne est automatiquement renvoyée en tant que colonne dans le document d'ensemble de lignes JSON. La valeur RowSerial peut être utilisée pour construire un URI d'élément de ligne vers cette ligne, comme le montre l'exemple suivant.

`http://hôte:port/objectserver/restapi/base_de_données/table/série_ligne/`

Où *base_de_données* est le nom de la base de données ObjectServer, *table* est le nom de la table ObjectServer et *série_ligne* est le numéro de série unique de la ligne.

La méthode RowSerial prend en charge les méthodes HTTP suivantes : GET, PATCH et DELETE.

Etant donné que RowSerial est un entier, il suffit à construire des URI et à évaluer des ObjectServer. Les URI RowSerial ne sont valides que pour l'ObjectServer à partir duquel ils ont été initialement déterminés. L'accès aux lignes via RowSerial est la méthode préconisée.

Zone clé

Chaque ligne de chaque table de l'ObjectServer contient une zone clé qui identifie de manière unique cette ligne dans la table où elle se trouve. Il est donc possible de référencer une ligne à l'aide de la zone clé plutôt que par la valeur RowSerial, comme indiqué dans l'exemple suivant.

`http://hôte:port/objectserver/restapi/base_de_données/table/kf/zone_clé/`

Où *base_de_données* est le nom de la base de données ObjectServer, *table* est le nom de la table ObjectServer et *zone_clé* est le numéro de série unique de la ligne.

La méthode de zone clé prend en charge les méthodes HTTP suivantes : GET, PATCH et DELETE.

Remarques sur la construction des zones clés

Lorsque cela est autorisé, un URI de zone clé est valide dans n'importe quel ObjectServer dans les environnements où l'interface HTTP est activée, de sorte que vous pouvez construire et utiliser un chemin commun vers une ligne dans plusieurs serveurs ObjectServer.

Les zones clés des lignes qui sont valides dans plusieurs serveurs ObjectServer varient en fonction de la table ObjectServer où elles se trouvent. Par exemple, dans la table `alerts.status`, la clé principale est la colonne `Identifier`. Etant donné que la valeur d'`Identifier` n'est pas unique pour une instance d'événement unique dans plusieurs serveurs ObjectServer, elle n'est pas utilisée pour construire la zone clé. Au lieu de cela, la zone clé des lignes dans la table `alerts.status` est construite à partir des valeurs des colonnes `ServerSerial` et `ServerName`.

Remarque : La génération des zones clés est complexe, notamment dans les tables contenant des clés principales dans plusieurs colonnes, car certains caractères doivent être codés pour être utilisés dans un URI. Utilisez l'URI RowSerial, excepté si vous devez utiliser un URI valide sur plusieurs serveurs ObjectServer.

Le tableau suivant montre comment la zone clé est construite pour les tables `alerts.status`, `alerts.journal`, `alerts.details` tables et pour toutes les autres tables générales dans le serveur ObjectServer.

Tableau 14. Construction de valeurs de zone clé pour des éléments de ligne

Table	Type	Mode de construction de la zone clé
alerts.status	Événement	<p>La zone ServerSerial est combinée à la zone ServerName et séparée par une virgule. L'exemple suivant montre comment les zones sont combinées. 1234:NCOMS</p> <p>L'exemple suivant montre comment la zone est construite après avoir été codée pour être utilisée dans un URI. 1234%3ANCOMS</p> <p>L'exemple suivant montre un modèle d'URI. http://localhost/ objectserver/ restapi/alerts /status/ kf/12510% 3ANCOMS</p>
alerts.journal	Journal	<p>La zone clé est définie par la colonne KeyField de la table de journal. Etant donné que les journaux sont resérialisés différemment dans les différents serveurs ObjectServer, une zone de clé de journal n'est pas valide dans plusieurs serveurs ObjectServer. L'exemple suivant montre un modèle de KeyField. 2684:0:1341416084</p> <p>L'exemple suivant montre comment la zone est construite après avoir été codée pour être utilisée dans un URI. 12684%3A0% 3A1341416084</p> <p>L'exemple suivant montre un modèle d'URI. http://localhost/ objectserver/ restapi/ alerts/ journal/kf/ 12684%3A0% 3A1341416084</p>

Tableau 14. Construction de valeurs de zone clé pour des éléments de ligne (suite)

Table	Type	Mode de construction de la zone clé
alerts.details	Détails	<p>La zone clé est définie par la colonne KeyField de la table de détails. L'exemple suivant montre un modèle de KeyField.</p> <p>EventIdentifieur@@@4####4</p> <p>L'exemple suivant montre comment la zone est construite après avoir été codée pour être utilisée dans un URI.</p> <p>EventIndentifieur%40%40%40%404%23%23%23%234</p> <p>L'exemple suivant montre un modèle d'URI.</p> <p>http://localhost/ objectserver/ restapi/alerts/ details/kf/ EventIndentifieur %40%40%40%4 04%23%23%23%234</p>
Toutes les autres tables générales dans l'ObjectServer	Dispositions générales	<p>La zone de clé est l'ensemble de colonnes qui constitue la clé principale de la table. Si la clé principale de la table est constituée de plusieurs clés, spécifiez les valeurs clés dans le même ordre que dans le schéma de table. Séparez les valeurs clés par la séquence #KF#.</p> <p>L'exemple suivant montre un modèle de zone clé dans une table contenant deux colonnes principales, avec les valeurs ColValue01 et 654.</p> <p>ColValue01#KF#654</p> <p>L'exemple suivant montre comment cette zone est construite après avoir été codée pour être utilisée dans un URI.</p> <p>ColValue01%23KF%23654</p> <p>L'exemple suivant montre cette zone clé dans un URI.</p> <p>http://localhost/ objectserver/ restapi/ alerts/mytable /kf/ColValue01 %23KF%23654</p>

Requête GET d'élément de ligne

Éléments d'une requête HTTP GET sur un élément de ligne pour extraire une ligne spécifique d'une table ObjectServer.

Tableau 15. Élément de ligne : requête GET

Élément	Description
Méthode HTTP	GET
Paramètres de requête	collist Définit les colonnes de la table dans les résultats de la réponse HTTP. Ce paramètre est équivalent au composant de liste de colonne (column list) d'une instruction SQL SELECT.
En-têtes de requête	Authorization Obligatoire Host Obligatoire
Accept	application/json
Corps de la requête	N/D

Référence associée:

«Mécanismes d'authentification», à la page 35

Toute connexion à l'interface HTTP ObjectServer nécessite un ensemble de données d'identification utilisateur Tivoli Netcool/OMNIBus pour l'authentification. Seule l'authentification HTTP de base est prise en charge.

«Exemple de requête GET d'élément de ligne via RowSerial», à la page 47

Réponse GET d'élément de ligne

Éléments d'une réponse HTTP GET pour l'extraction d'une ligne spécifique à partir d'une table ObjectServer.

Tableau 16. Élément de ligne : réponse GET

Élément	Description
En-têtes de réponse	Server Nom du moteur HTTPd. Date Date ou heure de la réponse. Connection Etat de la connexion. Les états possibles sont Close ou Keep-Alive.
Content-Type	application/json
Codes de réponse HTTP normale	200 (Créé) : le corps de la réponse contient un message de réussite JSON.

Tableau 16. Élément de ligne : réponse GET (suite)

Élément	Description
Codes de réponse HTTP erronée	<p>404 (Introuvable) : la ligne demandée est introuvable dans la table car elle a été supprimée.</p> <p>500 (Erreur de serveur interne) : le serveur n'a pas pu exécuter la requête en raison d'un problème interne inattendu. Le corps de la réponse contient le message d'erreur JSON et d'autres informations.</p> <p>Les codes réponse d'erreur HTTP courants sont 401 (Non autorisé), 403 (Interdit) et 406 (Inacceptable).</p>

Référence associée:

«Codes de réponse HTTP», à la page 33

Ensemble commun de codes de réponse HTTP pour une méthode HTTP provenant d'une interface HTTP ObjectServer.

«Charge de message d'erreur JSON», à la page 36

L'interface HTTP ObjectServer peut renvoyer une charge de message d'erreur JSON dans un code réponse d'échec, tel que 500. Ce message d'erreur fournit des informations sur l'échec du code retour ObjectServer interne, lié à la demande.

«Exemple de réponse GET d'élément de ligne via RowSerial», à la page 47

Requête PATCH d'élément de ligne

Éléments d'une requête HTTP PATCH sur un élément de ligne spécifique dans une table ObjectServer.

Tableau 17. Élément de ligne : requête PATCH

Élément	Description
Méthode HTTP	PATCH
Paramètres de requête	N/D
En-têtes de requête	<p>Authorization Obligatoire</p> <p>Host Obligatoire</p>
Accept	application/json
Corps de la requête	Ensemble de lignes JSON contenant une ligne unique. La ligne définit les colonnes de la ligne à mettre à jour.

Référence associée:

«Mécanismes d'authentification», à la page 35

Toute connexion à l'interface HTTP ObjectServer nécessite un ensemble de données d'identification utilisateur Tivoli Netcool/OMNIBus pour l'authentification. Seule l'authentification HTTP de base est prise en charge.

«Exemple d'ensemble de lignes JSON : PATCH», à la page 40

Cet exemple met à jour les colonnes Location, LastOccurrence, Acknowledged, OwnerUID et OwnerGID des lignes concordantes dans la table alerts.status.

«Exemple de requête PATCH d'élément de ligne», à la page 51
 Mettez à jour les colonnes Location, LastOccurrence, Acknowledged, OwnerUID et OwnerGID d'une ligne spécifique dans la table alerts.status.

Réponse PATCH d'élément de ligne

Éléments d'une réponse HTTP PATCH pour la mise à jour d'une ligne spécifique dans une table de l'ObjectServer.

Tableau 18. Élément de ligne : réponse PATCH

Élément	Description
En-têtes de réponse	<p>Server Nom du moteur HTTPd.</p> <p>Date Date ou heure de la réponse.</p> <p>Connection Etat de la connexion. Les états possibles sont Close ou Keep-Alive.</p>
Content-Type	application/json
Codes de réponse HTTP normale	200 (Créé) : le corps de la réponse contient un message de réussite JSON.
Codes de réponse HTTP erronée	<p>400 (Requête incorrecte) : la définition de jeu de lignes JSON de la ligne à insérer n'est pas valide. La ligne n'est pas valide car elle contient des colonnes manquantes, inconnues ou incorrectes ou parce que des valeurs de colonne insuffisantes ou incorrectes ont été indiquées.</p> <p>404 (Introuvable) : la ligne demandée est introuvable dans la table car elle a été supprimée.</p> <p>500 (Erreur de serveur interne) : le serveur n'a pas pu exécuter la requête en raison d'un problème interne inattendu. Le corps de la réponse contient le message d'erreur JSON et d'autres informations.</p> <p>Les autres codes réponse d'erreur HTTP courants sont 401 (Non autorisé), 403 (Interdit)406 (Inacceptable) et 415 (Type de support non pris en charge).</p>

Référence associée:

«Codes de réponse HTTP», à la page 33

Ensemble commun de codes de réponse HTTP pour une méthode HTTP provenant d'une interface HTTP ObjectServer.

«Charge de message JSON de réussite», à la page 36

L'interface HTTP ObjectServer renvoie un message de réussite JSON dans la charge. Le code réponse HTTP informe le demandeur du succès éventuel de la requête et le message JSON fournit des informations supplémentaires. Ces informations incluent le nombre de lignes d'une collection de tables qui ont été affectées par la requête.

«Charge de message d'erreur JSON», à la page 36

L'interface HTTP ObjectServer peut renvoyer une charge de message d'erreur JSON dans un code réponse d'échec, tel que 500. Ce message d'erreur fournit des informations sur l'échec du code retour ObjectServer interne, lié à la demande.

«Exemple de réponse PATCH d'élément de ligne», à la page 52

Requête DELETE d'élément de ligne

Éléments d'une requête HTTP DELETE sur un élément de ligne spécifique dans une table ObjectServer.

Tableau 19. Élément de ligne : requête DELETE

Élément	Description
Méthode HTTP	DELETE
Paramètres de requête	N/D
En-têtes de requête	Authorization Obligatoire Host Obligatoire
Accept	application/json
Corps de la requête	N/D

Référence associée:

«Mécanismes d'authentification», à la page 35

Toute connexion à l'interface HTTP ObjectServer nécessite un ensemble de données d'identification utilisateur Tivoli Netcool/OMNIBus pour l'authentification. Seule l'authentification HTTP de base est prise en charge.

«Exemple de requête DELETE d'élément de ligne», à la page 52

Supprimez une ligne spécifique dans la table alerts.status.

Réponse DELETE d'élément de ligne

Éléments d'une réponse HTTP DELETE pour la suppression d'une ligne spécifique dans une table de l'ObjectServer.

Tableau 20. Élément de ligne : réponse DELETE

Élément	Description
En-têtes de réponse	Server Nom du moteur HTTPd. Date Date ou heure de la réponse. Connection Etat de la connexion. Les états possibles sont Close ou Keep-Alive.
Content-Type	application/json
Codes de réponse HTTP normale	200 (Créé) : le corps de la réponse contient un message de réussite JSON.

Tableau 20. Élément de ligne : réponse DELETE (suite)

Élément	Description
Codes de réponse HTTP erronée	<p>404 (Introuvable) : la ligne demandée est introuvable dans la table car elle a été supprimée.</p> <p>500 (Erreur de serveur interne) : le serveur n'a pas pu exécuter la requête en raison d'un problème interne inattendu. Le corps de la réponse contient le message d'erreur JSON et d'autres informations.</p> <p>Les autres codes réponse d'erreur HTTP courants sont 401 (Non autorisé), 403 (Interdit)406 (Inacceptable) et 415 (Type de support non pris en charge).</p>

Référence associée:

«Codes de réponse HTTP», à la page 33

Ensemble commun de codes de réponse HTTP pour une méthode HTTP provenant d'une interface HTTP ObjectServer.

«Charge de message JSON de réussite», à la page 36

L'interface HTTP ObjectServer renvoie un message de réussite JSON dans la charge. Le code réponse HTTP informe le demandeur du succès éventuel de la requête et le message JSON fournit des informations supplémentaires. Ces informations incluent le nombre de lignes d'une collection de tables qui ont été affectées par la requête.

«Charge de message d'erreur JSON», à la page 36

L'interface HTTP ObjectServer peut renvoyer une charge de message d'erreur JSON dans un code réponse d'échec, tel que 500. Ce message d'erreur fournit des informations sur l'échec du code retour ObjectServer interne, lié à la demande.

«Exemple de réponse DELETE d'élément de ligne», à la page 53

Fabrique de commandes SQL

Utilisez la fabrique de commandes SQL de l'interface HTTP ObjectServer pour exécuter des commandes SQL arbitraires via HTTP.

Pour exécuter SQL via cette interface, les utilisateurs doivent disposer du droit utilisateur AllowISQL pour lire les commandes SQL et du droit utilisateur AllowISQLWrite pour écrire des commandes SQL.

Pour plus d'informations sur les droits utilisateur, voir *Guide d'administration d'IBM Tivoli Netcool/OMNIBus*.

Pour exécuter une commande SQL, envoyez un message de commande JSON SQL à l'URI de la fabrique de commandes SQL, comme illustré dans l'exemple suivant.

`http://hôte:port/objectserver/restapi/sql/factory`

La fabrique de commandes SQL prend en charge les méthodes HTTP suivantes : POST.

Requête POST de fabrique de commandes SQL

Conditions requises pour une requête POST sur une fabrique de commandes SQL pour exécuter des commandes SQL arbitraires dans l'ObjectServer.

Tableau 21. Fabrique de commandes SQL : requête POST

Élément	Description
Méthode HTTP	POST
Paramètres de requête	N/D
En-têtes de requête	Authorization Obligatoire Host Obligatoire
Accept	application/json
Content-Type	application/json
Corps de la requête	Commande de message JSON SQL

Référence associée:

«Mécanismes d'authentification», à la page 35

Toute connexion à l'interface HTTP ObjectServer nécessite un ensemble de données d'identification utilisateur Tivoli Netcool/OMNIBus pour l'authentification. Seule l'authentification HTTP de base est prise en charge.

«Exemple de fabrique de commandes SQL : POST», à la page 42

Cet exemple de message de fabrique de commandes SQL émet une commande de suppression d'utilisateur (drop user) pour exécution

«Exemple de requête POST de fabrique de commandes SQL», à la page 53

Réponse POST de fabrique de commandes SQL

Éléments d'une réponse HTTP POST pour l'exécution de commandes SQL arbitraires dans l'ObjectServer.

Tableau 22. Fabrique de commandes SQL : réponse POST

Élément	Description
En-têtes de réponse	Server Nom du moteur HTTPd. Date Date ou heure de la réponse. Connection Etat de la connexion. Les états possibles sont Close ou Keep-Alive.
Content-Type	application/json
Codes de réponse HTTP normale	200 (Créé) : le corps de la réponse contient un message de réussite JSON.

Tableau 22. Fabrique de commandes SQL : réponse POST (suite)

Élément	Description
Codes de réponse HTTP erronée	<p>400 (Requête incorrecte) : la définition de jeu de lignes JSON de la ligne à insérer n'est pas valide. La ligne n'est pas valide car elle contient des colonnes manquantes, inconnues ou incorrectes ou parce que des valeurs de colonne insuffisantes ou incorrectes ont été indiquées.</p> <p>500 (Erreur de serveur interne) : le serveur n'a pas pu exécuter la requête en raison d'un problème interne inattendu. Le corps de la réponse contient le message d'erreur JSON et d'autres informations.</p> <p>Les autres codes réponse d'erreur HTTP courants sont 401 (Non autorisé), 403 (Interdit) 406 (Inacceptable) et 415 (Type de support non pris en charge).</p>

Référence associée:

«Codes de réponse HTTP», à la page 33

Ensemble commun de codes de réponse HTTP pour une méthode HTTP provenant d'une interface HTTP ObjectServer.

«Charge de message JSON de réussite», à la page 36

L'interface HTTP ObjectServer renvoie un message de réussite JSON dans la charge. Le code réponse HTTP informe le demandeur du succès éventuel de la requête et le message JSON fournit des informations supplémentaires. Ces informations incluent le nombre de lignes d'une collection de tables qui ont été affectées par la requête.

«Charge de message d'erreur JSON», à la page 36

L'interface HTTP ObjectServer peut renvoyer une charge de message d'erreur JSON dans un code réponse d'échec, tel que 500. Ce message d'erreur fournit des informations sur l'échec du code retour ObjectServer interne, lié à la demande.

«Exemple de réponse POST de fabrique de commandes SQL», à la page 53

Services d'informations système

Utilisez cet URI pour interroger et déterminer les détails et les versions de compilation de l'interface HTTP et de l'interface OSLC dans l'ObjectServer.

Vous pouvez obtenir tous les éléments d'informations système disponibles en émettant une requête HTTP GET vers l'URI des services d'information système, comme illustré dans l'exemple suivant.

`http://hôte:port/objectserver/restapi/sysinfo`

Vous pouvez extraire les éléments d'informations système individuellement en faisant référence à l'élément d'informations système requis dans l'URI, comme illustré dans l'exemple suivant.

`http://hôte:port/objectserver/restapi/sysinfo/élément`

Où *élément* peut être l'un des éléments d'informations système suivants.

compile

Obtient les informations de construction détaillées des interfaces HTTP qui sont hébergées dans l'ObjectServer.

rest

Obtient des informations de version sur l'interface HTTP de l'ObjectServer.

oslc

Obtient des informations de version sur l'interface OSLC de l'ObjectServer.

L'URI de service d'informations système prend en charge la méthode HTTP suivante : GET.

Requête de collection GET

Eléments d'une requête HTTP GET vers un URI de collection d'informations système pour extraire toutes les informations système d'un ObjectServer.

Tableau 23. Services de collection d'informations système : requête GET

Élément	Description
Méthode HTTP	GET
Paramètres de requête	N/D
En-têtes de requête	Authorization Obligatoire Host Obligatoire
Accept	application/json
Corps de la requête	N/D

Référence associée:

«Mécanismes d'authentification», à la page 35

Toute connexion à l'interface HTTP ObjectServer nécessite un ensemble de données d'identification utilisateur Tivoli Netcool/OMNIBus pour l'authentification. Seule l'authentification HTTP de base est prise en charge.

«Exemple de requête GET d'informations système», à la page 53

Réponse de collection GET

Eléments d'une réponse HTTP GET pour l'extraction de toutes les informations de système ObjectServer.

Tableau 24. Services de collection d'informations système : réponse GET

Élément	Description
En-têtes de réponse	Server Nom du moteur HTTPd. Date Date ou heure de la réponse. Connection Etat de la connexion. Les états possibles sont Close ou Keep-Alive.
Content-Type	application/json

Tableau 24. Services de collection d'informations système : réponse GET (suite)

Élément	Description
Codes de réponse HTTP normale	200 (Créé) : le corps de la réponse contient un message de réussite JSON.
Codes de réponse HTTP erronée	500 (Erreur de serveur interne) : le serveur n'a pas pu exécuter la requête en raison d'un problème interne inattendu. Le corps de la réponse contient le message d'erreur JSON et d'autres informations. Les autres codes réponse d'erreur HTTP courants sont 401 (Non autorisé), 403 (Interdit) et 406 (Inacceptable).

Référence associée:

«Codes de réponse HTTP», à la page 33

Ensemble commun de codes de réponse HTTP pour une méthode HTTP provenant d'une interface HTTP ObjectServer.

«Charge de message d'erreur JSON», à la page 36

L'interface HTTP ObjectServer peut renvoyer une charge de message d'erreur JSON dans un code réponse d'échec, tel que 500. Ce message d'erreur fournit des informations sur l'échec du code retour ObjectServer interne, lié à la demande.

«Exemple de réponse GET d'informations système», à la page 54

Requête d'élément GET

Éléments d'une requête HTTP GET pour l'extraction d'un élément d'informations de système spécifique à partir d'un ObjectServer.

Tableau 25. Services d'élément d'informations système : requête GET

Élément	Description
Méthode HTTP	GET
Paramètres de requête	N/D
En-têtes de requête	Authorization Obligatoire Host Obligatoire
Accept	application/json
Corps de la requête	N/D

Référence associée:

«Mécanismes d'authentification», à la page 35

Toute connexion à l'interface HTTP ObjectServer nécessite un ensemble de données d'identification utilisateur Tivoli Netcool/OMNIBus pour l'authentification. Seule l'authentification HTTP de base est prise en charge.

«Exemple de requête GET d'élément d'informations système», à la page 54

Réponse d'élément GET

Éléments d'une réponse HTTP GET pour l'extraction d'un élément d'informations de système spécifique à partir d'un ObjectServer.

Tableau 26. Services d'élément d'informations système : réponse GET

Élément	Description
En-têtes de réponse	Server Nom du moteur HTTPd. Date Date ou heure de la réponse. Connection Etat de la connexion. Les états possibles sont Close ou Keep-Alive.
Content-Type	application/json
Codes de réponse HTTP normale	200 (Créé) : le corps de la réponse contient un message de réussite JSON.
Codes de réponse HTTP erronée	500 (Erreur de serveur interne) : le serveur n'a pas pu exécuter la requête en raison d'un problème interne inattendu. Le corps de la réponse contient le message d'erreur JSON et d'autres informations. Les autres codes réponse d'erreur HTTP courants sont 401 (Non autorisé), 403 (Interdit) et 406 (Inacceptable).

Référence associée:

«Codes de réponse HTTP», à la page 33

Ensemble commun de codes de réponse HTTP pour une méthode HTTP provenant d'une interface HTTP ObjectServer.

«Charge de message d'erreur JSON», à la page 36

L'interface HTTP ObjectServer peut renvoyer une charge de message d'erreur JSON dans un code réponse d'échec, tel que 500. Ce message d'erreur fournit des informations sur l'échec du code retour ObjectServer interne, lié à la demande.

«Exemple de réponse GET d'élément d'informations système», à la page 54

Chapitre 3. Comportements communs

Caractéristiques communes à toutes les demandes et les réponses de l'interface HTTP ObjectServer.

Support HTTP et HTTPS

L'interface HTTP ObjectServer prend en charge la connectivité HTTP ou HTTPS sur HTTP 1.0 or HTTP 1.1.

Codes de réponse HTTP

Ensemble commun de codes de réponse HTTP pour une méthode HTTP provenant d'une interface HTTP ObjectServer.

Codes de message de réussite

Le tableau suivant contient les codes de message de réussite HTTP communs.

Tableau 27. Codes de message de réussite HTTP communs

Méthode HTTP	Code de réponse HTTP	Commentaires
GET	200 (OK)	
POST	201 (Créé)	L'en-tête HTTP Location contient l'URI de la nouvelle ressource.
PATCH	200 (OK)	
DELETE	200 (OK)	

Codes de message d'erreur

Le tableau suivant contient les codes de message d'erreur HTTP communs.

Tableau 28. Codes de message d'erreur HTTP communs

Code de réponse HTTP	Commentaires
400	Requête incorrecte. Vérifiez la contenu de la requête et les paramètre de requête.
401	Non autorisé. La requête ne contient pas de données d'authentification valides.
403	L'accès à la ressource définie est refusé. Les données d'authentification qui ont été utilisées pour établir la connexion n'ont pas été autorisées à accéder aux ressources spécifiées dans la requête.
404	La ressource demandée est introuvable. La requête peut être supprimée.
406	Le type MIME Accept demandé n'est pas pris en charge.
409	Conflit. Tentative d'insertion d'une ligne qui existe déjà.

Tableau 28. Codes de message d'erreur HTTP communs (suite)

Code de réponse HTTP	Commentaires
415	Le type MIME Content indiqué n'est pas pris en charge.
500	Erreur interne du serveur. Pour plus d'informations, consultez le contenu du message d'erreur RDF/XML.

Référence associée:

«Charge de message d'erreur JSON», à la page 36

L'interface HTTP ObjectServer peut renvoyer une charge de message d'erreur JSON dans un code réponse d'échec, tel que 500. Ce message d'erreur fournit des informations sur l'échec du code retour ObjectServer interne, lié à la demande.

Paramètres de requête

Informations sur la syntaxe des paramètres de requête OSLC qui sont pris en charge par l'interface HTTP.

Tableau 29. Description des paramètres de requête de l'interface HTTP

Paramètres de requête	Description
filter	<p>Ce paramètre de requête définit les lignes sur lesquelles agit la requête, lorsque cette dernière porte sur l'ObjectServer. Utilisez le même format pour ce paramètre que pour une clause SQL WHERE. Vous pouvez utiliser ce paramètre uniquement avec les requêtes GET, PATCH et DELETE vers un URI de collection de tables ObjectServer.</p> <p>L'exemple suivant montre un modèle de paramètre filter. Node='hostname.domain'</p> <p>L'exemple suivant montre le même paramètre après qu'il a été codé pour être utilisé dans un URI. Node%3D%27hostname.domain%27</p> <p>L'exemple suivant montre un modèle d'URI contenant ce paramètre. http://localhost/objectserver/restapi/alerts/status?filter=Node%3D%27hostname.domain%27</p>

Tableau 29. Description des paramètres de requête de l'interface HTTP (suite)

Paramètres de requête	Description
collist	<p>Ce paramètre de requête est valide uniquement dans une requête HTTP GET. Ce paramètre définit quelles colonnes de la table doivent être renvoyées dans le message sur l'ensemble de lignes JSON. Pour définir la valeur du paramètre, utilisez la même syntaxe que le composant column list d'une instruction SQL SELECT.</p> <p>L'exemple suivant montre un modèle de paramètre collist. Serial,Node,Summary</p> <p>L'exemple suivant montre le même paramètre après qu'il a été codé pour être utilisé dans un URI. Serial%2CNode%2CSummary</p> <p>L'exemple suivant montre un modèle d'URI contenant ce paramètre. http://localhost/objectserver/restapi/alerts/status?collist=Serial%2CNode%2CSummary</p>
orderby	<p>Ce paramètre de requête est valide uniquement dans une requête HTTP GET vers un URI de collection de tables ObjectServer. Le paramètre définit l'ordre des lignes dans le message sur l'ensemble de lignes JSON qui est renvoyé. Pour définir la valeur du paramètre, utilisez la même syntaxe que la clause ORDER BY d'une instruction SQL SELECT.</p> <p>L'exemple suivant montre un modèle de paramètre orderby. Serial ASC</p> <p>L'exemple suivant montre le même paramètre après qu'il a été codé pour être utilisé dans un URI. Serial%20ASC</p> <p>L'exemple suivant montre un modèle d'URI contenant ce paramètre. http://localhost/objectserver/restapi/alerts/status/?orderby=Serial%20ASC</p>

L'exemple suivant montre un URI qui utilise tous les paramètres de requête dans tableau 29, à la page 34.

http://localhost/objectserver/restapi/alerts/status?filter=Node%3D%27hostname.domain%27&collist=Serial%2CNode%2CSummary&Serial%20ASC&orderby=Serial%20ASC

Mécanismes d'authentification

Toute connexion à l'interface HTTP ObjectServer nécessite un ensemble de données d'identification utilisateur Tivoli Netcool/OMNibus pour l'authentification. Seule l'authentification HTTP de base est prise en charge.

Si aucune donnée d'identification HTTP de base n'est fournie dans l'en-tête HTTP **Authorization**, une réponse HTTP 401 (Non autorisé) est renvoyée.

Etant donné que les données d'identification HTTP de base ne sont pas sécurisées, utilisez HTTPS pour garantir le chiffrement de la communication avec le connecteur.

Pour plus d'informations sur l'utilisation de SSL pour chiffrer les communications, consultez *Guide d'installation et de déploiement d'IBM Tivoli Netcool/OMNIBus*.

Référence associée:

«Requête POST de fabrication de commandes SQL», à la page 27

Conditions requises pour une requête POST sur une fabrique de commandes SQL pour exécuter des commandes SQL arbitraires dans l'ObjectServer.

Charge de message JSON de réussite

L'interface HTTP ObjectServer renvoie un message de réussite JSON dans la charge. Le code réponse HTTP informe le demandeur du succès éventuel de la requête et le message JSON fournit des informations supplémentaires. Ces informations incluent le nombre de lignes d'une collection de tables qui ont été affectées par la requête.

Tableau 30. Description de la charge de message de réussite RDF/XML

Nom	Type de valeur	Description
affectedRows	Entier	Nombre de lignes affectées par la requête. Pour les requêtes qui sont appliquées à une collection de tables, la valeur peut varier de zéro à plusieurs lignes.
keyField	Chaîne (String)	ID zone clé d'une ressource affectée. L'ID zone clé est présent uniquement dans les insertions de ligne ayant abouti.
uri	Chaîne (String)	URI de la ligne insérée ou URI de la requête ayant abouti. Pour une insertion de ligne réussie, cet URI doit correspondre à l'URI de l'en-tête HTTP Location .

Référence associée:

«Exemple de fabrication de commandes SQL : POST», à la page 42

Cet exemple de message de fabrication de commandes SQL émet une commande de suppression d'utilisateur (drop user) pour exécution

Charge de message d'erreur JSON

L'interface HTTP ObjectServer peut renvoyer une charge de message d'erreur JSON dans un code réponse d'échec, tel que 500. Ce message d'erreur fournit des informations sur l'échec du code retour ObjectServer interne, lié à la demande.

Tableau 31. Description de la charge de message d'erreur RDF/XML

Nom	Type de valeur	Description
statusCode	Integer	Code d'état HTTP renvoyé avec l'erreur.
message	Chaîne (String)	Message d'information décrivant l'erreur.

Tableau 31. Description de la charge de message d'erreur RDF/XML (suite)

Nom	Type de valeur	Description
moreInfo	Chaîne (String)	Informations supplémentaires, si disponibles. Si aucune information supplémentaire n'est disponible, cet élément de la charge du message n'est pas renvoyé.

Référence associée:

«Exemple de message d'erreur JSON», à la page 42

Exemple de message d'erreur JSON indiquant que la réponse HTTP 400 (Requête incorrecte) a été causée par un objet référencé tel qu'une colonne introuvable dans la table.

Chiffrement de message

Le chiffrement de la charge de message n'est pas pris en charge.

Types MIME Accept

Types MIME Accept pris en charge par l'interface HTTP ObjectServer.

Les types MIME pris en charge sont les suivants.

application/json (JSON)

Types MIME Content

Types MIME Content pris en charge par l'interface HTTP ObjectServer.

Les types MIME pris en charge sont les suivants.

application/json (JSON)

Mise en cache des réponses

Les lignes dans l'ObjectServer, notamment les données d'événement, varient constamment en raison des actions utilisateur ou de programmation. Etant donné que l'interface HTTP ObjectServer est hébergée directement avec les données de table, l'accès aux données n'entraîne pas de pénalité. L'interface ne met donc pas en cache de données, telles que les réponses, à quelque niveau que ce soit. Chaque requête HTTP ObjectServer est résolue séparément chaque fois, de la même manière qu'une requête sur les données de l'ObjectServer émise par une interface de l'ObjectServer.

Annexe A. Exemples de charges de message JSON

Exemples de charges de message JSON.

Exemple d'ensemble de lignes JSON : GET

Cet exemple d'ensemble de lignes JSON provient d'une instruction HTTP GET sur l'URI de collection de tables alerts.status.

```
{
  "rowset": {
    "osname": "NCOMS",
    "dbname": "alerts",
    "tblname": "status",
    "coldesc": [{
      "name": "Identifient",
      "type": "string",
      "size": 255
    }, {
      "name": "Serial",
      "type": "integer",
      "size": 4
    }, {
      "name": "Node",
      "type": "string",
      "size": 64
    }, {
      "name": "NodeAlias",
      "type": "string",
      "size": 64
    }, {
      "name": "AlertKey",
      "type": "string",
      "size": 255
    }, {
      "name": "Severity",
      "type": "integer",
      "size": 4
    }, {
      "name": "Summary",
      "type": "string",
      "size": 255
    }, {
      "name": "StateChange",
      "type": "utc",
      "size": 4
    }, {
      "name": "FirstOccurrence",
      "type": "utc",
      "size": 4
    }, {
      "name": "LastOccurrence",
      "type": "utc",
      "size": 4
    }, {
      "name": "RowSerial",
      "type": "integer",
      "size": 4
    }
  ],
  "rows": [{
    "Identifient": "Startup@sol9-build1",
    "Serial": 12469,
```

```

        "Node": "sol9-build1",
        "NodeAlias": "",
        "AlertKey": "",
        "Severity": 0,
        "Summary": "ObjectServer NCOMS on sol9-build1 started at
Wed Jul 04 15:27:57 2012",
        "StateChange": 1341412082,
        "FirstOccurrence": 1341411978,
        "LastOccurrence": 1341412077,
        "RowSerial": 12469
    }, {
        "Identifier": "ProfilerEnableToggle@NCOMS:sol9-build1",
        "Serial": 12468,
        "Node": "sol9-build1",
        "NodeAlias": "",
        "AlertKey": "",
        "Severity": 0,
        "Summary": "ObjectServer NCOMS Profiler enabled at Wed Jul 04
15:27:56 2012",
        "StateChange": 1341412077,
        "FirstOccurrence": 1341411976,
        "LastOccurrence": 1341412076,
        "RowSerial": 12468
    }, {
        "Identifier": "Shutdown@sol9-build1",
        ***** TRUNCATED *****
        "RowSerial": 12519
    }
  ],
  "affectedRows": 13
}
}

```

Exemple d'ensemble de lignes JSON : PATCH

Cet exemple met à jour les colonnes Location, LastOccurrence, Acknowledged, OwnerUID et OwnerGID des lignes concordantes dans la table alerts.status.

```

{
  "rowset": {
    "coldesc": [{
      "type": "integer",
      "name": "Acknowledged"
    }, {
      "type": "string",
      "name": "Location"
    }, {
      "type": "integer",
      "name": "OwnerUID"
    }, {
      "type": "integer",
      "name": "OwnerGID"
    }, {
      "type": "utc",
      "name": "LastOccurrence"
    }
  ],
  "rows": [{
    "Location": "UPDATED",
    "LastOccurrence": 1341412223,
    "Acknowledged": 1,
    "OwnerUID": 65534,
    "OwnerGID": 1
  }
  ],
  "affectedRows": 0
}

```

Exemple d'ensemble de lignes JSON : POST

Cet exemple d'ensemble de lignes JSON définit une ligne à insérer dans la table alerts.status de l'ObjectServer.

```
{
  "rowset": {
    "coldesc": [{
      "type": "string",
      "name": "Identifieur"
    }, {
      "type": "string",
      "name": "Node"
    }, {
      "type": "string",
      "name": "Manager"
    }, {
      "type": "string",
      "name": "Agent"
    }, {
      "type": "string",
      "name": "AlertKey"
    }, {
      "type": "integer",
      "name": "Severity"
    }, {
      "type": "integer",
      "name": "Type"
    }, {
      "type": "string",
      "name": "Summary"
    }, {
      "type": "integer",
      "name": "Acknowledged"
    }, {
      "type": "string",
      "name": "Location"
    }, {
      "type": "utc",
      "name": "FirstOccurrence"
    }, {
      "type": "utc",
      "name": "LastOccurrence"
    }, {
      "type": "integer",
      "name": "OwnerUID"
    }, {
      "type": "integer",
      "name": "OwnerGID"
    }
  ],
  "rows": [{
    "FirstOccurrence": 1341412087,
    "Node": "localhost",
    "AlertKey": "JUnitEventInstance",
    "Agent": "createEventNew()",
    "Summary": "This is a test event generated by the JUnit REST Event Tests.(0)",
    "LastOccurrence": 1341412087,
    "Acknowledged": 0,
    "Identifieur": "JUnitEventTestInstance###0",
    "Manager": "com.ibm.netcool.omnibus.ws.junit.rest.schema.utils.TableRowEvent",
    "OwnerGID": 0,
    "Location": "NOT UPDATED",
    "Type": 1,
    "Severity": 4,
    "OwnerUID": 0
  }
]
```

```
    } ],  
    "affectedRows":0  
  }  
}
```

Exemple de fabrication de commandes SQL : POST

Cet exemple de message de fabrication de commandes SQL émet une commande de suppression d'utilisateur (drop user) pour exécution

```
{  
  "sqlcmd" : "drop user 'testuser01';"  
}
```

Référence associée:

«Requête POST de fabrication de commandes SQL», à la page 27

Conditions requises pour une requête POST sur une fabrication de commandes SQL pour exécuter des commandes SQL arbitraires dans l'ObjectServer.

Exemple de message de réussite JSON

Exemple de message de réussite JSON pour une insertion de ligne dans la table alerts.status d'ObjectServer.

```
{  
  "entry": {  
    "affectedRows": 1,  
    "keyField": "14382%3ANCOMS",  
    "uri": "http://localhost/objectserver/restapi/alerts/status/kf/14382%3ANCOMS"  
  }  
}
```

Exemple de message d'erreur JSON

Exemple de message d'erreur JSON indiquant que la réponse HTTP 400 (Requête incorrecte) a été causée par un objet référencé tel qu'une colonne introuvable dans la table.

```
{  
  "exception": {  
    "statusCode": 400,  
    "message": "Object not found"  
  }  
}
```

Référence associée:

«Charge de message d'erreur JSON», à la page 36

L'interface HTTP ObjectServer peut renvoyer une charge de message d'erreur JSON dans un code réponse d'échec, tel que 500. Ce message d'erreur fournit des informations sur l'échec du code retour ObjectServer interne, lié à la demande.

Annexe B. Exemples de demandes et réponses HTTP

Exemples de demandes et réponses HTTP.

Exemple de requête POST de collection de tables

Insertion d'une ligne dans la table alerts.status.

```
Accept: application/json
Authorization: Basic dGVzdHVzZXIwMTpuZXRjb29s
Content-Type: application/json
Host: localhost
Connection: keep-alive
Content-Length: 984
```

```
{
  "rowset": {
    "coldesc": [ {
      "type": "string",
      "name": "Identifiser"
    }, {
      "type": "string",
      "name": "Node"
    }, {
      "type": "string",
      "name": "AlertKey"
    }, {
      "type": "integer",
      "name": "Severity"
    }, {
      "type": "string",
      "name": "Summary"
    }, {
      "type": "utc",
      "name": "FirstOccurrence"
    }, {
      "type": "utc",
      "name": "LastOccurrence"
    }, {
      "type": "integer",
      "name": "OwnerUID"
    }, {
      "type": "integer",
      "name": "OwnerGID"
    }
  ],
  "rows": [ {
    "FirstOccurrence": 1341412087,
    "Node": "localhost",
    "AlertKey": "JUnitEventInstance",
    "Summary": "This is a test event generated by the JUnit REST Event Tests.(1)",
    "LastOccurrence": 1341412087,
    "Identifiser": "JUnitEventTestInstance###1",
    "OwnerGID": 0,
    "Severity": 4,
    "OwnerUID": 0
  }
]
}
```

Exemple de réponse POST de collection de tables

```
HTTP/1.1 201 Created
Location: http://localhost/objectserver/restapi/alerts/status/kf/12481%3ANCOMS
Cache-Control: no-cache
Server: libnhttpd
Date: Wed Jul 4 15:31:53 2012
Connection: Keep-Alive
Content-Type: application/json;charset=UTF-8
Content-Length: 304
{
  "entry": {
    "affectedRows": 1,
    "keyField": "12481%3ANCOMS",
    "uri": "http://localhost/objectserver/restapi/alerts/status/kf/12481%3ANCOMS"
  }
}
```

Exemple de requête GET de collection de tables

Sélectionnez toutes les lignes de la table alerts.status.

```
GET /objectserver/restapi/alerts/status HTTP/1.1
Accept: application/json
Authorization: Basic dGVzdHVzZXIwMTpuZXRjb29s
Host: localhost
Connection: keep-alive
```

Exemple de réponse GET de collection de tables

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Server: libnhttpd
Date: Wed Jul 4 15:32:03 2012
Connection: Keep-Alive:
Content-Type: application/rdf+xml
Content-Length: 24860
{
  "rowset": {
    "osname": "NCOMS",
    "dbname": "alerts",
    "tblname": "status",
    "coldesc": [{
      "name": "Identifler",
      "type": "string",
      "size": 255
    }, {
      "name": "Serial",
      "type": "integer",
      "size": 4
    }, {
      "name": "Node",
      "type": "string",
      "size": 64
    }, {
      "name": "NodeAlias",
      "type": "string",
      "size": 64
    }, {
      "name": "AlertKey",
      "type": "string",
      "size": 255
    }, {
      "name": "Severity",
      "type": "integer",
      "size": 4
    }
  ]
}
```

```

    }, {
      "name": "Summary",
      "type": "string",
      "size": 255
    }, {
      "name": "StateChange",
      "type": "utc",
      "size": 4
    }, {
      "name": "FirstOccurrence",
      "type": "utc",
      "size": 4
    }, {
      "name": "LastOccurrence",
      "type": "utc",
      "size": 4
    }, {
      "name": "RowSerial",
      "type": "integer",
      "size": 4
    }
  ]],
  "rows": [{
    "Identifier": "Startup@sol9-build1",
    "Serial": 12469,
    "Node": "sol9-build1",
    "NodeAlias": "",
    "AlertKey": "",
    "Severity": 0,
    "Summary": "ObjectServer NCOMS on sol9-build1 started at
Wed Jul 04 15:27:57 2012",
    "StateChange": 1341412082,
    "FirstOccurrence": 1341411978,
    "LastOccurrence": 1341412077,
    "RowSerial": 12469
  }, {
    "Identifier": "ProfilerEnableToggle@NCOMS:sol9-build1",
    "Serial": 12468,
    "Node": "sol9-build1",
    "NodeAlias": "",
    "AlertKey": "",
    "Severity": 0,
    "Summary": "ObjectServer NCOMS Profiler enabled at
Wed Jul 04 15:27:56 2012",
    "StateChange": 1341412077,
    "FirstOccurrence": 1341411976,
    "LastOccurrence": 1341412076,
    "RowSerial": 12468
  }, {
    "Identifier": "JUnitEventTestInstance###0",
    "Serial": 12469,
    "Node": "sol9-build1",
    "NodeAlias": "",
    "AlertKey": "JUnitEventInstance",
    "Severity": 0,
    "Summary": "This is a test event generated by the
JUnit REST Event Tests. (0)",
    "StateChange": 1341412184,
    "FirstOccurrence": 1341411772,
    "LastOccurrence": 1341412074,
    "RowSerial": 12468
  }, {
    "Identifier": "Shutdown@sol9-build1",
    "RowSerial": 12519
  }
]
***** TRUNCATED *****

```

```
    }],  
    "affectedRows": 12  
  }  
}
```

Exemple de requête PATCH de collection de tables

Mettez à jour les colonnes Location, LastOccurrence, Acknowledged, OwnerUID et OwnerGID de toutes les lignes de la table alerts.status.

```
PATCH /objectserver/restapi/alerts/status HTTP/1.1
```

```
Accept: application/json
```

```
Authorization: Basic dGVzdHVzZXIwMTpuZXRjb29s
```

```
Content-Type: application/json
```

```
Host: localhost
```

```
Connection: keep-alive
```

```
Content-Length: 1092
```

```
{  
  "rowset": {  
    "coldesc": [  
      {  
        "type": "integer",  
        "name": "Acknowledged"  
      },  
      {  
        "type": "string",  
        "name": "Location"  
      },  
      {  
        "type": "integer",  
        "name": "OwnerUID"  
      },  
      {  
        "type": "integer",  
        "name": "OwnerGID"  
      },  
      {  
        "type": "utc",  
        "name": "LastOccurrence"  
      }  
    ],  
    "rows": [  
      {  
        "Location": "UPDATED",  
        "LastOccurrence": 1341412235,  
        "Acknowledged": 1,  
        "OwnerUID": 65534,  
        "OwnerGID": 1  
      }  
    ]  
  }  
}
```

Exemple de réponse PATCH de collection de tables

```
HTTP/1.1 200 OK
```

```
Cache-Control: no-cache
```

```
Server: libnhttpd
```

```
Date: Wed Jul 4 15:32:03 2012
```

```
Connection: Keep-Alive:
```

```
Content-Type: application/json;charset=UTF-8
```

```
Content-Length: 158
```

```
{  
  "entry": {
```

```
    "affectedRows": 10,  
    "uri": "http://localhost/objectserver/restapi/alerts/status"  
  }  
}
```

Exemple de requête DELETE de collection de tables

Supprimez toutes les lignes de la table alerts.status.
DELETE /objectserver/restapi/alerts/status HTTP/1.1
Accept: application/json
Authorization: Basic dGVzdHVzZXIwMTpuZXRjb29s
Host: localhost
Connection: keep-alive

Exemple de réponse DELETE de collection de tables

```
HTTP/1.1 200 OK  
Cache-Control: no-cache  
Server: libnhttpd  
Date: Wed Jul 4 15:38:53 2012  
Connection: Keep-Alive:  
Content-Type: application/json;charset=UTF-8  
Content-Length: 157  
{  
  "entry": {  
    "affectedRows": 10,  
    "uri": "http://localhost/objectserver/restapi/alerts/status"  
  }  
}
```

Exemple de requête GET d'élément de ligne via RowSerial

```
Accept: application/json  
Authorization: Basic dGVzdHVzZXIwMTpuZXRjb29s  
Host: localhost  
Connection: keep-alive
```

Exemple de réponse GET d'élément de ligne via RowSerial

```
Cache-Control: no-cache  
Server: libnhttpd  
Date: Wed Jul 4 15:32:03 2012  
Connection: Keep-Alive:  
Content-Type: application/json;charset=UTF-8  
Content-Length: 5964  
{  
  "rowset": {  
    "osname": "NCOMS",  
    "dbname": "alerts",  
    "tblname": "status",  
    "coldesc": [{  
      "name": "Identifiant",  
      "type": "string",  
      "size": 255  
    }, {  
      "name": "Serial",  
      "type": "integer",  
      "size": 4  
    }, {  
      "name": "Node",  
      "type": "string",  
      "size": 64  
    }, {  
      "name": "NodeAlias",
```



```

    "ExpireTime": 0,
    "ProcessReq": 0,
    "SuppressEsc1": 0,
    "Customer": "",
    "Service": "",
    "PhysicalSlot": 0,
    "PhysicalPort": 0,
    "PhysicalCard": "",
    "TaskList": 0,
    "NmosSerial": "",
    "NmosObjInst": 0,
    "NmosCauseType": 0,
    "NmosDomainName": "",
    "NmosEntityId": 0,
    "NmosManagedStatus": 0,
    "NmosEventMap": "",
    "LocalNodeAlias": "",
    "LocalPriObj": "",
    "LocalSecObj": "",
    "LocalRootObj": "",
    "RemoteNodeAlias": "",
    "RemotePriObj": "",
    "RemoteSecObj": "",
    "RemoteRootObj": "",
    "X733EventType": 0,
    "X733ProbableCause": 0,
    "X733SpecificProb": "",
    "X733CorrNotif": "",
    "ServerName": "NCOMS",
    "ServerSerial": 12510,
    "URL": "",
    "ExtendedAttr": "",
    "OldRow": 0,
    "ProbeSubSecondId": 0,
    "BSM_Identity": ""
  }],
  "affectedRows": 1
}
}

```

Exemple de requête GET d'élément de ligne via KeyField

Sélectionnez une ligne spécifique dans la table alerts.status via la zone clé.

```

GET /objectserver/restapi/alerts/status/kf/12510%3ANCOMS HTTP/1.1
Accept: application/json
Authorization: Basic dGVzdHVzZXIwMTpuZXRjb29s
Host: localhost
Connection: keep-alive

```

Exemple de réponse GET d'élément de ligne via une zone clé

```

HTTP/1.1 200 OK
Cache-Control: no-cache
Server: libnhttpd
Date: Wed Jul 4 15:32:03 2012
Connection: Keep-Alive:
Content-Type: application/json;charset=UTF-8
Content-Length: 5964
{
  "rowset": {
    "osname": "NCOMS",
    "dbname": "alerts",
    "tblname": "status",
    "coldesc": [{
      "name": "Identifiant",

```



```

    "InternalLast": 1341412207,
    "Poll": 0,
    "Type": 1,
    "Tally": 1,
    "Class": 0,
    "Grade": 0,
    "Location": "NOT UPDATED",
    "OwnerUID": 0,
    "OwnerGID": 0,
    "Acknowledged": 0,
    "Flash": 0,
    "EventId": "",
    "ExpireTime": 0,
    "ProcessReq": 0,
    "SuppressEsc1": 0,
    "Customer": "",
    "Service": "",
    "PhysicalSlot": 0,
    "PhysicalPort": 0,
    "PhysicalCard": "",
    "TaskList": 0,
    "NmosSerial": "",
    "NmosObjInst": 0,
    "NmosCauseType": 0,
    "NmosDomainName": "",
    "NmosEntityId": 0,
    "NmosManagedStatus": 0,
    "NmosEventMap": "",
    "LocalNodeAlias": "",
    "LocalPriObj": "",
    "LocalSecObj": "",
    "LocalRootObj": "",
    "RemoteNodeAlias": "",
    "RemotePriObj": "",
    "RemoteSecObj": "",
    "RemoteRootObj": "",
    "X733EventType": 0,
    "X733ProbableCause": 0,
    "X733SpecificProb": "",
    "X733CorrNotif": "",
    "ServerName": "NCOMS",
    "ServerSerial": 12510,
    "URL": "",
    "ExtendedAttr": "",
    "OldRow": 0,
    "ProbeSubSecondId": 0,
    "BSM_Identity": ""
  }],
  "affectedRows": 1
}
}

```

Exemple de requête PATCH d'élément de ligne

Mettez à jour les colonnes Location, LastOccurrence, Acknowledged, OwnerUID et OwnerGID d'une ligne spécifique dans la table alerts.status.

```
PATCH /objectserver/restapi/alerts/status/kf/12510%3ANCOMS HTTP/1.1
```

```
Accept: application/json
```

```
Authorization: Basic dGVzdHVzZXIwMTpuZXRjb29s
```

```
Content-Type: application/json
```

```
Host: localhost
```

```
Connection: keep-alive
```

```
Content-Length: 1092
```

```
{
  "rowset": {
    "coldesc": [
```

```

    {
      "type": "integer",
      "name": "Acknowledged"
    },
    {
      "type": "string",
      "name": "Location"
    },
    {
      "type": "integer",
      "name": "OwnerUID"
    },
    {
      "type": "integer",
      "name": "OwnerGID"
    },
    {
      "type": "utc",
      "name": "LastOccurrence"
    }
  ],
  "rows": [
    {
      "Location": "UPDATED",
      "LastOccurrence": 1341412235,
      "Acknowledged": 1,
      "OwnerUID": 65534,
      "OwnerGID": 1
    }
  ]
}

```

Exemple de réponse PATCH d'élément de ligne

```

HTTP/1.1 200 OK
Cache-Control: no-cache
Server: libnhttpd
Date: Wed Jul 4 15:32:03 2012
Connection: Keep-Alive
Content-Type: application/json;charset=UTF-8
Content-Length: 215
{
  "entry": {
    "affectedRows": 1,
    "uri": "/objectserver/restapi/alerts/status/kf/12510%3ANCOMS"
  }
}

```

Exemple de requête DELETE d'élément de ligne

Supprimez une ligne spécifique dans la table alerts.status.

```

DELETE /objectserver/restapi/alerts/status/kf/12621%3ANCOMS HTTP/1.1
Accept: application/json
Authorization: Basic dGVzdHVzZXIwMTpuZXRjb29s
Host: localhost
Connection: keep-alive

```

Exemple de réponse DELETE d'élément de ligne

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Server: libnhttpd
Date: Wed Jul 4 15:38:53 2012
Connection: Keep-Alive:
Content-Type: application/json;charset=UTF-8
Content-Length: 207
{
  "entry": {
    "affectedRows": 1,
    "uri": "/objectserver/restapi/alerts/status/kf/12621%3ANCOMS"
  }
}
```

Exemple de requête POST de fabrication de commandes SQL

```
POST /objectserver/restapi/sql/factory HTTP/1.1
Accept: application/json
Authorization: Basic dGVzdHVzZXIwMTpuZXRjb29s
Content-Type: application/json
Content-Length: 64
Host: localhost
Connection: keep-alive
{
  "sqlcmd": "drop user 'testuser01';"
}
```

Référence associée:

«Requête POST de fabrication de commandes SQL», à la page 27
Conditions requises pour une requête POST sur une fabrique de commandes SQL pour exécuter des commandes SQL arbitraires dans l'ObjectServer.

Exemple de réponse POST de fabrication de commandes SQL

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Server: libnhttpd
Date: Wed Jul 4 15:38:53 2012
Connection: Keep-Alive
Content-Type: application/json;charset=UTF-8
Content-Length: 124
{
  "rowset": {
    "osname": "NCOMS",
    "affectedRows": 0
  }
}
```

Exemple de requête GET d'informations système

```
GET /objectserver/sysinfo HTTP/1.1
Accept: application/json
Authorization: Basic dGVzdHVzZXIwMTpuZXRjb29s
Host: localhost
Connection: keep-alive
```

Exemple de réponse GET d'informations système

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Server: libnhttpd
Date: Wed Jul 4 15:38:53 2012
Connection: Keep-Alive
Content-Type: application/json;charset=UTF-8
Content-Length: 412
{
  "compile": {
    "full_details": "Tuesday June 26 17:12:01 BST 2012 on
hurccsol.hursley.ibm.com (SunOS 5.9 Generic_118558-30)",
    "date": "Tuesday June 26 17:12:02 BST 2012",
    "machine": "hurccsol.hursley.ibm.com",
    "system": "SunOS 5.9 sparc",
    "build_version": "750.CAPPL.01"
  },
  "rest": {
    "version": "v1.0",
    "major": 1,
    "minor": 0
  },
  "oslc": {
    "version": "v1.0",
    "major": 1,
    "minor": 0
  }
}
```

Exemple de requête GET d'élément d'informations système

```
GET /objectserver/sysinfo/compile HTTP/1.1
Accept: application/json
Authorization: Basic dGVzdHVzZXIwMTpuZXRjb29s
Host: localhost
Connection: keep-alive
```

Exemple de réponse GET d'élément d'informations système

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Server: libnhttpd
Date: Wed Jul 4 15:38:53 2012
Connection: Keep-Alive:
Content-Type: application/json;charset=UTF-8
Content-Length: 286
{
  "compile": {
    "full_details": "Tuesday June 26 17:12:01 BST 2012 on
hurccsol.hursley.ibm.com (SunOS 5.9 Generic_118558-30)",
    "date": "Tuesday June 26 17:12:02 BST 2012",
    "machine": "hurccsol.hursley.ibm.com",
    "system": "SunOS 5.9 sparc",
    "build_version": "750.CAPPL.01"
  }
}
```

Fichier de configuration JSON avec paramètres de type MIME et en-têtes HTTP

Cet exemple montre un fichier de configuration `$OMNIHOME/etc/libnhttpd/json` qui est édité pour définir des paramètres de type MIME et des en-têtes HTTP dans des réponses HTTP qui sont renvoyées par l'interface HTTP et l'interface OSLC. Pour activer les paramètres de type MIME et les en-têtes HTTP, activez la propriété `NHttpd.ConfigFile`.

Les sections qui activent les paramètres de type MIME et les en-têtes HTTP sont les suivantes :

httpResponse

Définit les en-têtes HTTP se trouvant dans les réponses HTTP qui sont renvoyées par l'interface HTTP et l'interface OSLC. Cette section comporte les sous-sections suivantes :

corsHeaders

Remplace les en-têtes HTTP CORS (Cross-Origin Resource Sharing). Par défaut, les en-têtes par défaut sont remplacés pour indiquer que l'en-tête HTTP **Location** est autorisé et exposé. Ce paramètre est requis pour les messages HTTP 201 Create responses.

httpHeaders

Pour les en-têtes HTTP définis par l'utilisateur. Ces en-têtes sont ajoutés à toutes les réponses HTTP. Utilisez cette section pour ajoutez des valeurs statiques pour les clients. Un modèle d'en-tête est fourni dans l'exemple.

mimeTypes

Cette section affecte une extension de fichier, par exemple `.html`, à un type MIME. Lorsque la gestion de fichiers est activée, ces définitions sont utilisées pour déterminer le type MIME pour le fichier. Elles définissent également l'en-tête HTTP Content-Type afin que les navigateurs puissent gérer le fichier correctement. Le fichier `$OMNIHOME/etc/libnhttpd/json` contient un jeu par défaut de paramètres de type MIME auquel vous pouvez ajouter des extensions de fichier.

Exemple

```
{
  "comment" : "Ce fichier fournit des données de configuration supplémentaires
à la bibliothèque de sockets HTTP imbriquée (libnhttpd).",
  "httpResponse" : {
    "comment" : "Cette section définit un ensemble d'éléments statiques définis
par l'utilisateur qui doivent être renvoyés dans une réponse HTTP, tels que les
en-têtes HTTP.",
    "corsHeaders" : [
      {
        "name" : "Access-Control-Allow-Headers",
        "value" : "Location"
      },
      {
        "name" : "Access-Control-Expose-Headers",
        "value" : "Location"
      }
    ],
    "httpHeaders" : [
    ]
  },
  "mimeTypes" : {
```

```

    "_comment" : "Cette section mappe des types MIME sur des extensions
de fichier.
Elle est utilisée par libnhttpd pour déterminer le type MIME pour un fichier
qui doit être géré à partir de son URI de gestion de fichiers.",
    "application/json" : [
        "json"
    ],
    "application/rdf+xml" : [
        "rdf"
    ],
    "application/xslt+xml" : [
        "xsl", "xslt"
    ],
    "image/jpeg" : [
        "jpg", "jpeg"
    ],
    "image/gif" : [
        "gif"
    ],
    "image/png" : [
        "png"
    ],
    "text/css" : [
        "css"
    ],
    "text/javascript" : [
        "js"
    ],
    "text/HTML" : [
        "htm", "html"
    ],
    "text/plain" : [
        "txt", "log"
    ],
    "text/xml" : [
        "xml"
    ]
}
}

```

Référence associée:

«Propriétés ObjectServer contrôlant l'interface HTTP et l'interface OSLC», à la page 2

Propriétés ObjectServer contrôlant les interfaces HTTP et OSLC.

Annexe C. Lists des abréviations

La documentation d'API de l'interface HTTP ObjectServer et de l'interface OSLC ObjectServer utilise les abréviations et termes suivants.

HTTP Hyper Text Transfer Protocol. HTTP version 1.1 est défini dans RFC2616. Sauf indication contraire, le terme HTTP dans le présent document désigne à la fois HTTP et HTTPS.

HTTPS Hyper Text Transfer Protocol Secure, défini dans RFC2818.

JazzSM Jazz for Service Management, disponible à l'adresse <https://www.ibm.com/developerworks/mydeveloperworks/groups/service/html/communityview?communityUuid=69ec672c-dd6b-443d-add8-bb9a9a490eba>.

JSON JavaScript Object Notation, défini dans ECMA-262.

MIME Multipurpose Internet Mail Extensions. Les types de support MIME sont définis dans *IANA MIME Media Types*.

OSLC Open Services for Lifecycle Collaboration, défini dans <http://open-services.net>.

REST Representational State Transfer, décrit initialement de façon informelle dans le document *Architectural Styles and the Design of Network-based Software Architectures*.

URI Uniform Resource Identifier, défini dans RFC3986.

XML eXtensible Markup Language, défini par W3C.

Remarques

Le présent document peut contenir des informations ou des références concernant certains produits, logiciels ou services IBM non annoncés dans ce pays. Pour plus de détails, référez-vous aux documents d'annonce disponibles dans votre pays, ou adressez-vous à votre partenaire commercial IBM. Toute référence à un produit, programme ou service IBM n'implique pas que seul ce produit, programme ou service IBM puisse être utilisé. Tout autre élément fonctionnellement équivalent peut être utilisé, s'il n'enfreint aucun droit d'IBM. Il est de la responsabilité de l'utilisateur d'évaluer et de vérifier lui-même les installations et applications réalisées avec des produits, logiciels ou services non expressément référencés par IBM.

IBM peut détenir des brevets ou des demandes de brevet couvrant les produits mentionnés dans le présent document. La remise de ce document ne vous octroie aucun droit de licence sur ces brevets ou demandes de brevet. Si vous désirez recevoir des informations concernant l'acquisition de licences, veuillez en faire la demande par écrit à l'adresse suivante :

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Pour le Canada, veuillez adresser votre courrier à :

IBM Director of Commercial Relations
IBM Canada Ltd
3600 Steeles Avenue East
Markham, Ontario
L3R 9Z7 Canada

Pour obtenir des informations sur les licences concernant les produits utilisant un jeu de caractères codé sur deux octets, contactez le service de propriété intellectuelle d'IBM de votre pays ou envoyez vos demandes par écrit à l'adresse suivante :

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

Le paragraphe suivant ne s'applique ni au Royaume-Uni, ni dans aucun pays dans lequel il serait contraire aux lois locales : LE PRESENT DOCUMENT EST LIVRE "EN L'ETAT" SANS AUCUNE GARANTIE EXPLICITE OU IMPLICITE. IBM DECLINE NOTAMMENT TOUTE RESPONSABILITE RELATIVE A CES INFORMATIONS EN CAS DE CONTREFACON AINSI QU'EN CAS DE DEFAUT D'APTITUDE A L'EXECUTION D'UN TRAVAIL DONNE. Certaines juridictions n'autorisent pas l'exclusion des garanties implicites, auquel cas l'exclusion ci-dessus ne vous sera pas applicable.

Le présent document peut contenir des inexactitudes ou des coquilles. Ce document est mis à jour périodiquement. Chaque nouvelle édition inclut les mises à jour. IBM peut, à tout moment et sans préavis, modifier les produits et logiciels décrits dans ce document.

Les références à des sites Web non IBM sont fournies à titre d'information uniquement et n'impliquent en aucun cas une adhésion aux données qu'ils contiennent. Les éléments figurant sur ces sites Web ne font pas partie des éléments du présent produit IBM et l'utilisation de ces sites relève de votre seule responsabilité.

IBM pourra utiliser ou diffuser, de toute manière qu'elle jugera appropriée et sans aucune obligation de sa part, tout ou partie des informations qui lui seront fournies.

Les licenciés souhaitant obtenir des informations permettant : (i) l'échange de données entre des logiciels créés de façon indépendante et d'autres logiciels (dont celui-ci), et (ii) l'utilisation mutuelle des données ainsi échangées, doivent adresser leur demande à :

IBM Corporation
958/NH04
IBM Centre, St Leonards
601 Pacific Hwy
St Leonards, NSW, 2069
Australia

IBM Corporation
896471/H128B
76 Upper Ground
London SE1 9PZ
United Kingdom

IBM Corporation
JBF1/SOM1
294 Route 100
Somers, NY, 10589-0100
United States of America

Ces informations peuvent être soumises à des conditions particulières, prévoyant notamment le paiement d'une redevance.

Le logiciel sous licence décrit dans ce document et tous les éléments sous licence disponibles s'y rapportant sont fournis par IBM conformément aux dispositions de l'ICA, des Conditions internationales d'utilisation des logiciels IBM ou de tout autre accord équivalent.

Les données de performance indiquées dans ce document ont été déterminées dans un environnement contrôlé. Par conséquent, les résultats peuvent varier de manière significative selon l'environnement d'exploitation utilisé. Certaines mesures évaluées sur des systèmes en cours de développement ne sont pas garanties sur tous les systèmes disponibles. En outre, elles peuvent résulter d'extrapolations. Les résultats peuvent donc varier. Il incombe aux utilisateurs de ce document de vérifier si ces données sont applicables à leur environnement d'exploitation.

Les informations concernant des produits non IBM ont été obtenues auprès des fournisseurs de ces produits, par l'intermédiaire d'annonces publiques ou via d'autres sources disponibles. IBM n'a pas testé ces produits et ne peut confirmer l'exactitude de leurs performances ni leur compatibilité. Elle ne peut recevoir aucune réclamation concernant des produits non IBM. Toute question concernant les performances de produits non IBM doit être adressée aux fournisseurs de ces produits.

Toute instruction relative aux intentions d'IBM pour ses opérations à venir est susceptible d'être modifiée ou annulée sans préavis, et doit être considérée uniquement comme un objectif.

Le présent document peut contenir des exemples de données et de rapports utilisés couramment dans l'environnement professionnel. Ces exemples mentionnent des noms fictifs de personnes, de sociétés, de marques ou de produits à des fins illustratives ou explicatives uniquement. Toute ressemblance avec des noms de personnes, de sociétés ou des données réelles serait purement fortuite.

LICENCE DE COPYRIGHT :

Le présent logiciel contient des exemples de programmes d'application en langage source destinés à illustrer les techniques de programmation sur différentes plateformes d'exploitation. Vous avez le droit de copier, de modifier et de distribuer ces exemples de programmes sous quelque forme que ce soit et sans paiement d'aucune redevance à IBM, à des fins de développement, d'utilisation, de vente ou de distribution de programmes d'application conformes aux interfaces de programmation des plateformes pour lesquels ils ont été écrits ou aux interfaces de programmation IBM. Ces exemples de programmes n'ont pas été rigoureusement testés dans toutes les conditions. Par conséquent, IBM ne peut garantir expressément ou implicitement la fiabilité, la maintenabilité ou le fonctionnement de ces programmes.

Des parties de ce produit contiennent du logiciel développé par Daniel Veillard.

- libxml2-2.7.8

Le logiciel libxml2-2.7.8 est distribué conformément au contrat de licence suivant :

© Copyright 1998-2003 Daniel Veillard.

All Rights Reserved. Toute personne possédant une copie de ce Logiciel et des fichiers de documentation associés (le «Logiciel») est autorisée gratuitement à exploiter le Logiciel sans restriction, y compris et sans limitation à utiliser, copier, modifier, fusionner, publier, distribuer, octroyer une sous-licence, et/ou vendre des copies du logiciel et à autoriser les personnes auxquelles le Logiciel est fourni à en faire de même, sous réserve des conditions suivantes :

Les déclarations relatives au copyright ci-dessus et cette déclaration de permission, doivent être incluses dans toutes les copies ou toute partie substantielle du Logiciel.

LE LOGICIEL EST FOURNI «EN L'ETAT» SANS AUCUNE GARANTIE EXPLICITE OU IMPLICITE. IBM DECLINE NOTAMMENT TOUTE RESPONSABILITE RELATIVE A CES INFORMATIONS EN CAS DE CONTREFAÇON AINSI QU'EN CAS DE DEFAUT D'APTITUDE A L'EXECUTION D'UN TRAVAIL DONNE. DANIEL VEILLARD NE POURRA EN AUCUN CAS ÊTRE TENU RESPONSABLE POUR TOUT DOMMAGE, QUELLES QU'EN SOIT LA CAUSE ET LES RESPONSABILITES, CONTRACTUELLES OU NON, OU PAR FAUTE LIEE A L'UTILISATION DE CE LOGICIEL.

Sauf indication contraire dans cette notice, le nom de Daniel Veillard ne doit pas être utilisé à des fins de publicité ou de promotion de ce Logiciel sans autorisation écrite préalable de Daniel Veillard.

Si vous visualisez ces informations en ligne, il se peut que les photographies et illustrations en couleur n'apparaissent pas à l'écran.

Marques

AIX, IBM, le logo IBM, ibm.com, Informix, Netcool, System z, Tivoli, et Tivoli Enterprise Console sont des marques d'International Business Machines Corporation aux Etats-Unis et/ou dans certains autres pays.

Adobe, Acrobat, Portable Document Format (PDF), PostScript ainsi que toutes les marques incluant Adobe sont des marques d'Adobe Systems Incorporated aux Etats-Unis et/ou dans certains autres pays.



Java ainsi que tous les logos et toutes les marques incluant Java sont des marques de Sun Microsystems, Inc. aux Etats-Unis et/ou dans certains autres pays.

Linux est une marque de Linus Torvalds aux Etats-Unis et/ou dans certains autres pays.

Microsoft, Windows, Windows NT et le logo Windows sont des marques de Microsoft Corporation aux Etats-Unis et/ou dans certains autres pays.

UNIX est une marque de The Open Group aux Etats-Unis et/ou dans certains autres pays.

Les autres noms de sociétés, de produits et de services peuvent appartenir à des tiers.



SC43-0826-00

