generix Collaborative Entreprise

Release note socle technique 8.1

Type diffusion Externe **Auteur** *Equipe Recherche et Innovation GCE*

Révision *7* **Date de création** *14/11/2013*

Ce document résume les nouvelles fonctionnalités disponibles sur le socle technique pour la version GCE 1.6.2

Ce socle se nomme 8.1.

Ce document est un condensé des nouveautés fournies par le socle.

Des guides utilisateurs plus complets sont disponibles sur certains sujets traités.

Société	:	***GENERIX Group***	Titre	:	*Projet : socle technique* 8.1
Dernière modification	:	Administrateur	Sujet	:	*Release note socle technique* 8.1
Temps de modification	:	*1104*	Résumé	:	
Nom du fichier		*Release Note technique Socle 8.1.docx*			
Liste de diffusion		Interne & externe			

Siège Social et Agence Nord
Immeuble le Verdi
6 rue du Moulin de Lezennes
BP 10215
59654 VILLENEUVE D'ASCQ Cedex
Tél. : +33 (0)3 20 41 48 00
Fax. : +33 (0)3 20 41 48 09
Service Technique :
Tél. : +33 (0)3 20 41 48 07
Fax. : +33 (0)3 20 41 48 08

Agence Région Parisienne

69/71 rue Beaubourg
75003 PARIS

Tél. : +33 (0)1 47 49 36 66
Fax. : +33 (0)1 47 51 22 17

Table des matières

1 Outils de mesures

1.1 Consommation mémoire instantanée

1.1.1 Objectif

Avoir facilement une idée de la mémoire instantanée consommée avec un historique facilement accessible par les développeurs.

1.1.2 Pré requis

Socle 8.1 et suivants.

1.1.3 Configuration

L'état de la mémoire est disponible au niveau du MDC de Log4J uniquement en niveau debug (Rappel : le MDC est une Map de données applicatives alimentée au fil de l'eau dans GCE).

Le MDC_CODE clé est %X{MEMORY}

Exemple de paramétrage :

```
[%d]¤[%X{DOREQUEST_START_TIME}]¤[%X{DOREQUEST_END_TIME}]¤[%X{MEMORY}]¤%p
```

Exemple de résultat :

```
[2013-10-07 11:41:59,840]¤[11:41:59:782]¤[11:41:59:839]¤[233024:31785]¤
[2013-10-07 11:41:59,902]¤[11:41:59:844]¤[11:41:59:901]¤[233024:28922]¤
[2013-10-07 11:42:00,192]¤[11:41:59:903]¤[11:42:00:191]¤[233024:41402]¤
[2013-10-07 11:42:00,407]¤[11:42:00:193]¤[11:42:00:407]¤[233024:25837]¤
[2013-10-07 11:42:01,145]¤[11:42:00:409]¤[11:42:01:144]¤[233152:45482]¤
[2013-10-07 11:42:01,329]¤[11:42:01:147]¤[11:42:01:328]¤[233536:54857]¤
```

Il est nécessaire de passer le logger activite au niveau debug pour que le MDC soit alimenté :

```
<logger name="activite" additivity="false">
    <level value="debug"/>
    <appender-ref ref="activiteAppender"/>
</logger>
```

1.2 Timestamp de début et de fin de requête

1.2.1 Objectif

Avoir en permanence le timestamp de début et de fin de requête. Le résultat se trouve dans le log activite.log

1.2.2 Pré requis

Socle 8.1 et suivants.

1.2.3 Configuration

Les timestamp sont disponibles au niveau du MDC de log4j (Rappel : le MDC est une Map de données applicatives alimentée au fil de l'eau dans GCE et disponible dans le layout de chaque appender de Log4J).

Les MDC_CODE clés sont :
%X{DOREQUEST_START_TIME} et %X{DOREQUEST_END_TIME}

Exemple de paramétrage :

```
[%d]¤[%X{DOREQUEST_START_TIME}]¤[%X{DOREQUEST_END_TIME}]¤
```

Exemple de résultat :

```
[2013-10-21 17:32:36,045]¤[17:32:35:378]¤[17:32:36:045]¤
[2013-10-21 17:32:36,296]¤[17:32:36:047]¤[17:32:36:295]¤
[2013-10-21 17:32:36,826]¤[17:32:36:422]¤[17:32:36:826]¤
[2013-10-21 17:32:37,569]¤[17:32:36:950]¤[17:32:37:569]¤
[2013-10-21 17:36:13,766]¤[17:36:13:061]¤[17:36:13:766]¤
[2013-10-21 17:36:15,043]¤[17:36:13:914]¤[17:36:15:043]¤
[2013-10-21 17:36:15,153]¤[17:36:15:045]¤[17:36:15:153]¤
[2013-10-21 17:36:15,247]¤[17:36:15:154]¤[17:36:15:247]¤
[2013-10-21 17:36:15,362]¤[17:36:15:250]¤[17:36:15:362]¤
```

Remarque : la valeur du DOREQUEST_END_TIME est très proche de celle donnée par le %d de Log4j, leur évaluation se faisant quasiment en même temps.

2 Outils techniques

2.1 Nom de la feuille de style disponible dans la feuille de style.

2.1.1 Objectif

Avoir à disposition le nom physique de la feuille de style.

2.1.2 Pré requis

Socle 8.0 et suivants.

2.1.3 Configuration

Balise à positionner dans la feuille de style sous les nœuds imports sous la forme :

```
<xsl:param name="fileref"/>
```

Le nom physique de la feuille est disponible sous le nom $fileref :

```
<xsl:value-of select="$fileref"/>
```

2.2 Servlets utilitaires Leon et Wolvie

2.2.1 Objectif

Disposer d'un outil indépendant de la session GCE pour libérer les ressources et éventuellement tuer la session http.
Les 2 Servlets effectuent le même travail de base qui est de :

- libérer les ressources, y compris les locks issus des requêtes SQL
- supprimer les requêtes http en attente

La Servlet nommée Leon va en plus invalider la session http ce qui provoquera la destruction de toutes les ressources.

Ces actions sont effectuées par défaut sur la session http du poste client qui demande l'exécution d'une des 2 Servlets.

Il est possible de demander le traitement de n'importe quel session à partir du moment où son Uuid est connu via un autre process nommé Nikita

2.2.2 Pré requis

Socle 8.1 et suivants.

2.2.3 Configuration

2.2.3.1 Déclaration

La déclaration des servlet est faite dans le web.xml comme toute autre servlet. Elles sont déjà présentes dans le web.xml fournis par GCE.

Exemple de déclaration de la Servlet Leon :

```
<servlet>
    <display-name>Leon</display-name>
    <servlet-name>Leon</servlet-name>
    <servlet-class>fr.generix.technicalframework.application.servlet.Leon</servlet-class>
    <load-on-startup>-1</load-on-startup>
    <security-role-ref>
        <description>Global access role required</description>
        <role-name>sr_gce_overall</role-name>
        <role-link>sr_gce_overall</role-link>
    </security-role-ref>
</servlet>
```

2.2.3.2 Exécution

Une simple URL de la forme suivante permet l'exécution des servlets :

http://localhost:8080/gce162-webgce/Leon
http://localhost:8080/gce162-webgce/Wolvie

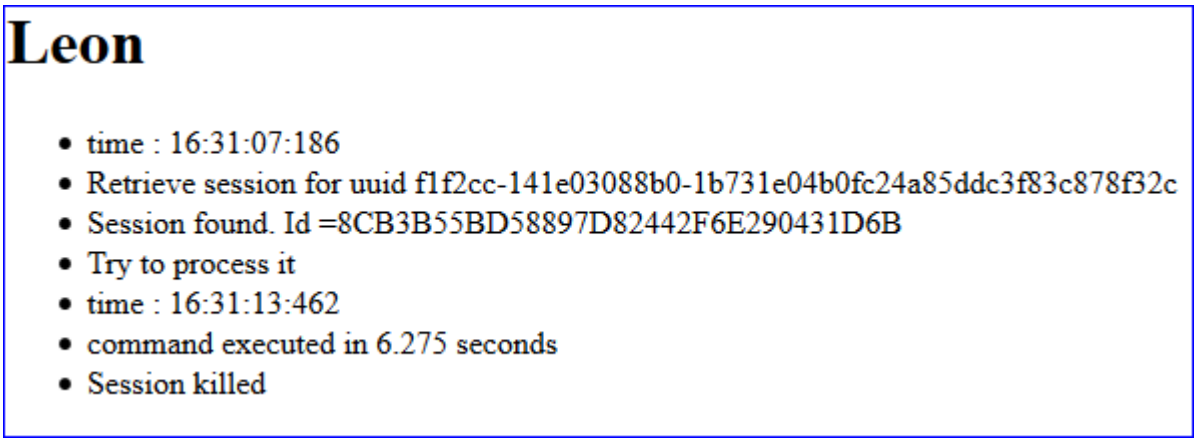
2.2.4 Résultat

Ces Servlet fournissent en retour d'exécution un flux HTML basique qui donne quelques informations essentielles sur le résultat de leur action.

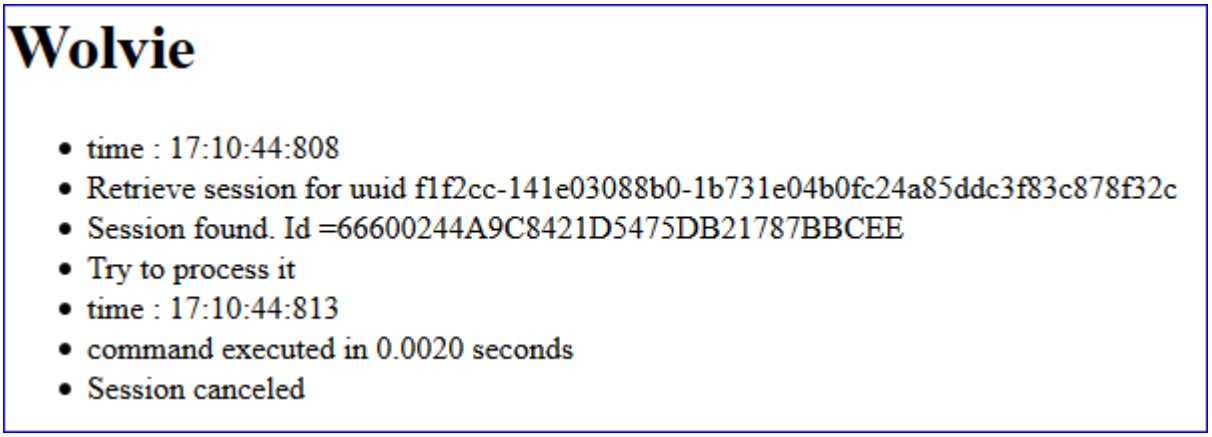
Leon :

Résultat pour une demande locale de type

http://localhost:8080/gce162-webgce/Leon



Wolvie



Le résultat du process Nikita est disponible dans les logs.

2.3 Process Nikita

2.3.1 Objectif

Provoquer à volonté le même traitement que les Servlets Leon et Wolvie.

2.3.2 Pré requis

Socle 8.1 et suivants.

2.3.3 Configuration

Il n'y a pas de configuration spécifique, ce process se lance de façon autonome à raison d'un process par JVM.
Nikita s'alimente via une mise à jour dans la BDD sur la table ut_uuid. Demander une action sur une session d'uuid 12345 revient à faire la requête suivante :

```
update ut_uuid set jsessionid = ? where uuid = 12345"
```

Les valeurs possibles à assigner à la colonne jsessionid sont :

- cancel (équivalent Servlet Wolvie)
- delete (équivalent Servlet Leon)

2.4 Servlet utilitaire Casper

2.4.1 Objectif

Reverse proxy servlet.

Remarque : permet d'éviter les problèmes de cross-domain sur les navigateurs.

2.4.2 Pré requis

Socle 8.1 et suivants.

2.4.3 Configuration

web.xml

```
<servlet>
	<display-name>Casper</display-name>
	<servlet-name>gnx.Casper</servlet-name>
	<servlet-
class>fr.generix.technicalframework.application.servlet.Casper</servlet-class>
	<load-on-startup>-1</load-on-startup>
	<security-role-ref>
		<description>Global access role required</description>
		<role-name>sr_gce_overall</role-name>
		<role-link>sr_gce_overall</role-link>
	</security-role-ref>
</servlet>
<servlet-mapping>
	<servlet-name>gnx.Casper</servlet-name>
	<url-pattern>/Casper</url-pattern>
</servlet-mapping>
```

L'url à exécuter est passée via le paramètre « url ».

Ci-dessous un exemple d'utilisation :

http://localhost:8888/extranet/GCE/Casper?url=http://www.developpez.com

2.5 Solidité du lien entre cache et feuille de style

2.5.1 Objectif

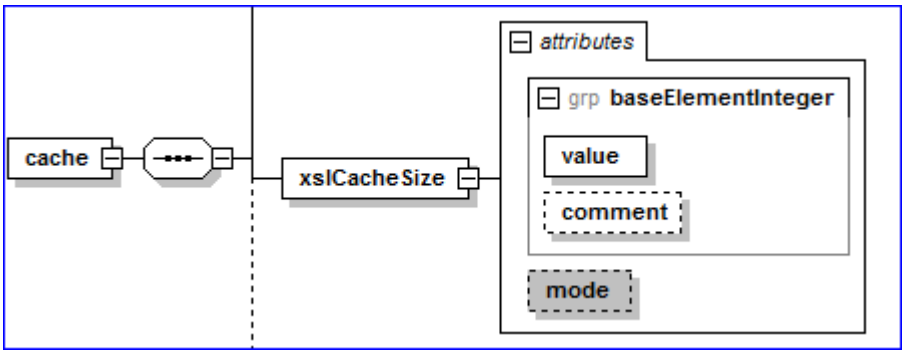
Pour des raisons de performances, les feuilles de style sont conservées dans un cache circulaire. La taille de ce cache est paramétrable dans le gce.properties.xml.
Une nouvelle option pour ce cache permet de définir un lien plus faible entre le cache et son contenu pour permettre à la JVM de récupérer temporairement de la mémoire afin d'absorber une consommation importante sans que les performances de la JVM s'écroule ou pire, qu'une OutOfMemoryException soit levée.

2.5.2 Pré requis

Socle 8.1 et suivants.

2.5.3 Configuration

Un nouvel attribut optionnel dans gce.properties :



Attribut	Utilisation	Signification
mode	Optionnel	3 valeurs possibles : hard : valeur par défaut. Correspond au fonctionnement actuel. Référence forte. Lien indestructible entre le cache et son objet. C'est le cache lui même qui décide de sa politique de séparation de l'objet (s'il est plein par exemple) Soft : Les liens Soft autorisent le GC à supprimer l'objet référencé si le besoin s'en fait sentir. Ainsi, si la charge mémoire est importante, il est garantit que toutes les soft références seront libérées AVANT qu'une OutOfMemoryException ne soit lancée. Cela permet de conserver en cache de gros objet afin d'éviter d'avoir à les recharger plus tard, tout en permettant au GC de les supprimer en cas de besoin. Weak : Les weak références permettent de stocker une référence, mais ne permettent pas d'assurer à elle seule que l'objet stocké sera conservé en mémoire. Les weak références peuvent être libérées par le GC à n'importe quel moment à condition qu'il n'y ait plus de référence forte ou soft sur cet objet. Le GC aura tendance à libérer plus facilement une weak référence qu'une soft, même si le besoin en mémoire n'est pas important.

2.5.4 Exemple de résultats

Pour illustrer l'efficacité de ce nouveau mode au niveau consommation mémoire du cache de feuille de style, un petit cas de test.

Scénario : balayer les onglets du menu, puis accès au portail produit : recherche, sélection d'un produit et balayage des onglets du produit.

Les tests sont fait pour une JVM à 1024 Mo qui est ensuite baissée à 256 Mo pour les 3 modes de gestion des références. Les résultats en nombre de feuilles de style conservées en cache sont les suivants :

JVM / mode	HARD	SOFT	WEAK
1024	46	46	46
256	19 (OOM) (1)	10	4

(1) : l'exception OutOfMemory survient au début de l'ouverture du portail produit. A ce moment-là il y a 19 feuilles de style en mémoire.

3 Nouvelles fonctionnalités

3.1 Log4J multi configuration

3.1.1 Objectif

Permettre plusieurs fichiers de configuration de log4J. Ajouter de la souplesse sur site sans modifier le standard.

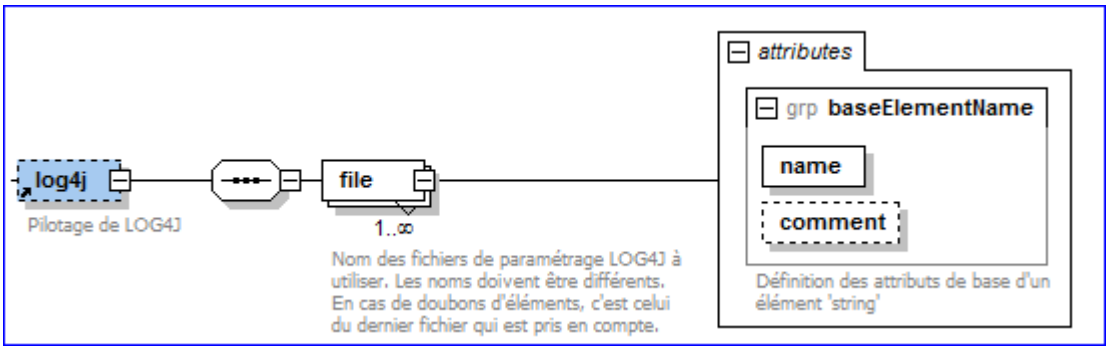
3.1.2 Pré requis

Socle 8.0 et suivants.

3.1.3 Configuration

La configuration s'effectue dans le gce.properties.xml

- La définition issue du schéma :



- Un exemple de configuration :

```
<log4j>
      <file name="log4j/develop_log4j.xml"/>
      <file name="log4j/processus_log4j.xml"/>
</log4j>
```

L'algorithme concatène les différents fichiers pour en créer un seul. En cas de duplication d'élément log4J (Logger, Appender), c'est le dernier dans la liste qui est pris en compte. En mode debug, le fichier de paramétrage résultat est écrit sur disque sous le nom *LOG4JFullConfigurationFile.xml*.

3.2 Listener autour des moteurs de rendu

3.2.1 Objectif

Permettre des traitements fonctionnels annexes autour de la génération finale du document de présentation (html ou pdf).

3.2.2 Pré requis

Socle 8.1 et suivants.

3.2.3 Configuration

3 éléments sont nécessaires pour ajouter ces listeners :

- Le listener lui-même : c'est une classe java spécifique qui dérive de la classe socle :

```
fr.generix.technicalframework.application.presentation.listener.AbstractRenderTransformerListener
```

L'interface RenderTransformerListener demande l'implémentation de la méthode suivante :

```
/**
 * Méthode principale du listener. Appel du traitement.
 *
 * @param eventObject L'événement complexe.
 */
public void handleEvent(RenderTransformerEventObject eventObject);
```

- Sa déclaration dans la configuration, sous forme d'alias de type value dans la section alias_def
- Les paramètres définies dans la feuille de style, dans la balise <xsl:output>, attribut cdata-section-elements. Ces paramètres doivent commencer par le préfix arc. Pour être pris en compte. (arc comme archive qui est le but premier de ces listeners, même si le concept peut être étendu à d'autres besoins).

3.2.3.1 Exemple de paramétrage :

```
<classalias_type name="RENDERTRANSFORMERLISTENER" defFullName="fr.generix.technicalframework.application.presentation.listener.RenderTransformerListenerExample" type="value"/>
```

```
<xsl:output encoding="UTF-8" indent="yes" method="html" cdata-section-elements="doctype-html5 arc.solver=RENDERTRANSFORMERLISTENER arc.filename=MONFILENAME arc.listener=beforeTransform"/>
```

Les 2 paramètres indispensables sont :

- ***arc.solver*** pour indiquer le nom logique du solver de type Listener,
- ***arc.listener*** pour indiquer sur quel événement ce listener doit intervenir.

Attention, ces paramètres et leurs valeurs sont sensibles à la casse.

2 valeurs possibles pour le listener :

- beforeTransform
- afterTransform

3.2.3.2 *Exemple d'implémentation*

La classe abstraite AbstractRenderTransformerListener donne accès aux paramètres issus de la feuille de style à travers la classe RenderTransformerStateHolder. Dans notre exemple, récupération du contenu du paramètre arc.filename :

```
// Récupération des paramètres issus de la feuille de style
Map<String, String> paramMap = getRenderTransformerStateholer().getParamMap();
String filename = paramMap.get("filename");
```

C'est également cette classe qui contiendra le résultat du listener :

```
getRenderTransformerStateholer().setResult(result);
```

L'événement reçu par le listener contient 2 informations :

- Le type du flux :

```
public SOURCE_TYPE getType()
```

- Le type d'événement :

```
public RenderTransformerEvent getEvent()
```

3.3 Gestion de la langue

3.3.1 Objectif

Nouveauté : selon le paramétrage, utiliser la langue du navigateur ou la langue de l'utilisateur comme langue d'affichage de GCE.
Changement de comportement : utiliser systématiquement la langue du navigateur comme langue d'affichage de la mire de login.

3.3.2 Pré requis

Socle 8.1 et suivants.

3.3.3 Configuration

Pour activer la prise en compte de la langue du navigateur, il faut positionner le mode de fonctionnement (policy) dans le gce.properties de la façon suivante :

```
<locale>
        <languages default="ENG" policy="languageFromBrowser">
                <language acceptLanguage="fr" lang="FRA"/>
                <language acceptLanguage="fr-FR" lang="FRA"/>
                <language acceptLanguage="fr-BE" lang="FRA"/>
                <language acceptLanguage="en" lang="ENG"/>
                <language acceptLanguage="en-US" lang="ENG"/>
        </languages>
</locale>
```

La persistance de l'information est gérée par un nouveau cookie : BROWSING_LANGUAGE

Remarque : le paramètre de cinématique 'language' est supprimé.

3.4 Fields calculés

3.4.1 Objectif

Une nouvelle gestion des Fields permet de calculer dynamiquement la valeur d'un <*field*> via un Solver de type ExprSolver. La demande initiale est la mise entre balise d'une partie de la valeur d'un attribut issu d'une recherche oracle text.

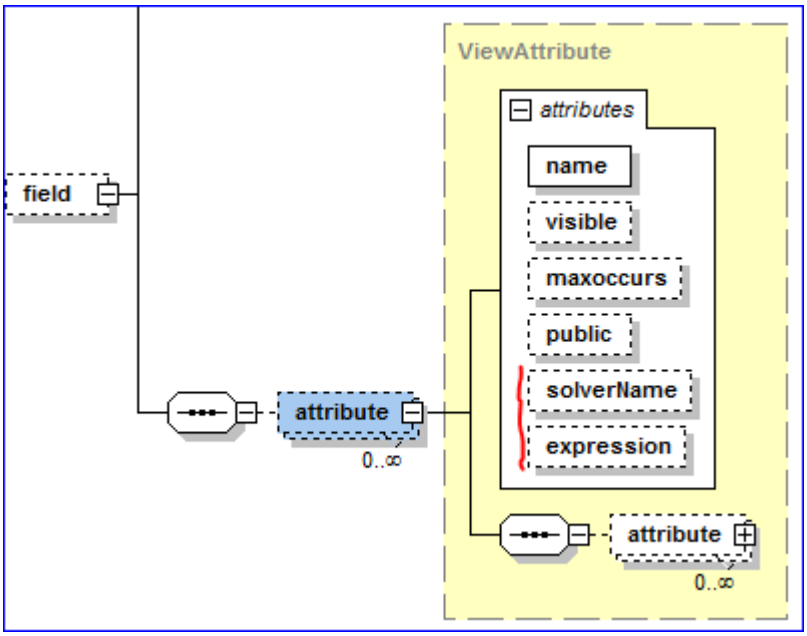
3.4.2 Pré requis

Socle 8.1 et suivants.

3.4.3 Configuration

3.4.3.1 Définition du field

Le schéma *configuration.xsd* évolue pour permettre le paramétrage du field :



Attribut	Utilisation	Signification
solverName	Optionnel	Nom logique du solver . Fait référence à un solver défini dans classalias_def : COMPUTED
expression	Optionnel	Expression de type exprSolver

Exemple :

```
<field>
    <attribute name="Test" public="Test" solverName="COMPUTED" expression="em(dlb('Fil'),'MM')" visible="true"/>
</field>
```

3.4.3.2 Définition du solver

Le ExprSolver nommé COMPUTED se défini comme suit dans la section classalias_def :

```
<classalias_type name="COMPUTED" defFullName="fr.generix.technicalframework.application.solver.business.ExprComputedFieldSolverImpl" type="value"/>
```

3.4.3.3 Des nouvelles fonctions

2 nouvelles fonctions sont disponibles :

Attribut	Disponibilité	Signification
dlb	Ligne de Document/ Hérite de View	Dlb = DocumentLineBusiness. Permet d'accéder à la valeur Business d'un attribut disponible sur la ligne de Document courante (= le row courant pour un Document de type BC4J ou Dynamique). Syntaxe : dlb('NomAttribut'). Exemple : dlb('Fil')
em	Ligne de Document / hérite de View.	Em : balise em. Permet d'encadrer une portion de la valeur d'un attribut par la balise em. Syntaxe : em('String', 'Section') Exemple : em('COMMERCE', 'MM') donne COMMERCE.

3.5 Log4J : nouvelles entrées dans le MDC

3.5.1 Objectif

Rendre disponible des informations supplémentaires dans les traces log4J via le mot clé %X{*MDC_CODE*}.

3.5.2 Pré requis

Socle 8.1 et suivants.

3.5.3 Configuration

Ajouter le mot clé %X{*CODE*} dans le layout du log concerné.

En plus des nouveaux mots clés déjà cités dans ce release note, nous avons les CODE suivants :

<table>
<tr><th>CODE</th><th>Signification</th></tr>
<tr><td>REQUEST_PUBLIC_BUSINESSVIEW</td><td>Le nom public de la BusinessView demandé</td></tr>
<tr><td>KILLSESSION_ORIGIN</td><td>L'origine de la fin de session. Les valeurs possibles sont :
<table>
<tr><th>Code</th><th>Signification</th></tr>
<tr><td>Unknown</td><td>Origine non spécifiée</td></tr>
<tr><td>Exception</td><td>Exception interne</td></tr>
<tr><td>RequestedSessionId invalid</td><td>Le JSessionID n'est pas valide</td></tr>
<tr><td>Id and Uuid inconsistency</td><td>Incohérence entre les valeurs du JSessionID et du Uuid</td></tr>
<tr><td>Action</td><td>Action KillSession</td></tr>
<tr><td>Destroy</td><td>Fin « naturelle » de la session.</td></tr>
<tr><td>ServletService</td><td>Fin de session demandée par la ServletService.</td></tr>
<tr><td>GXWebService</td><td>Fin de session demandée par le WebService.</td></tr>
</table>
</td></tr>
<tr><td>ERROR_CODE</td><td>Le code ut_mes de la dernière exception.</td></tr>
<tr><td>UUID_STATE</td><td>L'état du uuid. Valeurs possibles :
EMPTY, NEW, REUSED</td></tr>
</table>

3.6 QueryTimeOut paramétrable

3.6.1 Objectif

L'objectif est de pouvoir paramétrer finement le timeout de requête SQL. Ce paramétrage permet :

- De ne pas laisser l'utilisateur bloqué sur sa transaction en cours.
- De prévenir toute dérive des performances de GCE.

Cela ne remplace pas un monitoring de la base de données mais permet une gestion fine vue de l'applicatif.

3.6.2 Pré requis

Socle 8.1 et suivants.

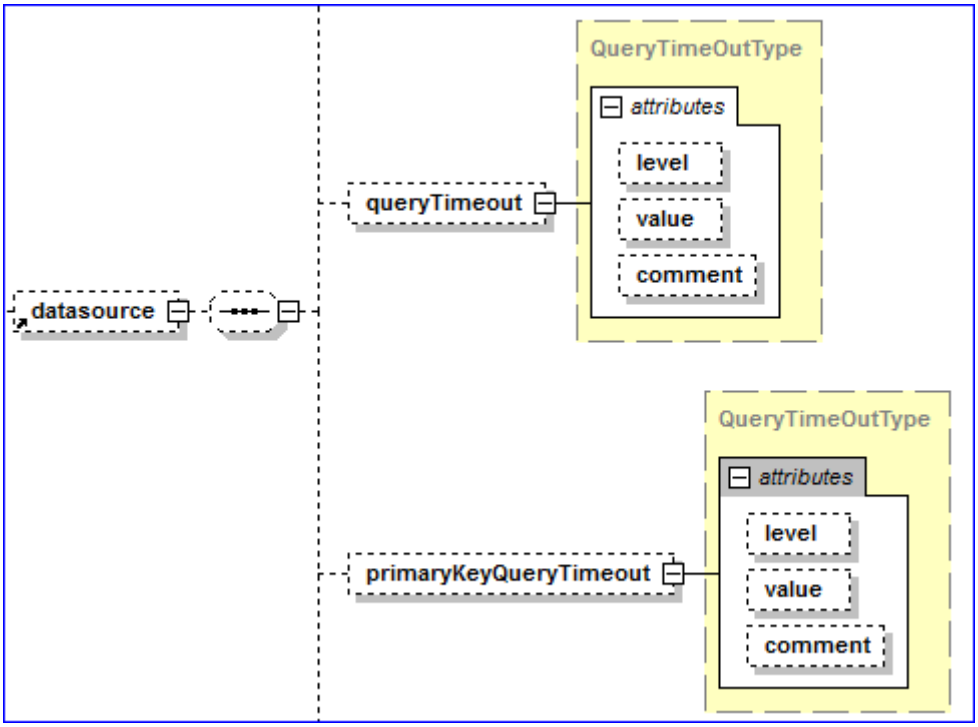
3.6.3 Configuration

Le timeout de requête se positionne sur 3 niveaux de finesse croissante.

3.6.3.1 Le gceProperties

Le paramétrage se trouve dans la section config/application/datasource.

3.6.3.1.1 configuration



Le paramétrage permet de différencier les requêtes sur clé primaire (primaryKeyQueryTimeout) des autres (queryTimeout). En l'absence d'information sur la qualité de la clé, c'est le paramétrage du queryTimeout qui s'applique.

Nœud/Attribut	Signification
queryTimeout	Paramétrage général du timeout de requête.
@level	Niveau d'alerte en cas de dépassement du timeout NONE : pas de paramétrage particulier. La valeur par défaut d'applique WARNING : En cas de dépassement de timeout, l'alerte se fera via log4J par le logger nommé « StatementMonitor ». Voir le paramétrage de ce logger pour spécifier la finesse des alertes. ERROR : valeur par défaut. En cas de dépassement de timeout, une SQLException est levée par le mécanisme BC4J.
@value	La valeur du timeout de la requête, en ms. Ne pas oublier de prendre en compte la période de scrutation du moniteur pour l'évaluation des résultats. Une valeur fixée à -1 signifie pas de timeout. C'est la valeur par défaut des BC4J.
primaryKeyQueryTimeout	Paramétrage du timeout de requête avec accès prévues sur clé primaire.
@level	Idem ci-dessus.
@value	Idem ci-dessus.

Remarque : par défaut, les moniteurs BC4J et Socle ont une période de scan de 1 seconde. Cela signifie que le temps de timeout fixée pour les requêtes est respecté à T + 1sec maximum. Il est possible de fixer cette période de scrutation à une valeur différente via le paramètre property suivant :

```
    <property name="jbo.mom.LongRunningStatementMonitor.idleTime" value="100"/>
</systemProperties>
```

La valeur est en ms. Elle ne peut pas être inférieure à 5 ms. Dans ce cas là, la valeur par défaut de 1000 ms est appliquée.

Dans cet exemple, la période est de 100 ms.

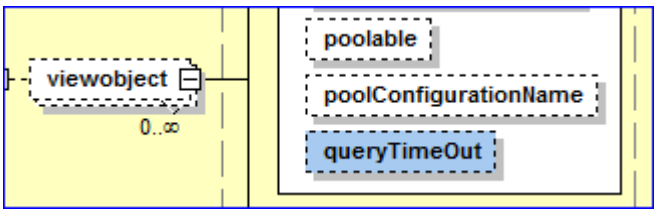
3.6.3.1.2 Exemple de paramétrage

```
<datasource>
    <presentationOverallMaxFetchSize value="1000"/>
    <queryTimeout value="100" level="WARNING"/>
    <primaryKeyQueryTimeout value="20" level="WARNING"/>
</datasource>
```

3.6.3.2 Les ViewObject

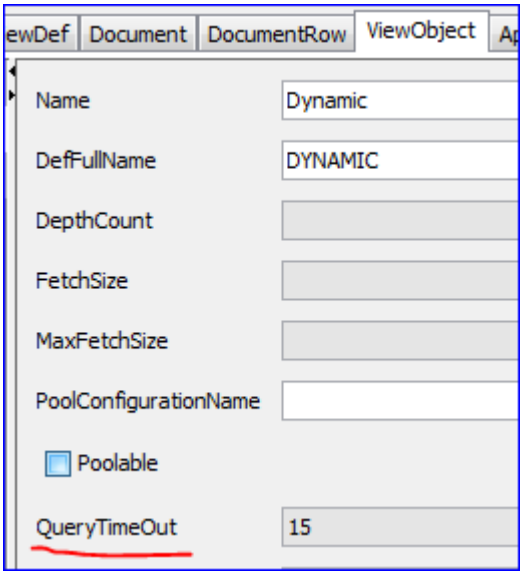
Un nouvel attribut queryTimeOut est disponible pour chaque ViewObject. Il est prioritaire sur celui du gce.properties mais ne fait pas la distinction entre clé primaire ou non (n'a pas de sens au niveau de la ViewObject).

3.6.3.2.1 Configuration



3.6.3.2.2 Exemple de paramétrage

Vu de XDME :



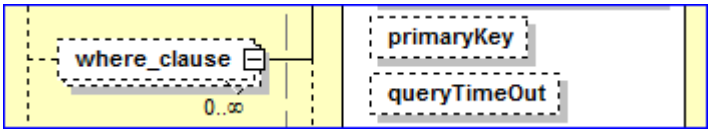
La valeur est en ms. Une valeur de -1 signifie « pas de timeOut »

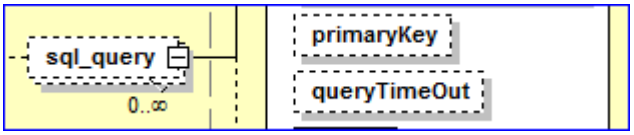
3.6.3.3 Les clés de clause where

Le niveau de paramétrage le plus fin se situe au niveau des clés de viewDef. Chaque clé possède sa propre valeur de QueryTimeOut à associer avec la valeur de l'attribut PrimaryKey. Dans tous les cas, c'est le niveau du gce.properties qui est pris en compte.

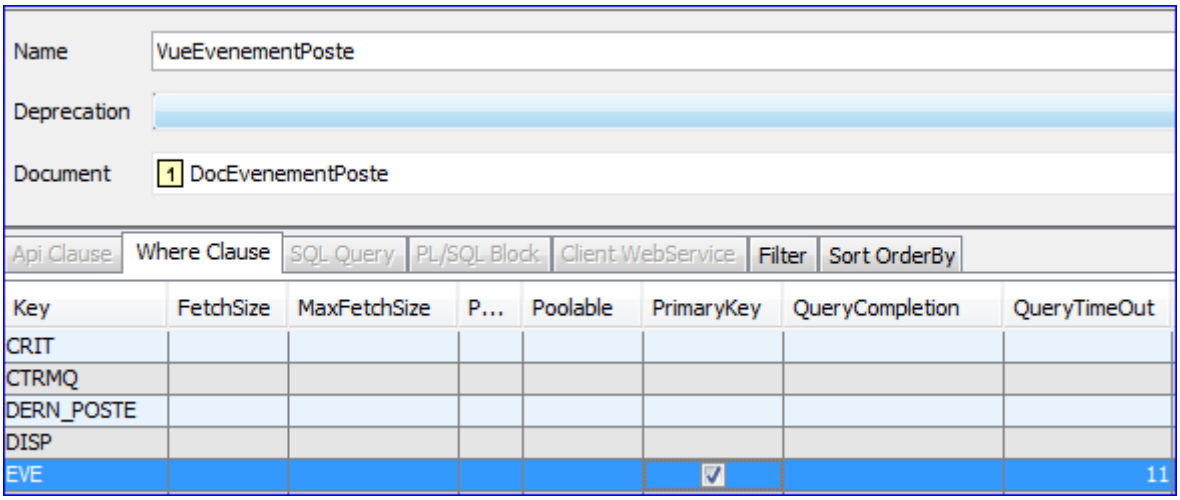
3.6.3.3.1 Configuration

Un nouvel attribut « queryTimeOut » est disponible pour les clés de type *WhereClause* et *SQL Query* :





3.6.3.3.2 Exemple



3.6.3.4 Paramétrage du log pour le level WARNING

Rappel : le niveau WARNING provoque une trace log en cas de dépassement de timeout. Les quantités de traces log disponibles sont variables en fonction du niveau du logger.
Le logger se nomme StatementMonitor, l'appender correspondant est StatementMonitorAppender.

Exemple de déclaration du logger :

```
<logger name="StatementMonitor" additivity="false">
    <level value="debug"/>
    <appender-ref ref="StatementMonitorAppender"/>
</logger>
```

Exemple de déclaration de l'appender :

```
<appender name="StatementMonitorAppender" class="org.apache.log4j.RollingFileAppender">
    <param name="File" value="log/${webmodule}/statementMonitor.log"/>
    <param name="MaxFileSize" value="10MB"/>
    <param name="MaxBackupIndex" value="10"/>
    <layout class="org.apache.log4j.PatternLayout">
        <param name="ConversionPattern" value="[%d]¤%-80m¤%X{VIEWOBJECT}¤%X{SQL_QUERY}¤%X{SQL_PARAM}%n"/>
    </layout>
```

En niveau de trace info, le log ressemble à ceci :

```
id = 4 request completed : timeOut : 100 ms. Executed in 7927.01    ¤JParavCompletViev
id = 7 request completed : timeOut : 100 ms. Executed in 2419.65    ¤JUt_UtiView¤MevUt
id = 8 request completed : timeOut : 100 ms. Executed in 3460.48    ¤JTieView¤MevTie.(
id = 9 request completed : timeOut : 100 ms. Executed in 233.63     ¤JAceView¤MevAce.(
id = 10 request completed : timeOut : 100 ms. Executed in 883.2     ¤JP_1exView¤MevP_1
id = 24 request completed : timeOut : 100 ms. Executed in 826       ¤JPevView¤MevPev.(
```

On y retrouve uniquement les requêtes en timeout, avec une ligne de log par requête en erreur.
On remarquera l'ID qui est un numéro incrémental de requête SQL. Chaque requête a son id propre au monitoring. Il est utile en mode trace, vu plus loin.

En niveau de trace debug le log ressemble à ceci :

```
id = 1 request completed : timeOut : 100 ms. Executed in 43.8        ¤MevView¤CODEN
id = 2 request completed : timeOut : 100 ms. Executed in 4.15        ¤Ut_SocView¤CO
id = 3 request completed : timeOut : 100 ms. Executed in 2.42        ¤MevView¤CODSO
id = 4 request completed : timeOut : 100 ms. Executed in 118.97      ¤JParavComplet
id = 5 request completed : timeOut : 1000 ms. Executed in 340.21     ¤JMediaView¤Me
id = 6 request completed : timeOut : 1000 ms. Executed in 1.06       ¤JMediaView¤Me
id = 7 request completed : timeOut : 100 ms. Executed in 538.93      ¤JUt_UtiView¤M
id = 8 request completed : timeOut : 100 ms. Executed in 3506.5      ¤JTieView¤MevT
id = 9 request completed : timeOut : 100 ms. Executed in 883.09      ¤JAceView¤MevA
id = 10 request completed : timeOut : 100 ms. Executed in 227.13     ¤JP_1exView¤Me
```

On y retrouve toutes les requêtes sql, en timeout ou non, avec une ligne par requête.

Au niveau trace, le log ressemble à ceci

```
id = 1 request completed : timeOut : 100 ms. Executed in 35.52       ¤MevView¤COD
id = 2 request completed : timeOut : 100 ms. Executed in 3.03        ¤Ut_SocView¤
id = 3 request completed : timeOut : 100 ms. Executed in 3.13        ¤MevView¤COD
id = 4 request expired   : timeOut : 100 ms. Elapse time : 112.54    ¤JParavCompl
id = 4 request completed : timeOut : 100 ms. Executed in 128.44      ¤JParavCompl
id = 5 request completed : timeOut : 1000 ms. Executed in 405.75     ¤JMediaView¤
id = 6 request completed : timeOut : 1000 ms. Executed in 1.3        ¤JMediaView¤
id = 7 request expired   : timeOut : 100 ms. Elapse time : 119.36    ¤JUt_UtiView
id = 7 request expired   : timeOut : 100 ms. Elapse time : 139.35    ¤JUt_UtiView
id = 7 request expired   : timeOut : 100 ms. Elapse time : 159.35    ¤JUt_UtiView
```

…

```
id = 7 request expired   : timeOut : 100 ms. Elapse time : 445.46    ¤JUt_UtiView
id = 7 request expired   : timeOut : 100 ms. Elapse time : 465.37    ¤JUt_UtiView
id = 7 request completed : timeOut : 100 ms. Executed in 481.5       ¤JUt_UtiView
id = 8 request expired   : timeOut : 100 ms. Elapse time : 109.44    ¤JTieView¤Me
```

On y retrouve une ligne pour chaque passage du monitor. Ce mode est extrêmement verbeux et est déconseillé hormis pour suivre pas à pas une requête qui reste bloquée.

3.7 Nouvelle action de cinématique

3.7.1 Objectif

Permettre le rechargement des BusinessFlow.

3.7.2 Pré requis

Socle 8.1 et suivants.

3.7.3 Configuration

Action de cinématique invalidateBF(BusinessView) à positionner dans la cinématique de l'URL.

Cette action est déclarée dans la section *action_def* dans le fichier de configuration *configuration_base.xml*.

```
<action name="invalidateBF" class="fr.generix.technicalframework.application.action.InvalidateBF">
    <required>
        <param name="businessView" maxOccurs="1" minOccurs="1" type="BusinessView"/>
    </required>
</action>
```

3.8 CrudManager

Pour rappel, le webservice WSCrudManager permet d'effectuer des transactions de base avec la base de données GCE : création, consultation, modification ou suppression d'une donnée (via une ViewObject).

3.8.1 Objectif

A l'instar des ResolverClass (SolverName) positionnés sur les clés de retrieve (Where Clause/SQL Query) au niveau des ViewDef, le ResolverClass définit au niveau ViewObject est maintenant pris en charge (pour le CrudManager).

3.8.2 Pré requis

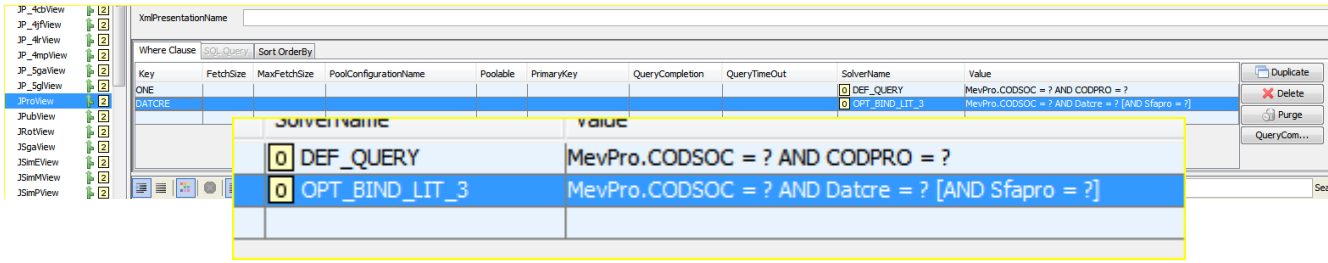
Socle 8.1 et suivants.

3.8.3 Mise en place

Le SolverName est défini dans la configuration :

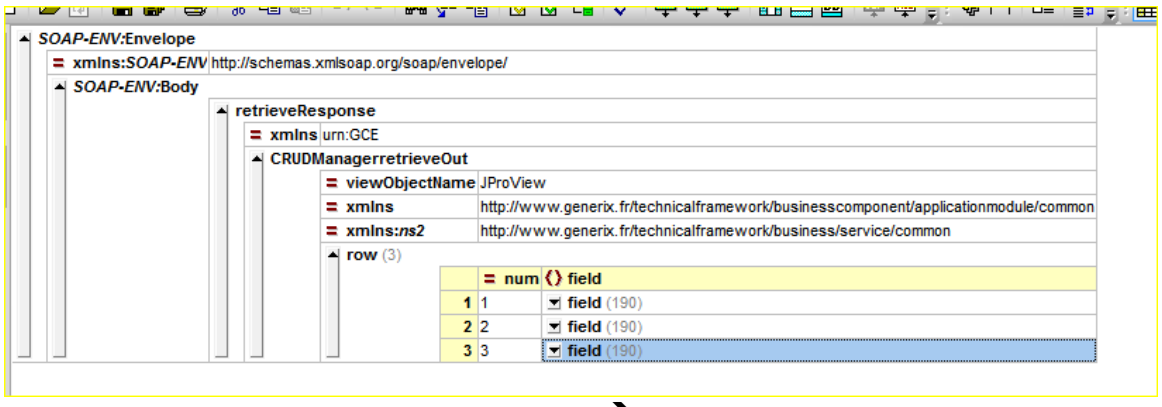
3.8.3.1 configuration

Vu de XDME :



3.8.3.2 Exemple d'utilisation

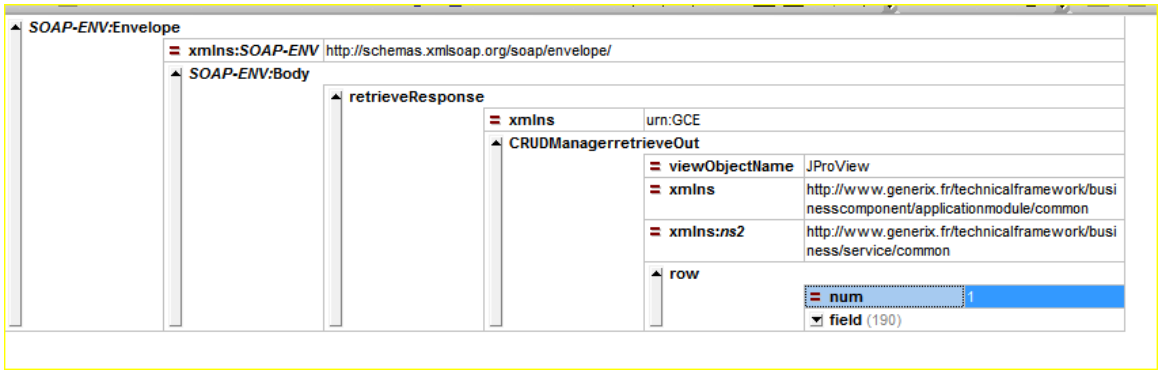
```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:GCE"
xmlns:com="http://www.generix.fr/technicalframework/businesscomponent/applicationmodu
le/common"
xmlns:ser="http://www.generix.fr/technicalframework/business/webservice/server">
   <soapenv:Header/>
   <soapenv:Body>
      <urn:retrieve>
         <com:ctx entity="1" target="GART" user="DEMO" password="DEMO">
          <ser:log>
              <ser:logger name="DATA" active="true"/>
              <ser:logger name="WS_SERVER_EXECUTE" active="true"/>
          </ser:log>
       </com:ctx>
         <com:CRUDManagerretrieveIn viewObjectName="JProView" errorIfNoRow="false">
            <com:retrieve key="DATCRE">
               <com:param num="1" value="1"/>
               <com:param num="2" value="20041008"/>
            </com:retrieve>
         </com:CRUDManagerretrieveIn>
      </urn:retrieve>
   </soapenv:Body>
</soapenv:Envelope>
```



➔

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:GCE"
xmlns:com="http://www.generix.fr/technicalframework/businesscomponent/applicationmodu
le/common"
xmlns:ser="http://www.generix.fr/technicalframework/business/webservice/server">
   <soapenv:Header/>
   <soapenv:Body>
      <urn:retrieve>
         <com:ctx entity="1" target="GART" user="DEMO" password="DEMO">
          <ser:log>
              <ser:logger name="DATA" active="true"/>
              <ser:logger name="WS_SERVER_EXECUTE" active="true"/>
          </ser:log>
       </com:ctx>
         <com:CRUDManagerretrieveIn viewObjectName="JProView" errorIfNoRow="false">
            <com:retrieve key="DATCRE">
               <com:param num="1" value="1"/>
               <com:param num="2" value="20041008"/>
               <com:param num="3" value="013"/>
            </com:retrieve>
         </com:CRUDManagerretrieveIn>
      </urn:retrieve>
   </soapenv:Body>
</soapenv:Envelope>
```

➔


SOAP-ENV:Envelope
xmlns:SOAP-ENV
http://schemas.xmlsoap.org/soap/envelope/
SOAP-ENV:Body
retrieveResponse
xmlns
urn:GCE
CRUDManagerretrieveOut
viewObjectName
JProView
xmlns
http://www.generix.fr/technicalframework/businesscomponent/applicationmodule/common
xmlns:ns2
http://www.generix.fr/technicalframework/business/service/common
row
num
1
field (190)

3.9 Envoi de mail en cas d'erreur

Fonctionnalité introduite avec le socle 8.0, elle s'enrichit :

3.9.1 Contenu

Le corps du mail s'agrémente des entrées du MDC en plus des « Environment properties » et des « JVM properties »

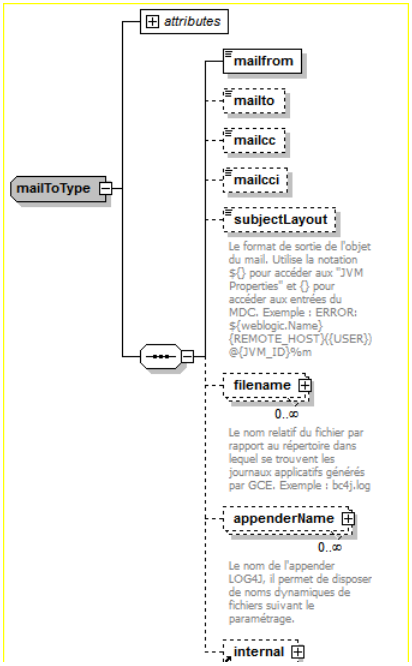
MDC properties	
Key	**Value**
DATE	20131107
REQUEST_CURRENT_BUSINESSVIEW	F_VMVC_L
ELAPSE_TIME_XMLRPC	0.0
ELAPSE_TIME	2.0649
QUERY_STRING	ctxPrefix=F_VCPT_F_F_VMVC_F&chp:TypeDeviseSelect=SEUR&chp:CodeEltUti=&chp:P3ctgrp=&entity=1&chp:P1excpp=PCG&chp:LblDev=Euro&LibCompte=&chp:QCIntraGroupe=T&chp:Journal=T&PieceType=RG,RM,RS,PB,PI,O p:QCTypeCouche=V&chp:Type=T&chp:TypeDevise=S&sourceview=F_VMVC_L&errorCtrlDevise=&frame=F_VCPT_F_F_VMVC_F_0001_liste&chp:P3ctcod=CLI&ListeLetPceRec=TN&ActiveDevise=&chp:Devise=&chp:P1exdvg=&chp =&chp:P1exdvc=&chp:Nomtie=PARADIS&ListeLettres=TN&LibelleDevise=&error=&chp:P1exdebope==&MultiEtab=N&chp:Cpttie=C000001&chp:P1ccgrp=&chp:SiTiersRatCompta=false&chp:LotLettrage=&forceDisplay=&P1ppbi
APPMODULE	RootApplicationModule
TRANSACTION_MODE	EndTransaction
CONNECTION	0.053894
COUNT_VO_EXECUTEQUERY	6
SQL_OPTIMIZATION	RELOAD
SERVER_HOST	172.21.102.242
SQL_QUERY	MevP_1pp.CODSOC = ? AND P_1PPCPP = ? AND P_1PPNAT = 'S'
REQUEST_CINEMATIC	[init(0), display(0)]
TIMER_NAME	TRANSACTION
VIEWOBJECT	null
REQUEST_URI	/gce/GCE/ServletControl
SQL_PARAM_PRIMARYKEY	
VIEW_NAME	UtilisateurCourantSociete
REMOTE_ADDR	0:0:0:0:0:0:0:1
UUID_STATE	REUSED
SQL_PARAM	1;PCG;
TARGET	P_1VIS
ACTION_NAME	init
LABEL	fr.generix.technicalframework.application.processus.listener.impl.DefaultAuthenticationListener
LANGUAGE	FRA
REMOTE_HOST	0:0:0:0:0:0:0:1
HTTP_METHOD	POST
ENTITYOBJECT	
ELAPSE_TIME_BUILD	6.419897
USER	GNC
UUID	1ebb6f2-1423328f282-115f57d80a691472cf7c2fa7c3bef148
SERVICE_NAME	null
JVM_ID	/gce_b30f2f3c-b85f-431a-a4cd-eac63fdd11cc

3.9.2 Paramétrage de l'objet

Il est dorénavant possible de paramétrer le format de l'objet du mail en fonction de données « variables ».

3.9.2.1 Le gceProperties

Comme d'habitude, le paramétrage se fait dans gce.properties :



Le format est :

- notation ${JVM_PROP} pour accéder aux "JVM Properties"
- notation {MDC_CODE} pour accéder aux entrées du MDC
- %m : le code de l'erreur

3.9.2.2 *Exemple de paramétrage*

```
<MailSettings active="true">
	<account admin="GENERIX Group in the sky&lt;tf_request@generixgroup.com&gt;"
user="tf_request@generixgroup.com" port="25" password="XXX"
host="smtp.generixgroup.com" auth="true" tls="true"/>
	<startup active="false">
		<mailfrom>GENERIX Group in the
sky&lt;tf_request@generixgroup.com&gt;</mailfrom>
		<mailto>arocher@generixgroup.com</mailto>
		<filename name="oc4j.log" maxSize="10000000"/>
		<appenderName name="egxAppender"/>
	</startup>
	<shutdown active="false">
		<mailfrom>GENERIX Group in the
sky&lt;tf_request@generixgroup.com&gt;</mailfrom>
		<mailto>arocher@generixgroup.com</mailto>
	</shutdown>
	<error active="true">
		<mailfrom>GENERIX Group in the
sky&lt;tf_request@generixgroup.com&gt;</mailfrom>
		<mailto>arocher@generixgroup.com</mailto>
		<mailcci>fcarton@generixgroup.com,gprince@generixgroup.com</mailcci>
		<subjectLayout>ERROR: ${weblogic.Name}
{REMOTE_HOST}({USER})@{JVM_ID}%m</subjectLayout>
		<filename name="oc4j.log"/>
		<appenderName name="egxAppender" maxSize="10000000"/>
		<appenderName name="bc4jAppender" maxSize="10000000"/>
		<appenderName name="activiteAppender" maxSize="10000000"/>
	</error>
</MailSettings>
```

3.10 Action de cinématique retrieve : nouveau paramètre

3.10.1 Objectif

Permettre de cibler la portée de l'action à une View en particulier.

3.10.2 Pré requis

Socle 8.1 et suivants.

3.10.3 Configuration

La définition de l'action dans le fichier configuration_base.xml prend en compte cette nouveauté :

```
<action name="retrieve" class="fr.generix.technicalframework.application.action.Retrieve">
    <required>
        <param name="businessView" maxOccurs="1" minOccurs="1" type="BusinessView"/>
    </required>
    <optional>
        <param name="key" maxOccurs="1" minOccurs="1" type="String"/>
        <param name="view" maxOccurs="1" minOccurs="1" type="View"/>
    </optional>
</action>
```

Aux 2 paramètres déjà présents, businessView et key, s'ajoute le paramètre optionnel view.

Il suffit d'ajouter dans la cinématique la vue ciblée :

cinematic=retrieve(MaBV, maClé, maView)

4 APIs pour le développement

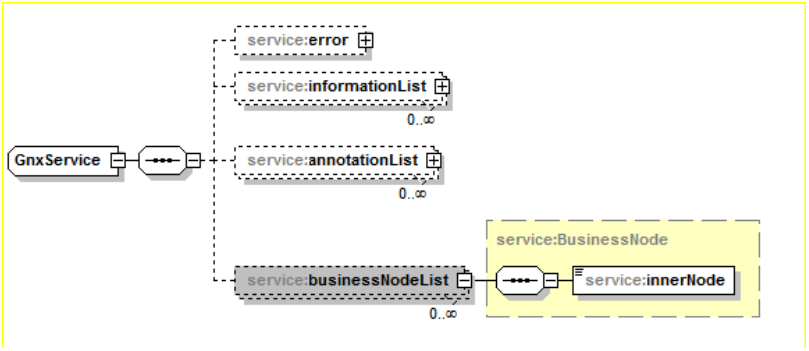
4.1 Flux XML « business »

Toute View (dans une ViewStruct) génère son bout d'arbre DOM dans le flux de présentation global. Ce bout d'arbre XML est généré par le socle en fonction du type de View/Document (BC4J/API/…).

A ce flux « automatique », il est dorénavant possible d'en accrocher un par code java sur chaque ligne de Document (RowDocument)

4.1.1 API

Chaque flux de sortie d'API dérive du type complexe *GnxService*. Celui-ci se voit agrémenter d'une liste de nœud XML facultative :




```
<xsd:complexType name="BusinessNode">
	<xsd:sequence>
		<xsd:element name="innerNode" type="service:Node"/>
	</xsd:sequence>
</xsd:complexType>
<xsd:simpleType name="Node">
	<xsd:annotation>
		<xsd:appinfo>
			<xjc:javaType name="org.w3c.dom.Node"
adapter="fr.generix.technicalframework.business.domain.NodeAdapter"/>
		</xsd:appinfo>
	</xsd:annotation>
	<xsd:restriction base="xsd:string"/>
</xsd:simpleType>
```

- La méthode *createElement()* facilite la création d'un objet *Node* xml.
- La méthode *createBusinessNode()* est le Builder d'un objet **BusinessNode** (à l'image de createError(), createInformation(), createAnnotation())
- L'accroche sur le beanOut de l'api est tout naturel via :
- `out.getBusinessNodeList().add()`
-

Ci-dessous un exemple d'utilisation :

```
Element elt1 = createElement("elt1");
elt1.setAttribute("attr1", "value1");
Element elt11 = createElement("elt11");
elt11.setAttribute("attr11", "value11");
elt1.appendChild(elt11);

Element elt2 = createElement("elt2");
elt2.setAttribute("attr2", "value2");

out.getBusinessNodeList().add(createBusinessNode(elt1));
out.getBusinessNodeList().add(createBusinessNode(elt2));
```

Code Java

```
<GestionTexteLibre type="ApplicationModule">
  <creerTexteLibre current="true" type="Service" index="1" business_row_index="1">
     <codcom>
        <business_data>CTRREA</business_data>
     </codcom>
     <codlan>
        <business_data>FRA</business_data>
     </codlan>
     <codtxt>
        <business_data>CTRREA</business_data>
     </codtxt>
     <texteLibre>
        <business_data>Test</business_data>
     </texteLibre>
     <typtxt>
        <business_data>TXT</business_data>
     </typtxt>
     <business_nodes>
        <elt1 attr1="value1">
           <elt11 attr11="value11"/>
        </elt1>
        <elt2 attr2="value2"/>
     </business_nodes>
  </creerTexteLibre>
</GestionTexteLibre>
```

Flux de présentation

4.1.2 ViewRow

Tout comme pour une API :

- La méthode *createElement()* facilite la création d'un objet *Node* xml.
- La méthode *createBusinessNode()* est le Builder d'un objet **BusinessNode** (à l'image de createError(), createInformation(), createAnnotation())
- La méthode *addBusinessNode()* permet l'accroche sur le flux « automatique »

Ci-dessous un exemple d'utilisation :

```
Element tfmen = createElement("TFMen");
tfmen.setAttribute("count", "3");

Element roa = createElement("roa");
roa.setAttribute("ville", "ville1");
tfmen.appendChild(roa);
Element caf = createElement("caf");
caf.setAttribute("ville", "ville2");
tfmen.appendChild(caf);
Element prg = createElement("prg");
prg.setAttribute("ville", "ville3");
tfmen.appendChild(prg);

Element rph = createElement("rph");
rph.setAttribute("pecheur", "oui");

addBusinessNode(createBusinessNode(tfmen));
addBusinessNode(createBusinessNode(rph));
```

Code Java

```
<JComViewRow current="true" type="ViewRow" index="1" business_row_index="1">
<Codcom precision="6" type="VARCHAR" pk="true">
  <business_data>CTRREA</business_data>
</Codcom>
<Coddoc precision="3" type="VARCHAR">
  <business_data/>
</Coddoc>
<Codent precision="16" type="VARCHAR" pk="true">
  <business_data>COM</business_data>
</Codent>
<Codlan precision="3" type="VARCHAR" pk="true">
  <business_data>FRA</business_data>
</Codlan>
<Codori precision="1" type="VARCHAR">
  <business_data>X</business_data>
</Codori>
<Codsoc precision="38" type="NUMERIC" pk="true">
  <business_data>1</business_data>
</Codsoc>
<Codsoc1 precision="38" type="NUMERIC" pk="true">
  <business_data>1</business_data>
</Codsoc1>
<CodsocPhy precision="38" type="NUMERIC">
  <business_data>1</business_data>
</CodsocPhy>
<Commen1 precision="40" type="VARCHAR">
  <business_data>contrôle de réalisation</business_data>
</Commen1>
<Datmod precision="8" type="VARCHAR">
  <business_data>20110524</business_data>
</Datmod>
<Segment precision="16" type="VARCHAR" pk="true">
  <business_data> </business_data>
</Segment>
<Utimod precision="8" type="VARCHAR">
  <business_data>GNC</business_data>
</Utimod>
<business_nodes>
  <TFMen count="3">
     <roa ville="ville1"/>
     <caf ville="ville2"/>
     <prg ville="ville3"/>
  </TFMen>
  <rph pecheur="oui"/>
</business_nodes>
```

```
<VueTexte type="View" name="TexteTraduction" habctr="true" total_business_row="1"
nbline="20" confNbline="20" numpage="1" nbpage="1">
```

Flux de présentation

5 Sécurité

5.1 Sécurisation des accès aux données techniques

L'accès aux données techniques de l'application via certaines valeurs du paramètre d'url *render_type* est maintenant réservé à l'agent de type « super user » (voir PPE PWDPOL).

Les valeurs du render_type sécurisées sont :

- xml_pre
- audit

En cas d'accès non autorisé, l'exception suivante sera affichée :