



LexML Brasil
Parte 4a – Kit Provedor de Dados

Versão 1.0 (RC1)
Brasília – Fevereiro de 2011

LexML Brasil
Parte 4a – Kit Provedor de Dados

Versão 1.0 (Maio/2010)

1.	Introdução.....	3
2.	Validador	3
2.1.	Instalação do Validador	3
2.2.	Manutenção dos Metadados	5
2.3.	Estrutura do XML.....	6
2.4.	Execução da Validação.....	9
2.5.	Grandes Quantidade de Registros	10
3.	Interface WEB OAI-PMH	10
3.1.	Instalação da Interface WEB OAI-PMH	10
	Anexo 1 – Exemplos de configuração de banco de dados	12

1. Introdução

O projeto LexML desenvolveu o **Kit Provedor de Dados** com o objetivo de facilitar a manutenção e validação dos metadados dos provedores de dados e de disponibilizar esses metadados através do protocolo OAI-PMH.

O Kit Provedor de Dados é composto dos módulos:

- **Validador**
 - Banco de Dados com registro de metadados
 - Programa que valida o conteúdo dos registros de acordo com o perfil
- **Interface WEB OAI-PMH**
 - Expõe os metadados utilizando o protocolo OAI-PMH

Este documento descreve o procedimento de instalação dessas ferramentas e como elas são utilizadas.

2. Validador

O **Validador** é uma aplicação de linha de comando para validar os metadados previamente cadastrados em um banco de dados relacional. Apenas registros validados poderão ser importados pelo coletor da rede LexML.

2.1. Instalação do Validador

Requisitos de Software

- Sistema Operacional: Qualquer sistema operacional compatível com a plataforma Java SE 1.5 ou superior.
- Banco de Dados: Qualquer banco de dados relacional suportado pela framework Hibernate (Veja lista em <https://www.hibernate.org/80.html>). O processo de instalação foi simplificado para os bancos de dados: Oracle, SQL Server, DB2, MySQL e PostgreSQL.
- Driver JDBC para seu banco de dados. Por restrições da licença de uso de alguns drivers JDBC, não pudemos distribuí-los junto com este pacote. Veja abaixo como obter o driver para seu banco:
 - **Oracle**: ojdbc14.jar (disponível com sua distribuição do banco de dados)
 - **SQL Server**: jtds-09.jar (distribuído com este pacote)
 - **DB2**: db2jcc4.jar (disponível com sua distribuição do banco de dados)
 - **MySQL**: mysql-connector-java-5.1.8-bin.jar (distribuído com este pacote)
 - **PostgreSQL**: postgresql-8.4-701.jdbc3.jar (distribuído com este pacote)

- **Outros SGBDs:** Verifique junto ao seu fornecedor.
- Máquina virtual Java compatível com a versão 1.5 ou superior.

Procedimento

- Obter perfil do provedor (arquivo lexml_nbconfig.xml) junto ao Prodasen
 - O arquivo de profile define as combinações possíveis dos elementos localidade, autoridade e tipo de documento de um publicador.
 - Na fase de implantação, este arquivo XML será enviado por e-mail a partir do preenchimento da planilha “Perfil Provedor Dados.xls”.
- Posicionar o arquivo lexml_nbconfig.xml na pasta /etc
- Criar uma conta no seu SGBD com acesso à criação de tabelas e manipulação dos dados. (Anexo 1 – Exemplos de configuração de banco de dados)
- Colocar na pasta /lib o driver JDBC do seu banco de dados
 - A pasta /lib já tem os drivers para o MySQL, PostgreSQL e SQL Server.
- Caso seu banco de dados não for Oracle, DB2, MySQL, PostgreSQL ou SQL Server você deverá providenciar o arquivo /etc/lexml-db.properties com as seguintes propriedades ajustadas ao seu banco de dados:

```
hibernate.dialect=org.hibernate.dialect.MySQLDialect
hibernate.connection.driver_class=com.mysql.jdbc.Driver
hibernate.connection.url=jdbc:mysql://127.0.0.1:3306/lexml?useUnicode=true\
&characterEncoding=UTF-8&autoReconnect=true
hibernate.connection.username=lexml
hibernate.connection.password=lexml
```

- Executar o script /bin/instalar.bat (windows) ou /bin/instalar.sh (linux)
 - Informar dados para conexão com seu banco de dados
 - O script criará o schema no banco de dados e os arquivos de configuração necessários para execução do Validador e do da Interface WEB OAI-PMH.
- Após a instalação você poderá utilizar o script abaixo para validar os registros no banco de dados.

```
(windows) /bin/validar.bat
(linux) /bin/validar.sh
```

2.2. Manutenção dos Metadados

Os metadados são mantidos em uma tabela de banco de dados relacional por aplicação de exportação de metadados desenvolvida pelo próprio provedor de dados e ficarão disponíveis para a coleta após validados.

2.1.1. Manutenção dos registros

A manutenção dos metadados é feita através de inclusões e alterações de registros na tabela **registro_item**.

Incluindo metadados

Exemplo MySQL:

```
INSERT INTO registro_item (
    id_registro_item, tx_metadado_xml, cd_status, cd_validacao,
    ts_registro_gmt)
VALUES(
    'oai:ssinf.senado.gov.br:njur/112321', '$OAI_LEXML', 'N', 'I',
    UTC_TIMESTAMP());
```

Exemplo Oracle:

```
INSERT INTO registro_item (
    id_registro_item, tx_metadado_xml, cd_status, cd_validacao,
    ts_registro_gmt)
VALUES(
    'oai:ssinf.senado.gov.br:njur/112321', '$OAI_LEXML', 'N', 'I',
    sysdate);
```

Sobre os dados

- A regra de formação da coluna `id_registro_item` é:

```
id_registro_item := prefixo_oai + "/" + chave
prefixo_oai      := 'oai:' + [orgao + "."] + dominio + ":" + sistema
chave           := identificador_interno + [ ";" + detalhe]
identificador_interno := é a chave do sistema de origem do registro
```

EXEMPLO: 'oai:ssinf.senado.gov.br:njur/112321'

- O valor 'N' da coluna `cd_status` significa que o registro é novo.
- O valor 'I' da coluna `cd_validacao` significa que o registro é pendente de validação.
- A coluna `ts_registro_gmt` deve receber sempre a data atual UTC ('2008-08-08 08:08:08') dependendo do horário de verão são 2 ou 3 horas a mais do que o horário oficial de Brasília.
- A coluna `tx_metadado_xml` deve receber o arquivo XML com os metadados de acordo com o schema `lexml_oai.xsd`.

- A coluna `id_conjunto_item` só é utilizada para particionamento nos casos em que o provedor de dados possui mais de 500.000 registros. Neste caso, deve-se entrar em contato com a equipe LexML para estudar o melhor critério de particionamento.

Atualizando metadados

Se por algum motivo for necessário executar UPDATE em algum registro da tabela `registro_item` há que se modificar as seguintes colunas:

- `cd_validacao` deve ser atualizada para 'I',
- `ts_registro_gmt` deve ser atualizado para data atual UTC.

```
UPDATE registro_item
SET cd_validacao = 'I',
    ts_registro_gmt = {data_atual_formato_utc}
    tx_metadada_xml = {novo conteúdo no formato xml}
WHERE id_registro_item = {chave};
```

Excluindo metadados

Se for necessária a exclusão de algum dos registros que já faz parte do Acervo LexML deve-se realizar um UPDATE alterando a coluna '`cd_status`' para 'D' para que a informação sobre a deleção seja enviada na próxima coleta de metadados para processamento e remoção do documento do acervo.

```
UPDATE registro_item
SET cd_status = 'D',
    ts_registro_gmt = {data_atual_formato_utc}
WHERE id_registro_item = {chave};
```

2.3. Estrutura do XML

Veja abaixo um exemplo de XML para a coluna `tx_metadado_xml` de acordo com o schema `oai_lexml.xsd` (http://projeto.lexml.gov.br/esquemas/oai_lexml.xsd):

```
<LexML
  xmlns="http://www.lexml.gov.br/oai_lexml"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.lexml.gov.br/oai_lexml
    http://projeto.lexml.gov.br/esquemas/oai_lexml.xsd">
  <Item formato="text/html" idPublicador="1" tipo="conteudo">
http://www6.senado.gov.br/legislacao/ListaPublicacoes.action?id=102415
  </Item>
  <Item formato="text/html" idPublicador="1" tipo="metadado">
http://www6.senado.gov.br/legislacao/DetalhaDocumento.action?id=102415
  </Item>
  <DocumentoIndividual>urn:lex:br:federal:lei:1990-09-11;8078</DocumentoIndividual>
  <Epigrafe>Lei nº 8.078, de 11 de setembro de 1990</Epigrafe>
  <Apelido>Código de Defesa do Consumidor</Apelido>
  <Apelido>Código de Proteção e Defesa do Consumidor</Apelido>
  <Ementa>
    Dispõe sobre a proteção do consumidor e dá outras providências
  </Ementa>
  <Indexacao>PROTEÇÃO, CONSUMIDOR. APROVAÇÃO, CODIGO, CONSUMIDOR.</Indexacao>
  <Relacionamento tipo="publicacao.oficial">
urn:lex:br:imprensa.nacional:publicacao.oficial;diario.oficial.uniao;secao.1:1990-09-
12;123!pag1
  </Relacionamento>
</LexML>
```

O elemento <Item> possui a URL do recurso disponibilizado na Internet sob a responsabilidade do publicador. A cada <Item> deve-se, obrigatoriamente, relacionar uma URN no elemento <DocumentoIndividual>. Esse relacionamento posiciona o recurso publicado no espaço de nomes definidos pela Parte 2 das especificações LexML. Na seqüência, são relacionados: a Epígrafe, os Apelidos registrados para o documento e a Ementa. Por fim, sempre que possível, deve-se relacionar ao Documento Individual identificado a URN da publicação oficial que o veiculou.

Os elementos textuais (Epígrafe, Apelido e Ementa) possuem o atributo `xml:lang` para especificação da língua. Esse atributo tem por default o valor “pt-BR”.

Caso uma URL contenha mais de um Documento Individual (por exemplo, a norma e seus anexos), deve-se criar um registro para cada Documento Individual relacionado. Nesse caso, pode-se acrescentar ao <identifier> um detalhamento que diferencie esse registro dos demais.

Elementos

Item: URL do Item. (opcional, múltiplo)

Atributos

`idPublicador`: Identificador do publicador, como no arquivo `lexml_nbconfig.xml` (obrigatório)

`tipo`: Tipo do conteúdo (Pode ser: `conteudo`, `metadado` ou `conteudo. anotado`. Default: `conteudo`)

`formato`: Tipo MIME do conteúdo da URL.

`qtdPaginas`: Número de páginas do documento. (opcional)

DocumentoIndividual: URN canônica no padrão URN LexML. Informar a URN do documento individual referente ao Item. No caso do Item conter mais de um documento individual, como no caso de anexos ou diferentes versões de texto, deve-se gerar um registro para cada documento. (obrigatório, único)

Atributos

Não possui atributos

Epigrafe: Epígrafe do documento individual. (opcional, múltiplo)

Atributos

`xml:lang`: Idioma utilizado no texto. (Default: `pt-BR`)

Apelido: Nome usual ou alternativo do documento individual. (opcional, múltiplo)

Atributos

`xml:lang`: Idioma utilizado no texto. (Default: `pt-BR`)

Ementa: Ementa do documento individual. Pode conter tags htmls `<s>`, `<u>`, ``, `<p>`, `<i>`, `<sup>`, `<sub>`. (opcional, múltiplo)

Atributos

xml:lang: Idioma utilizado no texto. (Default: pt-BR)

Indexacao: contém palavras-chave baseado em vocabulários controlados (opcional, múltiplo).

Atributos

xml:lang: Idioma utilizado no texto. (Default: pt-BR)

Catalogo: contém a classe do documento baseado em um sistema de classificação (opcional, múltiplo)

Atributos

xml:lang: Idioma utilizado no texto. (Default: pt-BR)

Relacionamento: Relacionamento do documento individual com outro documento individual. Contém uma URN no padrão URN LexML. (opcional, múltiplo)

Atributos

idPublicador: Identificador do publicador.

tipo: Tipo do relacionamento. (ver tabela abaixo)

Tipos de relacionamento

Relacionamento	Descrição
altera	Alteração entre documentos.
correlato.a	Correlação entre documentos.
declara.inconstitucional	Declaração de inconstitucionalidade de uma norma no todo.
declara.inconstitucional.dispositivo	Declaração de inconstitucionalidade de um dispositivo de uma norma.
emenda.de	Relação entre a emenda e a proposição legislativa.
equivalente.a	Equivalência entre identificadores de documentos.
membro.de	Relação de pertinência.
parecer.de	Relação entre um parecer e o documento objeto do parecer.
parte.de	Relação todo-parte.

peticao.inicial.de	Relação entre a petição inicial e o processo.
publicacao.oficial	Relação entre o documento e o identificador da publicação oficial.
Referencia	Referência entre documentos.
Regulamenta	Relação de regulamentação.
Revoga	Revogação de uma norma no todo.
revoga.dispositivo	Revogação de um dispositivo de uma norma.
sucessivo.ou.precedente.de	Relação de seqüência entre acórdãos.
sucessor.logico.de	Relacionamentos entre documentos que se sucedem logicamente.

2.4. Execução da Validação

Após a inclusão ou alteração de registros de metadados no banco de dados deve ser realizada uma validação dos mesmos utilizando o script abaixo:

```
(windows) /bin/validar.bat
(linux)   /bin/validar.sh
```

O script de validação deve ser agendado para execução diária, utilizando os recursos do sistema, como, por exemplo, o `cron` do Linux.

Apenas registros que passaram com sucesso pela validação serão importados pela rede LexML.

Arquivo de log

O resultado da validação é apresentado na tela e no arquivo `/log/toolkit.log`.

```
INFO : validação com sucesso do registro :
oai:ssinf.senado.gov.br:njur/132630
2008-10-16 12:48:04,842 [ValidadorBO.java:204]
INFO : validação com sucesso do registro :
oai:ssinf.senado.gov.br:njur/132631
2008-10-16 12:48:05,561 [ValidadorBO.java:242]
INFO : Total de registros processados: 525
2008-10-16 12:48:05,561 [ValidadorBO.java:243]
INFO : Processados com sucesso: 525
2008-10-16 12:48:05,561 [ValidadorBO.java:244]
INFO : Registros que apresentavam erros: 0
2008-10-16 12:48:05,561 [ValidadorBO.java:246]
INFO : Aproveitamento de aproximadamente: 100%
```

Tabela de Erro

Os erros de validação são registrados no banco de dados, na tabela registro_item_erro.

Os erros também podem ser consultados utilizando o aplicativo que implementa OAI-PMH.

2.5. Grandes Quantidade de Registros

No caso do provedor de dados possuir mais de 500.000 documentos, deve-se utilizar a tabela CONJUNTO_ITEM com o objetivo de criar identificadores que permitam particionar a tabela REGISTRO_ITEM.

3. Interface WEB OAI-PMH

A **Interface WEB OAI-PMH** é uma um serviço que permite a importação dos metadados do provedor de dados pela rede LexML além de disponibilizar uma interface WEB de navegação nos metadados.

A coleta de metadados é realizada diariamente em um horário a combinar com o provedor de metadados (normalmente, após às 19h00 e antes das 21h00).

3.1. Instalação da Interface WEB OAI-PMH

Requisitos de Software

- Servidor de aplicação Java compatível com as especificações Java Servlet 2.4 e Java JSP 2.0. (Recomendado Tomcat 5.5 ou superior).

Requisitos de Conectividade

- Conexão com a Internet com IP fixo para disponibilização do serviço.

Procedimento

- É necessário instalar antes o **Validador** (ver item **Erro! Fonte de referência não encontrada.**).
- Colocar no CLASSPATH do seu servidor de aplicação o driver JDBC do seu banco de dados.
 - Tomcat: pasta \$CATALINA_HOME/lib
 - JBoss: pasta \$JBASS_HOME/server/default/deploy
 - OC4J: pasta \$OAS_HOME/j2ee/home/shared-lib
- Fazer o deploy do arquivo /oai/oai.war em seu servidor de aplicação
 - Tomcat: pasta \$CATALINA_HOME/webapps
 - JBoss: pasta \$JBASS_HOME/server/default/deploy

- OAS: pasta \$OAS_HOME/j2ee/home/application-deployments

Obs: Após a instalação do Kit Provedor de Dados o arquivo oai.war estará configurado para se conectar diretamente ao banco de dados informado sem utilizar um pool de conexões do servidor de aplicação. Caso você prefira utilizar o pool de conexões do seu servidor de aplicação edite o arquivo oai.war:/WEB-INF/classes/lexml-db.properties faça as seguintes alterações:

```
# Adicionar o nome do datasource
hibernate.connection.datasource=jdbc/lexmlDS

# Retirar dados de conexão direta com o banco de dados
#hibernate.connection.driver_class=com.mysql.jdbc.Driver
#hibernate.connection.url=jdbc:mysql://127.0.0.1:3306/lexml?useUnicode=true\
&characterEncoding=UTF-8&autoReconnect=true
#hibernate.connection.username=lexml
#hibernate.connection.password=lexml
```

Edite o arquivo oai.war:/WEB-INF/classes/hibernate.properties e faça as seguintes alterações:

```
# Remova as linhas abaixo
#hibernate.connection.provider_class=org.hibernate...
#hibernate.c3p0.max_size=10
#hibernate.c3p0.min_size=0
#hibernate.c3p0.timeout=5000
#hibernate.c3p0.max_statements=1000
#hibernate.c3p0.idle_test_period=300
#hibernate.c3p0.acquire_increment=1
#hibernate.c3p0.debug=false
```

- Teste sua instalação:

Abra no browser: <http://nome.do.servidor/oai>

- a) clique em [Identify] para apresentar o perfil do provedor;
- b) clique em [ListMetadataFormats] e em seguida em [oai_lexml] para apresentar os registros.

Anexo 1 – Exemplos de configuração de banco de dados

Os exemplos abaixo apresentam como criar o usuário do banco de dados com os acessos necessários para o Kit Provedor de Dados.

Configurando o Oracle

Exemplo utilizando o sql*plus:

```
SQL> connect system
*digite a senha

SQL> CREATE USER lexmluser IDENTIFIED BY lexmlpassword
      DEFAULT TABLESPACE users
      TEMPORARY TABLESPACE temp
      QUOTA UNLIMITED ON users;

SQL> GRANT CONNECT TO lexmluser;
```

Configurando o MySQL

```
usuario@localhost:~$ mysql -u root -p
*digite a senha

mysql> CREATE DATABASE lexmldatabase;

mysql> GRANT ALL ON lexmldatabase.* to lexmluser@localhost
      IDENTIFIED BY 'lexmlpassword' WITH GRANT OPTION;

mysql> FLUSH PRIVILEGES;
```