

PL/SQL API для работы с XMLType (ORACLE)

Графеева Н.Г.

2017

DOM

- **DOM** (от англ. *Document Object Model* — «объектная модель документа») — это не зависящий от платформы и языка программный интерфейс, позволяющий получить доступ к содержимому HTML, XHTML и XML-документов, а также изменять содержимое, структуру и оформление таких документов.
- *Рекомендован к использованию консорциумом W3C.*

DOM API

- В ORACLE DOM API реализован в виде процедур и функций пакета DBMS_XMLDOM. Пакет обеспечивает:
- Создание XMLType таблиц, колонок и представлений.
- Создание экземпляров XMLType на основе соответствующих данных базы.
- Доступ к данным XMLType.
- Манипуляции с данными XMLType.
- *Пакет реализован в соответствии со стандартом DOM Level 2.0 (W3C DOM specifications).*

Как использовать PL/SQL DOM API для XMLType?

- 1. Процедура *newDOMDocument* обрабатывает экземпляр XMLType или пустой документ. Таким образом создается экземпляр *DOMDocument* .
- 2. Затем используются DOM API методы, такие как *createElement*, *createText*, *createAttribute* и *createComment* для того чтобы изменить или расширить структуру DOM .
- 3. Результаты методов, такие как *DOMElement* и *DOMText* отправляются к методу *makeNode* для того, чтобы получить доступ к *DOMNode* интерфейсу.

Пример

- DECLARE
- var XMLType;
- doc DBMS_XMLDOM.DOMDocument;
- ndoc DBMS_XMLDOM.DOMNode;
- buf VARCHAR2(2000);
- BEGIN
-
- select data into var from employees where id = 1;
-
- -- Create DOMDocument handle
- doc := DBMS_XMLDOM.newDOMDocument(var);
- ndoc := DBMS_XMLDOM.makeNode(doc);
-
- DBMS_XMLDOM.writeToBuffer(ndoc, buf);
- DBMS_OUTPUT.put_line(buf);
-
- END;

SQL Commands - Windows Internet Explorer

http://apex.oracle.... Google

Файл Правка Вид Избранное Сервис Справка

Избранное S... x Google Страница >>

Rows 10 Save Run

```
select data into var from employees where id = 1;

-- Create DOMDocument handle
doc := DBMS_XMLDOM.newDOMDocument(var);
ndoc := DBMS_XMLDOM.makeNode(doc);

DBMS_XMLDOM.writeToBuffer(ndoc, buf);
DBMS_OUTPUT.put_line(buf);

END;
```

Results Explain Describe Saved SQL History

```
<Employees>
  <Employee emplid="1111" type="admin">
    <firstname>John</firstname>
    <lastname>Watson</lastname>
    <age>30</age>
    <email>johnwatson@sh.com</email>
  </Employee>
  <Employee emplid="2222" type="admin">
    <firstname>Sherlock</firstname>
    <lastname>Homes</lastname>
    <age>32</age>
    <email>sherlock@sh.com</email>
  </Employee>
  <Employee emplid="3333" type="user">
    <firstname>Jim</firstname>
    <lastname>Moriarty</lastname>
    <age>52</age>
```

Готово Интернет 100%

Пример

- DECLARE
- var XMLType;
- doc DBMS_XMLDOM.DOMDocument;
- ndoc DBMS_XMLDOM.DOMNode;
- docelem DBMS_XMLDOM.DOMElement;
- node DBMS_XMLDOM.DOMNode;
- childnode DBMS_XMLDOM.DOMNode;
- nodelist DBMS_XMLDOM.DOMNodelist;
- buf VARCHAR2(2000);
- BEGIN
- var := XMLType('<PERSON><NAME>ramesh</NAME></PERSON>');
- -- **Create DOMDocument handle**
- doc := DBMS_XMLDOM.newDOMDocument(var);
- ndoc := DBMS_XMLDOM.makeNode(doc);
- DBMS_XMLDOM.writeToBuffer(ndoc, buf);
- DBMS_OUTPUT.put_line('Before: ' || buf);
- docelem := DBMS_XMLDOM.getDocumentElement(doc);
- -- **Access element**
- nodelist := DBMS_XMLDOM.getElementsByTagName(docelem, 'NAME');
- node := DBMS_XMLDOM.item(nodelist, 0);
- childnode := DBMS_XMLDOM.getFirstChild(node);
- -- **Manipulate element**
- DBMS_XMLDOM.setNodeValue(childnode, 'raj');
- DBMS_XMLDOM.writeToBuffer(ndoc, buf);
- DBMS_OUTPUT.put_line('After: ' || buf);
- DBMS_XMLDOM.freeDocument(doc);
- INSERT INTO EMPLOYEES(ID, DATA) VALUES (5,var);
- END;

SQL Commands - Windows Internet Explorer

http://apex.oracle... Google

Файл Правка Вид Избранное Сервис Справка

Избранное S... x Google

Home Application Builder SQL Workshop Team Development Administ

SQL Workshop SQL Commands

Rows 10 Save Run

```
-- Manipulate element
DBMS_XMLDOM.setNodeValue(childnode, 'raj');

DBMS_XMLDOM.writeToBuffer(ndoc, buf);

DBMS_OUTPUT.put_line('After: '||buf);

DBMS_XMLDOM.freeDocument(doc);

INSERT INTO EMPLOYEES(ID, DATA) VALUES (5,var);
END;
```

Results Explain Describe Saved SQL History

Before:<PERSON>
<NAME>ramesh</NAME>
</PERSON>

After:<PERSON>
<NAME>raj</NAME>
</PERSON>

Statement processed.

0.04 seconds

Готово Интернет 100%

Пример

- DECLARE
- doc DBMS_XMLDOM.DOMDocument;
- elem DBMS_XMLDOM.DOMEElement;
- node DBMS_XMLDOM.DOMNode;
- BEGIN
- doc := DBMS_XMLDOM.newDOMDocument;
- elem := DBMS_XMLDOM.createElement(doc, 'ELEM');
- node := DBMS_XMLDOM.makeNode(elem);
-
- DBMS_OUTPUT.put_line('Node name = ' || DBMS_XMLDOM.getNodeName(node));
- DBMS_OUTPUT.put_line('Node value = ' || DBMS_XMLDOM.getNodeValue(node));
- DBMS_OUTPUT.put_line('Node type = ' || DBMS_XMLDOM.getNodeType(node));
- DBMS_XMLDOM.freeDocument(doc);
- END;

SQL Commands - Windows Internet Explorer

http://apex.oracle... Google

Файл Правка Вид Избранное Сервис Справка

Избранное S... x Google Страница

Home Application Builder SQL Workshop Team Development Administ

SQL Workshop SQL Commands

Rows 10 Save Run

```
doc := DBMS_XMLDOM.newDOMDocument;
elem DBMS_XMLDOM.DOMElement;
node DBMS_XMLDOM.DOMNode;
BEGIN
doc := DBMS_XMLDOM.newDOMDocument;
elem := DBMS_XMLDOM.createElement(doc, 'ELEM');
node := DBMS_XMLDOM.makeNode(elem);

DBMS_OUTPUT.put_line('Node name = ' || DBMS_XMLDOM.getNodeName(node));
DBMS_OUTPUT.put_line('Node value = ' || DBMS_XMLDOM.getNodeValue(node));
DBMS_OUTPUT.put_line('Node type = ' || DBMS_XMLDOM.getNodeType(node));
DBMS_XMLDOM.freeDocument(doc);
END;
```

Results Explain Describe Saved SQL History

Node name = ELEM
Node value =
Node type = 1

Statement processed.

0.02 seconds

Workspace: GRAFEYEVA User: GRAFEYEVA@IS4B.RU

Готово Интернет 100%

Упражнение

- Создайте экземпляр `DOMDocument` на основе загруженного XML-документа (из одного из предыдущих заданий).
Содержимое созданного документа отобразите с помощью DOM-методов.