

JSON & XML SQL Server 2016

since SQL 2016 it is important
to weigh respective advantages

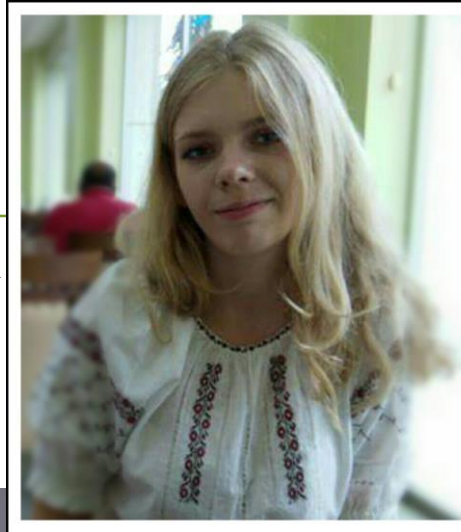


SQLSat Kyiv Team



Vitaliy Popovych

Olena Smoliak



Yevhen Nedashkivskiy



Mykola Pobyivovk



Oksana Borysenko



3 Sponsor Sessions at 12:40



- Don't miss them, they might be providing some interesting and valuable information!

	Room A	Room B	Room C
13:00 - 13:15	Intapp	DB Best	NULL
13:20 - 13:50	Microsoft	DevArt	JetBrains

Our Awesome Sponsors



Session will begin very soon :)



- Please complete the evaluation form from your pocket after the session. Your feedback will help us to improve future conferences and speakers will appreciate your feedback!
- Enjoy the conference!

About me



Alexander Karl

.net - CDE



SQL + BI Consultant

Microsoft
CERTIFIED
Trainer

Microsoft
CERTIFIED
IT Professional

Database Administrator 2008
Server Administrator on Windows Server® 2008
Database Administrator on SQL Server® 2005

.. and „2012er“ SQL MCSA + MCSE





my previous slides

www.net-CDE.de/presentations

publications



The screenshot shows a web browser window with the URL <https://www.video2brain.com/de/videotraining>. The page header includes the video2brain logo (a lynda.com brand) and navigation links: [Gratis Tutorials](#), [Abonnieren](#), [Trainings](#), [Trainer](#), [Lösungen](#), [Support](#), and [Apps](#). The main content area features a breadcrumb trail: [Alle Video-Trainings](#) » [IT](#) » [SQL Server](#). The title of the training is **SQL Server Integration Services – Grundlagen**, with a subtitle **Überblick und technischer Einstieg in den ETL-Prozess**. A video thumbnail shows a blue abstract graphic with the text **SQL Server Integration Services Grundlagen**. The description states: "Wenn Daten aus mehreren Datenquellen in eine Zieldatenbank, insbesondere in einem Datawarehouse zusammengeführt werden, nennt man diesen Prozess Extract-Transform-Load (ETL). Dafür gibt es im Microsoft SQL Server die Integration Services. Der Datenbank-Consultant und SQL-Entwickler Alexander Karl erläutert Ihnen in diesem Video-Training die Zusammenhänge und zeigt am Beispiel, wie Sie die SQL Server Integration Services (SSIS) erfolgreich einsetzen." The trainer is **Alexander Karl**, the release date is **27.09.2013**, and the duration is **4 Std. 0 min**.



publications



amazon.de [Mein Amazon](#) [Angebote](#) [Gutscheine](#) [Verkaufen](#) [Hilfe](#)

Suche: Alle

Hallo! Anmelden **Mein Konto** Prime **testen** **Einkaufswagen** **Wunschzettel**

Amazon.de Warehouse Deals Outlet Spar-Abo Amazon Apps Amazon Browser-Leiste Jetzt verkaufen Trade-In Geschenke

1-16 von 21 Ergebnissen in "Data quality services SQL" Sortieren in Bücher nach [Beste Ergebnisse](#) | [Beliebtheit](#) | [Preis: aufsteigend](#) | [Mehr](#)

Ergebnisse anzeigen für

- Fremdsprachige Bücher** > Computer & Internet
- Bücher** > Datenbanken
- Kindle-Shop** > Datenbanken
- + **Alle 4 Kategorien**

Filtern nach

Versandoption (Was ist das?)
 Kostenlose Lieferung ab EUR 20 Bestellwert

Blick ins Buch!

DQS step-by-step mit SQL-Server
von Alexander Karl (15. Mai 2014)
EUR 7,52 Kindle-Kauf
Jetzt als Download verfügbar.

Dieses Buch mit dem Kindle gratis ausleihen Amazon Prime: [Jetzt anmelden](#)
Kindle-Shop: [Alle 13 Artikel ansehen](#)

Blick ins Buch!

DQS step-by-step with SQL-Server
von Alexander Karl (15. Mai 2014)
EUR 7,52 Kindle-Kauf
Jetzt als Download verfügbar.

Dieses Buch mit dem Kindle gratis ausleihen Amazon Prime: [Jetzt anmelden](#)
Fremdsprachige Bücher: [Alle 17 Artikel ansehen](#)

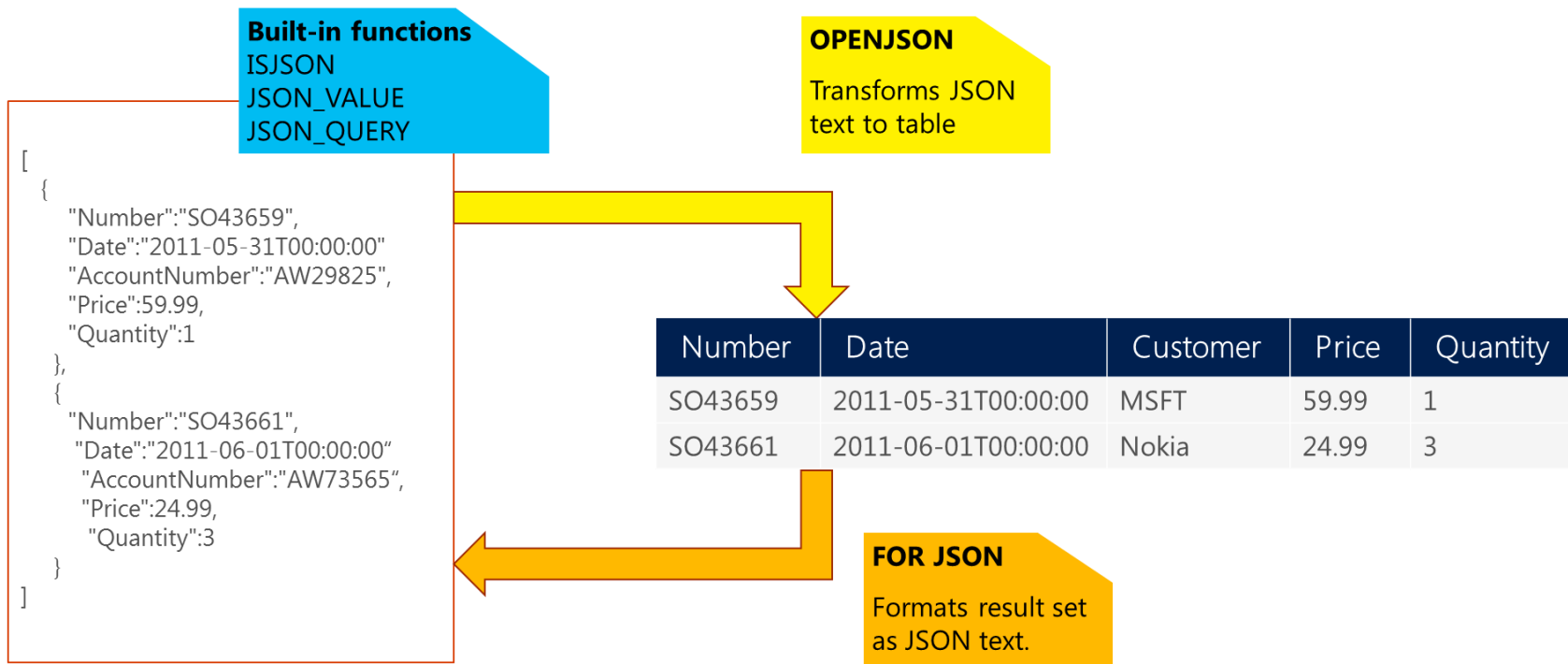




Agenda

- JSON „new feature in SQL Server 2016“
- appropriate tools
- syntax For JSON / For XML
- syntax OpenJSON / OpenXML
- datatype ? JSON vs. XML
- indexes on datatype

{JSON} „round trip“



<https://msdn.microsoft.com/en-us/library/dn921897.aspx>



<xml /> tooling

The screenshot shows the SQL Server Enterprise Manager interface. On the left, the Object Explorer displays the database structure for 'dbo.CarManufacturer_xml', with the 'xmlValue (XML(,), null)' column highlighted. The main window shows a SQL query in 'SQLQuery4.sql' and its results in a table.

```
1 SELECT [ID], [key], xmlValue
2 FROM   dbo.CarManufacturer_xml
```

	ID	key	xmlValue
1	1	Key01	<Customer CustomerID="1021"><!-- info 1021 custo...
2	2	Key02	<Customer CustomerID="1022"><!-- info 1022 custo...
3	3	Key03	<Customer CustomerID="1023"><!-- info 1023 custo...
4			
5			
6			
7			
8			
9			
10			
11			
..			

The results table shows the first row selected, and a preview of the XML data is shown below:

```
1 <Customer CustomerID="1021">
2   <!-- info 1021 customer since 2010-->
3   <CompanyName>Alfa Romeo</CompanyName>
4   <CustomerSince>2010</CustomerSince>
5 </Customer>
```



tooling



```
1 SELECT name          as 'name'
2     , database_id    as 'database_id'
3     , create_date    as 'create_date'
4     , is_published   as 'replication.is_published'
5     , is_subscribed  as 'replication.is_subscribed'
6     , is_merge_published as 'replication.is_merge_published'
7     , is_distributor as 'replication.is_distributor'
8 FROM sys.databases
9 FOR JSON PATH, ROOT('databases'), Include_NULL_Values;
10
```

100 % <

Results Messages

JSON_F52E2B61-18A1-11d1-B105-00805F49916B

1 {"databases":[{"name":"master","database id":1,"create date":...

JSON_F52E2B61-18...00805F49916B1.xml × demo01.sql

```
1 {"databases":[{"name":"master","database_id":1,"create_date":"2003-04-08T09:13:36.390"
```

Sign up for our free weekly **Web Developer Newsletter**.



ENTER THE **DOLBY AUDIO CHALLENGE**

GRAND PRIZE **\$10,000 USD**

[CLICK HERE](#)

Articles » Web Development » Client side scripting » General



Tip/Trick

Browse Code

Stats

Revisions (9)

Alternatives

Comments (11)

Add your own alternative version

Tagged as

- Javascript
- IE8
- IE10
- IE11
- IE7
- Browser

View JSON in Internet Explorer

Coding 101, 21 May 2014 CPOL

Rate this:

4.88 (17 votes)

A simple Registry change will enable IE to display JSON responses.

Need to view JSON responses in IE?



Hide Copy Code

```
Windows Registry Editor Version 5.00;
; Tell IE 7,8,9,10,11 to open JSON documents in the browser on Windows XP and later.
; 25336920-03F9-11cf-8FD0-00AA00686F13 is the CLSID for the "Browse in place" .
;
[HKEY_CLASSES_ROOT\MIME\Database\Content Type\application/json]
"CLSID"="{25336920-03F9-11cf-8FD0-00AA00686F13}"
"Encoding"=hex:08,00,00,00
```

1. Open Notepad and paste the following:
2. Save document as *IE-Json.reg* and then run it.

Internet Explorer browser window showing the URL: <http://www.codeproject.com/Tips/216175/View-JSON-in-Internet-Explorer>

Sign up for our free weekly **Web Developer Newsletter**.

ENTER THE **DOLBY AUDIO CHALLENGE**

GRAND PRIZE **\$10,000 USD**

[CLICK HERE](#)

Browser window showing the URL: <http://jsonview.com/example.json>

```

{"hey": "guy","anumber": 243,"anobject": {"whoa": "nuts","anarray": [1,2,"thr
ee"], "more":"stuff"},"awesome": true,"bogus": false,"meaning": null,
"japanese":"明日がある。", "link": "http://jsonview.com", "notLink":
"http://jsonview.com is great"}

```

Windows Registry Editor Version 5.00;

```

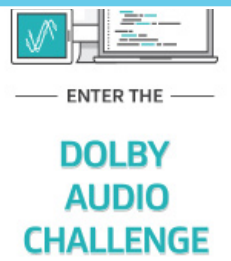
; Tell IE 7,8,9,10,11 to open JSON documents in the browser on Windows XP and later.
; 25336920-03F9-11cf-8FD0-00AA00686F13 is the CLSID for the "Browse in place" .
;
[HKEY_CLASSES_ROOT\MIME\Database\Content Type\application/json]
"CLSID"="{25336920-03F9-11cf-8FD0-00AA00686F13}"
"Encoding"=hex:08,00,00,00

```

1. Open Notepad and paste the following:
 2. Save document as *IE-Json.reg* and then run it.

Tagged as: Javascript, IE8, IE10, IE11, IE7, Browser

Hide Copy Code



ENTER THE **DOLBY AUDIO CHALLENGE**

Firefox Add-ons Manager: about:addons

Erweiterungen

JSONView 1.1.0

Von Ben Hollis

```
{
  hey: "guy",
  number: 243,
  - anObject: {
    whoa: "nuts",
    - anarray: [
      1,
      2,
      "threeeee"
    ],
    more: "stuff"
  },
  awesome: true,
  bogus: false,
  meaning: null,
  japanese: "明日が絶望。",
  link: http://jsonview.com,
  notlink: "http://jsonview.com is great"
}
```

JSON-Dokumente im Browser anzeigen.

Normalerweise bietet Firefox beim Aufrufen eines JSON-Dokuments (Content-Type "application/json") das Abspeichern der Datei an. Die JSONView-Erweiterung stellt JSON-Dokumente analog wie XML-Dokumente dar. Das Dokument verfügt über Formatierungen sowie Hervorhebungen und Felder und Objekte können erweitert und zusammengefasst werden. Enthält das JSON-Dokument Fehler, so zeigt JSONView nur den Quelltext an.

Um JSONView in Aktion zu sehen, rufen Sie nach der Installation <http://benhollis.net/software/jsonview/example.json> auf.

Automatische Updates: Standard Ein Aus

Zuletzt aktualisiert: Sunday, February 21, 2016

Homepage: <http://jsonview.com/>

Bewertung: ★★★★★ 143 Bewertungen

The screenshot shows a Firefox browser window with the address bar displaying `http://jsonview.com/example.j...`. The main content area shows a JSON object rendered in a viewer. The JSON object is as follows:

```
{
  hey: "guy",
  anumber: 243,
  anobject: {
    whoa: "nuts",
    anarray: [
      1,
      2,
      "thr<h1>ee"
    ],
    more: "stuff"
  },
  awesome: true,
  bogus: false,
  meaning: null,
  japanese: "明日がある。",
  link: http://jsonview.com,
  notLink: "http://jsonview.com is great"
}
```

{JSON} Formatter & Validator

A screenshot of a web browser showing the JSON Formatter & Validator website. The browser's address bar displays the URL https://jsonformatter.curiousconcept.com/. The website header includes the logo "JSON FORMATTER & VALIDATOR" and navigation links for "About", "Learn", "Bookmarklet", "Changelog", "Support", and "Contact". The main content area has a blue background. On the left, it says "Paste in JSON or a URL and away you go." In the center, there is a text area labeled "JSON Data/URL" containing a large block of JSON data. On the right, there are two dropdown menus: "JSON Standard" set to "RFC 4627" and "JSON Template" set to "3 Space Tab". At the bottom center, there is a blue button labeled "Process".

JSON Data/URL

```
{ "Header": [ { "symbol": "MSFT", "issuerName": "MICROSOFT CORP", "trdPrc": "51.82", "netChg": "-0.37", "pcntChg": "-0.71", "bid": "51.80", "ask": "51.81", "bidSize": "0", "askSize": "0", "cumVol": "33559073", "sectyType": "0" }, { "putcallInd": "C", "symbol": "MSFT1619B25", "exch": "US", "expr": "2016-02-20T00:00:00", "occExprDate": "2016-02-19T00:00:00", "strikePrc": "25.00", "trdPrc": "27.61", "netChg": "0.0", "bid": "26.70", "ask": "27.10", "cumVol": "0", "opnInt": "1", "shortDatedInd": "0", "putcallIndX": "P", "symbolX": "MSFT1619N25", "exchX": "US", "exprX": "2016-02-20T00:00:00", "occExprDateX": "2016-02-19T00:00:00", "strikePrcX": "25.00", "trdPrcX": "0.02", "netChgX": "0.0", "bidX": "0.0", "askX": "0.04", "cumVolX": "0", "opnIntX": "2", "shortDatedIndX": "0" }, { "putcallInd": "C", "symbol": "MSFT1619B28", "exch": "US", "expr": "2016-02-20T00:00:00", "occExprDate": "2016-02-19T00:00:00", "strikePrc": "28.00", "trdPrc": "20.30", "netChg": "0.0", "bid": "22.20", "ask": "22.70", "cumVol": "0", "opnInt": "1", "shortDatedInd": "0", "putcallIndX": "P", "symbolX": "MSFT1619N28", "exchX": "US", "exprX": "2016-02-20T00:00:00", "occExprDateX": "2016-02-19T00:00:00", "strikePrcX": "28.00", "trdPrcX": "0.02", "netChgX": "0.0", "bidX": "0.0", "askX": "0.04", "cumVolX": "0", "opnIntX": "2", "shortDatedIndX": "0" } ] }
```

Paste in JSON or a URL and away you go.

JSON Standard
RFC 4627

JSON Template
3 Space Tab

Process

{JSON} Formatter & Validator

A screenshot of a web browser displaying the JSON Formatter & Validator website. The browser's address bar shows the URL https://jsonformatter.curiousconcept.com/. The website's header includes the logo "JSON FORMATTER & VALIDATOR" and navigation links for "About", "Learn", "Bookmarklet", "Changelog", "Support", and "Contact". A green banner at the top of the main content area reads "VALID JSON (RFC 4627)". Below this, the text "Formatted JSON Data" is displayed above a code block containing the following JSON data:

```
{
  "Header": [
    {
      "symbol": "MSFT",
      "issuerName": "MICROSOFT CORP",
      "trdPrc": "51.82",
      "netChg": "-0.37",
      "pcntChg": "-0.71",
      "bid": "51.80",
      "ask": "51.81",
      "bidSize": "0",
      "askSize": "0",
      "cumVol": "33559073",
```



Demo



{JSON} FOR JSON



■ AUTO Mode

```
SELECT name, database_id, create_date
FROM sys.databases
FOR JSON Auto
```

■ PATH Mode

```
SELECT name as 'name',
       , database_id as 'database_id',
       , create_date as 'create_date',
       , is_published as 'replication.is_published',
       , is_subscribed as 'replication.is_subscribed',
       , is_distributor as 'replication.is_distributor',
FROM sys.databases
FOR JSON PATH, ROOT('databases'), Include_NULL_Values;
```



FOR JSON / Result



```

■ A file:///C:/ | . /JSON_demo/autoMode.json
SE [
FR  ▼ {
FO   name: "master",
      database_id: 1,
      create_date: "2003-04-08T09:13:36.390"
    },
■ P  ▼ {
SE   name: "tempdb",
      database_id: 2,
      create_date: "2016-02-24T11:40:54.827"
    },
FR  ▼ {
FO   name: "model",
      database_id: 3,
      create_date: "2003-04-08T09:13:36.390"
    },
      is_published,
      is_subscribed,
      is_distributor,
      de_NULL_Values;

```





```
file:///C:/.../JSON_demo/pathMode.json
file {
  databases: [
    {
      name: "master",
      database_id: 1,
      create_date: "2003-04-08T09:13:36.390",
      replication: {
        is_published: false,
        is_subscribed: false,
        is_merge_published: false,
        is_distributor: false
      }
    },
    {
      name: "tempdb",
      database_id: 2,
      create_date: "2016-02-24T11:40:54.827",
      replication: {
        is_published: false,
        is_subscribed: false,
        is_merge_published: false,
        is_distributor: false
      }
    }
  ]
}
```



<xml /> FOR XML

■ AUTO Mode

```
SELECT name, database_id, create_date
FROM   sys.databases
FOR    XML Auto ;
```

■ PATH Mode

```
SELECT name                as 'name',
       , database_id       as 'database_id',
       , create_date       as 'create_date',
       , is_published      as 'replication/is_published',
       , is_subscribed     as 'replication/is_subscribed',
       , is_distributor    as 'replication/is_distributor',
FROM   sys.databases
FOR    XML PATH ('db'), ROOT ('databases');
```

{JSON} OpenJSON



```
DECLARE @json NVARCHAR(MAX)
SET @json = '{
    "Name"      : "PASS Deutschland e.V.",
    "addInfo"   : null,
    "ID"        : 828,
    "Current"   : true,
    "Skills"    : ["SQL", "SSIS", "SSRS", 42, "MDX"],
    "Region"    : {"Country": "Germany", "Territory": "Hessen"}
}';

SELECT * -- [key], [value], type
FROM    OpenJSON (@json);

SELECT *
FROM    OpenJSON (@json, '$.Skills')

SELECT *
FROM    OpenJSON (@json, '$.Region')
```



OpenJSON



```

DECLARE @json
SET @json = '{
    "Name"
    "addInf
    "ID"
    "Curren
    "Skills
    "Region
    }';

```

```

SELECT * -- [k
FROM OpenJSON

```

```

SELECT *
FROM OpenJSON

```

```

SELECT *
FROM OpenJSON

```

	key	value	type
1	Name	PASS Deutschland e.V.	1
2	addInfo	NULL	0
3	ID	828	2
4	Current	true	3
5	Skills	["SQL","SSIS","SSRS",42,"MDX"]	4
6	Region	{'Country':"Germany",'Territory':"Hessen"}	5

essen"}

	key	value	type
1	0	SQL	1
2	1	SSIS	1
3	2	SSRS	1
4	3	42	2
5	4	MDX	1

	key	value	type
1	Country	Germany	1
2	Territory	Hessen	1





OpenJSON Data Type



```
1 DECLARE @json NVARCHAR(MAX)
2 SET @json = '{
3     "Name"      : "PASS Deutschland e.V.",
4     "addInfo"   : null,
5     "ID"        : 828,
6     "Current"   : true,
7     "Skills"    : ["SQL", "SSIS", "SSRS", 42, "MDX"],
8     "Region"    : {"Country": "Germany", "Territory": "Hessen"}
9 }';
10
```

100 %

Results

Messages

	key	value	type	Data Type
1	Name	PASS Deutschland e.V.	1	string
2	addInfo	NULL	0	null
3	ID	828	2	int
4	Current	true	3	true/false
5	Skills	["SQL", "SSIS", "SSRS", 42, "MDX"]	4	array
6	Region	{"Country": "Germany", "Territory": "Hessen"}	5	object

{JSON} Storing JSON



- NVARCHAR (MAX)
- Index
„abgeleitete Spalte“
mit `JSON_VALUE (@JsonCol, '$. ')`

>> CREATE INDEX ...



JSON Namespaces



The screenshot shows a web browser window displaying a Microsoft Docs page. The address bar shows the URL: `https://msdn.microsoft.com/de-de/library/system.runtime.serialization.json(v=vs.110).aspx`. The breadcrumb navigation indicates the path: `System.Runtime Namespaces > System.Runtime.Serialization.Json`. A yellow banner at the top of the article content states: "Dieser Artikel wurde maschinell übersetzt. Bewegen Sie den Mauszeiger über die Sätze im Artikel, um den Originaltext anzuzeigen. [Weitere Informationen](#)". Below this, there are radio buttons for "Übersetzung" (selected) and "Original". The main heading is "System.Runtime.Serialization.Json-namespace". Below the heading, it says ".NET Framework (current version) | [Andere Versionen](#)". The main text reads: "Stellt Klassen bereit, die mit Json-Serialisierung verknüpft sind." On the right side, there is a sidebar with navigation options: "Vorschläge?", "Drucken", "Exportieren (0)", and "Teilen". Below these, it says "IN DIESEM ARTIKEL" followed by a link "Klassen".



JSON schema validation



JSON Schema Generator × +
jsonschema.net/#/ Suchen
JSONSchema.net Home About Contact Resources Previous Version

JSON

URL

JSON

```
{
  "Header": [
    {
      "symbol": "MSFT",
      "issuerName": "MICROSOFT CORP",
      "trdPrc": "51.82",
      "netChg": "-0.37",
      "pcntChg": "-0.71",
      "bid": "51.80",
      "ask": "51.81",
      "bidSize": "0",
      "askSize": "0",
      "cumVol": "33559073",
    }
  ]
}
```

Well done! You provided valid JSON.

Metadata Include metadata keywords

Schema

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "id": "",
  "type": "object",
  "properties": {
    "Header": {
      "id": "/Header",
      "type": "array",
      "items": {
        "id": "/Header/0",
        "type": "object",
        "properties": {
          "symbol": {
            "id": "/Header/0/symbol",
            "type": "string"
          },
          "issuerName": {
            "id": "/Header/0/issuerName",
            "type": "string"
          },
          "trdPrc": {
            "id": "/Header/0/trdPrc",
            "type": "string"
          }
        }
      }
    }
  }
}
```





.. and Microsoft ❤️ to store JSON data here



{JSON} JSON functions



- **ISJSON()**
return 1 if JSON snippet is valid
- **JSON_VALUE()**
extract a scalar value from JSON snippet
2nd option [lax | strict] to handle missing values
- **JSON_QUERY()**
extract an object or an array from JSON snippet



<xml /> Select & Ins/Upd/Del

- T-SQL conformer Ins/Upd/Del
- T-SQL XQuery
 - <column>.query (return xml)
 - <column>.value (return sql-Type)
 - <column>.exist (return bit)
 - <column>.nodes



<xml /> Select & Ins/Upd/Del

- T-SQL Xquery Navigation im <xml>
 - child
 - descendant
 - parent
 - Attribut
 - self
 - descendant-or-self

- .node()
- .text()



<xml /> CRUD Samples

```
SQLQuery1.sql x
1  ---- 1st Query
2  SELECT [key], xmlValue.query('*') as Complete_Sequence
3  FROM   dbo.CarManufacturer_xml
4
5  ---- 2nd Query
6  SELECT [key], xmlValue.query('data(*)') as Complete_Data
7  FROM   dbo.CarManufacturer_xml
8
9  ---- 3rd Query /node() & /comment()
10 SELECT [key]
11        , xmlValue.query('/Customer/CompanyName/node()') as CompanyName
12        , xmlValue.query('/Customer/comment()')
13 FROM   dbo.CarManufacturer_xml
14
```



<xml /> Indexing

- PRIMARY XML INDEX IX_primaryXml
- CREATE XML INDEX IX_name
ON sch.table (xmlCol)
USING XML INDEX IX_primaryXml
 - FOR PATH;
 - FOR VALUE;
 - FOR PROPERTY;



<xml /> Indexing

```
xml_Indexes.sql x
1 CREATE PRIMARY XML INDEX IX_primaryXml
2 ON dbo.eConsumption_xml ( xmlValue );
3 GO
4
5
6 CREATE XML INDEX IX_Xml_PATH
7 ON dbo.eConsumption_xml ( xmlValue )
8 USING XML INDEX IX_primaryXml
9 FOR PATH;
10
11 CREATE XML INDEX IX_Xml_PROPERTY
12 ON dbo.eConsumption_xml ( xmlValue )
13 USING XML INDEX IX_primaryXml
14 FOR PROPERTY;
15
16 CREATE XML INDEX IX_Xml_VALUE
17 ON dbo.eConsumption_xml ( xmlValue )
18 USING XML INDEX IX_primaryXml
19 FOR VALUE;
20 GO
```



<xml /> Schema Validation

- XML SCHEMA COLLECTION schemaName
- (table) xmlColumn
xml (DOCUMENT schemaName)



<xml /> Schema Validation

```
xml_Schema.sql x
1 CREATE XML SCHEMA COLLECTION dbo.bookSchemaCollection
2 AS
3 N'<?xml version="1.0"?>
4 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
5   targetNamespace="http://www.test-fabrik.de/BooksSchema"
6   xmlns="http://www.test-fabrik.de/BooksSchema"
7   elementFormDefault="qualified">
8   <xs:element name="Book">
9     <xs:complexType>
10      <xs:attribute name="Title" type="xs:string"/>
11      <xs:attribute name="Price" type="xs:decimal"/>
12    </xs:complexType>
13  </xs:element>
14 </xs:schema>';
15 GO
16
17 CREATE TABLE dbo.Books_xml(
18   ID int Identity(1,1)
19   , [key] nvarchar(50)
20   , xmlRaw xml
21   , xmlValue xml(DOCUMENT bookSchemaCollection)
22 );
23 GO
24
```



<xml /> Schema Validation

xml_Schema.sql

```
1  INSERT dbo.Books_xml
2  ( [key] , xmlValue )
3  VALUES ( 'key02', '<Book xmlns="http://www.test-fabrik.de/BooksSchema"
4             Title="the booksample2"
5             Price="49.90" />' )
6
7  -----
8  -- !! decimal ,
9  INSERT dbo.Books_xml
10 ( [key] , xmlValue )
11 VALUES ( 'key03', '<Book xmlns="http://www.test-fabrik.de/BooksSchema"
12             Title="! v ! Book"
13             Price="49,90" />' )
14
15 --Msg 6926, Level 16, State 1, Line 40
16 --XML Validation: Invalid simple type value: '49,90'. Location: /*:Book[1]/*:Price
```




<xml /> queries & transformation

- XML „*FLWOR*“
for let where order return
- Aggregat functions
 - ✓ {count(\$j)}
 - ✓ {sum(\$j)}
 - ✓ {avg(\$j)}



<xml /> queries & transformation

```
xml_FLWOR.sql x
1  -- sample FLWOR
2  SELECT ID, [key], xmlValue
3      , xmlValue.value('(CustomerData/Customer/CustID)[1]', 'nvarchar(40)' ) as 'nvarchar_CustID'
4      , xmlValue.query(' for $i in CustomerData/Customer/CustID
5                          let $j := CustomerData/Customer/Consumption/consumptionValue
6                          where avg($j) >= 5
7                          return
8                          <result>
9                              {data($j)}
10                         </result>
11                         ') as 'data'
12      , xmlValue.query(' for $i in CustomerData/Customer/CustID
13                          let $j := CustomerData/Customer/Consumption/consumptionValue
14                          where avg($j) >= 5
15                          return
16                          <result>
17                              <count> {count($j)} </count>
18                              <sum> {sum($j)} </sum>
19                              <avg> {avg($j)} </avg>
20                          </result>
21                          ') as 'summary'
22  FROM  dbo.eConsumtion_xml;
```

???