

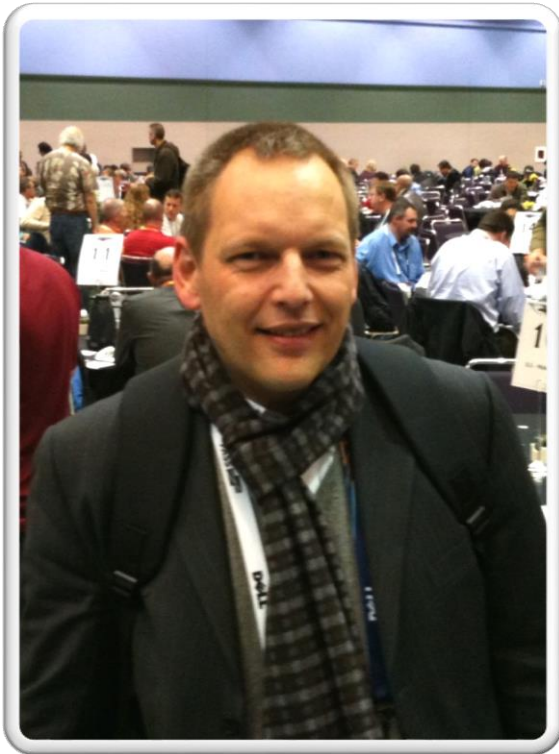


SQL Server Konferenz 2016

powered by PASS Deutschland e.V.

JSON, das neue und bessere <xml> ? Alexander Karl & Sascha Dittmann

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 MCSE



About me



Sascha Dittmann



Cloud Solution Architect



Twitter: @SaschaDittmann

Blog: <http://www.sascha-dittmann.de>

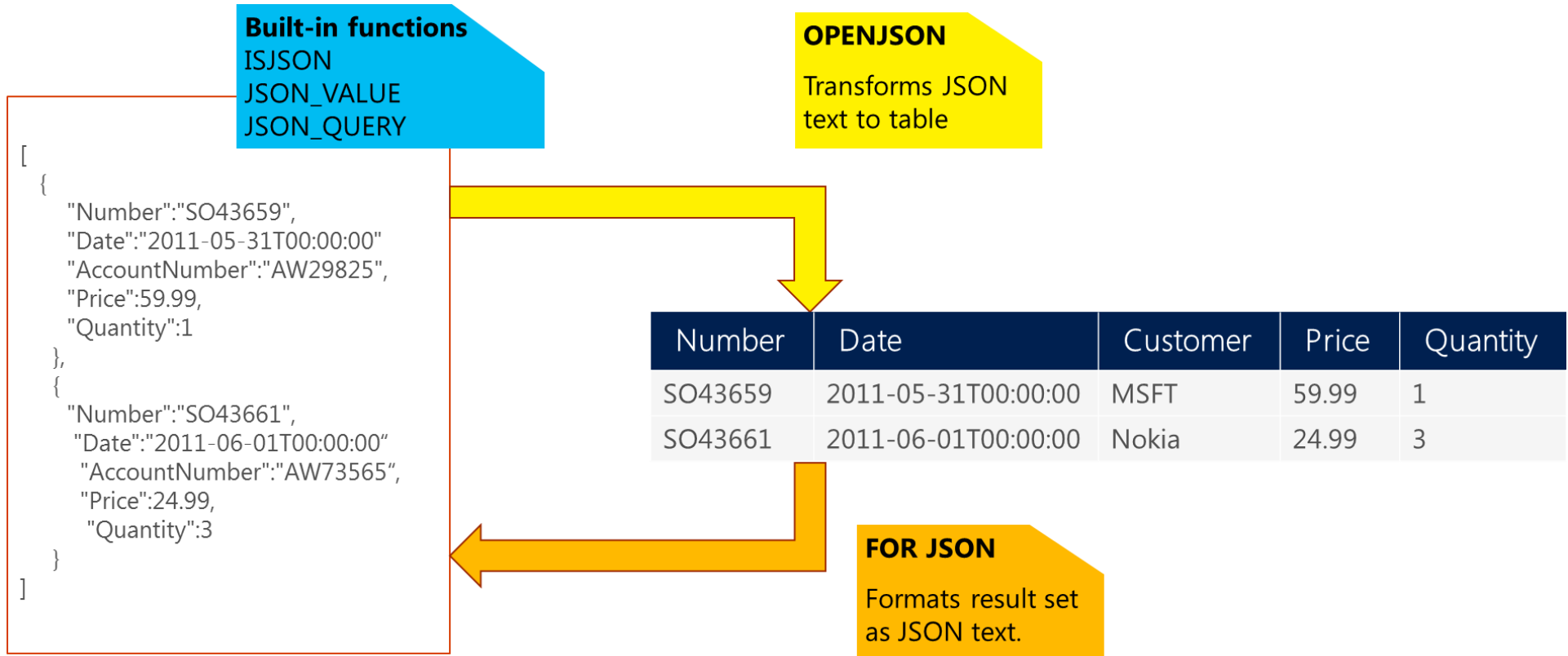


Agenda

- JSON „neu im SQL Server 2016“
- passende Tools
- Syntax For JSON / For XML
- Syntax OpenJSON / OpenXML
- Datentyp ? JSON / XML
- Indexierung



JSON „round trip“



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



<xml /> Tooling

The screenshot displays the SQL Server Enterprise Manager interface. On the left, the Object Explorer shows the database structure, with the `xmlValue` column of the `dbo.CarManufacturer_xml` table highlighted. The main window shows a SQL query in `SQLQuery4.sql` that selects `ID`, `key`, and `xmlValue` from `dbo.CarManufacturer_xml`. Below the query, the Results pane shows a table with three rows of data. A pop-up window titled `xmlValue2.xml` shows the XML content for the first row, which is a `<Customer>` element with attributes `CustomerID="1021"` and a comment `!-- info 1021 customer since 2010-->`, followed by `<CompanyName>Alfa Romeo</CompanyName>` and `<CustomerSince>2010</CustomerSince>`.

```
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...

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



{JSON}

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 x demo01.sql  
1 {"databases":[{"name":"master","database id":1,"create date":"2003-04-08T09:13:36.390"}
```



← → http://www.codeproject.com/Tips/216175/View-JSON-in-Internet-Explorer View JSON in Internet Explo... x

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

CODE PROJECT For those who code

DOLBY AUDIO

ENTER THE **DOLBY AUDIO CHALLENGE**

GRAND PRIZE **\$10,000 USD** [CLICK HERE](#)

home **articles** quick answers discussions features community help

Search for articles, questions, tips

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.

ENTER THE **DOLBY AUDIO CHALLENGE**



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

CODE PROJECT
For those who code

DOLBY AUDIO

ENTER THE **DOLBY AUDIO CHALLENGE**

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

CLICK HERE

http://www.codeproject.com/Tips/216175/View-JSON-in-Internet-Explorer

View JSON in Internet Explo... x

http://jsonview.com/example.json

jsonview.com x

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

Tagged as

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

```
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.

Hide Copy Code

ENTER THE **DOLBY AUDIO CHALLENGE**



Firefox Add-ons Manager: about:addons

Erweiterungen

JSONView 1.1.0

Von Ben Hollis

```
{
  key: "foo",
  number: 243,
  - anObject: {
    whoa: "nuts",
    - anarray: [
      1,
      2,
      "threeeee"
    ],
    waaa: "stuff"
  },
  awesome: true,
  bogus: false,
  meaning: null,
  japanese: "何日か@B.",
  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 containing 'http://jsonview.com/example.json'. The main content area displays a JSON object with the following structure:

```
{
  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 (input)

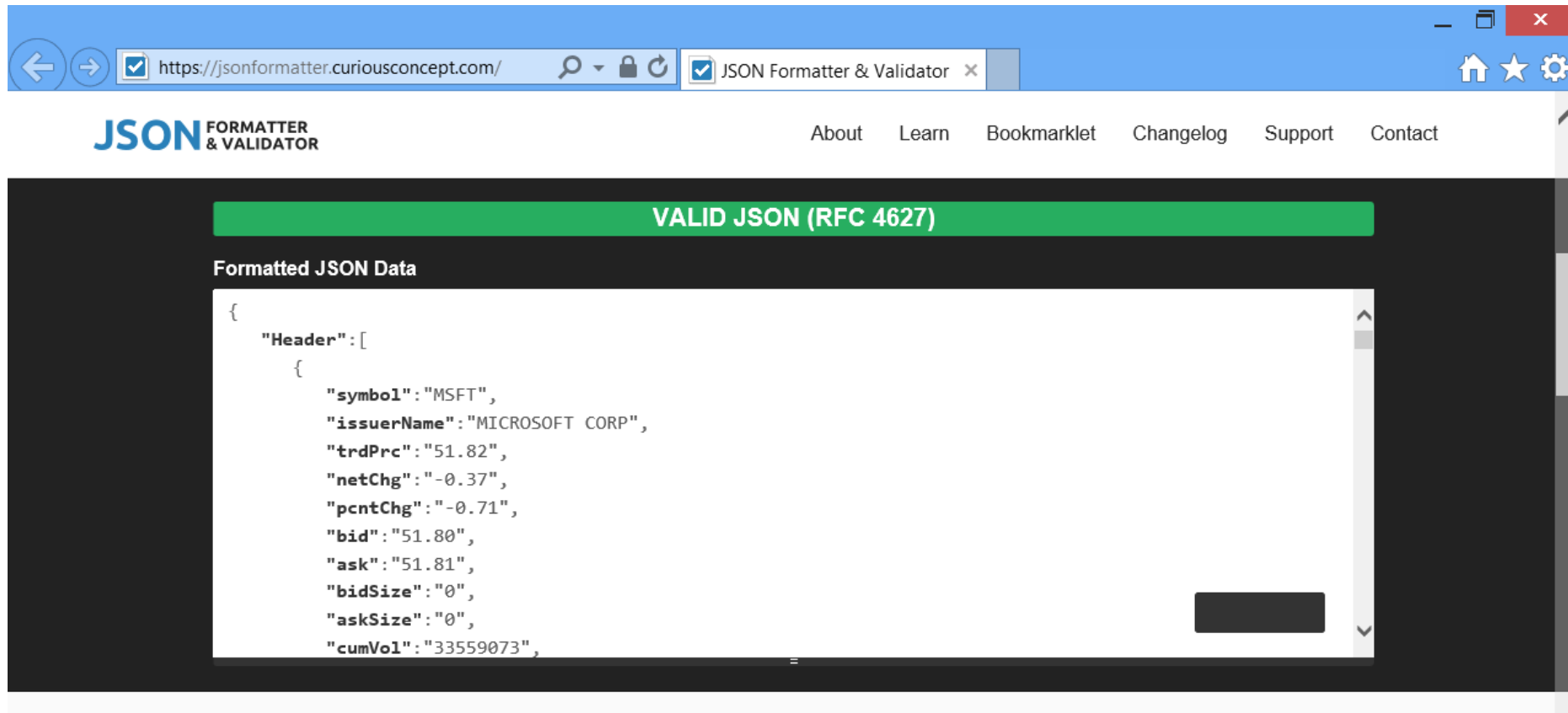
The screenshot shows a web browser window with the URL `https://jsonformatter.curiousconcept.com/`. The page title is "JSON FORMATTER & VALIDATOR". The navigation menu includes "About", "Learn", "Bookmarklet", "Changelog", "Support", and "Contact".

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 JSON object. 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 "Process" button.

```
{ "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" } ], "Data": [ { "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": "20.30", "netChgX": "0.0", "bidX": "22.20", "askX": "22.70", "cumVolX": "0", "opnIntX": "1", "shortDatedIndX": "0" } ] }
```



JSON Formatter & Validator (result)



The screenshot shows a web browser window with the URL <https://jsonformatter.curiousconcept.com/>. The page title is "JSON FORMATTER & VALIDATOR". The navigation menu includes "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:

```
{
  "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



FOR JSON

- **AUTO Mode**

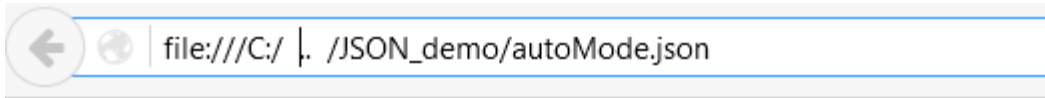
```
SELECT name, database_id, create_date
FROM sys.databases
FOR JSON Auto
```

- **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



```
[
  {
    name: "master",
    database_id: 1,
    create_date: "2003-04-08T09:13:36.390"
  },
  {
    name: "tempdb",
    database_id: 2,
    create_date: "2016-02-24T11:40:54.827"
  },
  {
    name: "model",
    database_id: 3,
    create_date: "2003-04-08T09:13:36.390"
  },
  FROM sys.databases
  FOR JSON PATH, ROOT('databases'), Include_NULL_Values;
  _id,
  ate,
  ion.is_published,
  ion.is_subscribed,
  ion.is_distributor,
```



FOR JSON / Result

```
file:///C:/.../JSON_demo/pathMode.json
```

```
{
  "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
      }
    }
  ]
}
```

```
FROM sys.databases
FOR JSON PATH
```



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 / Result

	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

	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

:",

42, "MDX"],
"Territory": "Hessen"}



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



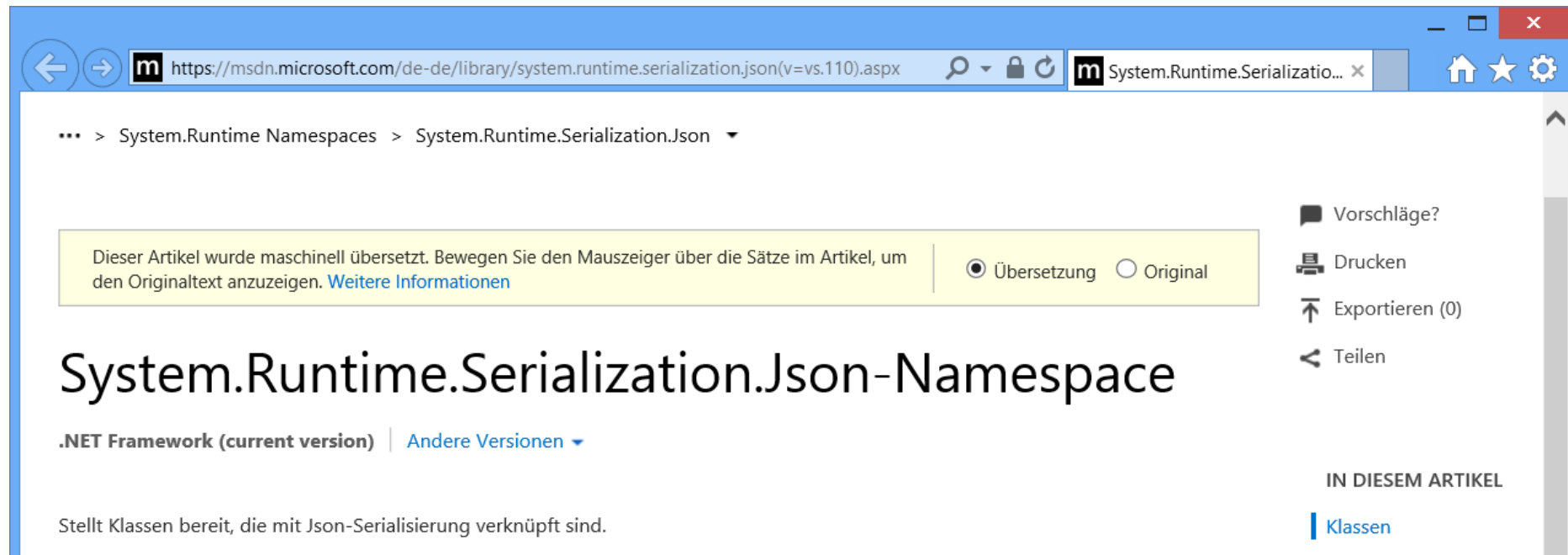
Storing JSON

- `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". Underneath, 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". At the bottom of the sidebar, it says "IN DIESEM ARTIKEL" followed by a link "Klassen".



JSON schema validation

The screenshot shows the JSON Schema Generator website. The browser address bar displays 'jsonschema.net/#/'. The page title is 'JSON Schema Generator'. The navigation menu includes 'Home', 'About', 'Contact', 'Resources', and 'Previous Version'. The main content area is divided into two panels: 'JSON' and 'Schema'.

JSON Panel:

- URL: `http://jsonschema.net`
- JSON Input:

```
{
  "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",
    }
  ]
}
```
- Message: **Well done! You provided valid JSON.**
- Buttons: **Generate Schema** and **Reset**
- Metadata: Include metadata keywords

Schema Panel:

- Buttons: **Code View**, **Edit View**, **String View**
- Schema Output:

```
{
  "$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"
          }
        }
      }
    }
  }
}
```



switch to





Setup

The screenshot displays the Azure portal interface. On the left, a navigation pane shows categories like 'NEU', 'STARTSEITE', 'BENACHRICHTL...', 'DURCHSUCHEN', 'AKTIV', 'ABRECHNUNG', and 'HILFE'. The main area is titled 'Erstellen' (Create) and 'Speicher, Cache und Sicherung' (Storage, Cache and Security). It lists various services: SQL Database, Azure DocumentDB (highlighted), Storage, Redis Cache, Search, and StorSimple. Below these are 'Zuletzt verwendet' (Recently used) items like Storage and Redis Cache. At the bottom, there's a link to 'Azure Marketplace'. On the right, a side panel titled 'DocumentDB-Konto' (DocumentDB Account) is open, showing configuration options: ID (with a text input field containing 'documents.azure.com'), KONTOSTUFE (Standard), RESSOURCENGRUPPE (New_Resource_Group-1), ABONNEMENT (MVP 1), and STANDORT (Nordeuropa). An information message states: 'Das Erstellen eines DocumentDB-Kontos dauert mindestens 10 Minuten.' (Creating a DocumentDB account takes at least 10 minutes). There is a checkbox for 'An Startmenü anheften' (Pin to Start menu) and an 'Erstellen' (Create) button.





Tooling

The screenshot shows a document explorer window with a dark blue header. The title bar reads "Document Explorer" and the current document is identified by the ID "78118a99-0c89-488e-82ef-3016b08c0424".

The interface includes a toolbar with the following actions: "Create Document", "Add Document", "Refresh", "Settings", "Save", "Discard", "Delete", and "Properties".

On the left side, there are three sections:

- Database:** A dropdown menu showing "SQLSat409".
- Collection:** A dropdown menu showing "CarManufacturer".
- Documents:** A search bar with the placeholder "Filter by id" and a list of document IDs. The selected ID, "78118a99-0c89-488e-82ef-3016b08c0424", is highlighted in blue.

The main area on the right displays the JSON content of the selected document, numbered from 1 to 7:

```
1 {
2   "id": "78118a99-0c89-488e-82ef-3016b08c0424",
3   "CustomerID": 1021,
4   "Comment": "info 1021 customer since 2010",
5   "CompanyName": "Alfa Romeo",
6   "CustomerSince": 2010
7 }
```





Tooling

Query Explorer

Load File

Run query

Database
SQLSat409

Collection
CarManufacturer

```
SELECT m.CompanyName, m.CustomerSince  
FROM Manufacturer m  
WHERE m.CustomerSince = 2010
```

QUERY COMPLETED SUCCESSFULLY

Results
SELECT m.CompanyName, m.CustomerSince FROM Manufacturer m WHERE m.CustomerSince = 2010

Previous page ... Next page ...

```
[  
  {  
    "CompanyName": "Alfa Romeo",  
    "CustomerSince": 2010  
  },  
  {  
    "CompanyName": "Aston Martin",  
    "CustomerSince": 2010  
  },  
  {  
    "CompanyName": "Audi",  
    "CustomerSince": 2010  
  },  
  {  
    "CompanyName": "Bentley",  
    "CustomerSince": 2010  
  },  
  {  
    "CompanyName": "BMW",
```





Tooling

The screenshot displays the 'Script Explorer' window in SQL Server Enterprise Manager. The left pane shows the 'Database' set to 'FamilyRegistry' and the 'Collection' set to 'FamilyCollection'. Under the 'SCRIPTS' section, the 'createMyDocument' stored procedure is selected and highlighted in blue. The right pane shows the code for this stored procedure, which is a JavaScript function that interacts with a REST API to create a document.

```
1 function(documentToCreate) {
2   var context = getContext();
3   var collection = context.getCollection();
4
5   var accepted = collection.createDocument(
6     collection.getSelfLink(),
7     documentToCreate,
8     function (err, documentCreated) {
9       if (err) throw new Error('Error' + err.message);
10      context.getResponse().setBody(documentCreated.id)
11    });
12   if (!accepted) return;
13 }
```





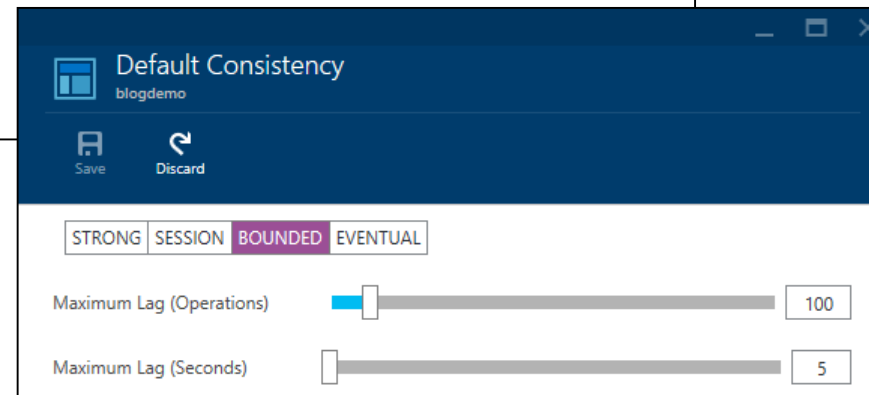
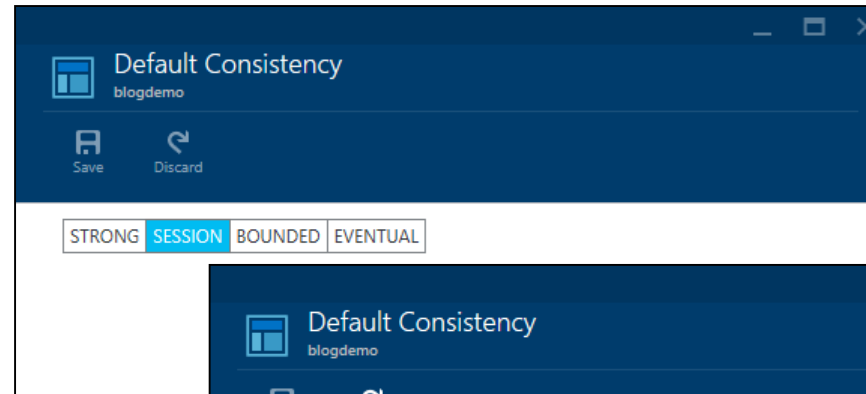
Consistency (BASE)

B asically

A vailable

S oft state

E ventual consistency





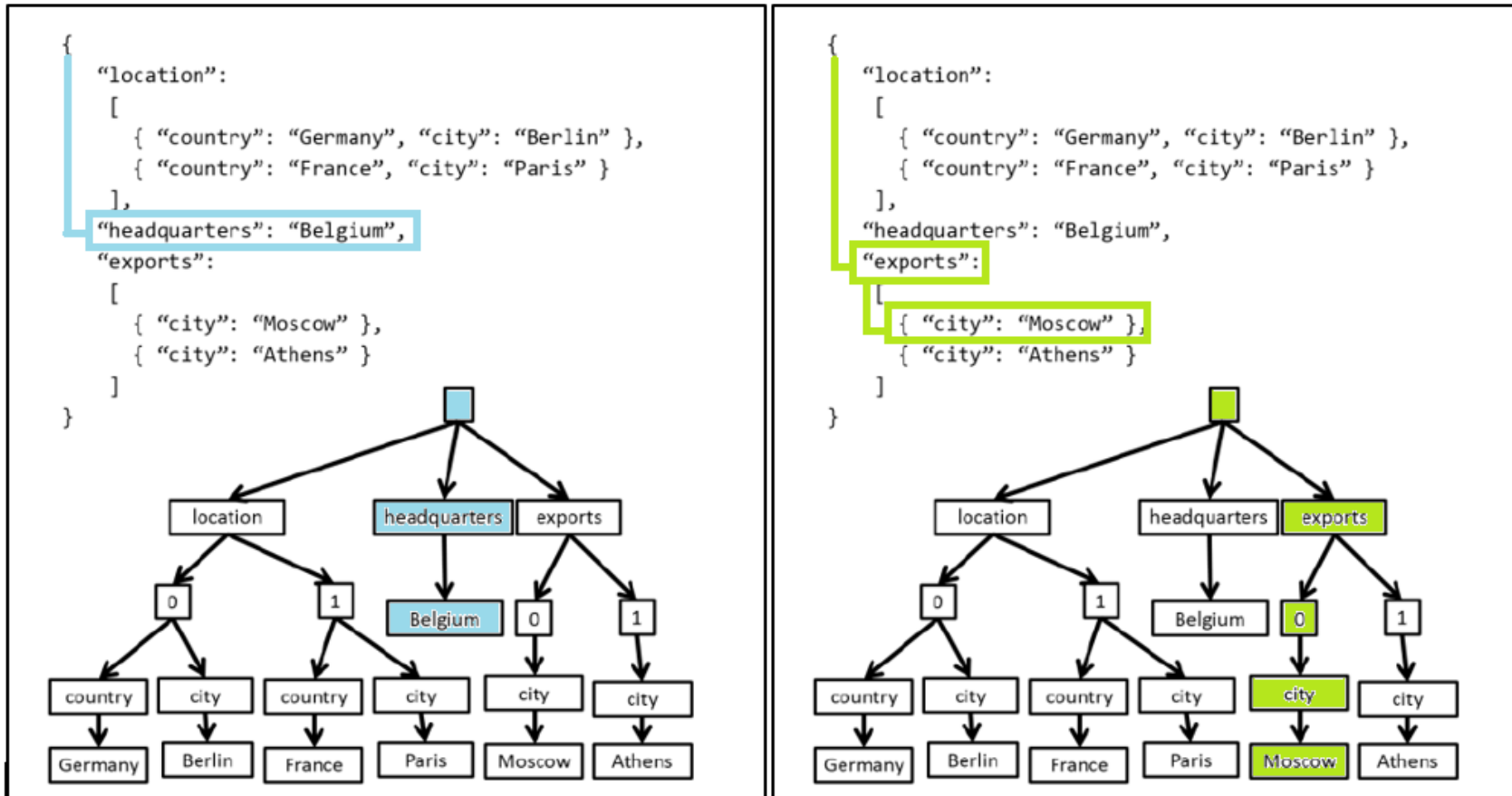
Select & Ins/Upd/Del

- LINQ, SQL
- REST API over HTTPS
 - .NET
 - Node.js
 - JavaScript
 - Python
 - ...





Indexing





Indexing Policies

```
{
  "id": "customIndexCollection",
  "indexingPolicy": {
    "automatic": true,
    "indexingMode": "Consistent",
    "IncludedPaths": [
      {
        "IndexType": "Hash",
        "Path": "/"
      }
    ],
    "ExcludedPaths": [
      "/\\nonIndexedContent\\/*"
    ]
  }
}
```





server side code

```
function(documentToCreate) {  
    var context = getContext();  
    var collection = context.getCollection();  
  
    var accepted = collection.createDocument(  
        collection.getSelfLink(),  
        documentToCreate,  
        function (err, documentCreated) {  
            if (err) throw new Error('Error' + err.message);  
            context.getResponse().setBody(documentCreated.id)  
        });  
    if (!accepted) return;  
}
```





compl. queries

```
SELECT c.id, v.ConsumptionDay  
FROM Customer c  
JOIN v IN c.Consumptions  
WHERE v.ConsumptionValue = 5
```



???



