

Using the SQL API for a Document Data Model



Leonard Lobel

CTO, SLEEK TECHNOLOGIES

lennilobel.wordpress.com



What Is a Document Database?

Object graph

Semi-structured JSON
(JavaScript Object
Notation) documents

```
{
  "business_id": "PK6a5",
  "full_address": "400",
  "hours": {},
  "open": true,
  "categories": [
    "Burgers",
    "Fast Food",
    "Restaurants"
  ],
  "city": "Homestead",
  "review_count": 5,
  "name": "McDonald's",
  "neighborhoods": [
    "Homestead"
  ],
  "longitude": -79.9106,
  "state": "PA",
  "stars": 2,
  "latitude": 40.412086,
  "attributes": {
    "Take-out": true,
    "Wi-Fi": "free",
    "Drive-Thru": true,
    "Good For": {
      "dessert": false,
      "latenight": false,
      "lunch": false,
      "dinner": false,
      "breakfast": false,
      "brunch": false
    },
    "Caters": false,
    "Noise Level": "average",
    "Takes Reservations": false,
    "Delivery": false
  }
}
```

```
{
  "id": 10,
  "name": "Clemens",
  "username": "M",
  "email": "Rey.",
  "address": {
    "street": "K",
    "suite": "Su",
    "city": "Leb",
    "zipcode": "1",
    "geo": {
      "lat": "-3",
      "lng": "57"
    }
  },
  "phone": "024-",
  "website": "am",
  "company": {
    "name": "Hoe",
    "catchPhrase",
    "bs": "target"
  }
}
```

```
{
  "id": "1242160000000072038",
  "description": "3",
  "website": "3",
  "numberOfEmployees": "3",
  "phone": "3",
  "name": "account3",
  "shippingAddress": {
    "country": "3",
    "stateOrProvince": "3",
    "city": "3",
    "postalCode": "3",
    "street1": "3"
  },
  "billingAddress": {
    "country": "3",
    "stateOrProvince": "3",
    "city": "3",
    "postalCode": "3",
    "street1": "3"
  }
}
```

```
{
  "train": {
    "date": "07/04/2016",
    "time": "09:30",
    "from": "New York",
    "to": "Chicago",
    "seat": "57B"
  },
  "passenger": {
    "name": "John Smith"
  },
  "price": 1234.25,
  "comments": ["Lunch & dinner incl.", "\"Have a nice day!\""]
}
```

What is a Document Database?

Object graph

Semi-structured JSON
(JavaScript Object
Notation) documents

Two API choices

- 1) SQL API
(fka DocumentDB API)
- 2) MongoDB API



Data Modeling: Relational vs. Document

Relational Store

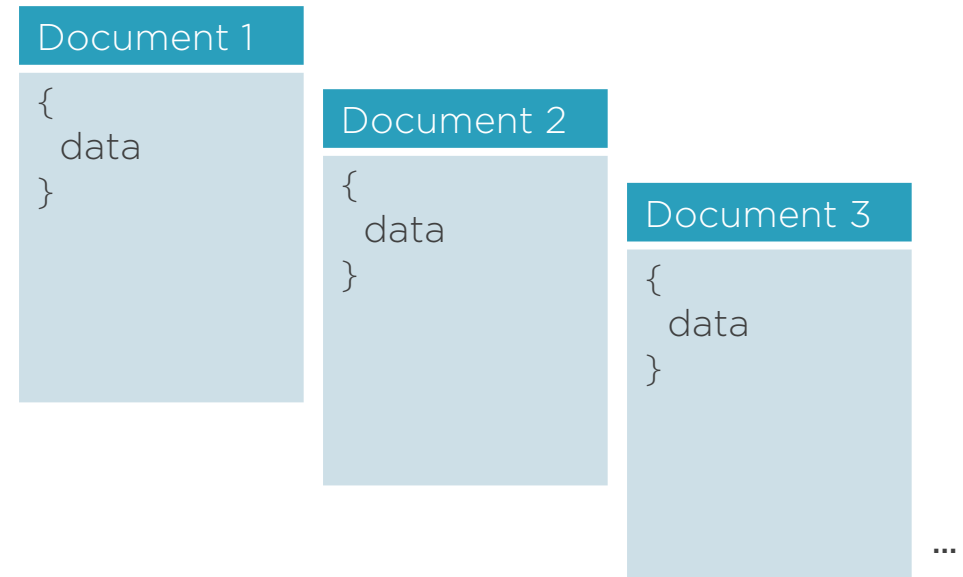
Document Store



Data Modeling: Relational vs. Document

Relational Store	Document Store
Rows	Documents

	Table
Row 1	Data
Row 2	Data
Row 3	Data
...	:



Data Modeling: Relational vs. Document

Relational Store	Document Store
Rows	Documents
Columns	Properties

Col1	Col2	Col3	Col4
Data	Data	Data	Data
Data	Data	Data	Data
Data	Data	Data	Data

Document 1

```
{  
  "prop1": data,  
  "prop2": data,  
  "prop3": data,  
  "prop4": data  
}
```

Document 2

```
{  
  "prop1": data,  
  "prop2": data,  
  "prop3": data,  
  "prop4": data  
}
```

Document 3

```
{  
  "prop1": data,  
  "prop2": data,  
  "prop3": data,  
  "prop4": data  
}
```



Data Modeling: Relational vs. Document

Relational Store	Document Store
Rows	Documents
Columns	Properties
Strongly-typed schemas	Schema-free

ID	Name	IsActive	Dob
1	John Smith	True	8/30/1964
2	Sarah Jones	False	2/18/2002
3	Adam Stark	True	7/13/1987

Document 1

```
{
  "id": 1,
  "name": "John Smith",
  "isActive": true,
  "dob": "1964-30-08"
}
```

Document 2

```
{
  "id": 2,
  "fullName": "Sarah Jones",
  "dob": "2002-02-18"
}
```

Document 3

```
{
  "id": 3,
  "fullName": {
    "first": "Adam",
    "last": "Stark"
  },
  "isActive": true,
  "dob": "2015-04-19"
}
```

Data Modeling: Relational vs. Document

Relational Store	Document Store
Rows	Documents
Columns	Properties
Strongly-typed schemas	Schema-free
Highly normalized	Typically denormalized

User Table

UserID	Name	Dob
1	John Smith	8/30/1964

Holdings Table

StockID	UserID	Qty	Symbol
1	1	100	MSFT
2	1	75	WMT

Document

```
{
  "id": 1,
  "name": "John Smith",
  "dob": "1964-30-08",
  "holdings": [
    { "qty": 100, "symbol": "MSFT" },
    { "qty": 75, "symbol": "WMT" }
  ]
}
```



Data Modeling: Relational vs. Document

Relational Store	Document Store
Rows	Documents
Columns	Properties
Strongly-typed schemas	Schema-free
Highly normalized	Typically denormalized

Document

```
{
  "postid": "1",
  "title": "My blog post",
  "body": "Post content...",
  "comments": [
    "comment #1",
    "comment #2",
    "comment #3",
    "comment #4",
    :
    "comment #1598873",
    :
  ]
}
```



Data Modeling: Relational vs. Document

Relational Store	Document Store
Rows	Documents
Columns	Properties
Strongly-typed schemas	Schema-free
Highly normalized	Typically denormalized

Document

```
{  
  "postId": "1",  
  "title": "My blog post",  
  "body": "Post content..."  
}
```

Document

Document

Document

```
{  
  "postId": "1",  
  "comment": "comment #3"  
}
```



Data Modeling: Relational vs. Document

Relational Store	Document Store
Rows	Documents
Columns	Properties
Strongly-typed schemas	Schema-free
Highly normalized	Typically denormalized

Document

```
{
  "postId": "1",
  "title": "My blog post",
  "body": "Post content...",
  "comments": [
    "comment #1",
    "comment #2",
    :
    "comment #100"
  ]
}
```

Document

Document

Document

```
{
  "postId": "1",
  "comments": [
    "comment #301",
    "comment #302",
    :
    "comment #400"
  ]
}
```



Data Modeling: Relational vs. Document

Relational Store	Document Store
Rows	Documents
Columns	Properties
Strongly-typed schemas	Schema-free
Highly normalized	Typically denormalized

Document

```
{
  "itemid": "1",
  "part": "1",
  "prop1": "Typically read",
  "prop2": "Typically read",
  "prop3": "Typically read",
  "prop4": "Typically read",
  :
  "prop200": "Typically read",
}
```

Document

```
{
  "itemid": "1",
  "part": "2",
  "prop1": "Typically updated",
  "prop2": "Typically updated",
  :
  "prop10": "Typically updated",
}
```



Data Modeling: Relational vs. Document

Relational Store	Document Store
Rows	Documents
Columns	Properties
Strongly-typed schemas	Schema-free
Highly normalized	Typically denormalized

Document

```
{
  "name": "John Smith",
  "orderDate": "2015-30-03",
  "details": [
    { "qty": 5, "code": "BC", "desc": "Black chair" },
    { "qty": 1, "code": "RT", "desc": "Red table" },
    { "qty": 2, "code": "YC", "desc": "Yellow clock" },
  ]
}
```

Document

```
{
  "name": "Sarah Jones",
  "orderDate": "2015-16-04",
  "details": [
    { "qty": 1, "code": "PL", "desc": "Purple lamp" },
    { "qty": 3, "code": "YC", "desc": "Yellow clock" },
  ]
}
```

Document

```
{
  "name": "Adam Stark",
  "orderDate": "2015-06-05",
  "details": [
    { "qty": 2, "code": "BC", "desc": "Black chair" },
    { "qty": 1, "code": "YC", "desc": "Yellow clock" },
  ]
}
```

Data Modeling: Relational vs. Document

Relational Store	Document Store
Rows	Documents
Columns	Properties
Strongly-typed schemas	Schema-free
Highly normalized	Typically denormalized

Document

```
{
  "name": "John Smith",
  "orderDate": "2015-30-03",
  "details": [
    { "qty": 5, "code": "BC" },
    { "qty": 1, "code": "RT" },
    { "qty": 2, "code": "YC" }
  ]
}
```

Document

```
{
  "name": "Sarah Jones",
  "orderDate": "2015-16-04",
  "details": [
    { "qty": 1, "code": "PL" },
    { "qty": 3, "code": "YC" }
  ]
}
```

Document

```
{
  "name": "Adam Stark",
  "orderDate": "2015-06-05",
  "details": [
    { "qty": 2, "code": "BC" },
    { "qty": 1, "code": "YC" }
  ]
}
```

Document

```
{
  "code": "BC",
  "desc": "Black chair"
}
```

Document

```
{
  "code": "YC",
  "desc": "Yellow clock"
}
```

Document

```
{
  "code": "RT",
  "desc": "Red table"
}
```



Data Modeling: Relational vs. Document

Relational Store	Document Store
Rows	Documents
Columns	Properties
Strongly-typed schemas	Schema-free
Highly normalized	Typically denormalized

Document

```
{
  "id": "1",
  "name": "John Smith",
  "dob": "1964-30-08",
  "holdings": [
    { "qty": 100, "symbol": "MSFT" },
    { "qty": 75, "symbol": "WMT" }
  ]
}
```



Cosmos DB Resource Model



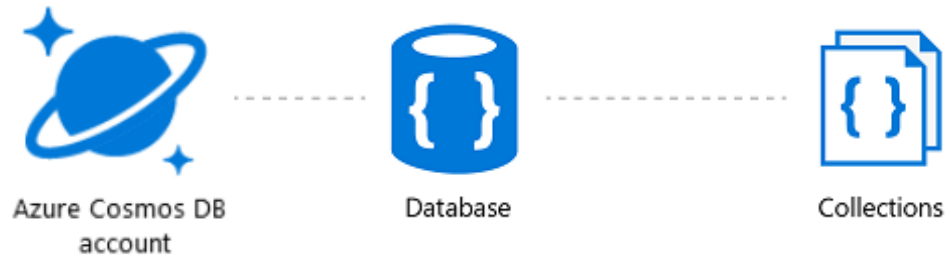
Azure Cosmos DB
account



Cosmos DB Resource Model



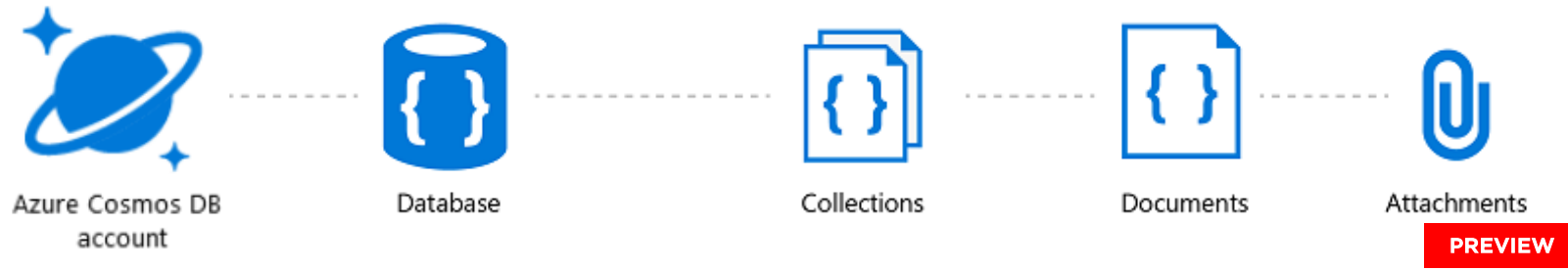
Cosmos DB Resource Model



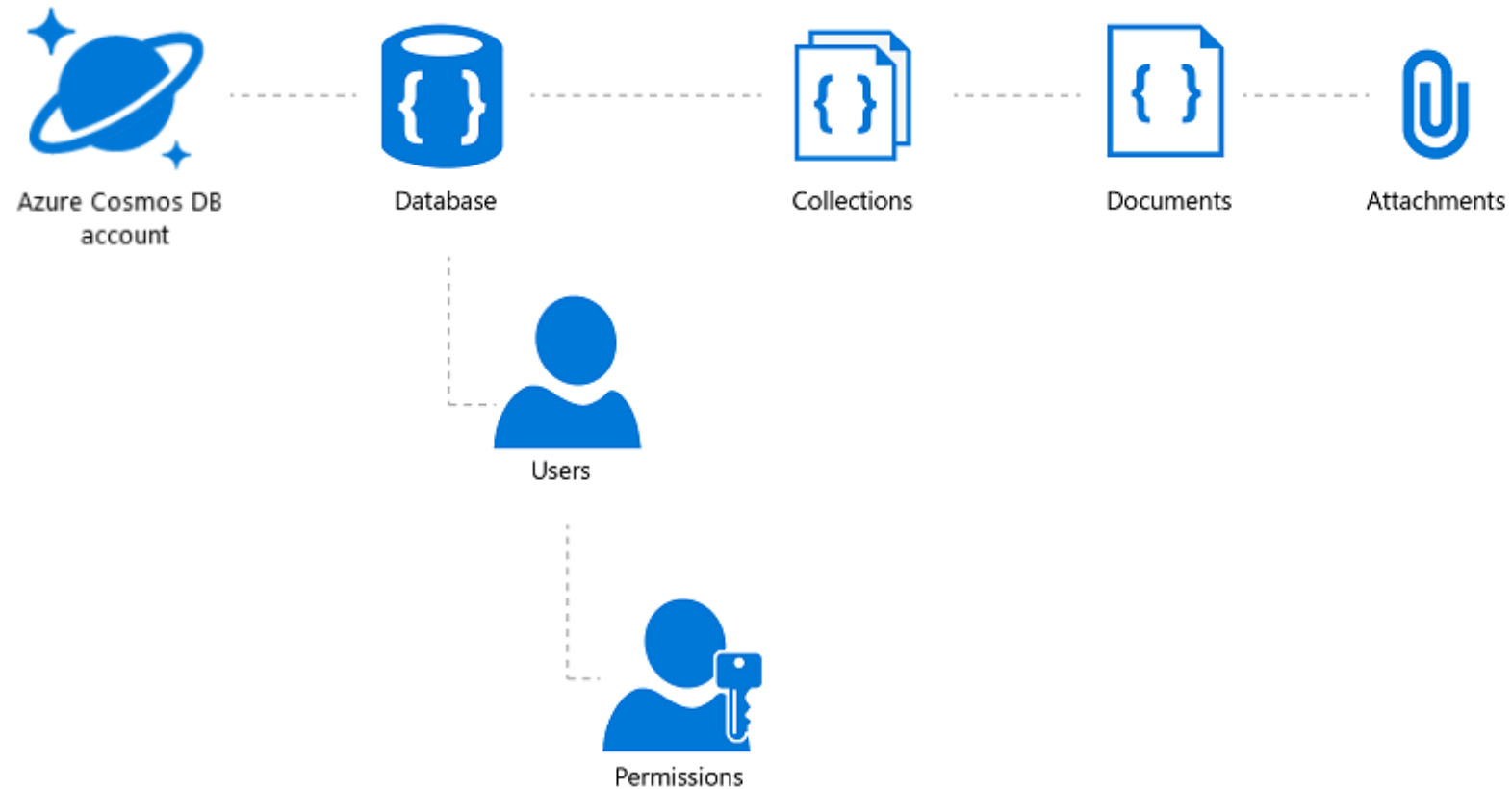
Cosmos DB Resource Model



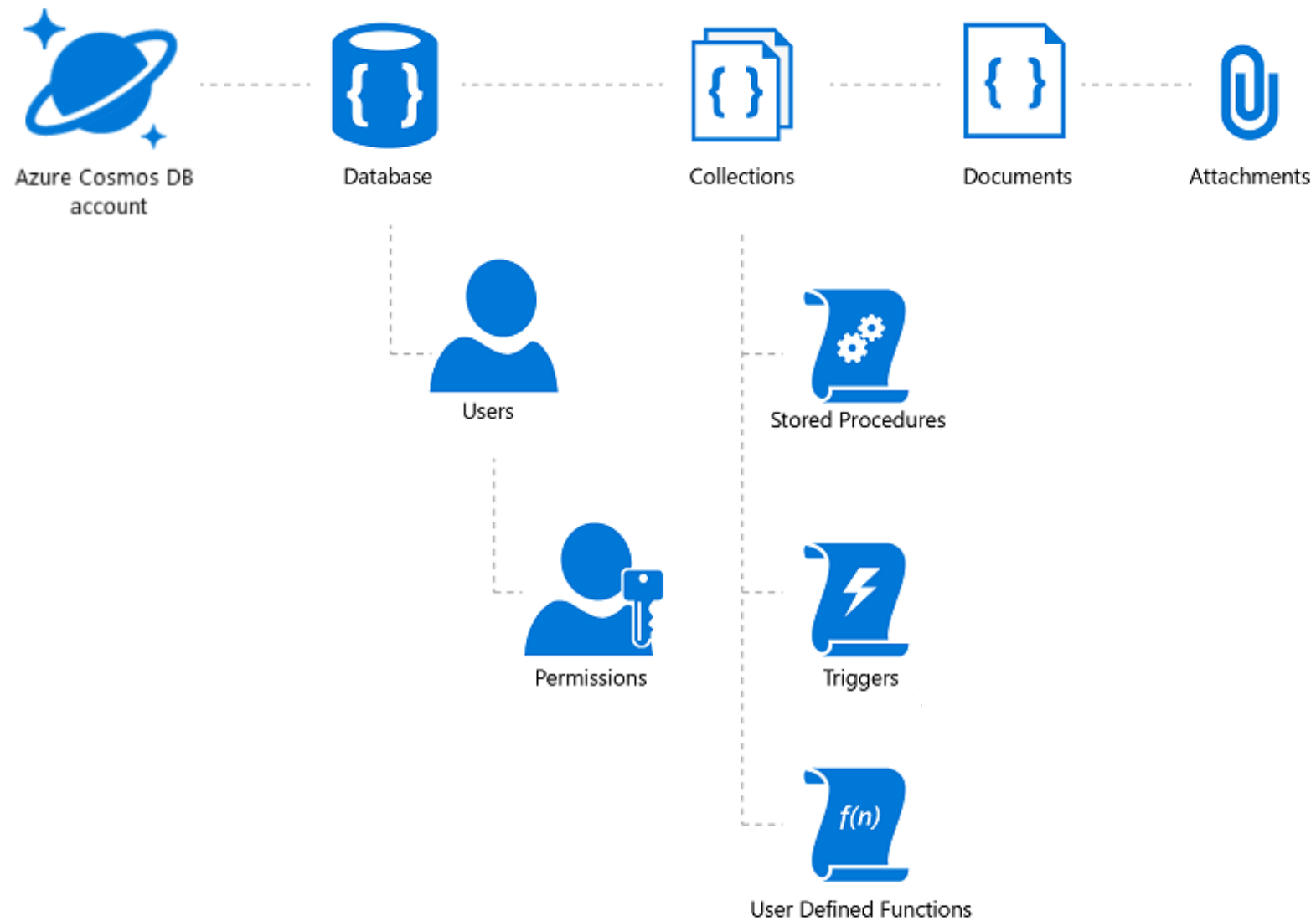
Cosmos DB Resource Model



Cosmos DB Resource Model



Cosmos DB Resource Model



Resource Properties, Self-links, and URIs

Every resource contains these properties



Resource Properties, Self-Links, and URIs

Every resource contains these properties

Property	Value
id	User-defined unique ID (string)



Resource Properties, Self-Links, and URIs

Every resource contains these properties

Property	Value
id	User-defined unique ID (string)
_rid	Resource ID



Resource Properties, Self-Links, and URIs

Every resource contains these properties

Property	Value
id	User-defined unique ID (string)
_rid	Resource ID
_ts	Timestamp (last updated) epoch value



Resource Properties, Self-Links, and URIs

Every resource contains these properties

Property	Value
id	User-defined unique ID (string)
_rid	Resource ID
_ts	Timestamp (last updated) epoch value
_etag	GUID used for optimistic concurrency



Resource Properties, Self-Links, and URIs

Every resource contains these properties

Property	Value
id	User-defined unique ID (string)
_rid	Resource ID
_ts	Timestamp (last updated) epoch value
_etag	GUID used for optimistic concurrency
_self	URI path to the resource



Resource Properties, Self-Links, and URIs

Every resource contains these properties

Property	Value
id	User-defined unique ID (string)
_rid	Resource ID
_ts	Timestamp (last updated) epoch value
_etag	GUID used for optimistic concurrency
_self	URI path to the resource
_attachments	URI path suffix to the resource attachments



Resource Properties, Self-Links, and URIs

```
{  
  "familyName": "Smith",  
  "address": {  
    "addressLine": "123 Main Street",  
    "city": "Chicago",  
    "state": "IL",  
    "zipCode": "60601"  
  },  
  "parents": [  
    "Peter",  
    "Alice"  
  ],  
  "kids": [  
    "Adam",  
    "Jacqueline",  
    "Joshua"  
  ]  
}
```



Resource Properties, Self-Links, and URIs

```
{  
  "familyName": "Smith",  
  "address": {  
    "addressLine": "123 Main Street",  
    "city": "Chicago",  
    "state": "IL",  
    "zipCode": "60601"  
  },  
  "parents": [  
    "Peter",  
    "Alice"  
  ],  
  "kids": [  
    "Adam",  
    "Jacqueline",  
    "Joshua"  
  ],  
  "id": "ab7aafc4-6a6e-f19e-bf5f-f7985b2f156d"  
}
```



Resource Properties, Self-Links, and URIs

```
{
  "familyName": "Smith",
  "address": {
    "addressLine": "123 Main Street",
    "city": "Chicago",
    "state": "IL",
    "zipCode": "60601"
  },
  "parents": [
    "Peter",
    "Alice"
  ],
  "kids": [
    "Adam",
    "Jacqueline",
    "Joshua"
  ],
  "id": "ab7aafc4-6a6e-f19e-bf5f-f7985b2f156d",
  "_rid": "v84pA07nPQABAAAAAABg==",
  "_self": "dbs/v84pAA==/colls/v84pA07nPQA=/docs/v84pA07nPQABAAAAAABg==/",
  "_etag": "\"00004700-0000-0000-0000-5a3e991a0000\"",
  "_attachments": "attachments/",
  "_ts": 1514051866
}
```

```
var updatedAt = new DateTime(1970, 1, 1, 0, 0, 0, DateTimeKind.Utc)
    .AddSeconds(1514051866)
    .ToLocalTime()
    .ToString("ddd M/d/yyyy h:m:s tt");
```



Resource Properties, Self-Links, and URIs

```
{
  "familyName": "Smith",
  "address": {
    "addressLine": "123 Main Street",
    "city": "Chicago",
    "state": "IL",
    "zipCode": "60601"
  },
  "parents": [
    "Peter",
    "Alice"
  ],
  "kids": [
    "Adam",
    "Jacqueline",
    "Joshua"
  ],
  "id": "ab7aafc4-6a6e-f19e-bf5f-f7985b2f156d",
  "_rid": "v84pA07nPQABAAAAAABg==",
  "_self": "dbs/v84pAA==/colls/v84pA07nPQA=/docs/v84pA07nPQABAAAAAABg==/",
  "_etag": "\"00004700-0000-0000-0000-5a3e991a0000\"",
  "_attachments": "attachments/",
  "_ts": 1514051866
}
```

```
var updatedAt = new DateTime(1970, 1, 1, 0, 0, 0, DateTimeKind.Utc)
    .Add(updatedAt)
    .ToLocalTime()
    .ToString("ddd M/d/yyyy h:m:s tt");
```



Resource Properties, Self-Links, and URIs

```
{
  "familyName": "Smith",
  "address": {
    "addressLine": "123 Main Street",
    "city": "Chicago",
    "state": "IL",
    "zipCode": "60601"
  },
  "parents": [
    "Peter",
    "Alice"
  ],
  "kids": [
    "Adam",
    "Jacqueline",
    "Joshua"
  ],
  "id": "ab7aafc4-6a6e-f19e-bf5f-f7985b2f156d",
  "_rid": "v84pA07nPQABAAAAAABg==",
  "_self": "dbs/v84pAA==/colls/v84pA07nPQA=/docs/v84pA07nPQABAAAAAABg==/",
  "_etag": "\"00004700-0000-0000-0000-5a3e991a0000\"",
  "_attachments": "attachments/",
  "_ts": 1514051866
}
```

```
// Identify document by...

// ...self link
string id1 = "dbs/v84pAA==/colls/v84pA07nPQA=/docs/v84pA07nPQABAAAAAABg==/";

// ...ID-based routing
string id2 = "dbs/Families/colls/Families/docs/ab7aafc4-6a6e-f19e-bf5f-f7985b2f156d";

// ...URI
Uri id3 = UriFactory.CreateDocumentUri("Families", "Families", "ab7aafc4-6a6e-f19e-bf5f-f7985b2f156d");
```



Resource Properties, Self-Links, and URIs

```
{
  "familyName": "Smith",
  "address": {
    "addressLine": "123 Main Street",
    "city": "Chicago",
    "state": "IL",
    "zipCode": "60601"
  },
  "parents": [
    "Peter",
    "Alice"
  ],
  "kids": [
    "Adam",
    "Jacqueline",
    "Joshua"
  ],
  "id": "ab7aafc4-6a6e-f19e-bf5f-f7985b2f156d",
  "_rid": "v84pA07nPQABAAAAAABg==",
  "_self": "dbs/v84pAA==/colls/v84pA07nPQA=/docs/v84pA07nPQABAAAAAABg==/",
  "_etag": "\"00004700-0000-0000-0000-5a3e991a0000\"",
  "_attachments": "attachments/",
  "_ts": 1514051866
}
```

```
// Identify document by...

// ...self link
string id1 = "dbs/v84pAA==/colls/v84pA07nPQA=/docs/v84pA07nPQABAAAAAABg==/";

// ...ID-based routing
string id2 = "dbs/Families/colls/Families/docs/ab7aafc4-6a6e-f19e-bf5f-f7985b2f156d";

// ...URI
Uri id3 = UriFactory.CreateDocumentUri("Families", "Families", "ab7aafc4-6a6e-f19e-bf5f-f7985b2f156d");
```

▶ id3 {dbs/Families/colls/Families/docs/ab7aafc4-6a6e-f19e-bf5f-f7985b2f156d}



Resource Properties, Self-Links, and URIs

```
{  
  "familyName": "Smith",  
  "address": {  
    "addressLine": "123 Main Street",  
    "city": "Chicago",  
    "state": "IL",  
    "zipCode": "60601"  
  },  
  "parents": [  
    "Peter",  
    "Alice"  
  ],  
  "kids": [  
    "Adam",  
    "Jacqueline",  
    "Joshua"  
  ],  
  "id": "SmithFamily"  
}
```



Resource Properties, Self-Links, and URIs

```
{
  "familyName": "Smith",
  "address": {
    "addressLine": "123 Main Street",
    "city": "Chicago",
    "state": "IL",
    "zipCode": "60601"
  },
  "parents": [
    "Peter",
    "Alice"
  ],
  "kids": [
    "Adam",
    "Jacqueline",
    "Joshua"
  ],
  "id": "SmithFamily",
  "_rid": "v84pA07nPQACAAAAAABg==",
  "_self": "dbs/v84pAA==/colls/v84pA07nPQA=/docs/v84pA07nPQACAAAAAABg==/",
  "_etag": "\"00004800-0000-0000-0000-5a3e9a210000\"",
  "_attachments": "attachments/",
  "_ts": 1514052129
}
```

```
// Identify document by...

// ...self link
string id1 = "dbs/v84pAA==/colls/v84pA07nPQA=/docs/v84pA07nPQACAAAAAABg==/";

// ...ID-based routing
string id2 = "dbs/Families/colls/Families/docs/SmithFamily";

// ...URI
Uri id3 = UriFactory.CreateDocumentUri("Families", "Families", "SmithFamily");
```



Resource Properties, Self-Links, and URIs

```
{
  "familyName": "Smith",
  "address": {
    "addressLine": "123 Main Street",
    "city": "Chicago",
    "state": "IL",
    "zipCode": "60601"
  },
  "parents": [
    "Peter",
    "Alice"
  ],
  "kids": [
    "Adam",
    "Jacqueline",
    "Joshua"
  ],
  "id": "SmithFamily",
  "_rid": "v84pA07nPQACAAAAAABg==",
  "_self": "dbs/v84pAA==/colls/v84pA07nPQA=/docs/v84pA07nPQACAAAAAABg==/",
  "_etag": "\"00004800-0000-0000-0000-5a3e9a210000\"",
  "_attachments": "attachments/",
  "_ts": 1514052129
}
```

// Identify document by...

// ...self link

string id1 = "dbs/v84pAA==/colls/v84pA07nPQA=/docs/v84pA07nPQACAAAAAABg==/";

// ...ID-based routing

string id2 = "dbs/Families/colls/Families/docs/SmithFamily";

// ...URI

Uri id3 = UriFactory.CreateDocumentUri("Families", "Families", "SmithFamily");

▶ id3 {dbs/Families/colls/Families/docs/SmithFamily} ⇐



Resource Properties, Self-Links, and URIs

```
{
  "familyName": "Smith",
  "address": {
    "addressLine": "123 Main Street",
    "city": "Chicago",
    "state": "IL",
    "zipCode": "60601"
  },
  "parents": [
    "Peter",
    "Alice"
  ],
  "kids": [
    "Adam",
    "Jacqueline",
    "Joshua"
  ],
  "id": "SmithFamily",
  "_rid": "v84pA07nPQACAAAAAABg==",
  "_self": "dbs/v84pAA==/colls/v84pA07nPQA=/docs/v84pA07nPQACAAAAAABg==/",
  "_etag": "\"00004800-0000-0000-0000-5a3e9a210000\"",
  "_attachments": "attachments/",
  "_ts": 1514052129
}
```

```
// Identify collection by...

// ...self link
string id1 = "dbs/v84pAA==/colls/v84pA07nPQA=/";

// ...ID-based routing
string id2 = "dbs/Families/colls/Families";

// ...URI
Uri id3 = UriFactory.CreateDocumentCollectionUri("Families", "Families");
```



Resource Properties, Self-Links, and URIs

```
{
  "familyName": "Smith",
  "address": {
    "addressLine": "123 Main Street",
    "city": "Chicago",
    "state": "IL",
    "zipCode": "60601"
  },
  "parents": [
    "Peter",
    "Alice"
  ],
  "kids": [
    "Adam",
    "Jacqueline",
    "Joshua"
  ],
  "id": "SmithFamily",
  "_rid": "v84pA07nPQACAAAAAABg==",
  "_self": "dbs/v84pAA==/colls/v84pA07nPQA=/docs/v84pA07nPQACAAAAAABg==/",
  "_etag": "\"00004800-0000-0000-0000-5a3e9a210000\"",
  "_attachments": "attachments/",
  "_ts": 1514052129
}
```

```
// Identify collection by...

// ...self link
string id1 = "dbs/v84pAA==/colls/v84pA07nPQA=/";

// ...ID-based routing
string id2 = "dbs/Families/colls/Families";

// ...URI
Uri id3 = UriFactory.CreateDocumentCollectionUri("Families", "Families");
▶ id3 {dbs/Families/colls/Families}
```



Data Migration Tool

Open source project

Microsoft Download Center

<http://www.microsoft.com/en-us/download/details.aspx?id=46436>

Documentation and How-To

<https://azure.microsoft.com/en-us/documentation/articles/documentdb-import-data>

Source code on GitHub

<https://github.com/azure/azure-documentdb-datamigrationtool>

Import from

- SQL Server
- JSON
- CSV
- MongoDB
- Azure Table Storage
- and more...



Demo



Importing data from SQL Server



Summary



Relational vs. document

Cosmos DB resource model

Data migration tool

