SIDI TOGETHER FUTURE HAPPENS

“SAP FIORI” INTRODUZIONE AI SERVIZI E AL PROTOCOLLO ODATA

Febbraio 2022

SAP Gold Partner

SAP® Certified Partner Center of Expertise

QUALITY SYSTEM CERTIFICATION DNV·GL ISO 9001:2015

INTRODUZIONE SERVIZI E PROTOCOLLO ODATA

- 01. SOAP, REST e Odata
- 02. Struttura Servizio OData
- 03. OData Query
- 04. Esercitazione Pratica

SOAP, REST E ODATA

In questa sessione vengono analizzate le varee strutture e metodi dei servizi Soap e Rest.
In particolare sui servizi Rest e Odata

Con introduzione ai formati XML e Json e la struttura di una URL.

01

XML, JSON

In informatica, sia JSON che XML sono formati leggibili dall'uomo basati su testo.

XML è un linguaggio di markup in grado di gestire in modo completo qualsiasi tipo di informazione.
JSON è un formato di interscambio dati che risulta essere di più agile utilizzo in determinati contesti.

XML
Extensive Markup Language

```
<widget>
    <debug>on</debug>
    <window title="Sample Konfabulator Widget">
        <name>main_window</name>
        <width>500</width>
        <height>500</height>
    </window>
    <image src="Images/Sun.png" name="sun1">
        <hOffset>250</hOffset>
        <vOffset>250</vOffset>
        <alignment>center</alignment>
    </image>
    <text data="Click Here" size="36" style="bold">
        <name>text1</name>
        <hOffset>250</hOffset>
        <vOffset>100</vOffset>
        <alignment>center</alignment>
        <onMouseUp>
            sun1.opacity = (sun1.opacity / 100) * 90;
        </onMouseUp>
    </text>
</widget>
```

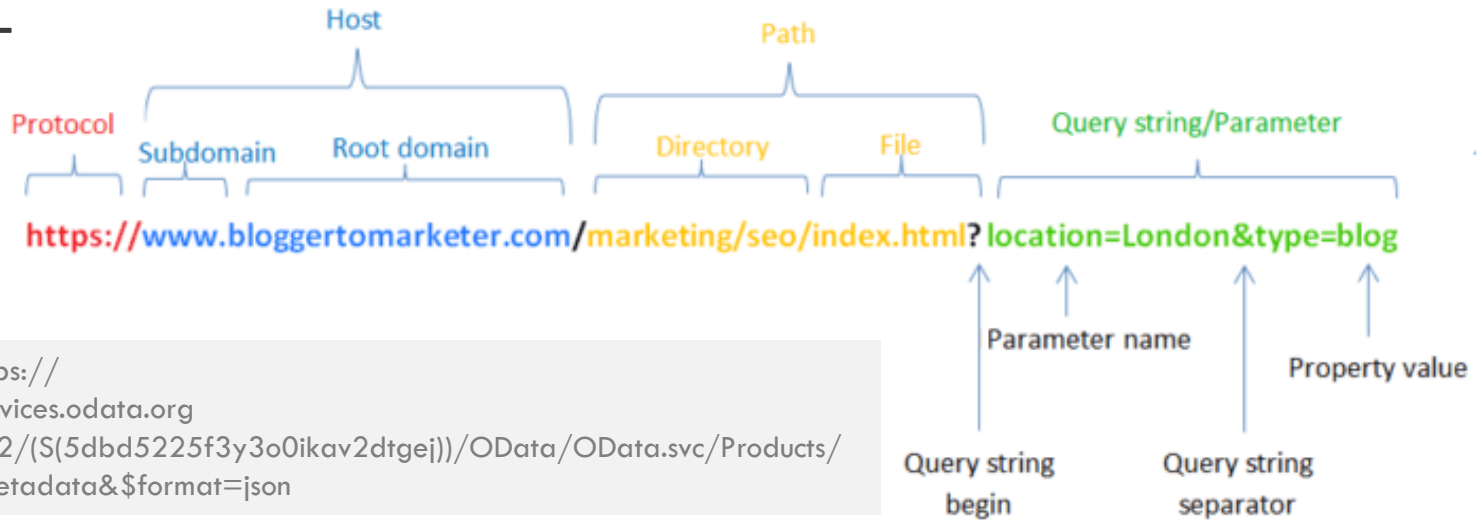
JSON
JavaScript Object Notation

```
{"widget": {
    "debug": "on",
    "window": {
        "title": "Sample Konfabulator Widget",
        "name": "main_window",
        "width": 500,
        "height": 500
    },
    "image": {
        "src": "Images/Sun.png",
        "name": "sun1",
        "hOffset": 250,
        "vOffset": 250,
        "alignment": "center"
    },
    "text": {
        "data": "Click Here",
        "size": 36,
        "style": "bold",
        "name": "text1",
        "hOffset": 250,
        "vOffset": 100,
        "alignment": "center",
        "onMouseUp": "sun1.opacity = (sun1.opacity / 100) * 90;"
    }
}}
```

URL

URL sta per **U**niform **R**esource **L**ocator è l'indirizzo globale dei documenti e altre risorse sul World Wide Web. Il suo scopo principale è identificare la posizione di un documento su Internet e specificare il meccanismo per accedervi tramite un browser web.

Codifica URL



Protocollo	https://
Host	services.odata.org
Path	/V2/(S(5dbd5225f3y3o0ikav2dtgej))/OData/OData.svc/Products/
Query String	?metadata&$format=json

SOAP, REST E ODATA

SOAP

Simple Object Access Protocol.
Protocollo che stabilisce una serie di regole grazie alle quali Server consumer e Service provider possono dialogare tra loro.
La comunicazione avviene attraverso il formato Xml. Il messaggio Xml deve essere formato da:

- ***Envelope*** (ovvero contenitore per Header e Body);
- ***Header*** (che è opzionale e contiene informazioni come routing e autenticazione)
- ***Body*** (dati che il Service consumer (o Client) ha chiesto al Server)

REST

REpresentational State Transfer.
Non prevede l'utilizzo di regole per permettere il dialogo tra Server consumer e Service provider.
La comunicazione può venire in vari formati (XML, JSON).
Per reperire una risorsa basta un link.

REST

Per poter definire un servizio RESTful deve soddisfare le seguenti condizioni:

- **Uniform Interface:** Risorse accessibili tramite URI
- **Stateless**: ogni richiesta tra client e server è indipendente e nessun contenuto viene salvato sul server.
- **Cacheable:** le risposte inviate dal server indicano se possono essere conservate o meno lato client in modo da evitare che vengano usati dati errati.
- **Layers:** il client può non comunicare direttamente con il server, ossia possono esistere livelli multipli tra client e server.
- **Code on Demand:** consentire di aggiornare il codice lato client senza dover modificare nulla lato server. (Proprietà facoltativa)

Ogni comando REST è una richiesta di una dei seguenti tipi:

- **GET:** prendi una singola entry o un collezione di entries
- **POST**: crea una nuova entry
- **PUT:** aggiorna un entry esistente
- **DELETE**: cancella un entry
- **PATCH**: aggiorna singole proprietà di un entry esistente

ODATA

Open Data Protocol (OData) è un protocollo basato su REST e di conseguenza rispetta i suoi principi e implementa i suoi metodi.

POST, GET, PUT e DELETE sono supportati dalle interfacce Create, Retrieve, Update, Delete (CRUD) sul server. Un servizio OData supporta tutte e 4 le interfacce ma non è obbligatorio averle definite tutte.

Uniform Resource Identifiers
Attraverso il protocollo OData, l'insieme delle entities o delle singole entity sono accessibili tramite URI

Uniform Interface
Anche il protocollo OData insieme all'identificatore tramite URI utilizza i "verbi" HTTP per implementare le azione da intraprendere su un entity (CRUD)

Stateless Communication
In OData lo stato della sessione non è mantenuto dal server ma può essere salvato lato client

Rappresentazione multipla di una risorsa
Un OData può essere spedito utilizzando diverse rappresentazioni (XML o JSON)

CRUD	HTTP
Create	POST
Read	GET
Update	PUT
Delete	DELETE

STRUTTURA ODATA

In questa sessione verra spiegata la struttura di un servizio OData

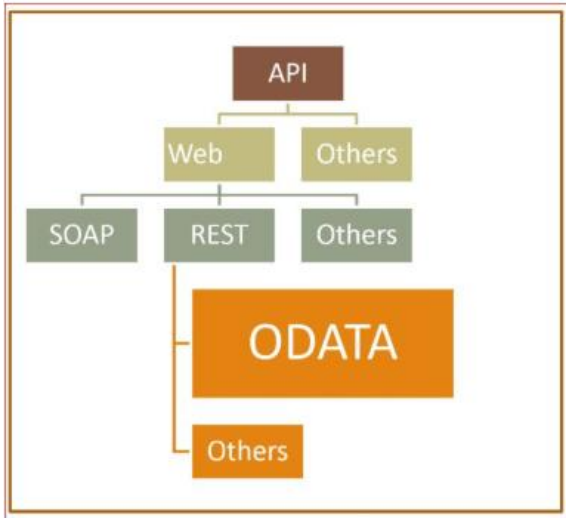
02

STRUTTURA ODATA

L' **Open Data Protocol (OData)** è un protocollo pubblicato da Microsoft che definisce best practices per creare e consumare principalmente RESTful APIs. Il protocollo si basa su **HTTP** e su protocolli precedenti **Open Database Connectivity (ODBC)** e **Java Database Connectivity (JDBC)**.

REST, nella sua forma completa, Representation State Transfer, è un tipo di architettura software che ha acquisito sempre più considerazione nel web, tanto da essere considerata un vero e proprio standard per la creazione di Web API. I servizi REST sfruttano il protocollo HTTP per lo scambio di dati. L'insieme dei metodi HTTP, e cioè GET, POST, PUT e DELETE, semplifica di molto il mapping tra le azioni CRUD (Create, Read, Update, Delete) e le chiamate HTTP che vengono effettuate.

SAP fornisce servizi OData attraverso il SAP Gateway o l' ODATA Provisioning.



STRUTTURA ODATA

Service document
è la rappresentazione del servizio dove vengono dichiarate: le risorse accessibili tramite il servizio, i loro URI e le operazioni.
Service metadata document
Espone il modello, i tipi, le azioni, le relazioni e i dettagli delle informazioni semantiche (elemento dati).

I precedenti due tipo di documento sono composti dai seguenti elementi:

- **Entity:** è una risorsa che può anche essere vuota. Può contenere diverse proprietà e deve avere almeno 1 campo chiave. Le entities possono essere individuate tramite il valore della chiave oppure da una collezione di entities attaverso l'esecuzione di una query
- **Entity type:** descrive una collezione. In molti casi il nome della collezione e dell'entity type sono lo stesso oppure sono relazionati da una convenzione. Gli entity type hanno una struttura definita dalle loro proprietà e posseggono anche una chiave che formata da un sottoinsieme delle proprietà.
- **Entity set**: rappresenta un insieme di entries. La cardinalità è 0:N. Molti entity set possono essere basati sullo stesso entity type. Se un enity set è individuabile dal valore delle sue chiavi allor il metodo GET_ENTITY deve essere implementato
- **Property**: è un tipo di elemento che rappresenta un tipo dato primitivo, una struttura dati o il link ad un'altra risorsa. In pratica corrisponde alla colonna di una tabella.
- **Navigation property**: è una specifica proprietà contenente un link che rappresenta un istanza di associazione. Il link punta ad altre tabelle o ad altre entries in base alla cardinalità definita nell'associazione. In pratica la navigazione consente di navigare da un entity ad altre
- **Association**: definisce una relazione tra entity types. Sono accettate anche relazioni ricorsive. In ogni associazione devono essere definiti: le entity types coinvolte e la cardinalità.

PROTOCOLLO ODATA

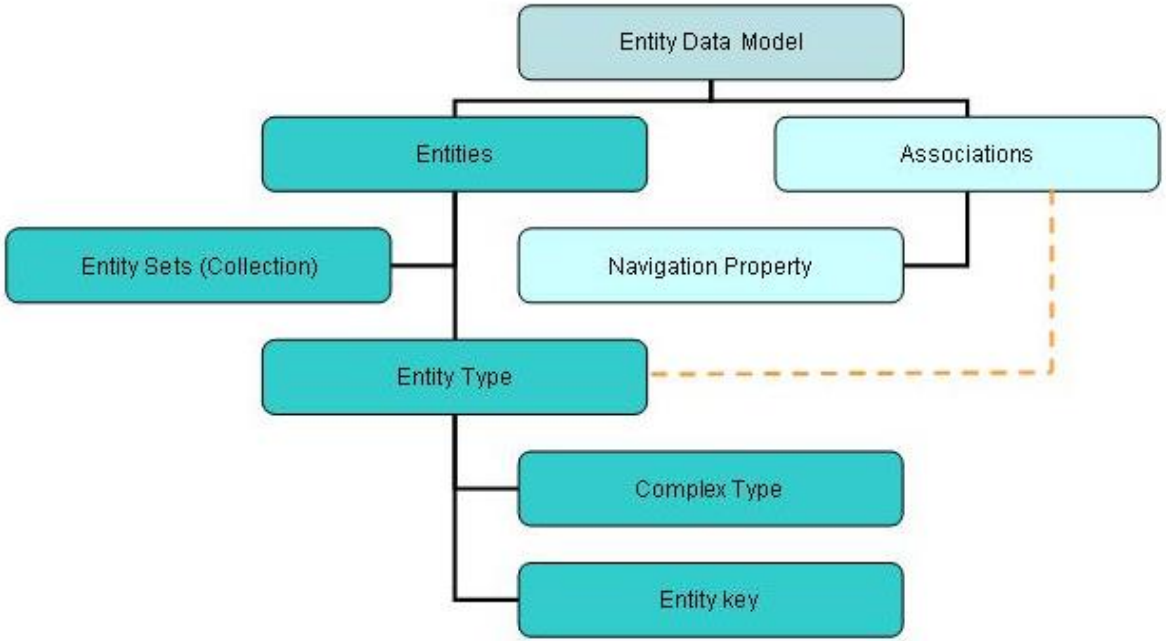
```
<EntityContainer Name="DemoService" m:IsDefaultEntityContainer="true">
  <EntitySet Name="Products" EntityType="ODataDemo.Product"/>
  <EntitySet Name="Categories" EntityType="ODataDemo.Category"/>
  <EntitySet Name="Suppliers" EntityType="ODataDemo.Supplier"/>
```

Entity Set

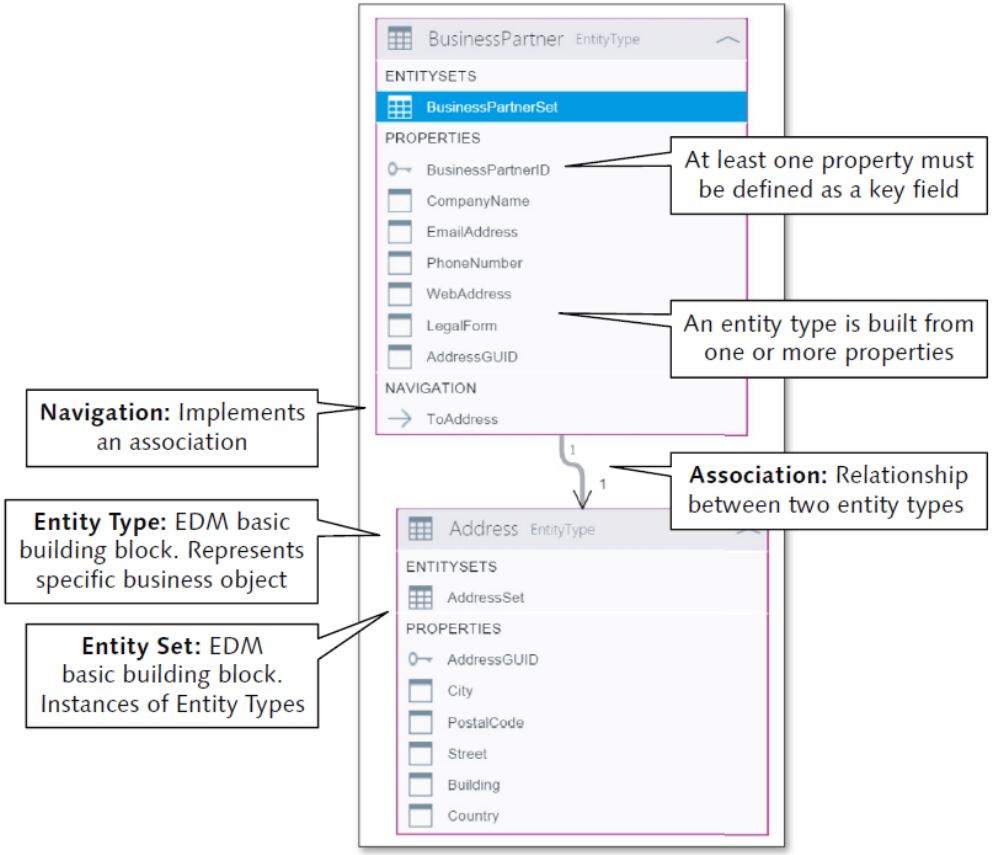
Entity Type

```
<EntityType Name="Product">
  <Key>
    <PropertyRef Name="ID"/>
  </Key>
  <Property Name="ID" Type="Edm.Int32" Nullable="false"/>
  <Property Name="Name" Type="Edm.String" Nullable="true" m:FC_TargetPath="SyndicationTitle" m:FC_ContentKind="text" m:FC_KeepInContent="false"/>
  <Property Name="Description" Type="Edm.String" Nullable="true" m:FC_TargetPath="SyndicationSummary" m:FC_ContentKind="text" m:FC_KeepInContent="false"/>
  <Property Name="ReleaseDate" Type="Edm.DateTime" Nullable="false"/>
  <Property Name="DiscontinuedDate" Type="Edm.DateTime" Nullable="true"/>
  <Property Name="Rating" Type="Edm.Int32" Nullable="false"/>
  <Property Name="Price" Type="Edm.Decimal" Nullable="false"/>
  <NavigationProperty Name="Category" Relationship="ODataDemo.Product_Category_Category_Products" FromRole="Product_Category" ToRole="Category_Products"/>
  <NavigationProperty Name="Supplier" Relationship="ODataDemo.Product_Supplier_Supplier_Products" FromRole="Product_Supplier" ToRole="Supplier_Products"/>
</EntityType>
```

STRUTTURA ODATA



STRUTTURA ODATA



ODATA QUERY

In questa sessione verra spiegato il linguaggio query per poter ricercare, filtrare e paginare dati sul backend.

03

OPZIONI ODATA QUERY

OData mette a disposizione un linguaggio query per poter ricercare, filtrare e paginare dati sul backend.
Le opzioni di query sono essenziali per ridurre o influenzare il risultato fornito dal server SAP Gateway.

https://services.odata.org/V2/(S(5dbd5225f3y3o0ikav2dtgej))/OData/OData.svc

Operation	Query Option
Filtering and projecting	`$filter` and `$select`
Sorting	`$orderby`
Client-side paging	`$top`, `$skip` and `$inlinecount`
Counting	`$count`
Inlining	`$expand`
Formatting	`$format`

OPZIONI ODATA QUERY

$format: utilizzata per definire il formato della response. L'OData v2.0 supporta solo 2 tipi di formato XML e JSON. Il formato di default è XML.

$count: è utilizzato per ottenere il numero di entries all'interno della collection e non firnisce altro.

$top: tipicamente usata insieme al parametro $skip. Specifica il numero di entry che si vuole ottenere nella response. Se, per esempio, il parametro è impostato $top=3, allora verranno ritornate solo le prime 3 entry della query.

$skip: definisce il numero di record da saltare dall'inizio della response. Se, ad esempio, il parametro è impostato $skip=2, la response non conterrà il primo e il secondo record ma partirà da terzo.

$orderby: consente di ordinare il risultato di una query sulla base di alcune proprietà. Per ogni proprietà è possibile indicare il tipo di ordinamento (ascendente o discendente). L'ordinamento ascendente è quello di default. Per l'ordinamento discendente basta aggiungere "desc" dopo la lista dei campi su cui ordinare.

OPZIONI ODATA QUERY

$filter: permette di selezionare solo i dati che soddisfano le condizioni impostare nel filtro. Il filtro va implementato nel relativo codice backend.

$select: è un opzione di query che consente di specificare quali proprietà della entity vengano ricevute nella response. Di default la response contiene tutte le proprietà dell'entity.

ODATA QUERY

Esercitazione Pratica

04

ESERCITAZIONE PRATICA

1. Quante EntitySets contiene il servizio?
2. Farsi tornare la lista degli EntitySets in formato JSON
3. Quali sono le proprietà dell' EntitySets "Products" ? Qual è la chiave?
4. Filtrare i "Products" che hanno un Rating = 4 e di questi selezionare solo la proprietà "Price". Qual è il prezzo?
5. Ordinare i "Products" in ordine ascendente per "Name". Qual è il secondo?
6. Estrarre solo la 3 e 4 Entry dei "Products"

VERIFICA ESERCITAZIONE PRATICA

Quante EntitySets contiene il servizio?

https://services.odata.org/V2/OData/OData.svc/

Farsi tornare la lista degli EntitySets in formato JSON

https://services.odata.org/V2/OData/OData.svc/?$format=json

Quali sono le proprietà dell' EntitySets " Products"? Qual è la chiave?

https://services.odata.org/V2/(S(5dbd5225f3y3o0ikav2dtgej))/OData/OData.svc/Products/$count

Filtrare i "Products" che hanno un Rating = 4 e di questi selezionare solo la proprietà "Price". Qual è il prezzo?

https://services.odata.org/(S(5dbd5225f3y3o0ikav2dtgej))/V2/OData/OData.svc/Products

?$format=json&$filter=Rating%20eq%204&$select=Price

Ordinare i "Products" in ordine ascendente e discendente per "Name". Qual è il secondo?

https://services.odata.org/(S(5dbd5225f3y3o0ikav2dtgej))/V2/OData/OData.svc/Products?$format=json&$orderby=Name

Estrarre solo la 3 e 4 Entry dei "Products"

https://services.odata.org/(S(5dbd5225f3y3o0ikav2dtgej))/V2/OData/OData.svc/Products/?$format=json&$top=2&$skip=2