

An OData Class for Tcl

Gerald W. Lester

Gerald.Lester@KnG-Consulting.net

What is OData?

- **Open Data Protocol (OData)** is a data access protocol designed to provide standard **CRUD** (create, read, uupdate and ddelete) access to a data source via a website, although it is not limited to SQL databases.
 - Accessed via HTTP/HTTPS
 - OData is built on the AtomPub protocol and XML (or JSON) where the Atom structure is the envelope that contains the data returned from each OData request.

What is OData?

- Provides MetaData (introspection) into full data model
 - Attributes
 - Relationships
 - Actions (aka methods)

What is OData?

- HTTP method of request specifies action
 - GET: Get a collection of entities (as a feed document) or a single entity (as an entry document).
 - POST: Create a new entity from a document.
 - PUT: Update an existing entity with a complete document.
 - PATCH: Update an existing entity with a partial document.
 - DELETE: Remove an entity.

Why would we want it?

- Recommended for the Open Government Data Initiative
- Web sites/services are providing it as an API.
 - Microsoft Azure and eBay provides as API to their data.
 - Access to SAP Business Suite and SAP Business Warehouse.
 - IBM WebSphere eXtreme Scale REST data service.
 - Client implementations for several languages.

Why using TclOO?

- Modern feel
- Wanted to play with something to learn TclOO

Implementation Overview

- Implements Client Side V4.0 of Odata.
 - Read only.
 - Supports full introspection/discovery
 - Fully dynamic (of course)
 - Submitted to Odata community

Implementation Overview

- Odata
 - Root (abstract) class for other Odata Classes.
- OdataService
 - Defines a service for a URL.
- OdataItem
 - Defines an Entity (like a class or table) inside of a service.

Implementation Overview

OdataService

- *new url*
- info **subcommand** *?args?*
Where subcommand is one of:
 - ServiceAddress
 - collections
 - collection
 - types
 - type
 - unsaved
 - debug
- insert *CollectionName ?property value?...*
- *CollectionName ?searchCriteria?*

Implementation Overview

OdataService - info

- ServiceAddress
 - Returns the service address
- collections
 - Returns what collections (tables) are available
- types
 - Returns what types are available
- unsaved
 - Returns a list of OdataItems that have not been saved
- debug
 - Returns a dictionary representing the service

Implementation Overview

OdataService - info

- collection *collectionName*
- type *typeName* **typeCommand**

Where **typeCommand** is one of:

- properties
- relationships
- links
- keys

Implementation Overview

OdataService – info type

- properties
 - Returns a list of properties (attributes).
- relationships
 - Returns a list of relationships to other types.
- links
 - Returns a list of links to things outside of the service.
- keys
 - Returns the properties that are the key.

Implementation Overview

OdataService – CollectionName

- Without an argument
 - Returns a list OdataItems contained in the collection
- With an argument
 - Argument is a search/selection criterial
 - Returns a list of OdataItems contained in the collection that match the search/selection criterial

Implementation Overview

OdataService – insert

insert *CollectionName ?property value?...*

- Creates and returns a new OdataItem of the type that the named collection is composed of and associates it with that collection.
 - Optional property value pairs may be passed. These will set the specified properties to the specified values.

Implementation Overview

OdataService – CollectionName

- Without an argument
 - Returns a list of OdataItems contained in the collection
- With an argument
 - Argument is a search/selection criterial
 - Returns a list of OdataItems contained in the collection that match the search/selection criterial

Implementation Overview

OdataItem

- info
- get
- set
- commit (not implement yet)
- rollback
- delete (not fully implement yet)
- *Action/Link*

Implementation Overview

OdataItem

- info **subcommand** *?args?*
 - Where subcommand is one of:
 - mode
 - state
 - type
 - properties
 - changed
 - links
 - actions
 - debug

Implementation Overview

OdataItem – info

- mode
 - Returns the mode the object is in, either “readonly” or “readwrite”
- state
 - Returns the state of the object:
 - new – the object is a new object and has not been written back to the store
 - clean – the object was read from the store and not modified
 - dirty – the object has been modified since it was read from the store or last written to the store

Implementation Overview

OdataItem – info

- type
 - Returns the type of the object
- properties
 - Returns the properties (attributes) of the object
- changed
 - Returns a list of changed values in the form of:
 - 1)propertyName
 - 2)oldValue
 - 3)newValue

Implementation Overview

OdataItem – info

- links
 - Returns a list of links.
- actions
 - Returns available actions.

Implementation Overview

OdataItem - get

get *propertyName*

- Returns the current value of the property

Implementation Overview

OdataItem - set

set propertyName value

- Sets the specified property to the value.
- Only visible if the item's mode is readwrite.
- Marks the item as “dirty”.
- Makes the commit and rollback commands visible.

Implementation Overview

OdataItem - commit

commit

- Writes all changed values to the store.
- Only visible if the item's state is “new” or “dirty”.
 - For new will fail if all the key properties are not set.
- Makes the commit and rollback commands not visible.

Implementation Overview

OdataItem - rollback

rollback

- Undoes all changed values to the object.
- Only visible if the item's state is “dirty”.
- Makes the commit and rollback commands not visible.

Implementation Overview

OdataItem - delete

delete

- Deletes the object.
- Only visible if the item's mode is readwrite or state is “new”.
- If the state is not “new”, item is deleted from store.
- Deletes the item.

Implementation Overview

OdataItem - Action

Action ?args?

- Invokes the URL for the *Action* on the object with the specified arguments.
- Definition of what happens is service dependent.

Implementation Overview

OdataItem - Link

Link

- Invokes the URL for the *Link* on the object.
- Definition of what happens is service dependent.

How to use it

```
source OdataClass.tcl
```

```
set url \
```

```
{http://services.odata.org/V4/OData/(S(qe0mzsxp2vzxc23zdd2rxg0))/OData.svc/}
```

```
set obj [ODataService new $url]  
puts stdout [$obj info collections]
```

⇒ Products ProductDetails Categories Suppliers Persons
PersonDetails Advertisements

```
puts stdout [$obj Categories]
```

⇒ ::oo::Obj15 ::oo::Obj16 ::oo::Obj17

```
set p2 [$obj Products 2]
```

```
set s2 [$p2 Supplier]
```

```
puts stdout "'[$p2 get Name]' is supplied by '[$s2 get Name]'"
```

⇒ 'Vint soda' is supplied by 'Exotic Liquids'!

Implementation Tricks

- Use `[oo::objdefine $obj (un)export]` to expose hide methods depending on state.
- Use `[oo::objdefine $obj forward]` plus private “helper” methods in `odata::OData` class to create “dynamic” methods for collections, actions and links.

The Future

- Looking for long term maintainer
 - Just done as an experiment/learning
 - Odata community seems excited about it

Overview

- What is Odata?
- Why would we want it?
- Why using TclOO?
- Implementation Overview
- How to use it
- Implementation Tricks
- The Future

Questions?

Contact Information

- Email

Gerald.Lester@KnG-Consulting.net

- Fossil Url:

<http://chiselapp.com/user/gwlester/repository/ODataTclOO>