

# Creating and Consuming an OData V4 Service

---



**KEVIN DOCKX**

ARCHITECT

@KevinDockx <https://www.kevindockx.com>



# Coming Up



Entity Data Model and Conceptual Model

Selecting entities, entity sets and properties

Routing

Manipulating entities

Working with relations and navigation properties



# Entity Data Model and Conceptual Model



## Entity Data Model (EDM)

- Abstract data model
- Describes data exposed by an OData service
- Stored form of data is irrelevant

# Entity Data Model and Conceptual Model



## Conceptual Model

- Representation as entities and relationships
- OData-CSDL implements concepts of EDM



# A Few Important Definitions

## **Entity**

instance of Entity  
Type

## **Entity Type**

named structured  
type with key

## **Complex Type**

named structured  
type without key

## **Type Definition**

named primitive  
type

## **Navigation Property**

relationship from one  
entity to another

## **Entity Set**

named collection of  
entities



# Demo



## Defining an Entity Data Model

- Add a new project for our OData services
- Add packages
- Define the EDM
- Inspect the service document and metadata document



# Demo

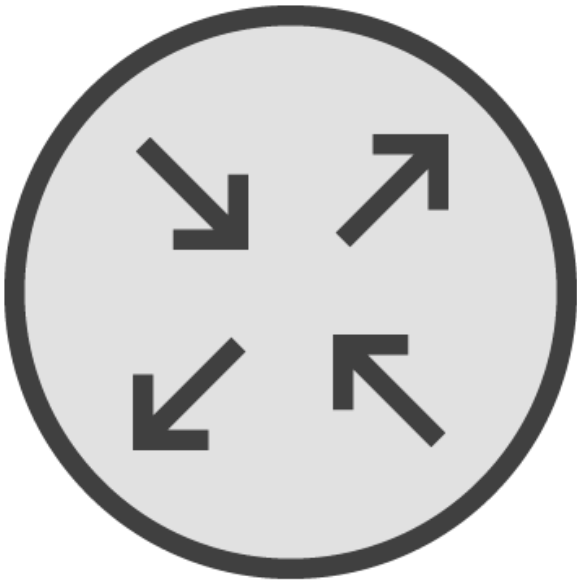


## Selecting an EntitySet

- Create an ODataController
- Create an action to return an entity set



# Routing



**Matches request uri to controller action**

**Routing templates in routing table**

- MapODataServiceRoute



# Routing

HTTP Method	URI	Action name	Sample URI	Sample action
-------------	-----	-------------	------------	---------------



# Demo



## Working with Attribute-based Routing - [ODataRoute]



# Demo

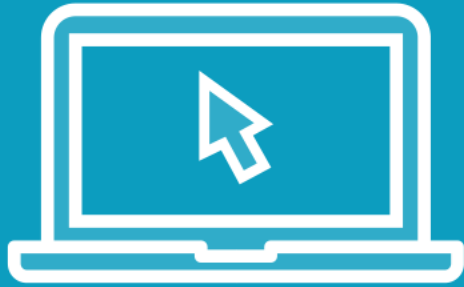


## Controlling the Amount of Control Information

- Amount of metadata in response



# Demo



## Selecting an Individual Entity

- GET root/EntitySet(entityKey)



# Demo

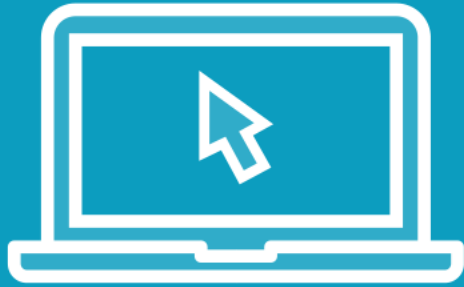


## Selecting Properties and Raw Property Values

- GET root/EntitySet(entityKey)/PropertyName
- GET root/EntitySet(entityKey)/PropertyName/\$value



# Demo



## Creating an Entity

- POST root/EntitySet



# Demo

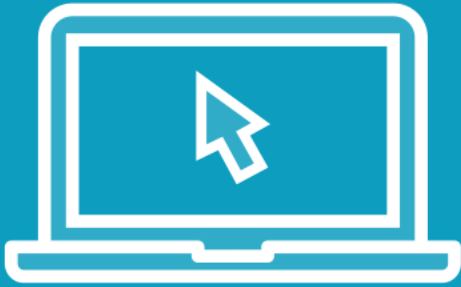


## Updating an Entity

- PUT: only for full updates
- PUT root/EntitySet(entityKey)



# Demo



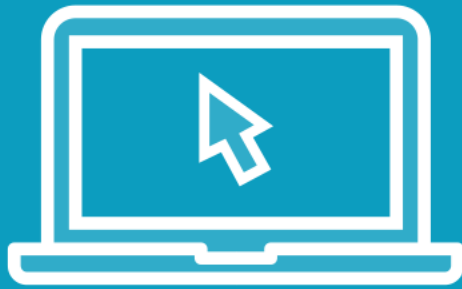
## Partially Updating an Entity

- PATCH: preferred way of updating
- PATCH root/EntitySet(entityKey)





# Demo



## Deleting an Entity

- DELETE root/EntitySet(entityKey)



# Demo



## Working with Relations and Navigation Properties

- Manipulation through association links



# Summary



Entity Data Model and Conceptual Model

GET, POST, PUT/PATCH, DELETE

We're implementing a standard

Routing

Relationships

