

# More Examples

## Get all payment methods

### Example

```
GET http://localhost:1260/api/v1/Payments/Methods
```

Note that this request returns providers rather than entities and therefore we must use the `/api/v1/` service here.

## Partially update address with ID 1

### Example

```
PATCH http://localhost:1260/odata/v1/Addresses(1)?SmNetFulfillCountry=US&SmNetFulfillStateProvince=NY
{"City":"New York","Address1":"21 West 52nd Street","ZipPostalCode":"10021","FirstName":"John","LastName":"Doe"}
```

The example uses the `SmNetFulfillCountry` and `SmNetFulfillStateProvince` options to update the country (USA) and province (New York). This avoids extra querying of country and province records and passing its IDs in the request body.

## Get ID of store with name *my nice store*

### Example

```
GET http://localhost:1260/odata/v1/Stores?$top=1&$filter=Name eq 'my nice store'&$select=Id
```

Note the select option, which tells OData just to return the `Id` property.

## Getting localized property values

### Example

```
GET http://localhost:1260/odata/v1/LocalizedProperties?$top=120&$filter=LocaleKeyGroup eq 'Product' and
EntityId eq 224 and Language/LanguageCulture eq 'en-US'
```

This is where the OData filter enters the equation. That request filters all English property values for a product with the ID 224. `LocaleKeyGroup` is typically the entity name (Product, Category, Manufacturer, ProductBundleItem, DeliveryTime etc.). The result looks like this:

### Example

```
{ "odata.metadata": "http://localhost:1260/odata/v1/$metadata#LocalizedProperties", "value": [
  {
    "EntityId": 224, "LanguageId": 1, "LocaleKeyGroup": "Product", "LocaleKey": "Name", "LocaleValue": "My
name for the product", "Id": 111
  }, {
    "EntityId": 224, "LanguageId": 1, "LocaleKeyGroup": "Product", "LocaleKey": "FullDescription", "
LocaleValue": "<p>Hello world! My product description goes here...</p>", "Id": 113
  }
] }
```

`LocaleKey` is the context property name (the name and full description of the product entity). `LocaleValue` offers the localized value.