Progress®

Beautiful Native Mobile Apps with NativeScript and OpenEdge

Edsel Garcia
OpenEdge Development Team
October 2019

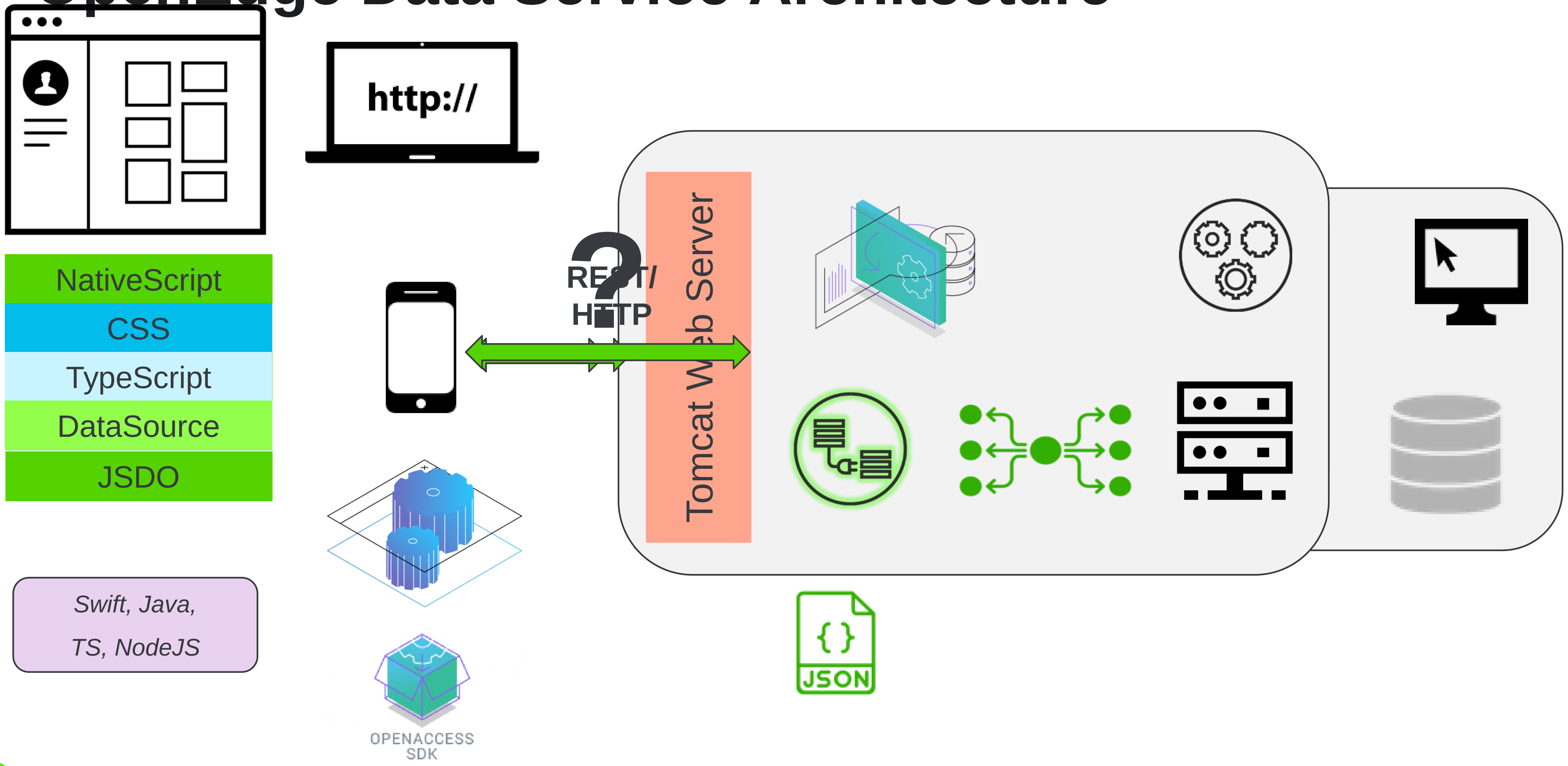
Agenda

- OpenEdge Data Service Architecture
- JSDO
- DataSource
- Using the Blank Template
- Demo
- Next Steps

Progress®

OpenEdge Data Service Architecture

OpenEdge Data Service Architecture



JSDO

JSDO

“A JSDO is an object designed to simplify access to relational data in a mobile app. It does this by providing JavaScript methods to execute the Data Object operations supported by a single Data Object resource and by supporting an internal data store (JSDO memory) to cache the data that is defined by and returned from the Data Object resource to the mobile app.”

Progress Data Objects Guide and Reference

JSDO

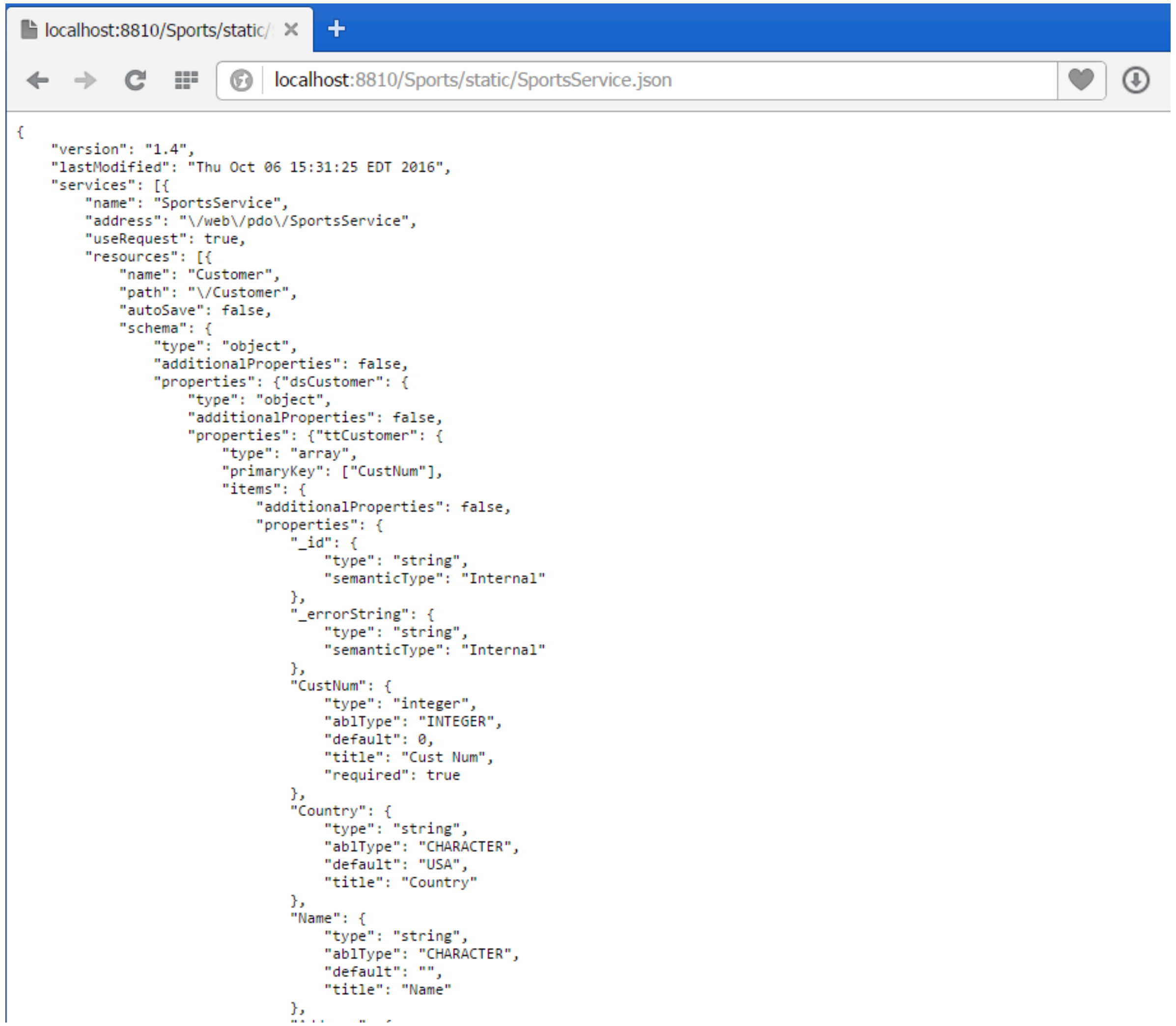
- ES6 Promises
- npm packages:
    - @progress/jsdo-core
    - @progress/jsdo-angular
    - @progress/jsdo-nativescript
    - @progress/jsdo-node

JSDO

- CRUD + Invoke
  - add() *(CREATE)*
  - **fill()** *(READ)*
  - assign() *(UPDATE)*
  - remove() *(DELETE)*
  - **Invoke()** *(INVOKE)*
- Properties
  - autoSort
  - caseSensitive
  - name
  - record
  - useRelationships
- Methods
  - addRecords()
  - find()
  - foreach()
  - getData()
  - getId()
  - getSchema()
  - **saveChanges()**
  - sort()
  - subscribe()
  - unsubscribe()

Data Service Catalog

- Location: <project-name>/PASOEContent/static
- URI: Service URI: http://<host>:<port>/<web-app>/static/<service-name>.json


localhost:8810/Sports/static/
localhost:8810/Sports/static/SportsService.json

```
{
    "version": "1.4",
    "lastModified": "Thu Oct 06 15:31:25 EDT 2016",
    "services": [{
        "name": "SportsService",
        "address": "\/web\/pdo\/SportsService",
        "useRequest": true,
        "resources": [{
            "name": "Customer",
            "path": "\/Customer",
            "autoSave": false,
            "schema": {
                "type": "object",
                "additionalProperties": false,
                "properties": {"dsCustomer": {
                    "type": "object",
                    "additionalProperties": false,
                    "properties": {"ttCustomer": {
                        "type": "array",
                        "primaryKey": ["CustNum"],
                        "items": {
                            "additionalProperties": false,
                            "properties": {
                                "_id": {
                                    "type": "string",
                                    "semanticType": "Internal"
                                },
                                "_errorString": {
                                    "type": "string",
                                    "semanticType": "Internal"
                                },
                                "CustNum": {
                                    "type": "integer",
                                    "ablType": "INTEGER",
                                    "default": 0,
                                    "title": "Cust Num",
                                    "required": true
                                },
                                "Country": {
                                    "type": "string",
                                    "ablType": "CHARACTER",
                                    "default": "USA",
                                    "title": "Country"
                                },
                                "Name": {
                                    "type": "string",
                                    "ablType": "CHARACTER",
                                    "default": "",
                                    "title": "Name"
                                },
```

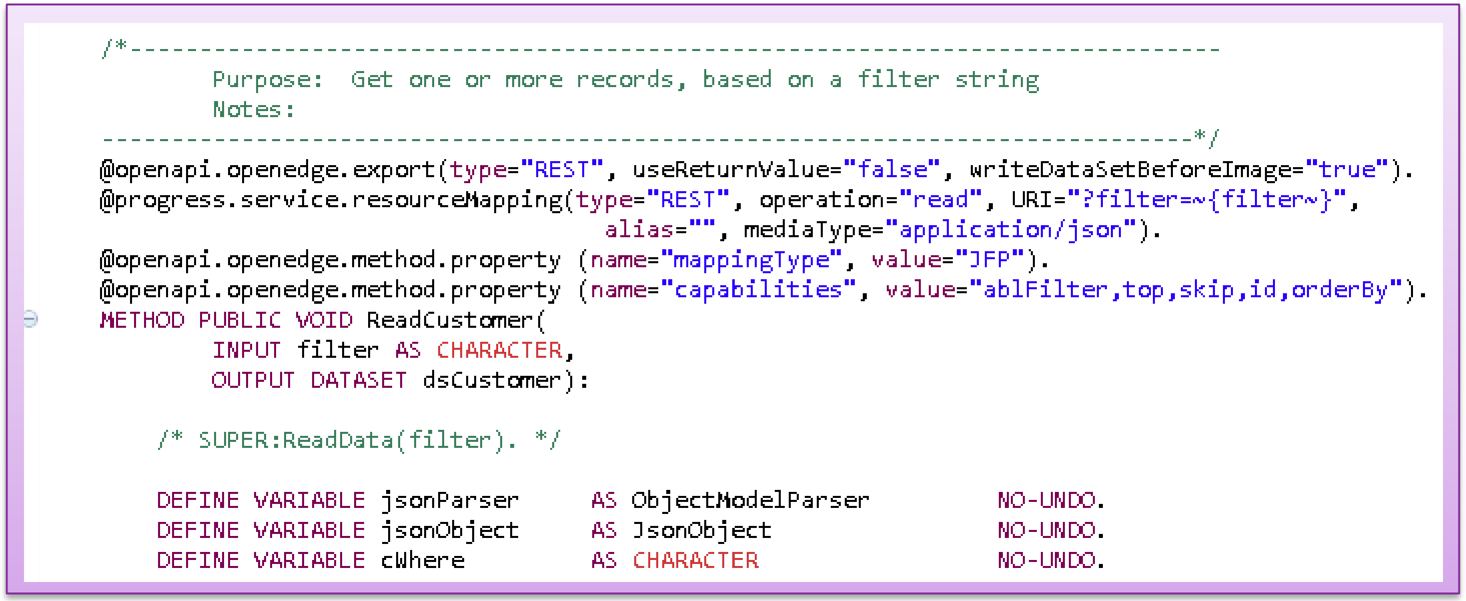
JSON Filter Pattern - Protocol

- Annotations:
  - mappingType (JFP)
  - capabilities

Search using a filter and sorting for page 3 with page size of 20 records:

{“ablFilter”: “CustNum < 50”, “top”: “20”, “skip”: 40, “orderBy”: “CustNum”}

- Capabilities:
  - ablFilter
  - top
  - skip
  - Id
  - orderBy

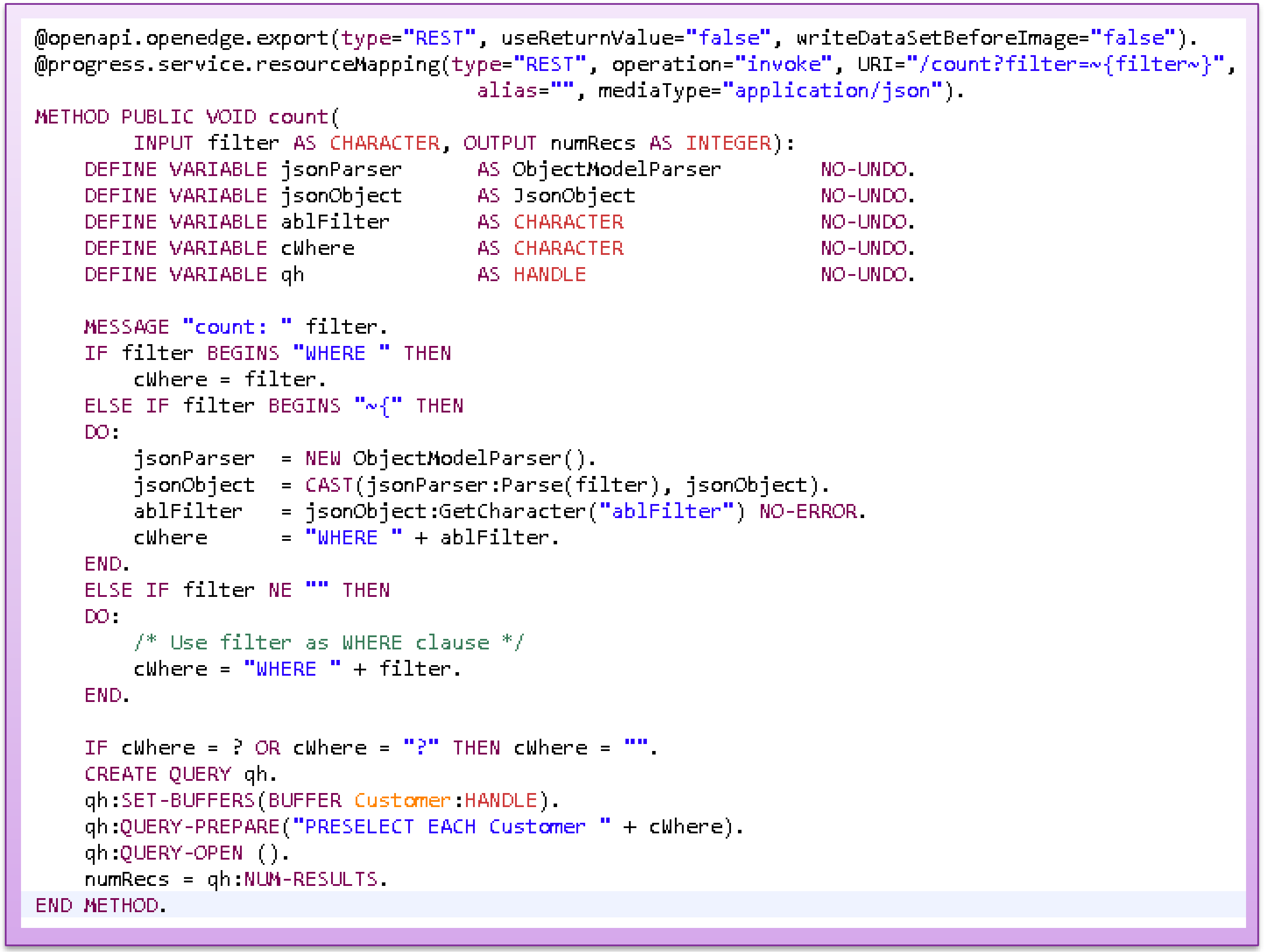


```
/*------------------------------------------------------------------------------
        Purpose:  Get one or more records, based on a filter string
        Notes:
------------------------------------------------------------------------------*/
@openapi.openedge.export(type="REST", useReturnValue="false", writeDataSetBeforeImage="true").
@progress.service.resourceMapping(type="REST", operation="read", URI="?filter=~{filter~}",
                                    alias="", mediaType="application/json").
@openapi.openedge.method.property (name="mappingType", value="JFP").
@openapi.openedge.method.property (name="capabilities", value="ablFilter,top,skip,id,orderBy").
METHOD PUBLIC VOID ReadCustomer(
        INPUT filter AS CHARACTER,
        OUTPUT DATASET dsCustomer):

    /* SUPER:ReadData(filter). */

    DEFINE VARIABLE jsonParser     AS ObjectModelParser         NO-UNDO.
    DEFINE VARIABLE jsonObject     AS JsonObject                NO-UNDO.
    DEFINE VARIABLE cWhere         AS CHARACTER                 NO-UNDO.
```

- Note: You can also define your own mappingType

Count Support

- countFnName property
- Operation="count"



```
@openapi.openedge.export(type="REST", useReturnValue="false", writeDataSetBeforeImage="false").
@progress.service.resourceMapping(type="REST", operation="invoke", URI="/count?filter=~{filter~}",
                                    alias="", mediaType="application/json").
METHOD PUBLIC VOID count(
        INPUT filter AS CHARACTER, OUTPUT numRecs AS INTEGER):
    DEFINE VARIABLE jsonParser      AS ObjectModelParser        NO-UNDO.
    DEFINE VARIABLE jsonObject      AS JsonObject               NO-UNDO.
    DEFINE VARIABLE ablFilter       AS CHARACTER                NO-UNDO.
    DEFINE VARIABLE cWhere          AS CHARACTER                NO-UNDO.
    DEFINE VARIABLE qh              AS HANDLE                   NO-UNDO.

    MESSAGE "count: " filter.
    IF filter BEGINS "WHERE " THEN
        cWhere = filter.
    ELSE IF filter BEGINS "~{" THEN
    DO:
        jsonParser  = NEW ObjectModelParser().
        jsonObject  = CAST(jsonParser:Parse(filter), jsonObject).
        ablFilter   = jsonObject:GetCharacter("ablFilter") NO-ERROR.
        cWhere      = "WHERE " + ablFilter.
    END.
    ELSE IF filter NE "" THEN
    DO:
        /* Use filter as WHERE clause */
        cWhere = "WHERE " + filter.
    END.

    IF cWhere = ? OR cWhere = "?" THEN cWhere = "".
    CREATE QUERY qh.
    qh:SET-BUFFERS(BUFFER Customer:HANDLE).
    qh:QUERY-PREPARE("PRESELECT EACH Customer " + cWhere).
    qh:QUERY-OPEN ().
    numRecs = qh:NUM-RESULTS.
END METHOD.
```

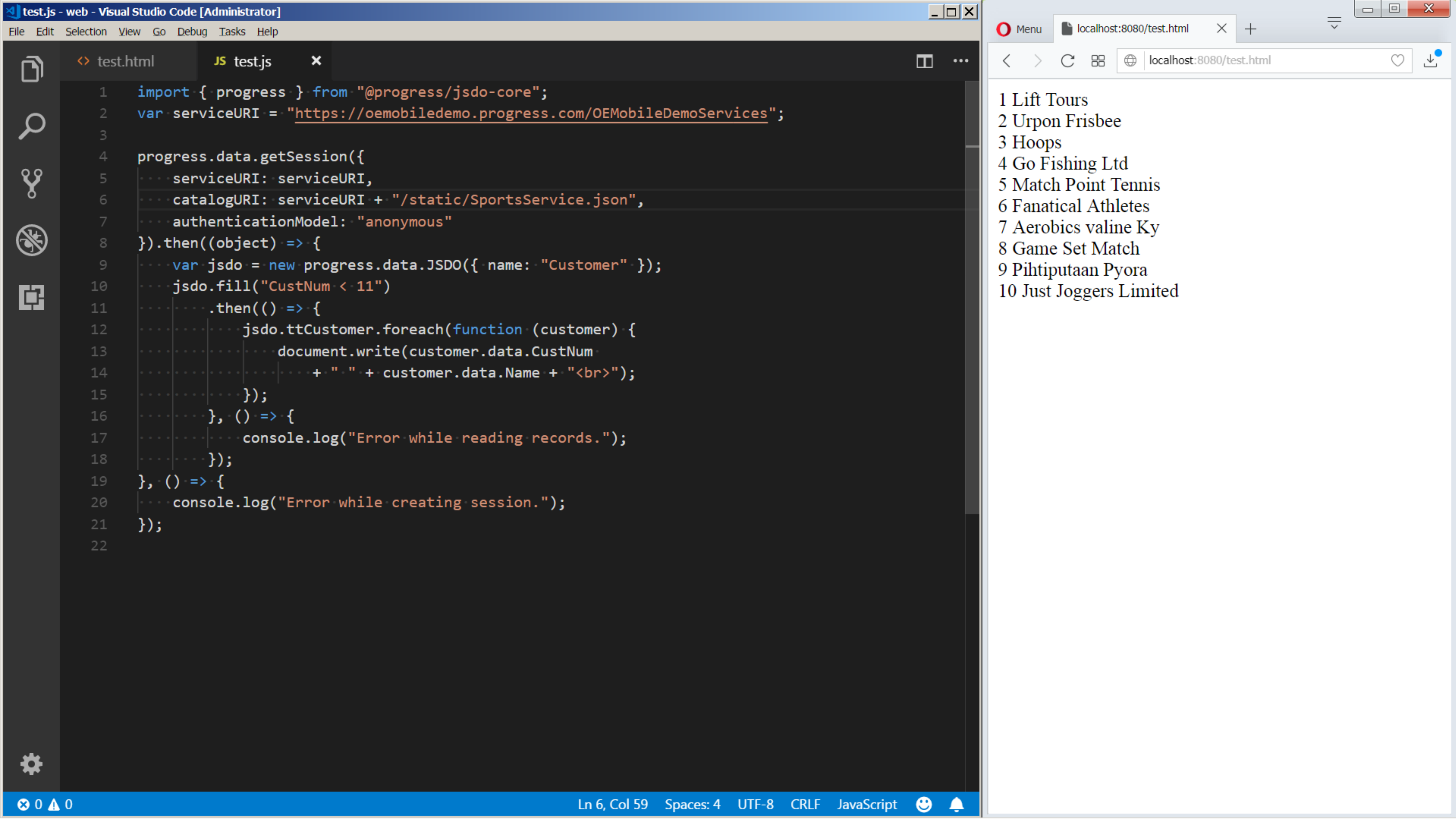
DataSource

DataSource

- CRUD
  - create() *(CREATE)*
  - **read()** *(READ)*
  - update() *(UPDATE)*
  - remove() *(DELETE)*
- Properties
  - jsdo

- Methods
  - findById()
  - getData()
  - hasCUDSupport()
  - hasSubmitSupport()
  - **saveChanges()**

Getting Started

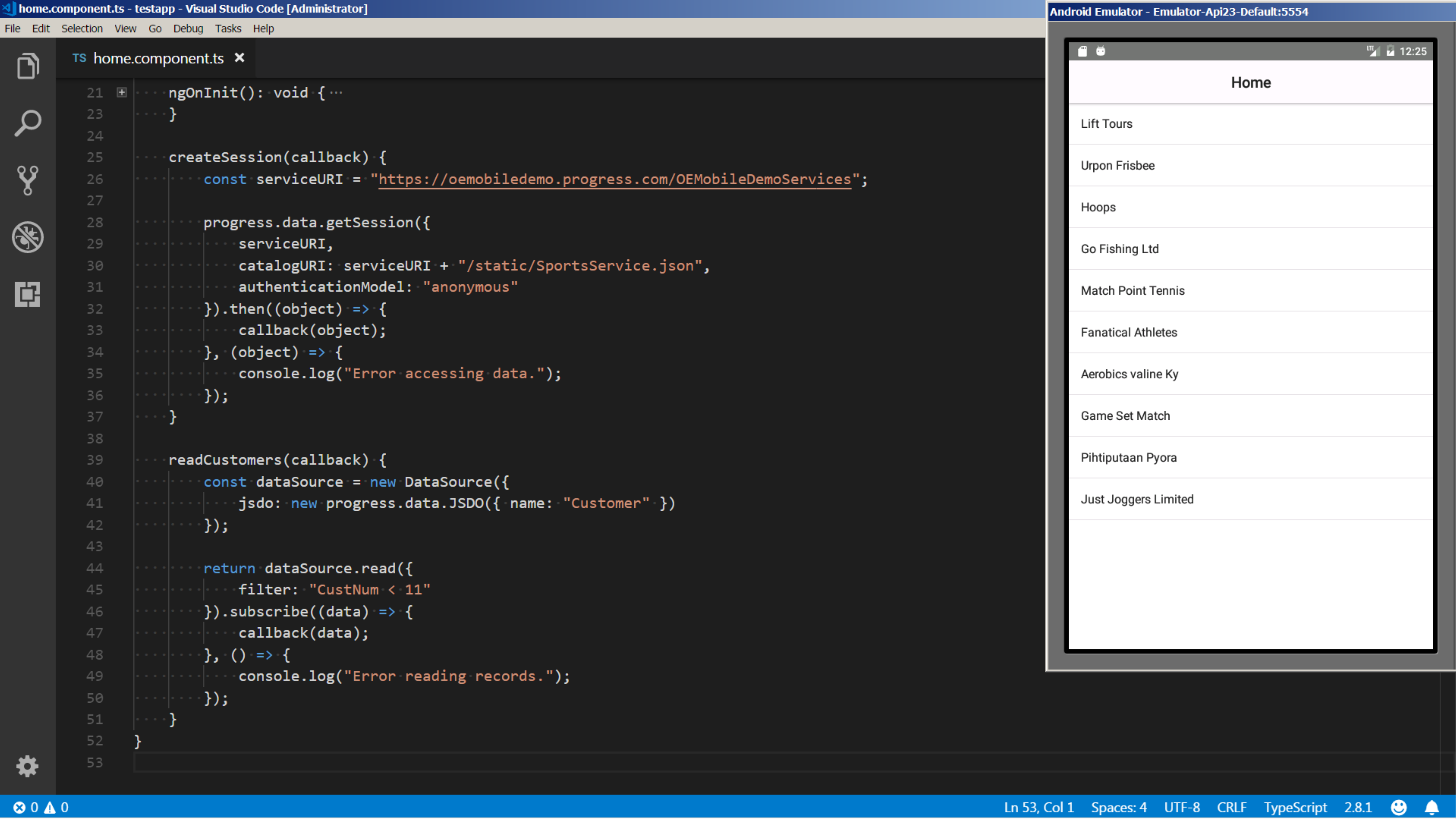

test.js - web - Visual Studio Code [Administrator]
File Edit Selection View Go Debug Tasks Help
test.html
test.js
Menu
localhost:8080/test.html
Ln 6, Col 59 Spaces: 4 UTF-8 CRLF JavaScript

```javascript
import { progress } from "@progress/jsdo-core";
var serviceURI = "https://oemobiledemo.progress.com/OEMobileDemoServices";

progress.data.getSession({
    serviceURI: serviceURI,
    catalogURI: serviceURI + "/static/SportsService.json",
    authenticationModel: "anonymous"
}).then((object) => {
    var jsdo = new progress.data.JSDO({ name: "Customer" });
    jsdo.fill("CustNum < 11")
        .then(() => {
            jsdo.ttCustomer.foreach(function (customer) {
                document.write(customer.data.CustNum
                    + " " + customer.data.Name + "<br>");
            });
        }, () => {
            console.log("Error while reading records.");
        });
}, () => {
    console.log("Error while creating session.");
});

```

```
1 Lift Tours
2 Urpon Frisbee
3 Hoops
4 Go Fishing Ltd
5 Match Point Tennis
6 Fanatical Athletes
7 Aerobics valine Ky
8 Game Set Match
9 Pihtiputaan Pyora
10 Just Joggers Limited
```

Using the Blank Template

- Create project using:
  - tns create <App Name> --template tns-template-blank-ng
  - npm install @progress/jsdo-nativescript
- Add Code
- Run using:
  - tns preview
    - https://docs.nativescript.org/tooling/docs-cli/project/testing/preview


home.component.ts - testapp - Visual Studio Code [Administrator]
File Edit Selection View Go Debug Tasks Help
home.component.ts

```
    ngOnInit(): void {…
    }

    createSession(callback) {
        const serviceURI = "https://oemobiledemo.progress.com/OEMobileDemoServices";

        progress.data.getSession({
            serviceURI,
            catalogURI: serviceURI + "/static/SportsService.json",
            authenticationModel: "anonymous"
        }).then((object) => {
            callback(object);
        }, (object) => {
            console.log("Error accessing data.");
        });
    }

    readCustomers(callback) {
        const dataSource = new DataSource({
            jsdo: new progress.data.JSDO({ name: "Customer" })
        });

        return dataSource.read({
            filter: "CustNum < 11"
        }).subscribe((data) => {
            callback(data);
        }, () => {
            console.log("Error reading records.");
        });
    }
}
```

Ln 53, Col 1 Spaces: 4 UTF-8 CRLF TypeScript 2.8.1
Android Emulator - Emulator-Api23-Default:5554
12:25
Home

Lift Tours
Urpon Frisbee
Hoops
Go Fishing Ltd
Match Point Tennis
Fanatical Athletes
Aerobics valine Ky
Game Set Match
Pihtiputaan Pyora
Just Joggers Limited

DEMO

Progress®

Next Steps

Progress®

Beautiful Apps

NativeScript UI

Enhance your app

- Chart
- ListView
- SideDrawer
- Calendar
- Gauges
- AutoComplete

ListView

Different layout modes. Pull-down to refresh. Continuous scrolling.

Calendar

Week, month and year views. Single, multiple and range date Selection.

Chart

Beautiful and flexible charts: area, line, pie, scatter and more.

https://www.nativescript.org/ui-for-nativescript

Progress

The JSDO is Open Source

JSDO on GitHub

- JSDO repository:
  - https://github.com/progress/JSDO
- Open Source:
  - Develop branch
  - Pull Requests (sign CLA)
- Issues (bugs, enhancements, tech preview conversations):
  - https://github.com/progress/JSDO/issues
- Developer Wiki:
  - https://github.com/progress/JSDO/wiki

Resources

- JSDO on GitHub:
  - https://github.com/progress/JSDO/issues
  - https://github.com/progress/JSDO/wiki
  - https://github.com/progress/JSDO/wiki/Using-the-JSDO-and-DataSource-with-an-existing-NativeScript-app
- Documentation:
  - https://documentation.progress.com/output/pdo/index.html
- Progress Community:
  - https://community.progress.com/community_groups/mobile/m/documents/2677

Summary

- Easy access to OpenEdge data via the JSDO
- Share Data Service support for web and mobile apps
- New DataSource component for NativeScript
- Use NativeScript templates to get started
- Use NativeScript UI components to enhance your mobile apps
- The JSDO is open source

Thank You.

Progress®