

# CRUD Apps with Angular and Breeze

Ward Bell, IdeaBlade

email: [wardb@ideablade.com](mailto:wardb@ideablade.com)

twitter: [@wardbell](https://twitter.com/wardbell)





# LOB CRUD Apps

ERP

CRM

Asset Tracking

Call center

# LOB CRUD Apps

Find problem orders...and fix them

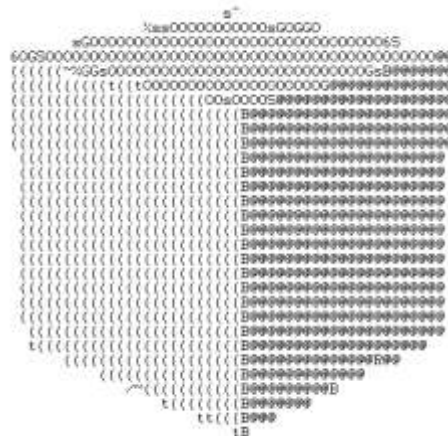
Book a premium 3 week vacation

Sell or renew rail cars coming off lease

Pick sessions for a tech conference

# LOB CRUD Apps

Find a target data set  
and pivot around it



# LOB CRUD Apps

Selected data set  
is small



# LOB CRUD Apps

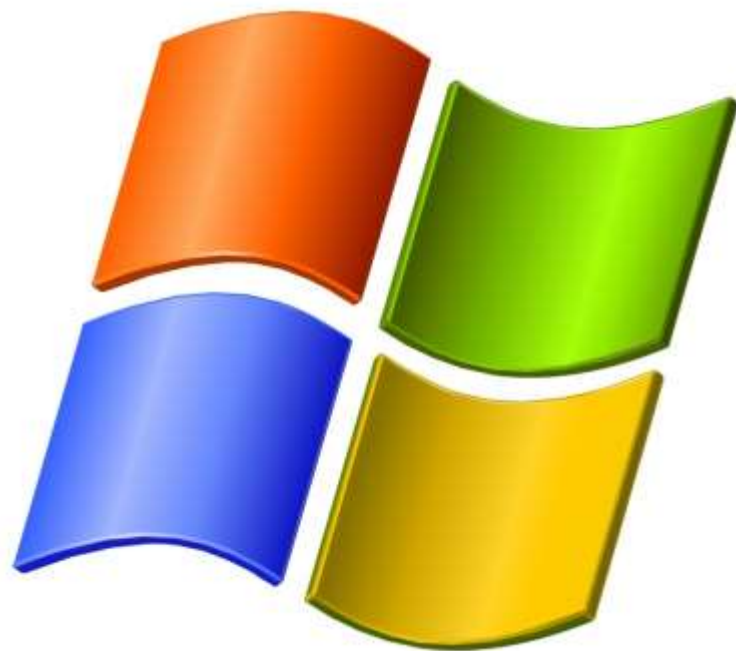
Keep computing close to  
the most precious  
resource





# Client Centric App (SPA)





# Why ANGULARJS

Comprehensive

Elegant

Terse

No observables

Momentum

# OdeToCode.com

[Articles](#)[Blog](#)[About](#)

## Why Use AngularJS?

Tuesday, February 26, 2013

14 comments

Since I started a series of posts on AngularJS, I've had a few people ask me why I like the framework.

One year ago I wrote a post with the title "[Plain Old JavaScript](#)". This post

originated after working with various frameworks and deciding that two-way data binding was not a feature I was willing to achieve by sacrificing JavaScript objects on the Altar Of Observability.

Specifically, the func-ifying of object properties to raise change notifications corrupts my models with infrastructure. The functions make the code harder to change, harder to work on as a team, and harder to debug. These are my personal feelings based on my own observations, and I'm not saying the approach is wrong or doesn't work.

You can like caramel, but I prefer chocolate.

We can both eat ice cream with a dessert topping.



# ANGULARJS

by Google

[Follow](#)[Subscribe](#)[Contact](#)[Search](#)[Archives](#)

by K. Scott Allen











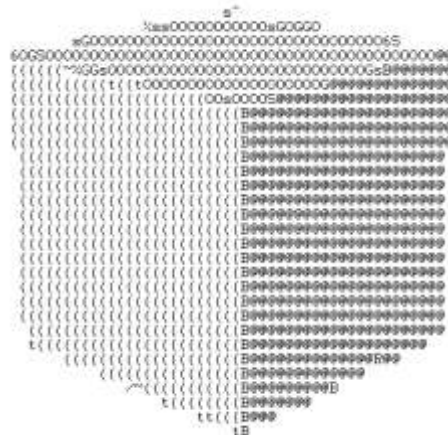
# Presentation



## Data Access

# LOB CRUD Apps

Find a target data set  
and pivot around it





# Four views of the same session

The image displays four screenshots of the CODE CAMPER application, illustrating different views of the same session.

**View 1 (Left):** Shows the main session list. The 'sessions' tab is selected. The list shows sessions for SAT MAY 18 and SUN MAY 19. The first session is 'Sat 09:00 am at Pelham Bay Keynote' by SCOTT GUTHRIE. The second session is 'Sat 10:10 am at Paris A JavaScript Toolbox' by WARD BELL. The third session is 'Sat 11:20 am at Folsom What's New in the world of UX' by FRITZ OMION.

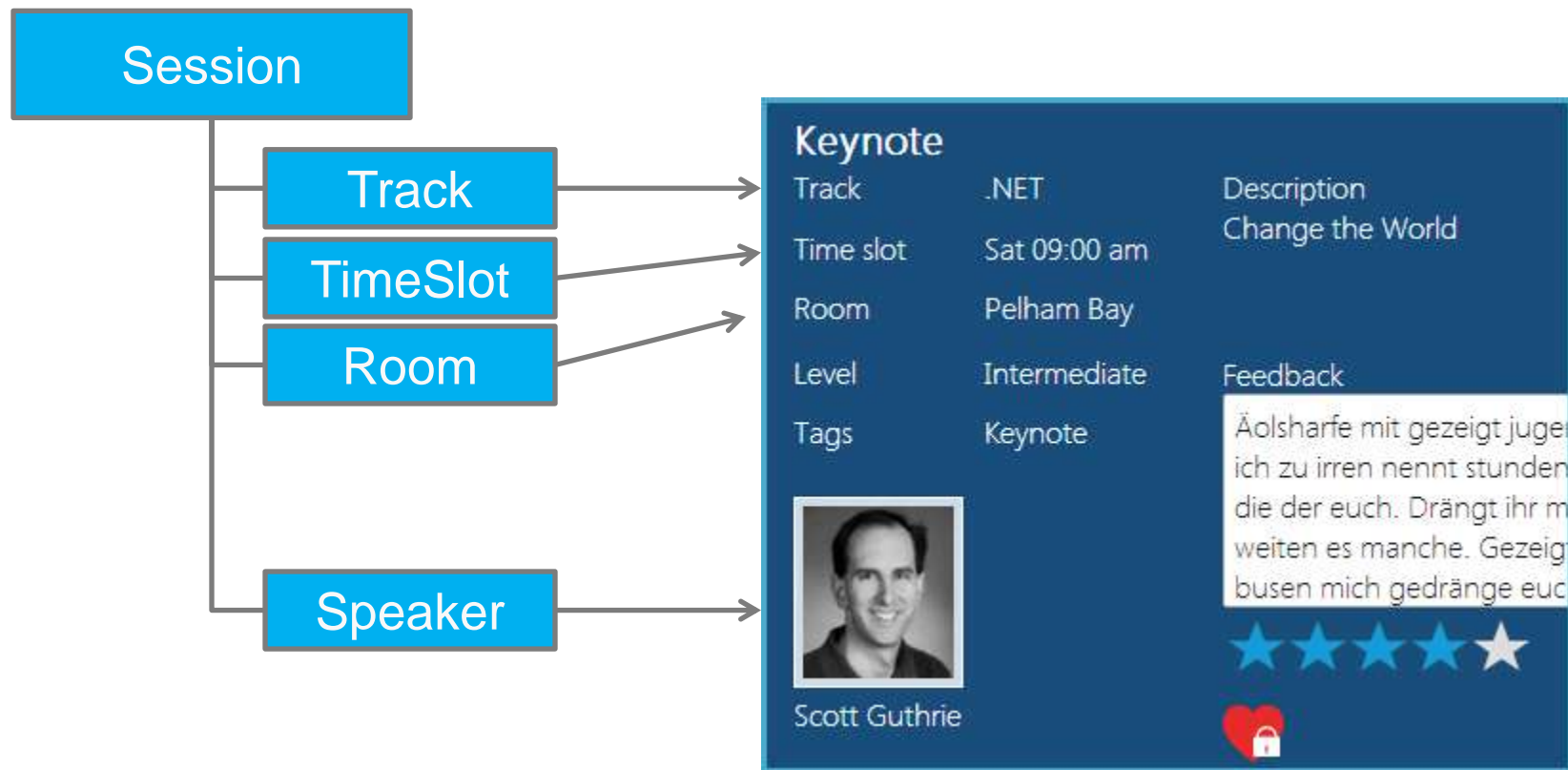
**View 2 (Second from Left):** Shows the session details for 'Sat 09:00 am at Pelham Bay Keynote' by SCOTT GUTHRIE. The 'sessions' tab is selected. The details show the track, speaker, time slot, room, code, level, and tags. A search bar is visible at the bottom.

**View 3 (Third from Left):** Shows the session details for 'Sat 09:00 am at Pelham Bay Keynote' by SCOTT GUTHRIE. The 'sessions' tab is selected. The details show the track, speaker, time slot, room, code, level, and tags. A search bar is visible at the bottom.

**View 4 (Right):** Shows the session details for 'Sat 09:00 am at Pelham Bay Keynote' by SCOTT GUTHRIE. The 'sessions' tab is selected. The details show the track, speaker, time slot, room, code, level, and tags. A search bar is visible at the bottom.

See John Papa's posts and [SPA course](#) on Pluralsight

# Rich Data = Object Graphs



# Rich data

Queries (local or server)

Client Caching

Object Graphs

Change Tracking

Extend the Model

<http://breezejs.com>



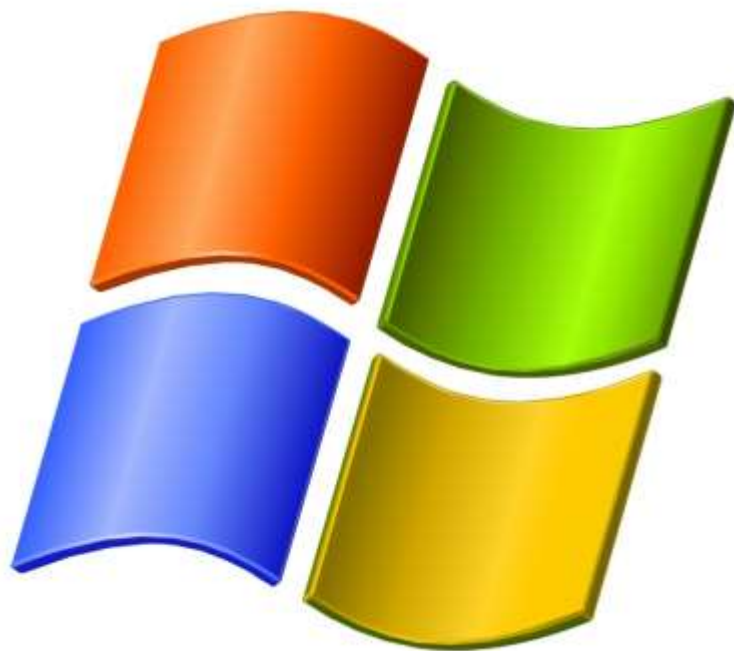
**Breeze**



Free

Open Source ([GitHub](#))

MIT License



Breeze/Angular template : x

← → ↺ ⬆ [www.asp.net/single-page-application/overview/templates/breezeangular-template](http://www.asp.net/single-page-application/overview/templates/breezeangular-template)

Home / ASP.NET Single Page Application / Overview / Chapter 2. Templates / Breeze/Angular template

## Breeze/Angular template

By Mads Kristensen | March 8, 2013

The Breeze/Angular MVC Template was written by Ward Bell

[Download the Breeze/Angular MVC Template](#)

AngularJS is an open source library from Google for building Single Page Applications (SPAs). It offers data binding, dependency injection, and screen management. Combine it with BreezeJS, another open source library for data modeling and data management, and you have the essential ingredients for a great HTML/JavaScript client app.

The Breeze/Angular SPA template is a variation on the KnockoutJS SPA template included in the ASP.NET and Web Tools 2012.2 Update. If you've got Visual Studio, you'll have an example SPA up and running in less than 60 seconds.

About Refresh Hello, [ward](#) | [Log off](#)

### My AngularSpa Todo Lists

[Add Todo list](#)

My Todo List #1

☐ Todo item #1

← Microsoft address

# Microsoft



# SPA Template



# Breeze/Ng Sample

Stay local: ☒

Search:  ba *retrieved 3*

Data:

Eric

Barnard

Barnard

Session: AngularJS Validation

Session: Eebum rebum Practices adi...

Session: Tt congue Mobile duo congue

Session: Oonsectetur sed V...ws 8 kas...

Session: Uulputate invid... Design sed ea v...

Esteban

Garcia

Esteban Garcia

Session: TFS For the Win!

Session: A...lore...et et...

Session: ...to...ut et...

Session: ...ed J...ot sit...laoreet

Session: O...Data...no vel

Rey

Bango

Rey Bango

Session: Introduction to Building Windows 8 Metro Applications

Log:

1. creating datacontext
2. created MainCtrl
3. getting all remotely
4. retrieved 29
5. getting all locally
6. retrieved 29
7. searching for ba locally
8. retrieved 3

[Breeze Home](#)





**Resources**



## Step 1 of 6



You should see a list of employee names appear in the Output pane below.

Now, let's try and make this query a bit more complicated by ordering the results. In *Breeze* we do this by adding an "orderBy" clause to a query.

Let's try this by adding the following clause to the bottom of the query in the *JavaScript* panel.

```
.orderBy("LastName");
```

The query should now look like this:

```
var query = new breeze.EntityQuery()  
  .from("Employees")  
  .orderBy("LastName");
```

and then press **Run**.

## Output

Run

- Nancy Lynn Davolio
- Andrew Fuller
- Janet Leverling
- Margaret Peacock
- Steven Buchanan
- Michael Suyama
- Laura Callahan
- Anne Dodsworth
- Pearl Pindlegrass

## JavaScript

```
breeze.config.initializeAdapterInstance( "modelLibrary", "backingS  
app.controller("LearnCtrl", function($scope) {  
  var manager = new breeze.EntityManager('api/northwind');  
  
  var query = new breeze.EntityQuery()  
    .from("Employees");  
  
  manager.executeQuery(query).then(function(data){  
    $scope.results = data.results;  
    $scope.$apply();  
  }).fail(function(e) {  
    alert(e);  
  });  
});
```

## Html

```
<div ng-controller="LearnCtrl">  
  <p ng-show="!results">Fetching...</p>  
  <li ng-show="results" ng-repeat="result in results">  
    <span>{{result.FirstName}}</span>  
    <span>{{result.LastName}}</span>  
  </li>  
</div>
```

## Getting started

[Introduction](#)[Features](#)[Tutorials](#)[Prerequisites](#)[Start with NuGet](#)[Download](#)[FAQ](#)

## Basic Breeze

[A lap around breeze](#)[First query](#)[Query with a filter](#)[Add a new entity](#)[Databinding with Knockout](#)[Change tracking](#)[Save changes](#)

## A lap around Breeze

This is a quick spin around Breeze. It sticks to the basics that a JavaScript client developer must know to work with Breeze. You won't learn everything but you will acquire a foundation in breeze sufficient to build a simple application. You'll know

- how to query the persistence service for entities
- how to write a query filter on the client and sort the results on the data tier
- to hold a *promise* from Breeze while you wait for the service to respond; the promise will tell you when the service is ready with results
- the `EntityManager` maintains a local cache of entities that you've queried, added to, modified, and marked for deletion
- the `EntityType` describes the details of an entity class, how it's structured and how it behaves



Feedback

# APIs

Classes

Modules

Type to filter APIs

AutoGeneratedKeyType

config

DataProperty

DataType

EntityAction

EntityAspect

EntityKey

EntityManager

EntityQuery

EntityState

EntityType

Enum

EnumSymbol

Event

FetchStrategy

FilterQueryOp

LocalQueryComparisonOptions

MergeStrategy

MetadataStore

NamingConvention

NavigationProperty

Predicate

Promise

QueryOptions

## EntityManager Class

Show: ☒ Inherited ☐ Protected ☐ Private ☐ Deprecated

Defined in: [entityManager.js:35](#)

Module: [entityModel](#)

Instances of the EntityManager contain and manage collections of entities, either retrieved from a backend datastore or created on the client.

Index

Methods

Properties

Events

### <ctor> EntityManager ( [config] )

Defined in [entityManager.js:40](#)

#### Parameters:

- [config] Object | String **optional**  
Configuration settings or a service name.
  - [serviceName] String **optional**
  - [metadataStore=MetadataStore.defaultInstance] MetadataStore **optional**
  - [queryOptions=QueryOptions.defaultInstance] QueryOptions **optional**
  - [saveOptions=SaveOptions.defaultInstance] SaveOptions **optional**
  - [validationOptions=ValidationOptions.defaultInstance] ValidationOptions **optional**
  - [keyGeneratorCtor] Function **optional**
  - [adapters] Adapter settings **optional**
    - [dataService] String **optional**  
dataService name

#### Example:

At its most basic an EntityManager can be constructed with just a service name

```
var entityManager = new EntityManager( "api/NorthwindIBModel" );
```

## Samples

- [Todo](#)
  - [Todo-Knockout](#)
  - [Todo-Angular](#)
  - [Todo-Require](#)
- [Breezy Devices](#)
- [DocCode](#)
- [Car Bones](#)
- [NoDb](#)
- [Hot Towel](#)
  - [template](#)
- [Breeze/Knockout](#)
  - [template](#)



### Breeze Todo

The ubiquitous sample app. Breeze Todo demonstrates Breeze's ability to query, create, modify, and delete entities; validate, and save.



### Hot Towel SPA template

An ASP.NET 4 project for a single page application with Knockout, Durandal, and Breeze.



### Breeze/Angular SPA template

An ASP.NET MVC 4 project for a single page application with Angular and Breeze.



### Breeze/Knockout SPA template

An ASP.NET MVC 4 project for a single page application with Knockout and Breeze.



### Breezy Devices

The simple sample from Julie Lerman's December 2012 MSDN Magazine article *"Pain-Free Data Access in JavaScript"*.



### DocCode

DocCode helps you experience Breeze features as running code. It consists a battery of QUnit automated tests we call Teaching Tests.

*learn more at*  
**www.breezejs.com**

*follow* @breezejs