

Let's Tame a Dragon!!

Tom Van Doorslaer, Fiddle
June 30, 2017



#F #Fiddle

Let's Tame a Dragon!!

I mean a mobile app

Tom Van Doorslaer, Fiddle
June 30, 2017





A long **LONG** time ago

Let's slay a Dragon

The Beast

The Hero

A Princess?

The Quest

Treasure!

A wise ruler

I must protect my people!



The three ~~musket~~ers components



Central instance

Messaging

Agents

The three ~~musketeers~~ components



Central instance

Log of sightings

Submission forms

Messaging

Agents

The three ~~musketeers~~ components



Central instance

Messaging

Agents



The three ~~musketeers~~ components



Central instance

Log of sightings

Submission forms

Messaging

Detect and report

Agents

The three ~~musket~~ers components



Central instance



Messaging



Agents



The three ~~musketeers~~ components

Central instance

Log of sightings

Submission forms

Messaging

Detect and report

Agents

Recent sightings

Submission forms

Stealth

The three ~~musket~~ers components



Central instance



Messaging



Agents



The three musketeers components



Central instance

Data model

API

Messaging

Change detection

What SAP Does

Agents

Data model

API

UI



The three musketeers components

Central Instance

Data model

API

Big, Bulky,
Unpractical

Build it yourself

Messaging

Change detection

Meh...

Agents

Data model

API

UI

Offline plugin

Wait, what?
Frontend API?

UI5 is
Awesome!



Really, we can do **better**

Let's slay a Dragon

The Beast

The Hero

A Princess?

The Quest

Treasure!

The three musketeers components



Central Instance

Data model

API

Timesliced data

Still Build it yourself

Messaging

Change detection

No middleware!

**Offline model
using indexedDB**

Agents

Data model

API

UI

**Business
objects**

**UI5 is still
Awesome!**



From a kingdom **far far away**

Let's slay a Dragon

The Beast

The Hero

A Princess?

The Quest

Treasure!

The three musketeers components



Backend

Data model

API

Timesliced data

Still Build it yourself

Synchronization

Change detection

No middleware!

Offline model
using indexedDB

Frontend

Data model

API

UI

Business
objects

UI5 is still
Awesome!



Timesliced tables

In a CRUD-model, 1 object is represented by 1 entry in a table

This record can be read, changed and removed, or a new object created



Timesliced tables

In a **Timesliced** CR-Model, an object is represented by a set of records, which each represent a change at a certain time.

A new change can be added, or a delete record inserted

Timesliced tables



Id	Timestamp	received_on_server	Name	Type	...
1	20170101143510	20170103000000	Smag	Drag	

Timesliced tables



Id	Timestamp	received_on_server	Name	Type	...
1	20170101143510	20170103000000	Smag	Drag	
1	20170101143615	20170103000000	null	Dragon	
1	20170101143615	20170103000000	Smag	Dragon	

Timesliced tables



Id	Timestamp	received_on_server	Name	Type	...
1	20170101143510	20170103000000	Smag	Drag	
1	20170101143615	20170103000000	null	Dragon	
1	20170101153000	20170103000000	Smaug	Null	
1	20170101153000	20170103000000	Smaug	Dragon	

Timesliced tables



Id	Timestamp	received_on_server	Name	Type	status
1	20170101143510	20170103000000	Smag	Drag	
1	20170101143615	20170103000000	null	Dragon	
1	20170101153000	20170103000000	Smaug	Null	
1	20170101153000	20170115000000	null	Null	Tamed
1	20170101153000	20170115000000	Smaug	Dragon	Tamed



Timesliced tables

No locking issues!

No data conflicts!

Change documents included

Undo!

Review/approve/reject

API's become super simple!

The three musketeers components



Backend

Data model

API

Timesliced data

Still Build it yourself

Synchronization

Change detection

No middleware!

Offline model
using indexedDB

Frontend

Data model

API

UI

Business
objects

UI5 is still
Awesome!



You'll want her **number**

Let's slay a Dragon

The Beast

The Hero

A Princess?

The Quest

Treasure!

Offline BusinessObjectModel



First, a little rant: What is a Model?

A model contains data and business logic

But in a webservice, the logic is on the server

Offline **BusinessObject**Model



It's offline! Using IndexedDB

It **automagically** stores all entries in a local IndexedDB, upon sync

Using changeRecords (remember timeslices?)

Offline **BusinessObject**Model



Every entry, is actually an object instance

It acts like a silly JSON entry

But is really a **business object**

```
<Label text="Name"/>
<Input value="{monsters>name}" width="15rem"/>

<Label text="Id"/>
<Input value="{monsters>id}" width="15rem" enabled="false"/>

<Label text="Type"/>
<Input value="{monsters>type}" width="15rem" enabled="false"/>

<Label text="Special Ability"/>
<TextArea value="{monsters>specialAbility}" width="15rem" height="5rem"/>

<Label text="Add last sighting"/>
<Button src="sap-icon://map" text="Last seen here!" press="onLastSeenPress"

DragonCtrl.prototype.onPressSave = function(oEvent){
    var oMonster = this.getView().getBindingContext("monsters").getObject();
    oMonster.save();

    this.navBack();
};
```

Offline **BusinessObjectModel**





And it has built-in sync

```
<fiddle:Sync text="Sync" icon="sap-icon://synchronize" />
```

Automatic synchro ☐ OFF

monsters

	0 / 23	01/Jan/1970
	0 / 0	01/Jan/1970



I saw a **Dragon!**

Let's slay a Dragon

The Beast

The Hero

A Princess?

The Quest

Treasure!

How to **spot** a dragon



Offline Business Object Model

ZippleBack

ZippleBack

Name ZippleBack

Id id-1498251699010-11


Type Dragon

Special Ability Monologues in stereo

Add last sighting Last seen here!


Picture Browse...

Picture



+ Save Cancel

+ Save Cancel





How to **spot** a dragon

First, you need a Monster model

```
"dataSources": {  
  "monsters": {  
    "uri": "/service/db.php?entity=monsters",  
    "type": "JSON",  
    "settings": {  
      "annotations": []  
    }  
  }  
}  
  
"models": {  
  "monsters": {  
    "type": "be.fiddle.BusinessObjectModel.model.BusinessObjectModel",  
    "dataSource": "monsters",  
    "settings": {  
      "testData": "/service/monsters.json",  
      "ObjectClass": "be.fiddle.BusinessObjectModel.model.businessobject.Monster",  
      "db": "ui5busobjModel",  
      "store": "monsters",  
      "metadataUriParams": {  
        "sap-documentation": "heading"  
      }  
    }  
  },  
}
```

How to **spot** a dragon



In essence, a Dragon is a monster, so start by defining a monster

```
sap.ui.define([
    "be/fiddle/BusinessObjectModel/model/BusinessObject"
], function(BusinessObject) {
    "use strict";

    var Monster = BusinessObject.extend("be.fiddle.BusinessObjectModel.model.businessobject.Monster", {
        constructor: function(oData){
        }
    });

    Monster.prototype.setPicture = function(sDataUrl){
    };

    //static constructor
    Monster.ClassConstructor = function () {
        sap.ui.require(["be/fiddle/BusinessObjectModel/model/businessobject/monster/Dragon"],
            function(Dragon){
            });
    };

    Monster.getObject = function(oData){
        switch(oData.type){
            case "Dragon":
                return be.fiddle.BusinessObjectModel.model.businessobject.monster.Dragon.getObject(oData);
            default:
                return new Monster(oData);
        }
    }; //factory to get the right child object

    //statically call the class constructor
    setTimeout( Monster.ClassConstructor, 0); //to prevent the constructor from running before the BusinessObject is defined

    return Monster;
});
```

The getObject factory will provide the right monster type

How to **spot** a dragon



In essence, a

```
sap.ui.define([
    "be/fiddle/BusinessObjectModel/model/BusinessObject"
], function(BusinessObject) {
    "use strict";
```

```
    Monster.getObject = function(oData){
        switch(oData.type){
            case "Dragon":
                return be.fiddle.BusinessObjectModel.model.businessobject.monster.Dragon.getObject(oData);
            default:
                return new Monster(oData);
        }
    }; //factory to get the right child object
```

The getObject
factory will provide
the right monster
type

```
        case "Dragon":
            return be.fiddle.BusinessObjectModel.model.businessobject.monster.Dragon.getObject(oData);
        default:
            return new Monster(oData);
        }
    }; //factory to get the right child object
```

```
    //statically call the class constructor
    setTimeout( Monster.ClassConstructor, 0); //to prevent the constructor from running before the BusinessObject is defined

    return Monster;
});
```



How to **spot** a dragon

Then, define your dragon

```
sap.ui.define([
    "be/fiddle/BusinessObjectModel/model/businessobject/Monster"
], function(Monster) {
    "use strict";

    var Dragon = Monster.extend("be.fiddle.BusinessObjectModel.model.businessobject.monster.Dragon", {
        constructor: function(oData){}
    });

    Dragon.prototype.addSighting = function(){};

    Dragon.getObject = function(oData){
        return new Dragon(oData);
    };//factory to get the right child object

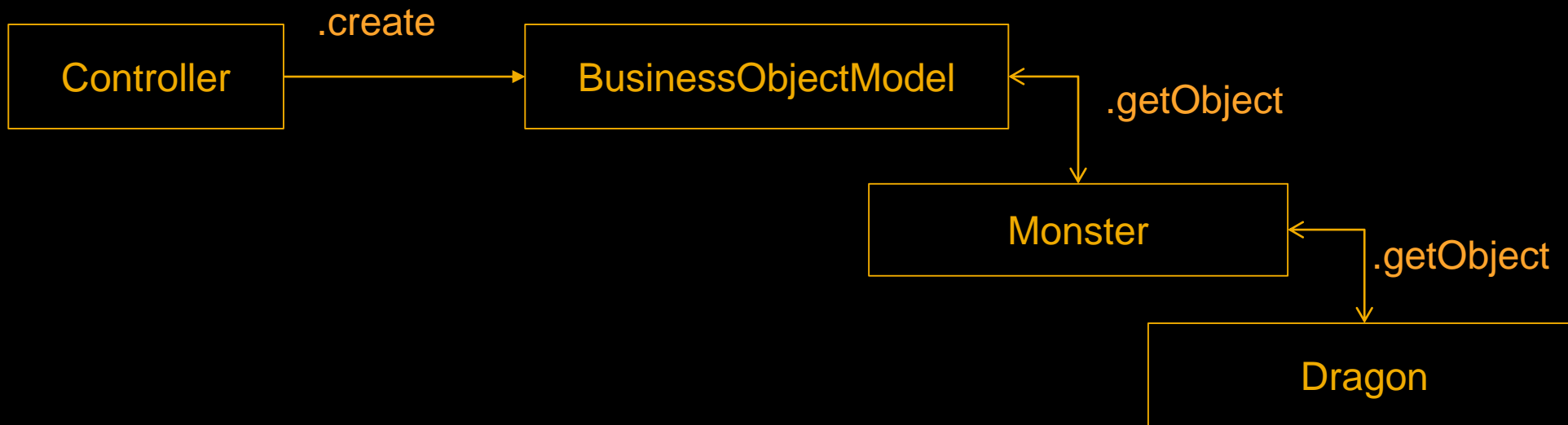
    return Dragon;
});
```



How to **spot** a dragon

Creating a new dragon, is easy.

```
MonstersCtrl.prototype.addDragon = function(){  
    var oDragon = this.getView().getModel("monsters").create( {"type":"Dragon" });  
    this.getOwnerComponent().getRouter().navTo("Dragon", {"id":oDragon.id });  
};
```





How to **spot** a dragon

Adding a sighting to the dragon:

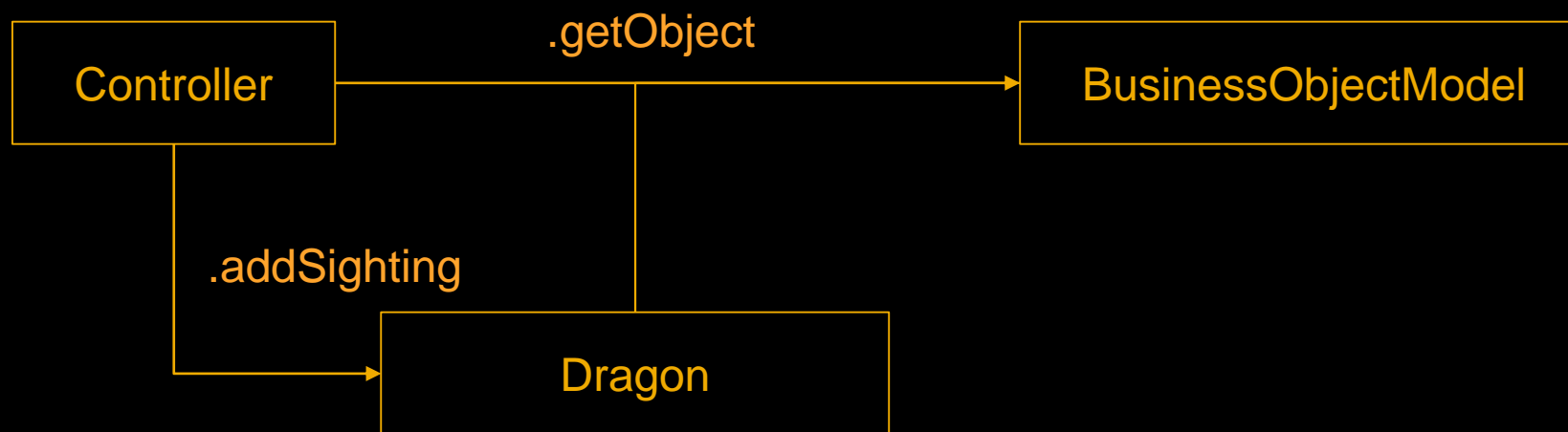
```
Dragon.prototype.addSighting = function(){
    navigator.geolocation.getCurrentPosition(function(oPos){
        this.longitude = oPos.coords.longitude;
        this.latitude = oPos.coords.latitude;
        this.save();
    }).bind(this))
};
```

```
DragonCtrl.prototype.onLastSeenPress = function(oEvent){
    oEvent.getSource().getBindingContext("monsters").getObject().addSighting();
};
```

How to **spot** a dragon



Adding a sighting to the dragon:





Every Dragon has Treasure

Let's slay a Dragon

The Beast

The Hero

A Princess?

The Quest

Treasure!

Resources



<http://businessobjectmodel.fiddle.be> (Yes, I made a typo in the domainname)

{ User: Monster_doctor , pass: ILoveSAP }

<https://github.com/FiddleBe/UI5-BusinessObjectModel>

<http://fiddle.be> (For a couple of other cool UI5 custom controls with alternative input)

And yes, I still have to package it all into a [be.fiddle.library](#)

Whitepaper on Timesliced date in SAP ER On Hana: https://netorg689932-my.sharepoint.com/personal/tom_vandoorslaer_fiddle_be/_layouts/15/guestaccess.aspx?docid=1abdbd995b4da4343985c7ada31e26158&authkey=AZaMtX7PMyaLLKMSC0ntwvM



Innocence was lost



Windows Mobile

- Edge Mobile doesn't fully support indexedDB
- Edge Mobile doesn't support Fiori well
- Fiori Client on Windows mobile uses Edge engine
- There's no chrome, firefox or opera on windows Mobile



Windows Mobile

- Edge Mobile doesn't fully support indexedDB
- Edge Mobile doesn't support Fiori well
- Fiori Client on Windows mobile uses Edge engine
- There's no chrome, firefox or opera on windows Mobile
- Yesterday morning, I went shopping. For an Android Phone.



To be Continued



Next steps

- 1) Bugs—
- 2) Make it work on **Windows Mobile!**
- 3) Figure out Automatic Sync
- 4) Publish **be.fiddle.library**
- 5) Introduce **serviceworkers** and make it a truly **P**rogressive **W**eb **A**pp
- 6) Ditch PHP & MySQL in favour of Hana eXpress
- 7) **[**Offline, Gestures, Eye tracking, **AR goggles****]**.Combine().catch(oDragon)
- 8) Give another preso at UI5Con **year++**

Thank you.

Contact information:

Tom Van Doorslaer

@TomVanDoo

Fiddle.be



You are welcome to give feedback for this session
in the UI5con Event App

Which does not exist for Windows
Mobile!!