Super Fast Application Development with Mura CMS

What’s Up?

- Build a fully functional application using Mura ORM & Mura.js

Target Audience

- ColdFusion/CFML Developer
- Using Mura CMS
- Familiar with
  - .CFCs
  - Mura's base objects
  - Popular SQL DBMS
- May (or may not) have used ColdFusion's ORM

What is Mura CMS?

- Open source Content Management System (CMS)
- Let users add/edit content, while you build applications
- Used by clients large & small

Mura Display Objects

- Self-contained display
  - Portable
  - Distributable (via plugins)
- Could be an entire application!

The Mura Scope

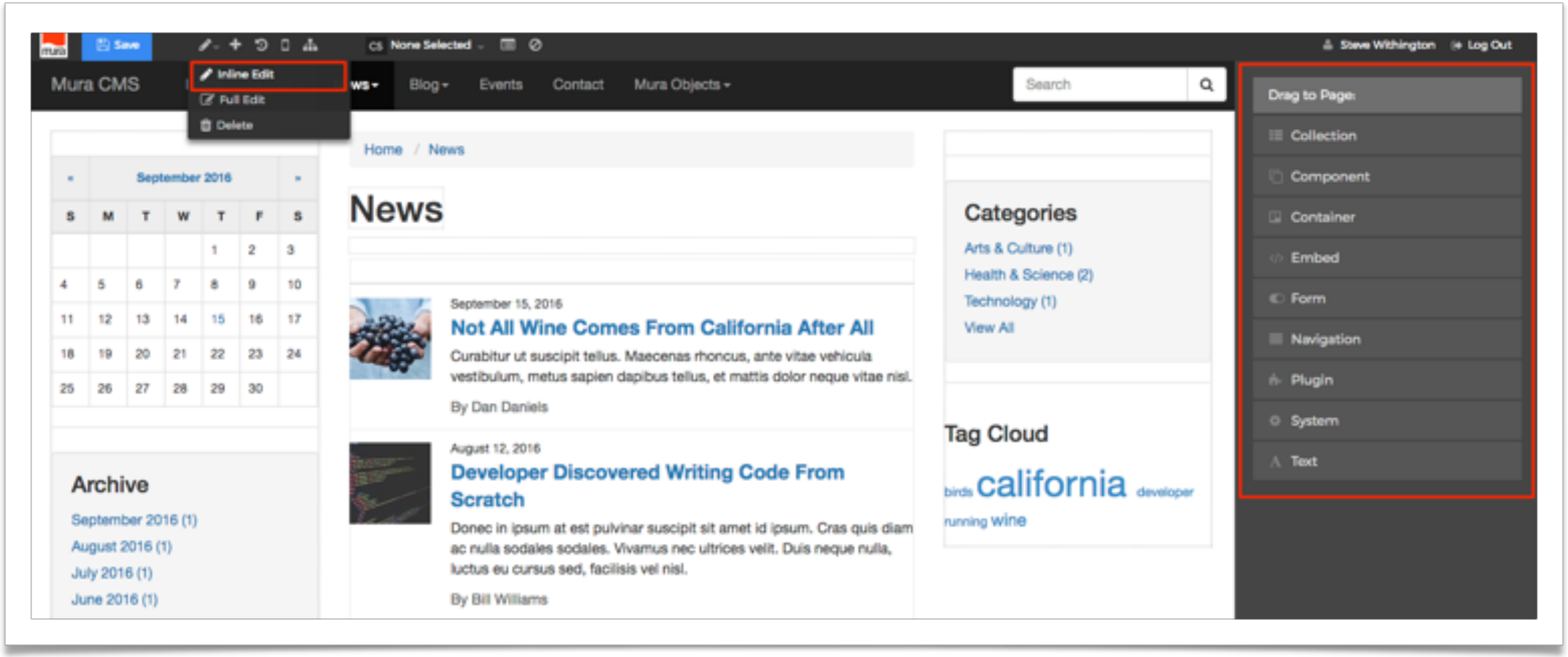
- A wrapper for interacting with Mura CMS
  - Access data
  - Create/modify data
  - Helper methods
- “Mura” or “m” and still supports “$”

Mura Display Objects

Part 1

Mura Display Objects

- Managed via front-end only (*in Mura v7+*)

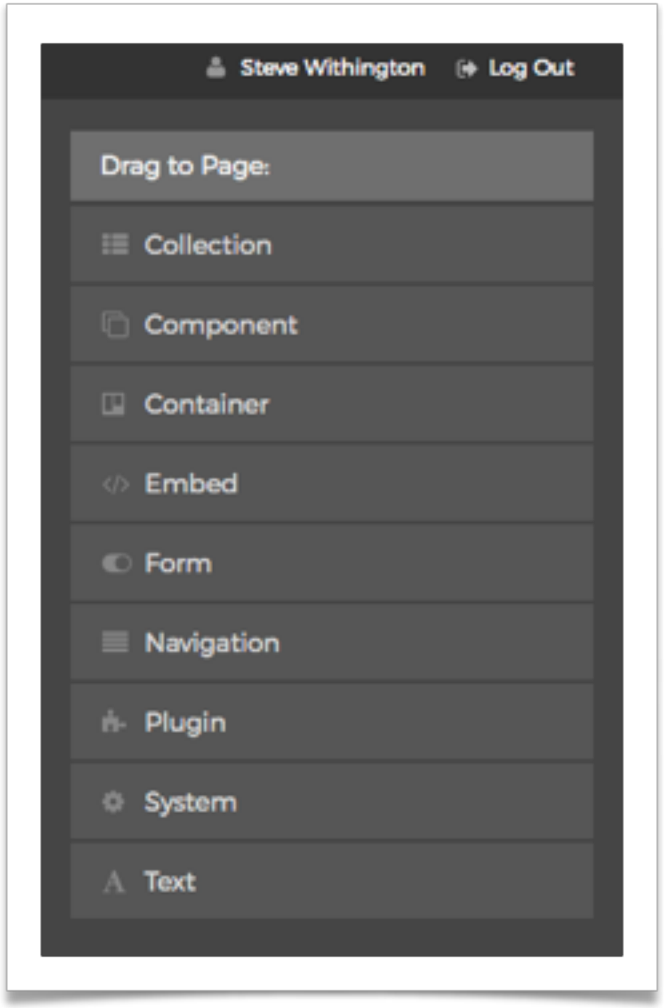


News

September 15, 2016
Not All Wine Comes From California After All
Curabitur ut suscipit tellus. Maecenas rhoncus, ante vitae vehicula vestibulum, metus sapien dapibus tellus, et mattis dolor neque vitae nisl.
By Dan Daniels

August 12, 2016
Developer Discovered Writing Code From Scratch
Donec in ipsum at est pulvinar suscipit sit amet id ipsum. Cras quis diam ac nulla sodales sodales. Vivamus nec ultrices velit. Duis neque nulla, luctus eu cursus sed, facilisis vel nisl.
By Bill Williams

Mura Display Objects



Mura Display Objects

- Directory based format

```
{SiteID}/includes/themes/{ThemeName}/display_objects/
{SiteID}/includes/display_objects/
```

Mura Display Objects

- Directory based format **(works in plugins too)**

```
plugins/{YourPlugin}/display_objects/
```

Mura Display Objects

- Or, register any directory you want …

```
m.siteConfig().registerDisplayObjectDir(
  '/path/to/your/display_objects/'
);

// Path is a logical path to a CFML directory
// Usually registered in onApplicationLoad();
```

Mura Display Objects

Adobe

- **Anatomy** of a display object

```
/mydisplayobject/
   config.xml.cfm
   index.cfm
   configurator.cfm (optional)
```

Mura Display Objects

- config.xml.cfm

```
<mura name="My Display Object"
      contenttypes="*" />
```

Mura Display Objects

- config.xml.cfm

```
<mura name=“My Display Object”
      contenttypes=“*” />
```

** = all content types*

Mura Display Objects

- config.xml.cfm

```
<mura name="My Display Object"
      contenttypes="*" />
```

```
* = all content types
contenttypes="{Type|Type/Subtype},Folder/Blog,Folder/News"
```

Mura Display Objects

- **config.xml.cfm**

```
<mura name="My Display Object"
      contenttypes="*" />


* = all content types

contenttypes="{Type|Type/Subtype},Folder/Blog,Folder/News"

contenttypes="" Blank = never display as draggable option
```

Mura Display Objects

- config.xml.cfm

```
<mura name="My Display Object"
      contenttypes="*"
      condition="(m.content('title') eq 'News')" />
```

Mura Display Objects

- **config.xml.cfm**

```
<mura name="My Display Object"
      contenttypes="*"
      iconclass="mi-somefontawesomeiconclass" />
```

Mura Display Objects

- config.xml.cfm

```
<mura name="My Display Object"
      contenttypes="*">

  <extensions>…</extensions>

</mura>
```

Mura Display Objects

- config.xml.cfm

```
<mura name="My Display Object"
      contenttypes="*">

  <extensions>…</extensions>

  <imagesizes>
    <imagesize name="yoursize" width="100" height="100"/>
  </imagesizes>

</mura>
```

Mura Display Objects

- **index.cfm** *(if rendering via CFML)*

```
<cfparam name="objectparams.mytext" default="">


<cfoutput>
  <h3>#esapiEncode('html', objectparams.mytext)#</h3>
</cfoutput>
```

Mura Display Objects

- **index.cfm** *(if rendering via JS)*

```
<cfset objectparams.render="client">
```

Mura Display Objects

- **index.cfm** *(if rendering via JS)*

```
<cfset objectparams.render="client">
```

- mydisplayobject/**model/handlers/myhandler.cfc**

```
component extends="mura.cfobject" {
  function onRenderStart(m){
    m.addToHTMLHeadQueue('<script src="/path/to/js.js"></script>');
 }
}
```

Mura Display Objects

- **configurator.cfm** (*optional*)

```
<cfparam name="objectparams.mytext" default="">


  <cfoutput>
    <div class="mura-control-group">
      <label class="mura-control-label">My Text</label>
      <input type="text"
             name="mytext"
             class="objectParam"
             value="#esapiEncode('html_attr', objectparams.mytext#">
  </cfoutput>
```

Mura Display Objects

- **configurator.cfm** (*optional*)

```
<cfparam name="objectparams.mytext" default="">

<cf_objectconfigurator>
  <cfoutput>
    <div class="mura-control-group">
      <label class="mura-control-label">My Text</label>
      <input type="text"
             name="mytext"
             class="objectParam"
             value="#esapiEncode('html_attr', objectparams.mytext#">
  </cfoutput>
</cf_objectconfigurator>
```

https://github.com/stevewithington/
murahelloworld/tree/**cfml**

What is **ORM**?

What is ORM?

- **Popular SQL DBMS**
  - Only store/manipulate scalar values such as integers and strings organized within tables

What is ORM?

- **Popular SQL DBMS**
  - Only store/manipulate scalar values such as integers and strings organized within tables
  - Object values must either be converted into groups of simpler values, or only use simple scalar values

What is ORM?

- **Object-Relationtional Mapping** (ORM) **addresses the main issue with SQL DBMS**
  - Translates the logical representation of objects into an atomized form capable of being stored in a database, while preserving the properties of objects and their relationships so they can be reloaded as objects when needed

What is ORM?

- Object-Relational Mapping (**ORM**) is essentially a “**virtual object database**”

What is **Mura ORM**?

What is Mura ORM?

- Mura ORM is a “**virtual object database**” that can be used within Mura

What is Mura ORM?

- Mura ORM is a “**virtual object database**” that can be used within Mura
- Takes advantage of Di1 (dependency injection)
  - https://github.com/framework-one/di1

What is Mura ORM?

- Mura ORM is a “**virtual object database**” that can be used within Mura
- Takes advantage of Di1 (dependency injection)
  - https://github.com/framework-one/di1
- Access & modify objects quickly & easily

What is Mura ORM?

- Mura ORM is a "**virtual object database**" that can be used within Mura
- Takes advantage of Di1 (dependency injection)
    - https://github.com/framework-one/di1
- Access & modify objects quickly & easily
- No need for custom DAOs, class extensions, or database-vendor-specific CRUD statements

What is Mura ORM?

- Mura ORM is a “**virtual object database**” that can be used within Mura
- Takes advantage of Di1 (dependency injection)
  - https://github.com/framework-one/di1
- Access & modify objects quickly & easily
- No need for custom DAOs, class extensions, or database-vendor-specific CRUD statements
- Similar to ColdFusion-based Hibernate ORM (*but not the same!*)

Mura ORM **Configuration**

Mura ORM Configuration

- Convention based vs. having to explicitly register entities

```
{SiteID}/includes/model/
{SiteID}/includes/themes/{ThemeName}/model/
```

Mura ORM Configuration

- Convention based vs. having to explicitly register entities

```
{SiteID}/includes/model/
{SiteID}/includes/themes/{ThemeName}/model/
```

Or any *display object's* "model" directory!

```
../display_objects/mydisplayobject/model/
```

Mura ORM Configuration

- Or, register any directory you want …

```
m.globalConfig().registerModelDir(
  '/path/to/your/model/'
);

// Path is a logical path to a CFML directory
```

Mura ORM Configuration

- Or, register any directory you want …

```
m.globalConfig().registerModelDir(
   '/path/to/your/model/'
);

// Path is a logical path to a CFML directory
// Usually registered in onApplicationLoad();
```

Mura ORM Configuration

- Any **.cfc** under the “**model**” directory will automatically get registered with Di1

Mura ORM Configuration

- Any .cfc under the “model” directory will automatically get registered with Di1
- If it’s under a “**beans**” directory, it will be registered as a **transient**, ***not*** as a singleton

Mura ORM Configuration

- Any .cfc under the “model” directory will automatically get registered with Di1
- If it’s under a “beans” directory, it will be registered as a transient, *not* as a singleton
  - A **transient** exists only for a request and then is discarded
    - You get a new copy every time

Mura ORM Configuration

- Any .cfc under the "model" directory will automatically get registered with Di1
- If it's under a "beans" directory, it will be registered as a transient, *not* as a singleton
    - A transient exists only for a request and then is discarded
        - You get a new copy every time
    - A **singleton** is shared among all threads and requests … there's only one of them
        - You're ***not*** getting a new copy every time
        - Good for service type objects

Mura ORM Configuration

- Any .cfc under the “model” directory will automatically get registered with Di1
- If it’s under a “beans” directory, it will be registered as a transient, *not* as a singleton

```
../model/beans/person.cfc
../model/beans/personaddress.cfc
../model/beans/personphonenumber.cfc
```

Mura ORM Configuration

- If a directory found under “model” is named “**handlers**” then any **.cfc** under this directory will be registered as eventHandlers

Mura ORM Configuration

- If a directory found under “model” is named “**handlers**” then any .cfc under this directory will be registered as eventHandlers
- For example:

```
../model/handlers/myhandler.cfc

../model/services/handlers/myhandler.cfc
```

Mura ORM Configuration

- Reload Mura to get them registered

Mura ORM Configuration

- Reload Mura to get them registered
- The first time you reload, *and anytime you add new properties*, be sure to append ‘**`&applydbupdates`**’ to the URL

Mura ORM Configuration

- Reload Mura to get them registered
- The first time you reload, *and anytime you add new properties*, be sure to append ‘`&applydbupdates`’ to the URL
- For example:
  `./?appreload&applydbupates`

Mura Display Objects

Part 2

Mura Display Objects

- Display object's can have their own "**model**" directory

```
../display_objects/mydisplayobject/model/
```

Mura Display Objects

- Display object's can have their own "model" directory

```
../display_objects/mydisplayobject/model/
```

- And their own "**beans**" directory

```
../display_objects/mydisplayobject/model/beans/
```

Mura Display Objects

- Display object's can have their own "model" directory

```
../display_objects/mydisplayobject/model/
```

- And their own "beans" directory

```
../display_objects/mydisplayobject/model/beans/
```

- And "**handlers**" too

```
../display_objects/mydisplayobject/model/handlers/
```

Address Book

Address Book

- Person

Address Book

- **Person**
  - Zero or more phone numbers

Address Book

- **Person**
  - Zero or more phone numbers
  - Zero or more addresses

Address Book

- **Person** object with attributes/fields

Address Book

- **Person** object with attributes/fields
  - Person’s name

Address Book

- **Person** object with attributes/fields
  - Person's name
  - List of phone numbers
    - **PhoneNumber** objects

Address Book

- **Person** object with attributes/fields
  - Person's name
  - List of phone numbers
    - **PhoneNumber** objects
  - List of addresses
    - **Address** objects

Address Book

- The address book entry (Person) is treated as a single object

Address Book

- The address book entry (Person) is treated as a single object
    - It can be referenced by a single variable containing a pointer to the object

Address Book

- The address book entry (Person) is treated as a single object
  - It can be referenced by a single variable containing a pointer to the object
- Various helper methods can be associated with the object, such as a method to return the preferred phone number, the home address, and so on

Mura ORM **Entity**

Mura ORM Entity

```
component
    extends="mura.bean.beanORM"
    entityname="person"
    table="custom_persons"
    bundleable="true"
    displayname="PersonBean"
    public=true
    orderby="namelast,namefirst" {

    // Properties & Methods

}
```

Mura ORM Entity

- Component (CFC) Attributes

```
entityname
table
datasource
discriminatorcolumn
discriminatorvalue
orderby
readonly
```

Mura ORM Entity

- Component (CFC) Attributes

```
entityname
table
datasource
discriminatorcolumn
discriminatorvalue
orderby
readonly
```

- Mura-specific attributes

```
public
bundleable
cachename
dbtype
manageschema
usetrash
```

Mura ORM Entity

```
component
    extends="mura.bean.beanORM"
    entityname="person" … {

    // primary key (are UUIDs)
    property name="personid" fieldtype="id";

}
```

Mura ORM Entity

```
component
    extends="mura.bean.beanORM"
    entityname="**person**" … {

    // primary key
    property name="personid" fieldtype="id";

    // person attributes
    property name="namefirst" datatype="varchar";
    property name="namelast" datatype="varchar";
}
```

Mura ORM Entity

```
component
    extends="mura.bean.beanORM"
    entityname="person" … {
    …
    // foreign key(s)
    property name="user"
             cfc="user"
             fieldtype="many-to-one"
             fkcolumn="userid";
}
```

Mura ORM Entity

```
component
    extends=“mura.bean.beanORM”
    entityname=“person” … {
    …
    // relationship(s)
    property name=“phonenumbers”
             singularname=“phonenumber”
             cfc=“personphonenumber”
             fieldtype=“one-to-many”
             cascade=“delete”;
}
```

Mura ORM Entity

```
component
    extends=“mura.bean.beanORM”
    entityname=“personphonenumber” … {
    …
    // relationship(s)
    property name=“person”
             cfc=“person”
             fieldtype=“many-to-one”
             fkcolumn=“personid”;
}
```

Mura ORM Entity

```
component
    extends="mura.bean.beanORM"
    entityname="personphonenumber" … {
    …
    // attribute(s)
    property name="phonenumber"
             datatype="varchar"
             length="255"
             required=true
             message="Phone Number is required.";
}
```

Mura ORM Entity

- Property Attributes

```
name
persistent
fieldtype
cfc
fkcolumn
type
cascade
singularname
orderby
length
default
ormtype
```

Mura ORM Entity

- Property Attributes

  ```
  name
  persistent
  fieldtype
  cfc
  fkcolumn
  type
  cascade
  singularname
  orderby
  length
  default
  ormtype
  ```

- Mura-specific attributes

  ```
  loadkey
  datatype
  nullable
  required
  validate
  message
  regex
  comparable
  ```

Mura ORM Entity

- Property validation attributes

`minValue`	`inList`
`maxValue`	`method`
`minLength`	`lte`
`maxLength`	`lt`
`minCollection`	`gte`
`maxCollection`	`gt`
`minList`	`eq`
`maxList`	`neq`

Mura ORM Entity

```
entity = m.getBean('entityName')
            .loadBy(someAttribute='Something');
```

Mura ORM Entity

```
entity = m.getBean(‘entityName’)
            .loadBy(someAttribute=‘Something’);


entity.get(‘attribute’);
```

Mura ORM Entity

```
entity = m.getBean('entityName')
            .loadBy(someAttribute='Something');

entity.get('attribute');
entity.set('attribute', 'Some Value');
```

Mura ORM Entity

```
entity = m.getBean('entityName')
           .loadBy(someAttribute='Something');


entity.get('attribute');

entity.set('attribute', 'Some Value').save();
```

Mura ORM Entity

```
entity = m.getBean('entityName')
              .loadBy(someAttribute='Something');


entity.get('attribute');
entity.set('attribute', 'Some Value').save();
entity.get{RelatedEntity}();
```

Mura ORM Entity

```
entity = m.getBean('entityName')
           .loadBy(someAttribute='Something');

entity.get('attribute');
entity.set('attribute', 'Some Value').save();
entity.get{RelatedEntity}();
entity.get{RelatedEntity}Iterator();
```

Mura ORM Entity

```
entity = m.getBean(‘entityName’)
           .loadBy(someAttribute=‘Something’);

entity.get(‘attribute’);
entity.set(‘attribute’, ‘Some Value’).save();
entity.get{RelatedEntity}();
entity.get{RelatedEntity}Iterator();
entity.get{RelatedEntity}Query();
```

Mura ORM Entity

```
entity = m.getBean(‘entityName’)
           .loadBy(someAttribute=‘Something’);


entity.get(‘attribute’);
entity.set(‘attribute’, ‘Some Value’).save();
entity.get{RelatedEntity}();
entity.get{RelatedEntity}Iterator();
entity.get{RelatedEntity}Query();
m.getFeed(‘entityName’);
```

Mura ORM Entity

```
contactBean = m.getBean(‘person’)
                .loadBy(personid=m.event(‘pid’));
```

Mura ORM Entity

```
contactBean = m.getBean('person')
                  .loadBy(personid=m.event('pid'));

if ( contactBean.exists() ) {
  // This bean exists!
} else {
  // Do something if it's a new bean
}
```

Mura ORM Entity

```
contactBean = m.getBean('person')
              .loadBy(personid=m.event('pid'));

// Phone Numbers Iterator
itPhones = contactBean.getPhoneNumbersIterator();
```

Mura ORM Entity

```
contactBean = m.getBean('person')
                  .loadBy(personid=m.event('pid'));

// Phone Numbers Iterator
itPhones = contactBean.getPhoneNumbersIterator();

while ( itPhones.hasNext() ) {
  phone = itPhones.next();
  WriteOutput( phone.get('phonenumber') );
}
```

Mura ORM **Feed**

Mura ORM Feed (Content)

```
m
  .getFeed(‘content’);
```

Mura ORM Feed (Content)

```
m
  .getFeed(‘content’)
  .where()  // optional … but readable
  .prop(‘title’)
  .isEQ(‘News’)
  .orProp(‘title’)
  .isEQ(‘Blog’)
  .sort(‘title’, ‘desc’)
  .getIterator();
```

Mura ORM Feed (Entity)

```
m
  .getFeed('person')
  .where()
  .prop('namefirst')
  .containsValue(m.event('namefirst'))
  .orProp('namelast')
  .isEQ(m.event('namelast'))
  .getIterator();
```

Mura ORM Feed (Sorting)

```
m
  .getFeed('person')
  .where()
  .prop('userid')
  .isEQ(m.currentUser('userid')
  .sort('namelast', 'asc')
  .sort('namefirst', 'desc')
  .getIterator();
```

Mura ORM Feed (Methods)

```
m
  .getFeed([entityname])
  .where()
  .prop([property])
  .andProp([property])
  .orProp([property])
  .isEQ([criteria])
  .isNEQ([criteria])
  .isLT([criteria])
  .isLTE([criteria])
  .isGT([criteria])
  .isGTE([criteria])
  .containsValue([criteria])
  .null()
```

```
.beginsWith([criteria])
.endsWith([criteria])
.openGrouping()
.andOpenGrouping()
.closeGrouping()
.sort([property],[asc||desc])
.itemsPerPage([itemsPerPage])
.maxItems([maxItems])
.isIn([list criteria])
.isNotIn([list criteria])
.innerJoin([relatedEntity])
.leftJoin([relatedEntity])
.getQuery()
.getIterator()
```

Mura ORM **Events**

Mura ORM Events

- `../model/`**`handlers/myhandler.cfc`**

Mura ORM Events

- ../model/handlers/myhandler.cfc

```
onBefore{Entity}Save(m) {
   var bean = m.event(‘bean’);
   …
}
```

Mura ORM Events

- ../model/handlers/myhandler.cfc

```
onBefore{Entity}Save(m) {
   var bean = m.event('bean');
   …
}

// Plus any other Mura event listeners you want!
```

Mura ORM Events

- Support for ColdFusion/CFML ORM Events

```
preLoad();                    postLoad();
preUpdate();                  postUpdate();
preCreate();                  postCreate();
preInsert();                  postInsert();
preDelete();                  postDelete();
```

https://github.com/stevewithington/
muracontacts/tree/**cfml**

Mura.js

Mura.js

- A lightweight utility to decouple dependency on jQuery

Mura.js

- A lightweight utility to decouple dependency on jQuery
- A JS framework for interacting with the Mura JSON API

Mura.js

- Familiar jQuery syntax

```
Mura(function(m) {
  m('.target').each(function() {
    m(this).html('Mura found you!');
  });
});
```

Mura.js

- Baked-in **AJAX** support

```
Mura
  .ajax({
    type: ‘post’,
    url: ’https://domain.com/path/',
    data: { key: value },
    success: function(resp) { console.log(resp) },
    error: function(resp) { console.log(resp) }
  });
```

Mura.js

- Baked-in AJAX support & **JS promises**

```
Mura
   .get('https://domain.com/path/')
   .then(function(resp) {
     // success
     Mura('#target').html(resp.data.html);
   }, function(e) {
     // fail
   });
```

Mura.js

- Baked-in AJAX support & JS promises

```
Mura
  .post(
    ‘https://domain.com/path/',
    { key: value }
  )
  .then(function(resp) {
    // success
  }, function(e) {
    // error
  });
```

Mura.js

- JS promises can be stacked like Russian dolls

```
Mura.get('https://domain.com/path/')
   .then(function(resp) {
     // success
     Mura.get('https://domain.com?id=' + resp.id)
       .then(function(newresp) {
         Mura('#target').html(newresp.html);
       });
   }, function(e) {
     // error
   });
```

Mura.js

- Register event handlers

```
Mura('#mybutton')
   .on('click', function(e) {
     e.preventDefault();
     console.log(e);
   });
```

Mura.js

- **Mura.DOMSelection** class
  - Wraps selected target via the Mura() method

Mura.js

- **Mura.DOMSelection** class
  - Wraps selected target via the Mura() method
  - Allows you to handle selection as a *single object* or a *collection*

Mura.js

- **Mura.DOMSelection** class
  - Wraps selected target via the Mura() method
  - Allows you to handle selection as a *single object* or a *collection*
    - **Single object**

```
Mura('.target').html('Hello world!');
```

Mura.js

- **Mura.DOMSelection** class
  - Wraps selected target via the Mura() method
  - Allows you to handle selection as a *single object* or a *collection*
    - **Single object**
```
Mura('.target').html('Hello world!');
```
    - **Collection**
```
Mura('.target').each(function() {
  Mura(this).html('Hello world!');
});
```

Mura.js

- Supported methods can be found under:

  `{SiteID}/js/src/`**`mura.domselection.js`**

Mura.js

- **Load** entities

```
Mura
   .getEntity('person')
   .loadBy('personid', pid)
   .then(function(person) {
     // success
     console.log(person);
   }, function(person) {
     // fail
     console.log(person.get('errors'));
   });
```

Mura.js

- **Create/Update** entities

```
Mura
   .getEntity('person')
   .loadBy('personid', pid)
   .then(function(person) {
     // load success
     person.set('namelast', 'Withington')
       .save()
       .then(function(person) {
         // save success
       }, function(person) {
         // save fail
       });
   });
```

Mura.js

- **Delete** entities

```
Mura
   .getEntity('person')
   .loadBy('personid', pid)
   .then(function(person) {
     // load success
     person
        .delete()
        .then(function(person) {
          // delete success
        }, function(person) {
          // delete fail
        });
   });
```

Mura.js

- Feed API

```
Mura
   .getFeed('person')
   .where()   // optional
   .prop('namelast').isEQ('Levine')
   .orProp('namelast').beginsWith('Withing')
   .getQuery()
   .then(function(people) {
     // success
     people.each(function(prsn, idx) {
       console.log(prsn.get('namefirst') + ' ' + prsn.get('namelast'));
     });
   });
```

Mura.js (Feed Methods)

```
Mura
  .getFeed([entityname])
  .where()
  .prop([property])
  .andProp([property])
  .orProp([property])
  .isEQ([criteria])
  .isNEQ([criteria])
  .isLT([criteria])
  .isLTE([criteria])
  .isGT([criteria])
  .isGTE([criteria])
  .containsValue([criteria])
  .null()
  .beginsWith([criteria])
  .endsWith([criteria])
  .openGrouping()
  .andOpenGrouping()
  .closeGrouping()
  .sort([property],[asc||desc])
  .itemsPerPage([itemsPerPage])
  .maxItems([maxItems])
  .isIn([list criteria])
  .isNotIn([list criteria])
  .innerJoin([relatedEntity])
  .leftJoin([relatedEntity])
  .getQuery()
```

Mura Display Objects

Part 3

Mura Display Objects

- Loading JS/CSS files

```
Mura(function(m) {
  m.loader()
    .loadcss(m.themepath + ‘/path/all.css’, {media:’all’})
    .loadcss(m.themepath + ‘/path/print.css’, {media:’print’})
    .loadjs(
      m.themepath + ’/path/script1.js’,
      m.themepath + ’/path/script2.js’,
      function() {
       // Now do something with the loaded JS
      }
    );
});
```

Mura Display Objects

- Mura.**loader()**
  - Only loads files once (no duplicates)

Mura Display Objects

- Mura.loader()
  - Only loads files once (no duplicates)
  - **loadjs()** can take **string** arguments or **arrays** of strings

Mura Display Objects

- Mura.loader()
    - Only loads files once (no duplicates)
    - **loadjs()** can take string arguments or arrays of strings
        - if `array`, files are loaded *asynchronously*

Mura Display Objects

- Mura.loader()
  - Only loads files once (no duplicates)
  - **loadjs()** can take string arguments or arrays of strings
    - if `array`, files are loaded *asynchronously*
    - if `string`, files are loaded *synchronously*

Mura Display Objects

- loadjs() **Synchronous** Example
  - When the 2nd file is loading, it ***can*** reference variables from 1st

```
Mura(function(m) {
  m.loader()
    .loadjs(
      m.themepath + '/path/script1.js',
      m.themepath + '/path/script2.js',
      function() {
       // Now do something with the loaded JS
      }
    );
});
```

Mura Display Objects

- loadjs() **Asynchronous** Example
  - When the 2nd file is loading, it ***cannot*** reference variables from 1st

```
Mura(function(m) {
  m.loader()
    .loadjs(
      [
        m.themepath + '/path/script1.js',
        m.themepath + '/path/script2.js'
      ],
      function() {
       // Now do something with the loaded JS
      }
    );
});
```

Mura Display Objects

- **index.cfm** *(Full example if rendering via Server/CFML)*

```
<cfparam name="objectparams.mytext" default="">

<cfoutput>
  <h3>#esapiEncode('html', objectparams.mytext)#</h3>
</cfoutput>

<script>
Mura(function(m) {
  m.loader()
    .loadcss(m.themepath + '/path/my.css')
    .loadjs(m.themepath + '/path/my.js');
});
</script>
```

Mura Display Objects

- **index.cfm** *(Full example if rendering via Client/JS)*

```
Mura.DisplayObject.myobject = Mura.UI.extend({
  render: function() {
    this.container = Mura(this.context.targetEl);

    Mura.loader()
      .loadcss(Mura.themepath + '/path/my.css')
      .loadjs(Mura.themepath + '/path/my.js',
        function(){
          //DO STUFF
        }
      );
    }
});
```

https://github.com/stevewithington/
murahelloworld/tree**/js**

https://github.com/stevewithington/
muracontacts/tree/**js**

content_types

content_types

- Control body by Type/Subtype

content_types

- Control body by Type/Subtype
  - Folder/Contacts
  - Page/Contact

content_types

- Control body by Type/Subtype

  `{SiteID}/includes/`**`content_types`**`/`

content_types

- Control body by Type/Subtype

```
{SiteID}/includes/content_types/
{SiteID}/includes/themes/{ThemeName}/content_types/
```

content_types

- Or, register any directory you want …

```
m.siteConfig().registerContentTypeDir(
   ‘/path/to/your/content_types/‘
);

// Path is a logical path to a CFML directory
```

content_types

- Or, register any directory you want …

```
m.siteConfig().registerContentTypeDir(
  ‘/path/to/your/content_types/‘
);

// Path is a logical path to a CFML directory
// Usually registered in onApplicationLoad();
```

content_types

- Control body by Type/Subtype

```
{SiteID}/includes/content_types/
{SiteID}/includes/themes/{ThemeName}/content_types/
```

  - Target Type/Subtype by directory structure:

    ```
    content_types/{type}
    ```

content_types

- Control body by Type/Subtype

```
{SiteID}/includes/content_types/
{SiteID}/includes/themes/{ThemeName}/content_types/
```

  - Target Type/Subtype by directory structure:

```
content_types/{type}
content_types/page/index.cfm
```

content_types

- Control body by Type/Subtype

```
{SiteID}/includes/content_types/
{SiteID}/includes/themes/{ThemeName}/content_types/
```

- Target Type/Subtype by directory structure:

```
content_types/{type}
content_types/page/index.cfm

content_types/{type}_{subtype}
```

content_types

- Control body by Type/Subtype

```
{SiteID}/includes/content_types/
{SiteID}/includes/themes/{ThemeName}/content_types/
```

  - Target Type/Subtype by directory structure:

```
content_types/{type}
content_types/page/index.cfm

content_types/{type}_{subtype}
content_types/page_contact/index.cfm
```

content_types

- **Anatomy** of a content_type

```
content_types/folder_contacts/index.cfm
content_types/folder_contacts/config.xml.cfm
```

content_types

- **Anatomy** of a content_type

```
content_types/folder_contacts/index.cfm
content_types/folder_contacts/config.xml.cfm
```

  - **`index.cfm`** = the body/view

content_types

- **Anatomy** of a content_type

```
content_types/folder_contacts/index.cfm
content_types/folder_contacts/config.xml.cfm
```

  - `index.cfm` = the body/view
  - **`config.xml.cfm`** = the configuration

content_types

- **Anatomy** of a content_type

```
content_types/folder_contacts/index.cfm
content_types/folder_contacts/config.xml.cfm

content_types/folder_contacts/model/
```

content_types

- **Anatomy** of a content_type

```
content_types/folder_contacts/index.cfm
content_types/folder_contacts/config.xml.cfm

content_types/folder_contacts/model/
content_types/folder_contacts/model/beans/
content_types/folder_contacts/model/handlers/
```

Q+A

Resources

- http://www.getmura.com
- https://github.com/stevewithington/cfsummit-2016
- https://github.com/stevewithington/muracontacts
- https://github.com/stevewithington/murahelloworld
- https://groups.google.com/forum/#!forum/mura-cms-developers

Thank you!