Mura ORM

Stephen J. Withington, Jr.

mura

What is ORM?

- Object-relational mapping (ORM) in computer science is a programming technique for converting data between incompatible type systems in object-oriented programming languages.
- This creates, in effect, a “virtual object database” that can be used from within the programming language.

mura

What is ORM?

- Object-relational mapping (ORM) in computer science is a programming technique for converting data between incompatible type systems in object-oriented programming languages.
- This creates, in effect, a “**virtual object database**” that can be used from within the programming language.
- https://en.wikipedia.org/wiki/Object-relational_mapping

What is ORM?

- Address book entry
  - Person
    - Zero or more phone numbers
    - Zero or more addresses

mura

What is ORM?

- “Person object” with attributes/fields to hold each data item that the entry comprises
  - Person’s name
  - List of phone numbers
    - “PhoneNumber objects”
  - List of addresses
    - “Address objects”

What is ORM?

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

mura

What is ORM?

- However, many popular database products such as SQL database management systems (DBMS) can only store and manipulate scalar values such as integers and strings organized within tables.
- The programmer must either convert the object values into groups of simpler values for storage in the database (and convert them back upon retrieval), or only use simple scalar values within the program.

mura

What is ORM?

- The heart of the problem is translating the logical representation of the objects into an atomized form that is capable of being stored in the database, while preserving the properties of the objects and their relationships so that they can be reloaded as objects when needed. If this storage and retrieval functionality is implemented, the objects are said to be **persistent**.

  https://en.wikipedia.org/wiki/Object-relational_mapping

What is Mura ORM?

- It's a "**virtual object database**" that can be used within Mura
- Takes advantage of Di1 (dependency injection)

Mura ORM

- 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/someDisplay/model/`

Mura ORM

- Any .cfc under the “**model**” directory will automatically get registered with Di1.
- If it’s under “**beans**” or “**entities**” directory it will ***not*** be registered as a “singleton.”
  - A singleton is shared among all threads and requests … there’s only one of them. In other words, you’re not getting a new copy every time.
- Reload Mura to get them registered.

Mura ORM

- If a directory is called “event_handlers” or “handlers” then any .cfc under these directories will be registered as eventHandlers.
- For example:

```
../model/handlers/
../model/event_handlers/
../model/services/event_handlers/
```

mura

Demo

person.cfc

Mura ORM

```
entity = m.getBean('entityName');
entity.loadBy(someAttribute='Something');
entity.get('attribute');
m.getFeed('entityName');
entity.get{RelatedEntity}();
entity.get{RelatedEntity}Iterator();
entity.get{RelatedEntity}Query();
```

Mura ORM Content Feed

```
m
  .getFeed('content')
  .where()
  .prop('title')
  .isEQ('about')
  .orProp('title')
  .isEQ('events')
  .sort('title', 'desc')
  .getQuery();
```

Mura ORM Entity Feed

```
m
  .getFeed('personphonenumber')
  .where()
  .prop('phonenumber')
  .isEQ('916.555.1212')
  .andProp('personid')
  .isEQ(person.get('personid'))
  .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])
```

```
.contains([criteria])
.beginsWith([criteria])
.endsWith([criteria])
.openGrouping()
.andOpenGrouping()
.closeGrouping()
.sort([property],[desc||asc])
.itemsPerPage([itemPerPage])
.maxItems([maxItems])
.isIn([list criteria])
.isInNoIn([list criteria])
.getQuery()
.getIterator()
```

Mura ORM Feed Sorting

```
m
  .getFeed(‘content’)
  .sort(‘title')
  .sort(‘releaseDate’, 'desc');
```

mura

Demo

Yes!

Thanks!

- Code samples:

  `https://github.com/stevewithington/mura-orm-preso/`

- Download:

  `https://github.com/stevewithington/mura-orm-preso/archive/master.zip`