

Building web apps with Drupal and MongoDB

Bartram Nason
@bartramnason

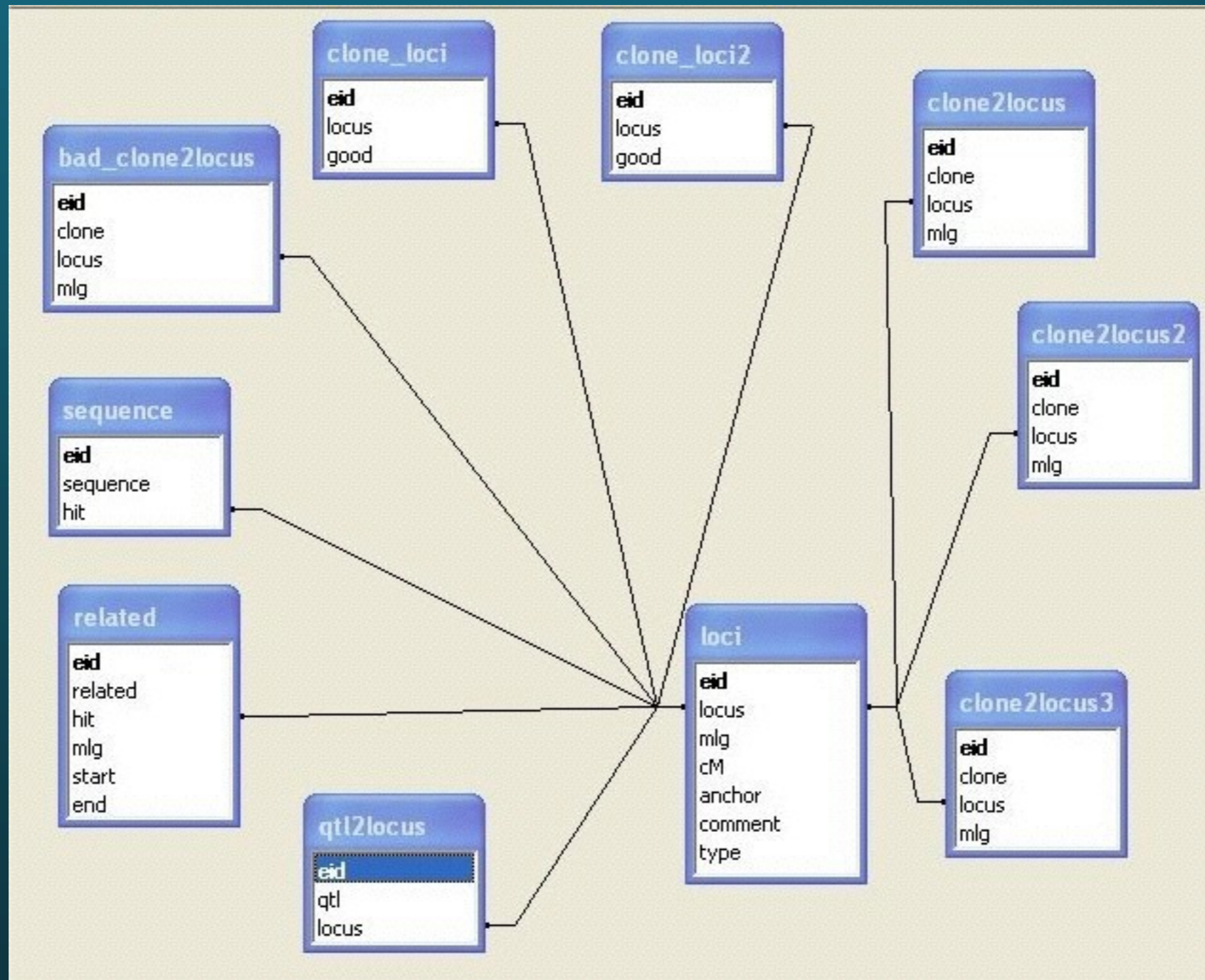
The Entity API

- Define new data types
- Supports fields and properties
- CRUD operations, custom view types, administration UI
- Developer defines base table schema

NoSQL with MongoDB

- Document-based database
- Hierarchical
- Schema-less
- Open-source

SQL tables



MongoDB collections

```
{
  "_id" : ObjectId("505947e7afeb23380a0000bf"),
  "id" : "3263632",
  "type" : "book",
  "title" : "Ulysses",
  "links" : {
    "goodreads" : {
      "url" : "http://www.goodreads.com/book/show/3263632-ulysses",
      "label" : "View on Goodreads"
    }
  },
  "author" : "James Joyce",
  "weight" : 14415,
  "pattern" : /\b((ulysses))\b/i
}
{
  "_id" : ObjectId("505947f1afeb23380a0000fb"),
  "id" : "24337",
  "type" : "book",
  "title" : "Ella Enchanted",
  "links" : {
    "goodreads" : {
      "url" : "http://www.goodreads.com/book/show/24337.Ella_Enchanted",
      "label" : "View on Goodreads"
    }
  },
  "author" : "Gail Carson Levine",
  "weight" : 250,
  "pattern" : /\b((ella enchanted))\b/i
}
{
  "_id" : ObjectId("505947faafeb23380a000135"),
  "id" : "93",
  "type" : "book",
  "title" : "Heidi",
  "links" : {
    "goodreads" : {
      "url" : "http://www.goodreads.com/book/show/93.Heidi",
      "label" : "View on Goodreads"
    }
  },
  "author" : " Johanna Spyri"
```

Requirements

- Drupal 7
- MongoDB from 10Gen
- PHP Mongo extension
- MongoDB and Mongo Entity modules

MongoDB with Drupal

- Replaces SQL tables for caching, logging, sessions, field storage, etc.
- Faster queries when loading many fields
- Simple installation

Using Mongo Entity

- Store entire entity in MongoDB collection
- Quickly create new entity types without database schema
- Embed entities for complex data structures

Defining a new entity type

```
/**
 * Implements hook_entity_info().
 */
function mongo_entity_bookstore_entity_info() {

  $items = array();
  $items['book'] = array(
    'label' => t('Book'),
    'entity class' => 'MongoEntity',
    'controller class' => 'MongoEntityController',
    'access callback' => 'mongo_entity_bookstore_access',
    'label callback' => 'book_label',
    'fieldable' => TRUE,
    'revisionable' => TRUE,
    'bundles' => array(
      'book' => array(
        'label' => t('Book'),
        'admin' => array(
          'path' => 'admin/book/manage',
        ),
      ),
    ),
  );
  return $items;
}
```

Adding properties and fields

```
/**
 * Implements hook_entity_property_info().
 */
function bookstore_entity_property_info() {

  $items['book']['properties'] = array(
    'title' => array(
      'label' => t('Title'),
      'type' => 'text',
    ),
    'status' => array(
      'label' => t('Status'),
      'type' => 'boolean',
    ),
  );
  return $items;
}
```

Adding properties and fields

The screenshot shows the Drupal 7.31 admin interface for managing fields on the Books entity. The browser address bar shows the URL: localhost/bookstore/drupal-7.31/admin/book/manage/fields. The navigation menu includes Content, Structure, Appearance, People, Modules, Authors, Books, Chapters, Configuration, and Reports. The user is logged in as admin.

The main content area is titled "Books" and has three tabs: LIST, MANAGE FIELDS (active), and MANAGE DISPLAY. There is a "Show row weights" link on the right.

LABEL	MACHINE NAME	FIELD TYPE	WIDGET	OPERATIONS
✚ Title	field_title	Text	Text field	edit delete
✚ Author	field_author	Entity Reference	Select list	edit delete
✚ Chapters	field_chapters	Embedded entity	Embedded entity widget	edit delete
✚ Add new field				
<input type="text"/>		- Select a field type -	- Select a widget -	
Label		Type of data to store.	Form element to edit the data.	
✚ Add existing field				
<input type="text"/>	- Select an existing field -		- Select a widget -	
Label	Field to share		Form element to edit the data.	

Save

Creating and saving entities

```
function book_form($form, $form_state, $entity = NULL) {  
  
  if (!$entity) {  
    $entity = entity_create('book', array());  
  }  
  field_attach_form('book', $entity, $form, $form_state);  
  $form['#entity'] = $entity;  
  
  $form['submit'] = array(  
    '#type' => 'submit',  
    '#value' => 'Save Book',  
  );  
  return $form;  
}
```

Creating and saving entities

```
function book_form_validate(&$form, &$form_state) {  
  
    $entity = $form['#entity'];  
    field_attach_form_validate('book', $entity, $form,  
$form_state);  
  
}  
  
function book_form_submit(&$form, &$form_state) {  
  
    $entity = $form['#entity'];  
    entity_form_submit_build_entity('book', $entity, $form,  
$form_state);  
    entity_save('book', $entity);  
  
}
```

Querying

- Works with EntityFieldQuery class
- Also supports passing conditions to entity_load

Querying

```
// Creates a new EFQ
$query = new EntityFieldQuery;
// Limits the query to books
$query->entityCondition('entity_type', 'book');
// Finds titles
$query->fieldCondition('field_title', 'value', 'Bloodsucking
Fiends');

// Run the query
$result = $query->execute();
// Load results, if any
if (isset($result['book'])) {
    $books = entity_load('book', array_keys($result['book']));
}
```


Embedded entities

- Unique to Mongo Entity
- Works with both fields and properties

Embedded entity properties

```
/**
 * Implements hook_entity_info().
 */
function bookstore_entity_info() {
  $items['chapter'] = array(
    'label' => t('Chapter'),
    'entity class' => 'MongoEntity',
    'controller class' => 'MongoEmbeddedEntityController',
    ...
  );
}
```

Embedded entity properties

```
/**
 * Implements hook_entity_property_info().
 */
function bookstore_entity_property_info() {
  $items['book']['properties'] = array(
    'chapters' => array(
      'type` => 'list<embedded_entity:chapter>'
      'description' => t('A list of chapters'),
    ),
  );
  return $items;
}
```

Embedded entity properties

```
// Create the book
$book = entity_create('book', array());
$book->field_title[LANGUAGE_NONE][0]['value'] = 'A Dirty Job';

// Create the chapter
$chapter = entity_create('chapter', array());
$chapter->field_num_pages[LANGUAGE_NONE][0]['value'] = 54;

// Add the chapter to the book
$book->chapters[] = $chapter;

// Save the book
entity_save('book', $book);
```

Embedded entity fields

The screenshot shows the 'Add book' form in a Drupal 7.31 administration interface. The browser address bar is `localhost/bookstore/drupal-7.31/admin/book/add`. The navigation menu includes 'Content', 'Structure', 'Appearance', 'People', 'Modules', 'Authors', 'Books', 'Chapters', 'Configuration', and 'Reports'. The user is logged in as 'admin'.

The form has the following fields and elements:

- Title ***: A text input field.
- Author**: A dropdown menu currently set to '- None -'.
- CHAPTERS**: A section containing an embedded entity field. It includes:
 - A plus sign icon (+) to the left of the field.
 - Title ***: A text input field for each chapter.
 - Number of pages**: A text input field for each chapter.
- Show row weights**: A link to the right of the chapters section.
- Add another item**: A button below the chapters section.
- Save Book**: A button at the bottom of the form.

Querying with embedded entities

- Use field names as array to find entities by embedded entity conditions
- Use `MongoEmbeddedEntityFieldQuery` `parentFieldCondition` and `parentPropertyCondition` methods for finding embedded entities by parent

Querying entities with embedded entities

```
$query = new EntityFieldQuery;  
$query->entityCondition('entity_type', 'book');  
$query->fieldCondition(array('field_chapters',  
'field_num_pages'), 'value', 20, '>');
```


Querying embedded entities by parent

```
$query = new MongoEmbeddedEntityFieldQuery;  
$query->entityCondition('entity_type', 'chapter');  
$query->parentPropertyCondition('field_title', 'The Stupidest  
Angel');  
$query->fieldCondition('field_num_pages', 'value', 20, '>');
```

MongoDB under the hood

- Document structure
- Uses integer IDs

Document structure

```
drupal-7.31 — mongo — 119x36
db.book.find().pretty()
{
  "_id" : 1,
  "field_title" : {
    "value" : "A Dirty Job"
  },
  "field_chapters" : [
    {
      "_id" : 1,
      "_bundle" : "chapter",
      "field_title" : {
        "value" : "Chapter 1"
      },
      "field_num_pages" : {
        "value" : 100
      }
    },
    {
      "_id" : 2,
      "_bundle" : "chapter",
      "field_title" : {
        "value" : "Chapter 2"
      },
      "field_num_pages" : {
        "value" : 200
      }
    }
  ],
  "field_author" : {
    "target_id" : 1
  },
  "_revision" : 10
}
```

Index collections for performance

```
function bookstore_entity_info() {
  $items['book'] = array(
    ...
    'indexes' => array(
      array('field_title.value' => 1),
      array(array('isbn' => 1), array('unique' => TRUE)),
    ),
    ...
  );
  return $items;
}
```

http://drupal.org/project/mongo_entity