

Sinatra

Tobias Schmid & Andreas Knöpfle



Persistenz

- MongoDB
 - Dokumente statt Tabellen
 - Zugriff mit Hilfe von MongoMapper
- Daten eines Ereignisses sind jeweils in einem Dokument

```
# Beispiel: Teilnehmer zum Event hinzufügen
```

```
@event = Event.find(id)
if @event.open
  @event.push({:members => current_user._id})
end
```

- API ähnelt ActiveRecord API aus RoR

MongoDB Document "Event"

```
{
  "_id":ObjectId("51114d858148094277000137"),
  "mm_user_id":ObjectId("510bed1181480912f3000022"),
  "deleted":false,
  "name":"Grill und Gartenfest",
  "location":"Dort",
  "open":false,
  "creator_pays":false,
  "add_bill":false,
  "date":  ISODate("2012-01-10T00:00:00  Z"),
  "members":[
    ObjectId("510bed1181480912f3000022"),
    ObjectId("51047efc8148090c35000001")
  ],
  "created_at":  ISODate("2013-02-05T18:20:53.887  Z"),
  "updated_at":  ISODate("2013-02-05T18:32:05.127  Z"),
  "items":[
    {
      "_id":ObjectId("51114d988148094277000157"),
      "bill_id":null,
      "mm_user_id":ObjectId("510bed1181480912f3000022"),
      "name":"Kartoffelsalat",
      "amount":15
    },
    ...
  ]
}
```

Formulare, Validierung

Formulare:

- **Gem:** sinatra-formhelpers
- Helper für `<input>` - Tags

Validierung:

- Mongomapper implementiert Validierungshelper von ActiveRecord (Rails)
- Eigene Validatoren sind leicht implementierbar

Allerdings: Keine Helper für die Darstellung von Validierungsfehlern in Formularen

Templates

- Templatesprache **HAML**
- Quasi-Standard für Sinatra

```
%html
  %head
    %meta{:charset => "utf-8"}
    %title Splitbill
    %link{:href => "/css/bootstrap.css", :rel => "stylesheet"}
    %link{:href => "/css/splitbill.css", :rel => "stylesheet"}

  %body
    .navbar.navbar-inverse
      .navbar-inner
        .container
          %a.brand{:href => '/'} Splitbill
# ...
```



I18N

Gem: sinatra-r18n

Im Template:

```
t.event.name
```

In de.yml:

```
event:  
  name: Name
```

Authentifizierungs-Plugin musste von Hand internationalisiert werden!

Routing

Sinatra's Sweet Spot

```
post '/events/:id/add' do |id|  
  # ...  
end
```

```
post '/events/:id/:item_id/edit' do |id,item_id|  
  # ...  
end
```

Sicherheitsmechanismen

- **XSS:**

HTML-Escape über Helper möglich

- **CSRF:**

Gem: rack_csrf

- **SQL-Injection**

Durch MongoDB kein Problem

BSON-Objekte statt SQL-Statements

- **Session Hijacking**

Sessions können ebenfalls in der MongoDB abgelegt werden

Scaffolding & Bootstrapping

- Scaffolding nicht möglich
- Bootstrapping

```
# app.rb  
require 'sinatra'
```

```
# Routes
```

```
-- > ruby app.rb
```

Use-Case Statistiken

Use-Case	Dauer	Verwendete Dokumentation (Nützlichkeit,Häufigkeit)	Schwierigkeiten
Oberflächensprache ändern	1h + Übersetzen	Sinatra (100%, wenig) R18n- gem (100% , häufig)	-
Login	3h	Ruby (100%, 1 mal) Sinatra (100%, 1 mal) sinatra-authentication (70%, häufig) + <i>Codereading</i> stackoverflow/sonstiges (70%, häufig) MongoMapper (100%, wenig)	Authentifizierungs-Gem nicht internationalisiert --> Fork auf github
Registration	1h	-	-

Use-Case Statistiken

Use-Case	Dauer	Verwendete Dokumentation (Nützlichkeit,Häufigkeit)	Schwierigkeiten
Erstellung eines neuen Ereignis	2h	MongoMapper (100%, 5 mal) Sinatra (70%, 1 mal)	-
Editieren eines Ereignis	1h	-	Nur möglich wenn Event noch nicht abgerechnet ist.
Entfernen eines Ereignis	1h	-	-

Use-Case Statistiken

Use-Case	Dauer	Verwendete Dokumentation (Nützlichkeit,Häufigkeit)	Schwierigkeiten
Posten zu Ereignis hinzufügen	5h	Sinatra (100%,2 mal) joint-gem (30%, 2 mal) stackoverflow/ etc. (80%, häufig) Beispielprojekt: MongoMapper-joint (100%, häufig) MongoMapper (30%, 3 mal)	Hochladen von Dateien: joint gem - nicht leicht zu finden - Dokumentation schlecht - Keine Alternative <i>GridFS Fehler mit Index</i>
Kostenpunkt bzw. Posten bearbeiten/löschen	2h	Sinatra (100%, wenig) MongoMapper (70 %, häufig)	Rechte: MongoMapperObjekte müssen über ID verglichen werden.
Rechnungen einsehen	1h	Ruby (100%, 1 mal) joint-gem (30%, 3 mal) stackoverflow/sonstiges (70%, häufig)	Bilder aus der MongoDB laden und als Content rendern

Use-Case Statistiken

Use-Case	Dauer	Verwendete Dokumentation (Nützlichkeit,Häufigkeit)	Schwierigkeiten
Benutzer tritt Ereignis bei	1h	MongoMapper(100%, 1 mal)	-
Benutzer kommentiert ein Ereignis	2h	-	-
Aktueller Kostenstand eines Ereignisses	0,5 h	MongoMapper(100%, 1 mal)	-

Use-Case Statistiken

Use-Case		Dauer	Verwendete Dokumentation (Nützlichkeit,Häufigkeit)	Schwierigkeiten
~2h	Zuzahlende Beträge eines Ereignisses einsehen	< 1h	-	-
	Meine zu erhaltenden Beträge einsehen	<1h	-	-
	Meine zu zahlenden Beträge einsehen	<1h	-	-
	Betrag als eingegangen markieren	<1h	-	-
~1,5h	Betrag als beglichen markieren	1h	-	-

Use-Case Statistiken

Use-Case	Dauer	Verwendete Dokumentation (Nützlichkeit,Häufigkeit)	Schwierigkeiten
Gesamtstatistik anzeigen	1h	-	-
Ereignisverlauf ansehen	1h	Ruby (100%, 1)	MongoMapperAPI mit push & save reagierte nicht so wie erwartet beim Ereignis-Array.
Benutzereingaben werden validiert	4 h	MongoMapper(100%, 10 mal) Ruby (70%, 1mal)	Validatoren für Rechnungspflicht schwer umzusetzen! Anzeige für fehlerhafte Benutzereingaben muss von Hand geschrieben werden

Gesamtstatistik

Alle Use-Cases zusammen: 30 Stunden
+ 2-3 Stunden Sinatra Einarbeitung
+ 1 min Bootstrapping :)

Verwendete Dokumentationen:

- Sinatra
- **MongoMapper-gem**
- joint-gem
- sinatra-authentication-gem
- sinatra-formhelpers-gem
- Ruby
- stackoverflow + Foren
- Beispielprojekte

Fazit

Fazit Zwischenpräsentation:

- Viele Ruby-Gems nicht nutzbar, da abhängig vom Rails-Stack
- Weniger Hilfsmittel als in Full-Stack-Frameworks verfügbar
- Sinatra kommt eher für spezielle Applikationen wie Video-Streaming oder als Backend für Java-Script Oberfläche zum Einsatz

Fazit zusätzlich:

- Sinatra sehr abhängig vom verwendeten Persistenz-gem
 - Beeinflusst alle weiteren Entscheidungen und nutzbare Tools
- MongoMapper leider noch nicht ganz an ActiveRecord-Rails
 - Nicht-relationale Datenbank vereinfacht allerdings an anderen Stellen die Entwicklung

Lessons Learned

- **HAML sehr nützlich**
 - Vereinfacht Arbeit an HTML-Templates stark
 - Jederzeit wiederverwenden
- **MongoDB**
 - Kann eine Relationale Datenbank in vielen Fällen ersetzen
 - Allerdings nicht immer !
 - Ergänzung zu relationalen Datenbanken
- **Sinatra**
 - Für kleinere Projekte und spezielle Aufgaben geeignet
 - Für große Webplattformen eher anderes Framework nutzen

Quellen

Bilder:

Logo MongoDB: <http://www.mongodb.org/>

Logo Sinatra: http://a0.twimg.com/profile_images/80388980/logo.png

Logo HAML: http://upload.wikimedia.org/wikipedia/commons/3/3b/Haml_1-5_logo.png