2012-09-15 Enju開発WS演習メモ

いろんなxml形式やjson形式など、
様々な形式の結果を返せるようにカスタマイズをやります。

http://enju2.slis.keio.ac.jp/manifestations/26028

↑の結果について、

xml形式、json形式などを返せます（他にもいろいろあります）。
http://enju2.slis.keio.ac.jp/manifestations/26028.xml
http://enju2.slis.keio.ac.jp/manifestations/26028.json

URLに、 .xml などとつけると返せるようになっています。

検索結果一覧でも同様です。
以下はlibraryで検索した結果です。

http://enju2.slis.keio.ac.jp/manifestations?utf8=%E2%9C%93&query=library

http://enju2.slis.keio.ac.jp/manifestations.xml?utf8=%E2%9C%93&query=library
http://enju2.slis.keio.ac.jp/manifestations.json?utf8=%E2%9C%93&query=library

いきなりEnjuでやると難しいので、
まずは、railsの小さなアプリケーションをつくって
そこで演習をします。

"enjuapi" という小さなrailsのアプリケーションを作ることにします。

```
$ rails new enjuapi
```

```
Your bundle is complete! Use `bundle show [gemname]` to see where a bundled gem is installed.
```

などとメッセージが出ればOK

```
$ cd enjuapi
```

(今回は4人一度にやるので、全員同じデフォルトのポートでやると競合するので、
一人づつそれぞれ別のポート番号を設定し、その番号を使うこととする。
ちなみに、私のポートは3004)

Webサーバーを起動してみる

```
$ rails s -p 3004
```

（もし、別のポート番号を設定しないなら、 "$ rails s" とするだけでよい）

ブラウザで
http://192.168.11.14:3004/ でアクセスしてみる

エラーメッセージが表示されず、なにかそれっぽいものが表示されたらOK

で、いったん、Ctl-C でWebサーバーをとめる

とりあえず、以下のようなものでつくってみることとする

属性名	データ型	内容の説明
original_title	text	タイトル
isbn	string	ISBN
date_of_publication	datetime	出版日（時刻）
url	text	URL

モデル名は"Manifestation"（体現形）とします。ひな形を作成します。

```
$ rails g scaffold Manifestation original_title:text isbn:string date_of_publication:datetime url:text
```

テーブルを作成します。

```
$ rake db:migrate
```

Webサーバを起動します。

```
$ rails s -p 3004
```

http://192.168.11.14:3004/manifestations/ にアクセスして、

New manifestation のリンクをたどった先で、3件くらいレコードを入力しておく

http://192.168.11.14:3004/manifestations.json にアクセスすると
json形式の検索結果一覧が表示される

個々の検索結果詳細情報でも
http://192.168.11.14:3004/manifestations/3.json にアクセスすると
json形式で表示される

で、いったん、Ctl-C でWebサーバーをとめる

```
$ ls -l app/views/manifestations/
total 40
-rw-r--r--+ 1 yuka  staff  864 Sep 15 13:18 _form.html.erb
-rw-r--r--+ 1 yuka  staff  137 Sep 15 13:18 edit.html.erb
-rw-r--r--+ 1 yuka  staff  754 Sep 15 13:18 index.html.erb
-rw-r--r--+ 1 yuka  staff   93 Sep 15 13:18 new.html.erb
-rw-r--r--+ 1 yuka  staff  407 Sep 15 13:18 show.html.erb
```

こんなかんじでファイルがあるが、jsonはないよね？
でもなぜ、jsonができるか？

じゃあ、controller をひらいてみます
```
$ vi app/controllers/manifestations_controller.rb
```

```
----

  def index
    @manifestations = Manifestation.all

    respond_to do |format|
      format.html # index.html.erb
      format.json { render json: @manifestations }
    end
  end
----
```
↑ここがポイント

ここで、html くれとわれたら、なにもかいてないので、
app/view/manifestation/index.html.erb のテンプレートをみにいく
（index は　def index だから）

ここで、jsonくれといわれたら、
{ render json: @manifestations }
とかいてあるので、内部で to_jsonというメソッドを読んでいる。

render ってのはいろんな書き方ができる

```
      format.json { render json: @manifestations }
      format.json { render :nothing => true } # に書き換えてみる
```

http://192.168.11.14:3004/manifestations.json にアクセスして
なにもでなかったら成功

```
      format.json { render json: @manifestations }
      format.json { render :text => 'hogehoge' } # に書き換えてみる
```

http://192.168.11.14:3004/manifestations.json にアクセスして
hogehogeと出たら成功

こういうふうに、いろいろRubyのプログラムを書けばそのまま表示される。

```
      format.json { render json: @manifestations } #元に戻す
```

http://192.168.11.14:3004/manifestations.json にアクセスして
ちゃんとjson形式で出たら成功

（今は、デベロッパー環境でWebサーバーを起動(rails s) しているので、
いちいち起動しなおさなくてもOK)

じゃあ、XMLも出るようにしましょう。

```
$ vi app/controllers/manifestations_controller.rb
```

```
----

  def index
    @manifestations = Manifestation.all

    respond_to do |format|
      format.html # index.html.erb
      format.json { render json: @manifestations }
      format.xml { render xml: @manifestations }
    end
  end
----
```

赤字の部分を追記します。

http://192.168.11.14:3004/manifestations.xml にアクセスします。
XMLが表示されたら成功

名前空間の指定などはないが、
とりあえず、なんでもいいからXMLで出せればいいやという場合は、
これでOK
railsがやってくれているのでOK。

じゃあ、デフォルト出力じゃやだーという場合
XMLをカスタマイズする場合は、テンプレートをつくらないといけない

どこにどんな名前でテンプレートをつくるかをきめないといけない。
今回は、manifestation で index です。
index メソッドだったから、 HTMLのときは、index.html.erbだった。
なら、XMLのときは？ index.xml.erb

```
$ vi app/views/manifestations/index.xml.erb
----
<book>
<created-at type="datetime">2012-09-15T04:23:38Z</created-at>
<date-of-publication type="datetime">2012-09-15T04:23:00Z</date-of-publication>
<id type="integer">2</id>
<isbn>9784004313800</isbn>
<original-title>チャーリーとチョコレート工場</original-title>
<updated-at type="datetime">2012-09-15T04:23:38Z</updated-at>
<url>http://sample.com/bababa.html</url>
</book>
----
```

上のような感じで書いてとりあえずてきとうに書いて保存しておく

```
$ vi app/controllers/manifestations_controller.rb
----
```

```
  def index
    @manifestations = Manifestation.all

    respond_to do |format|
      format.html # index.html.erb
      format.json { render json: @manifestations }
#       format.xml { render xml: @manifestations }
      format.xml
    end
  end
```

赤字の部分を追記します。

(TIPS: EUCからUTF-8に変える方法 $iconv -f EUC-JP -t UTF-8 inputfile > outputfile)

http://192.168.11.14:3004/manifestations.xml にアクセスします。
そうすると、さっきかいたとりあえずてきとうにかいたものが表示される。

```
$ vi app/views/manifestations/index.xml.erb
```

```
<book>
<created-at type="datetime">2012-09-15T04:23:38Z</created-at>
<date-of-publication type="datetime">2012-09-15T04:23:00Z</date-of-publication>
<id type="integer">2</id>
<isbn>9784004313800</isbn>
<original-title>チャーリーとチョコレート工場</original-title>
<updated-at type="datetime">2012-09-15T04:23:38Z</updated-at>
<url>http://sample.com/bababa.html</url>
<note><%= Time.now %></note>
</book>
```

赤字の部分を追記して、Rubyのプログラムをかいてみます。
 = をかくのをわすれないように。=かかないと表示されません！

http://192.168.11.14:3004/manifestations.xml にアクセスします。
そうすると、

```
<book>
<created-at type="datetime">2012-09-15T04:23:38Z</created-at>
<date-of-publication type="datetime">2012-09-15T04:23:00Z</date-of-publication>
<id type="integer">2</id>
<isbn>9784004313800</isbn>
<original-title>チャーリーとチョコレート工場</original-title>
<updated-at type="datetime">2012-09-15T04:23:38Z</updated-at>
<url>http://sample.com/bababa.html</url>
<note>2012-09-15 14:15:53 +0900</note>
</book>
```

こんなかんじで表示現在時刻がUTCで表示されました！

つぎのように書き換えてみましょう。

```
$ vi app/views/manifestations/index.xml.erb
```

```
<books>
  <% @manifestations.each do |manifestation| %>
    <book>
      <title><%= manifestation.original_title %></title>
    </book>
  <% end %>
</books>
```

こんどは　<% @manifestations.each do |manifestation| %>　などは　= がないことに注意！
表示される部分以外は = を書かないこと。

http://192.168.11.14:3004/manifestations.xml にアクセスします。
そうすると、以下のようにタイトルが表示されると成功！
each do をつかっているので、複数のすべての本のタイトルが表示されます。

```
<books>
<book>
<title>ハリーポッターと賢者の石</title>
</book>
<book>
<title>チャーリーとチョコレート工場</title>
</book>
<book>
<title>若草物語</title>
</book>
</books>
```

```
$ vi app/views/manifestations/index.xml.erb
<books>
  <% @manifestations.each do |manifestation| %>
    <book>
      <title><%= manifestation.original_title %></title>
      <isbn><%= manifestation.isbn %></isbn>
    </book>
  <% end %>
</books>
```

赤字を追加すると

http://192.168.11.14:3004/manifestations.xml にアクセスします。
そうすると、以下のようにisbnが表示されると成功！

```
<books>
<book>
<title>ハリーポッターと賢者の石</title>
<isbn>9784004313762</isbn>
</book>
<book>
<title>チャーリーとチョコレート工場</title>
<isbn>9784004313800</isbn>
</book>
<book>
<title>若草物語</title>
<isbn>978400430000</isbn>
</book>
</books>
```

IF文も使える

```
$ vi app/views/manifestations/index.xml.erb

<books>
  <% @manifestations.each do |manifestation| %>
    <book>
      <title><%= manifestation.original_title %></title>
      <% if manifestation.isbn? %>
          <isbn><%= manifestation.isbn %></isbn>
      <% else %>
          <isbn></isbn>
      <% end %>
    </book>
  <% end %>
</books>
```

isbnがnilか空でなければ、表示する。

jsonファイルを作り変えたい場合は。

```
$ vi app/views/manifestations/index.json.erb
---
[
  {
    "title": "aaaa",
    "date_of_publication": "2012-01-01",
    "note": "<%= Time.zone.now %>"
  },
  {
    "title": "bbb",
    "date_of_publication": "2012-02-03",
    "note": "<%= Time.zone.now %>"
  }
]
---
```
とりあえず、てきとうなjson記述をかいてみる
```
$vi app/controllers/manifestations_controller.rb

#      format.json { render json: @manifestations }
       format.json
```
赤字のようにする。（{ rennder json... を削除する）

http://192.168.11.14:3004/manifestations.json
にアクセスします。
そうすると、さっきてきとうにかいたjsonが表示されると成功！

jsonが正しいかどうかをチェックするには、

```
$ rails c
(rails コンソールの起動)

irb(main):005:0> require "open-uri"
=> true
irb(main):006:0> JSON.parse(open('http://192.168.11.14:3004
/manifestations.json').read)
=> [{"title"=>"aaaa", "date_of_publication"=>"2012-01-01", "note"=>"2012-09-15
05:51:23 UTC"}, {"title"=>"bbb", "date_of_publication"=>"2012-02-03",
"note"=>"2012-09-15 05:51:23 UTC"}]
```

↑のように、エラーがでなければ、ただしいjsonがかけている。

以下のように、rubyプログラムとしてもかける。

```
$  vi app/views/manifestations/index.json.erb
[
  <% @manifestations.each do |manifestation| %>
  {
    "title": "<%= manifestation.original_title %>",
    "date_of_publication": "<%= manifestation.date_of_publication %>",
    "note": "<%= Time.zone.now %>"
  },
  <% end %>
  {}
]
```

http://192.168.11.14:3004/manifestations.json
にアクセスします。
そうすると、以下のようにjsonが表示されると成功！

```
[
  {
    "title": "ハリーポッターと賢者の石",
    "date_of_publication": "2012-09-15 04:21:00 UTC",
```

```
    "note": "2012-09-15 06:07:16 UTC"
  },
  {
    "title": "チャーリーとチョコレート工場",
    "date_of_publication": "2012-09-15 04:23:00 UTC",
    "note": "2012-09-15 06:07:16 UTC"
  },
  {
    "title": "若草物語",
    "date_of_publication": "2012-09-15 04:23:00 UTC",
    "note": "2012-09-15 06:07:16 UTC"
  },
  {}
]
```

カスタムなAPIをやるにはどうやったらいいか

一行で引用できる形で表示されたらうれしいな
TSVでダウンロードできるとうれしいな

endnote とか、文献管理用のフォーマットでだしたい
昔ながらのMARCでだしたい
など。

今まで、検索結果一覧のものをカスタマイズする例だったので、index.xxx.erb だったけど
ちなみに、詳細結果のものをやりたいばあいは、show.xxx.erb

演習：じゃあ、なにかテーマをきめて、やりましょう。

江草：SIST02風に一行で表示させる
高久：ESIDOC形式で出す

まず、フォーマットを定義します

```
$ vi config/initializers/mime_types.rb

# Mime::Type.register "text/richtext", :rtf
```

↑は例。
.rtf とあったら、text/richtext ですよ　と定義になる

```
# Mime::Type.register_alias "text/html", :iphone
```

モバイル専用のテンプレートを作りたい場合は、↑のような記述も。

```
$  vi ~/.rbenv/versions/1.9.3-p194/lib/ruby/gems/1.9.1/gems/actionpack-3.2.8/lib
/action_dispatch/http/mime_types.rb
```

↑にすでに登録済みのmime_typeがある。ここにない場合は、

↓にmimeタイプを追加する。

```
$ vi config/initializers/mime_types.rb
```

Type.register_alias は、すでに登録済みのmimeタイプのエイリアスを保存する。

例えば、

```
Mime::Type.register_alias "text/plain", :sist
```

というのを追加する。

江草：SIST02風に一行で表示させるをやる場合は、

http://192.168.11.14:3004/manifestations/3.text
にアクセスすると

若草物語.ISBN:978400430000.<URL=http://bababa.zzz.jp/zzz.html>

と表示されるようにカスタマイズすることとする。

★手順1. Mimeの確認、なければ登録

$ vi ~/.rbenv/versions/1.9.3-p194/lib/ruby/gems/1.9.1/gems/actionpack-3.2.8/lib/action_dispatch/http/mime_types.rb

↑をみて、登録済みかどうか確認する。

Mime::Type.register "text/plain", :text, [], %w(txt)

とあり、.text はすでに登録されているので、

$ vi config/initializers/mime_types.rb

は、特に修正しない。
たとえば、
http://192.168.11.14:3004/manifestations/3.sist
としたいのであれば、

Mime::Type.register_alias "text/plain", :sist

を追加し、Webサーバーの再起動をする（Ctrl+c, $ rails s -p 3004）

★手順2. まずは、show.text.erb のダミーをつくる

$ vi app/views/manifestations/show.text.erb

bababa.bbbb.zzzz.<URL http://zzz.jjj.pp>

↑のように適当にまずはかいておく

★手順3. manifestations_controller.rb で、.text のときの指定を書く。

http://192.168.11.14:3004/manifestations/3.text
にアクセスするとダミーが表示されるようにする

$ vi app/controllers/manifestations_controller.rb

```
----
  # GET /manifestations/1
  # GET /manifestations/1.json
  def show
    @manifestation = Manifestation.find(params[:id])

    respond_to do |format|
      format.html # show.html.erb
      format.json { render json: @manifestation }
       format.text
    end
  end
----
```

赤字を追加する

★手順4.
http://192.168.11.14:3004/manifestations/3.text
にアクセスするとダミーが表示されることを確認

★手順5. Rubyプログラムを使ってSIST02風に表示されるようにする

とりあえず、以下のように表示されるようにする

若草物語.ISBN:978400430000.<URL=http://bababa.zzz.jp/zzz.html>

```
$ vi app/views/manifestations/show.text.erb
```

```
<%= @manifestation.original_title %>.ISBN: <%= @manifestation.isbn %>.<URL <%=
@manifestation.url %>>
```

@manifestation = Manifestation.find(params[:id])

となっているので、 @manifestation とする。

★手順6.
http://192.168.11.14:3004/manifestations/3.text
にアクセスすると

若草物語.ISBN:978400430000.<URL=http://bababa.zzz.jp/zzz.html>

と表示されると成功

ここからは、Enjuにこれを移す作業をする。

注：

私は依然のワークショップですでに、 enju_leaf というフォルダを作成していたので、
そこに先ほど作ったデータをコピーして作業した。
そうでない場合は、以下のコマンドを実行して、
enju_leafのソースコードをダウンロードする必要がある

```
$ gem install bundler
$ git clone git://github.com/nabeta/enju_leaf.git
```

★手順7. show.text.erb を Enju にコピーする

```
$ cd ~/enju_leaf
$ cp ../enjuapi/app/views/manifestations/show.text.erb app/views/manifestations/.
```

★手順8. url は Enju では access_adress なので修正する

```
$ vi app/views/manifestations/show.text.erb
```

```
<%= @manifestation.original_title %>.ISBN: <%= @manifestation.isbn %>.<URL <%=
@manifestation.access_address%>>
```

urlからaccess_address に修正する（赤字の部分）

★手順9. manifestations_controller.rb で、.text のときの指定を書く。

http://192.168.11.14:3004/manifestations/3.text
にアクセスすると表示されるようにする

```
$ vi app/controllers/manifestations_controller.rb

----
・・・・省略・・・
  # GET /manifestations/1
  # GET /manifestations/1.json
  def show
    if params[:isbn]
      @manifestation = Manifestation.find_by_isbn(params[:isbn])
・・・・省略・・・
    respond_to do |format|
      format.html # show.html.erb
      format.mobile
      format.text
・・・・省略・・・
    end
  end
----
```
赤字を追加する

★手順10.Enju の検索サーバーの起動

```
$ rake sunspot:solr:start RAILS_ENV=production
```

（すでにサーバが起動している場合もある。その場合は、そのまますすめる）

★手順11. Webサーバと非同期処理用サーバを起動します

```
$ rake resque:work RAILS_ENV=production QUEUE=* BACKGROUND=yes
$ rails server -e production -p 3004
```

★手順
http://192.168.11.14:3004/manifestations/1.text
にアクセスすると表示されるようにする

できた！

次は、もっとデータ項目を追加してよりSIST02らしくする

★どんなデータ項目名とデータ型があるか調べる。

まずは、コンソールを起動する

```
$ rails c production
```

```
irb(main):001:0> Manifestation
```

とすすると、以下のように項目名と方がでてくる。

```
=> Manifestation(id: integer, original_title: text, title_alternative: text,
title_transcription: text, classification_number: string, manifestation_identifier:
string, date_of_publication: datetime, date_copyrighted: datetime, created_at:
datetime, updated_at: datetime, deleted_at: datetime, access_address: string,
language_id: integer, carrier_type_id: integer, extent_id: integer, start_page:
integer, end_page: integer, height: integer, width: integer, depth: integer, isbn:
string, isbn10: string, wrong_isbn: string, nbn: string, lccn: string, oclc_number:
string, issn: string, price: integer, fulltext: text, volume_number_string: string,
issue_number_string: string, serial_number_string: string, edition: integer, note:
text, repository_content: boolean, lock_version: integer, required_role_id: integer,
state: string, required_score: integer, frequency_id: integer, subscription_master:
boolean, attachment_file_name: string, attachment_content_type: string,
attachment_file_size: integer, attachment_updated_at: datetime,
```

title_alternative_transcription: text, description: text, abstract: text, available_at: datetime, valid_until: datetime, date_submitted: datetime, date_accepted: datetime, date_caputured: datetime, ndl_bib_id: string, pub_date: string, edition_string: string, volume_number: integer, issue_number: integer, serial_number: integer, ndc: string, content_type_id: integer, year_of_publication: integer, attachment_fingerprint: string, attachment_meta: text, month_of_publication: integer, online_isbn: string, review: text)

ほかには、

http://192.168.11.14:3004/manifestation.josn
などでもいいかも？

ところで、

http://192.168.11.14:3004/manifestations/1.text

は以下のような書き方もできる

http://192.168.11.14:3004/manifestations/1?format=text