

## フレームワークがリクエストを処理する流れ

この文書では Ruby on Rails が担当するフレームワークでのリクエストを処理する流れについて記している。

以下の記述は Rails 1.2.6 を対象にしている。

### 基本

Rails ではユーザーからの HTTP リクエストを解釈して controller に渡し、view に基づいて出力された結果を Web サーバからユーザーにレスポンスとして返す。この処理を受け持つのは ActionPack [1]と呼ばれるライブラリであり、ActionController や ActionView をまとめている。

アプリケーションをフレームワーク上に作成する場合には、これらの流れの一部を実装するだけで済む; ちょうど controller と view にあたる処理である。

したがって以下の内容は

- フレームワークの処理の流れを変更する場合や、
  - 複雑な状況でのデバッグを行う場合
- に特に関係してくる。

### Web サーバがリクエストを受け付ける

Rails は Web サーバとして動作するために、標準的には

- WEBrick
- Mongrel

のいずれかを用いている。

これらの Web サーバはリクエストを request オブジェクトに変換する。

request オブジェクトから

- 環境変数

の情報にアクセスできる。

またこの段階で既にレスポンスのオブジェクト response が作成されており、以降の処理でその内容が設定される。

実装としては ActionController::CgiRequest が ActionController::AbstractRequest を継承しており、CGI 経由でリクエストを処理する場合に利用される。リクエストごとにこのインスタンスが作成され、@session などのインスタンス変数を持つ。

また、ActionController::CgiResponse が ActionController::AbstractResponse を継承しており、メソッド CgiResponse#out で出力を行う。

### リクエストをルーティングする

その際に参照されるルーティングに関する設定は

config/routes.rb  
にある。

ルーティングの処理はモジュール ActionController::Routing で実装されている。

### controller を呼び出す

ルーティングの結果 request が変換されると同時に、適切な controller の public メソッドが呼び出される。これらメソッドはアクションをなす。

各アクションは

1. view のレンダリングを明示的に指示する
2. view のレンダリングを暗黙に指示する
3. リダイレクトする

のいずれかを行う。

3 の場合は "302 Moved" HTTP リクエストを生成してユーザー側にリダイレクトを促す。これによって実質的に別のアクションを呼び出すことになる。

またこの段階で  
:params  
リクエストのパラメータから抽出される情報のオブジェクト  
:session  
セッション情報のオブジェクト  
が利用できる。

controller の呼び出しはメソッド ActionController::Base#process で定義されている。  
既定の場合には ActionController::Base#perform\_action へと制御が移る。  
上にあげられている変数は ActionController::Base の中でattr\_internal によって  
定義されている。

## view に基づいて出力する

view はテンプレートと helper によって出力される。  
request の生存期間はここまでである。

出力は ActionController::Base#render で行っている。

## レスポンスを届ける

Web サーバから出力がレスポンスとして返される; このとき controller が指示した  
content type などが利用される。

レスポンスの情報は ActionController::Base#response にまとめられている。

## リクエスト/レスポンス単位の処理以外

基本的に上のような流れで MVC を構成するが、以下のような例外的な処理が行われる余地がある。

## flash

Rails には flash という概念があり、同一セッションの次のアクセスのみで  
利用できる情報をそれに格納できる。

このためには、controller の中で flash という変数に対して  
flash[:message] = "some of messages."  
という代入を行うだけでよい。  
この値は当のリクエストに対しては利用できず、次のリクエストに限りアクセスできる。

フレームワークではこれを主に  
- リダイレクト後のメッセージ表示  
に用いている。

実装は ActionController::Flash であり、session["flash"] に格納されている。

## 参考

[1] ActionPack  
<http://api.rubyonrails.org/files/vendor/rails/actionpack/README.html>