

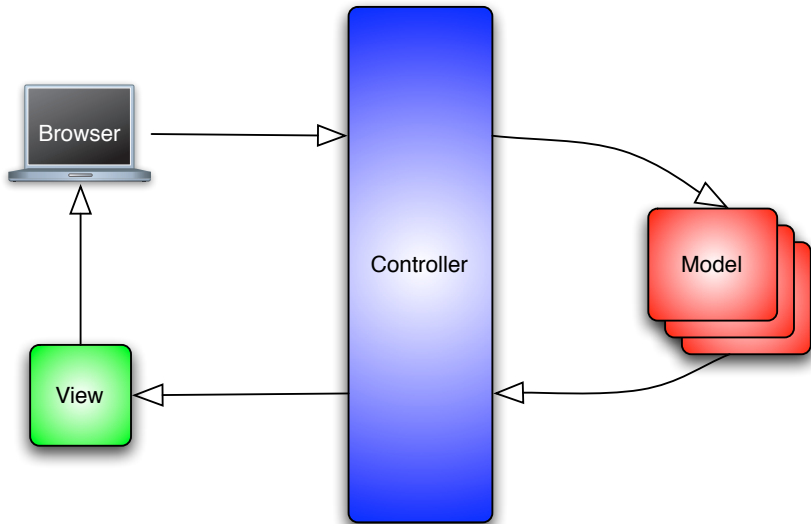
## Lezione 6: Validazione, Layout, Scaffold

Matteo Vaccari

<http://matteo.vaccari.name/>  
matteo.vaccari@uninsubria.it



(cc) Matteo Vaccari. Published in Italy.  
Attribution – Non commercial – Share alike 2.5



# Validazione

```
class User < ActiveRecord::Base
  def validate
    if name.blank?
      errors.add(:name, "is missing")
    end
  end
end

u = User.new
u.save           # => false
u.errors        # => {"name"=>["is missing"]}
u.errors.full_messages # => ["Name is missing"]
```

## Validazione a mano

```
class User < ActiveRecord::Base
  def validate_on_create
    if User.find_by_name(name)
      errors.add(:name, "is already being used")
    end
  end
end
```

## Validation methods

```
class User < ActiveRecord::Base
  validates_presence_of :name
  validates_uniqueness_of :name
end
```



## Ancora validazioni

```
class User < ActiveRecord::Base
  validates_numericality_of :height_in_meters
  validates_numericality_of :age, :only_integer => true
end
```

```
class User < ActiveRecord::Base
  validates_presence_of :name, :address
end
```

## Aggiungere le validazioni al server di citazioni

```
class Quote < ActiveRecord::Base
  validates_presence_of :author
  validates_presence_of :body
end
```

## Aggiungere un link per l'azione "new"

```
<%= link_to "Aggiungi un motto", :action => "new" %>
```

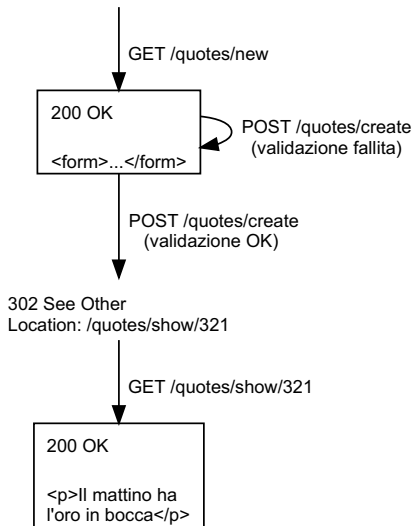
```
<%= link_to "Aggiungi un motto", :action => "new",  
  :controller => "quotes" %>
```

Chi la fa l'aspetti

— Anonimo

[Aggiungi un motto](#)

# Form editing



## La form

app/views/quotes/new.rhtml

---

```
<% form_for(@quote, :url => {:action => 'new'}) do |f| %>
  Autore:<br/>
  <%= f.text_field :author %><br/>
  Testo:<br/>
  <%= f.text_area :body %><br/>
  <%= f.submit_tag 'Create' %>
<% end %>
```

## L'azione `new` nel controller

```
def new
  @quote = Quote.new
  # default: render new.html.erb
end

def create
  @quote = Quote.new(params[:motto])
  if @quote.save
    flash[:notice] = 'Quote was created successfully'
    redirect_to :action => 'show', :id => @quote.id
  else
    render :action => 'new'
  end
end
```

## II controller

```
def index
  @quote = Quote.find(:first, :order => 'rand()')
end

def show
  @quote = Quote.find(params[:id])
  render :action => 'index'
end

def new
  if request.post?
    @quote = Quote.new(params[:motto])
    if @quote.save
      redirect_to :action => 'show', :id => @quote.id
    end
  end
end
```

# Cosa succede se fallisce la validazione?

## Inserisci un nuovo motto

**2 errors prohibited this motto from being saved**

There were problems with the following fields:

- Autore can't be blank
- Testo can't be blank

Autore:

Testo:

## Inserire il blocco di messaggi di errore

```
<h3>Inserisci un nuovo motto</h3>
```

Qui appariranno i messaggi di errore

```
<%= error_messages_for :quote %>
```

```
<% form_for(@quote, :url => {:action => 'new'}) do |f| %>
```

Oppure qui

```
<%= f.error_messages %>
```

...

```
<% end %>
```

## Il codice HTML generato dagli helpers

```
<div class="errorExplanation" id="errorExplanation">
  <h2>2 errors prohibited this motto from being saved</h2>
  <p>There were problems with the following fields:</p>
  <ul>
    <li>Autore can't be blank</li>
    <li>Testo can't be blank</li>
  </ul>
</div>
<form action="/quotes/new" method="post">
  Autore:<br/>
  <div class="fieldWithErrors">
    <input id="motto_autore" name="motto[autore]" type="text" value="" />
  </div><br/>
  ...
</form>
```

## CRUD: list



The screenshot shows a web browser window titled "Motti celebri e meno". The address bar contains "http://localhost:3000/motti". The main content area displays a list of quotes with columns for "Autore", "Testo", and "Created at". Each quote has three links: "Show", "Edit", and "Destroy".

<b>Autore</b>	<b>Testo</b>	<b>Created at</b>	
Anonimo	Chi la fa l'aspetti	Thu Oct 18 10:27:47 +0200 2007	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>
Anonimo	Veni vidi vici	Thu Oct 18 10:27:58 +0200 2007	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>
Foo	Bar	Sun Oct 21 18:17:24 +0200 2007	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>

[New motto](#)

## CRUD: edit



Motti celebri e meno

http://localhost:3000/motti/edit/1

# Editing motto

Autore

Testo

Created at

   —  : 

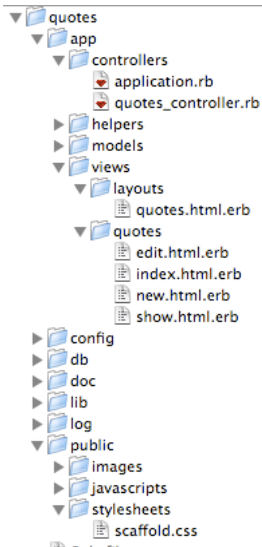
[Show](#) | [Back](#)

# Scaffold

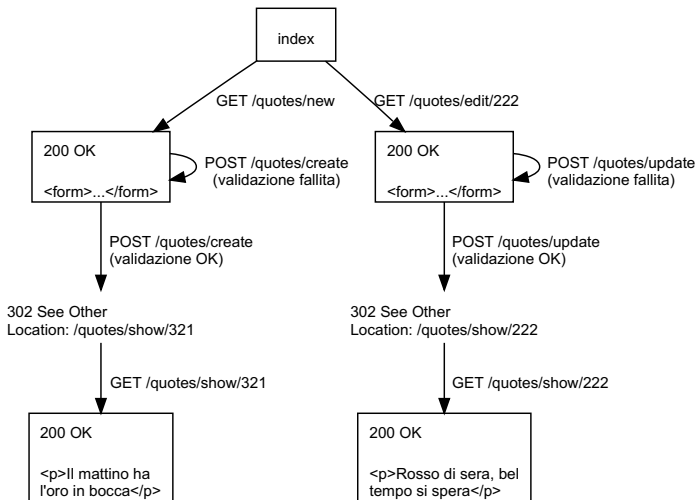
```
$ script/generate scaffold quote author:string body:text
  create  app/views/quotes
  create  app/views/quotes/index.html.erb
  create  app/views/quotes/show.html.erb
  create  app/views/quotes/new.html.erb
  create  app/views/quotes/edit.html.erb
  create  app/views/layouts/quotes.html.erb
  create  public/stylesheets/scaffold.css
  create  app/controllers/quotes_controller.rb
  create  test/functional/quotes_controller_test.rb
  create  app/helpers/quotes_helper.rb
dependency model
  create  app/models/quote.rb
  create  test/unit/quote_test.rb
  create  test/fixtures/quotes.yml
  create  db/migrate/20090108080726_create_quotes.rb

$
```

## File generati dallo scaffold



# Le azioni generate dallo *scaffold*



## II controller I

app/controllers/quotes\_controller.rb

---

```
class QuotesController < ApplicationController
  # GET /quotes
  # GET /quotes.xml
  def index
    @quotes = Quote.find(:all)

    respond_to do |format|
      format.html # index.html.erb
      format.xml { render :xml => @quotes }
    end
  end
end
```

## II controller II

```
# GET /quotes/1
# GET /quotes/1.xml
def show
  @quote = Quote.find(params[:id])

  respond_to do |format|
    format.html # show.html.erb
    format.xml  { render :xml => @quote }
  end
end
```

## II controller III

```
# GET /quotes/new
# GET /quotes/new.xml
def new
  @quote = Quote.new

  respond_to do |format|
    format.html # new.html.erb
    format.xml { render :xml => @quote }
  end
end
```

## II controller IV

```
# POST /quotes
# POST /quotes.xml
def create
  @quote = Quote.new(params[:quote])

  respond_to do |format|
    if @quote.save
      flash[:notice] = 'Quote was successfully created.'
      format.html { redirect_to(@quote) }
      format.xml { render :xml => @quote, :status => :created, :location => @quote }
    else
      format.html { render :action => 'new' }
      format.xml { render :xml => @quote.errors, :status => :unprocessable_entity }
    end
  end
end
```

## II controller V

```
# GET /quotes/1/edit
def edit
  @quote = Quote.find(params[:id])
end

# PUT /quotes/1
# PUT /quotes/1.xml
def update
  @quote = Quote.find(params[:id])

  respond_to do |format|
    if @quote.update_attributes(params[:quote])
      flash[:notice] = 'Quote was successfully updated.'
      format.html { redirect_to(@quote) }
      format.xml  { head :ok }
    else
      format.html { render :action => 'edit' }
      format.xml  { render :xml => @quote.errors, :status => :unprocessable_entity }
    end
  end
end

# DELETE /quotes/1
# DELETE /quotes/1.xml
def destroy
```

## II controller VI

```
@quote = Quote.find(params[:id])
@quote.destroy

respond_to do |format|
  format.html { redirect_to(quotes_url) }
  format.xml  { head :ok }
end
end
end
```

## II layout

### app/views/layouts/quotes.html.erb

---

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
  <meta http-equiv="content-type" content="text/html; charset=UTF-8" />
  <title>Quotes: <%= controller.action_name %></title>
  <%= stylesheet_link_tag 'scaffold' %>
</head>
<body>

<p style="color: green"><%= flash[:notice] %></p>

<%= yield %>

</body>
</html>
```

## Il template per l'azione *index*

app/views/quotes/index.html.erb

---

```
<h1>Listing quotes</h1>
<table> <tr> <th>Author</th> <th>Body</th> </tr>
<tr> <td><td><td><td><td></tr> </table>
<br />
<
```

## Il layout per l'azione *new*

`app/views/quotes/new.html.erb`

---

```
<h1>New quote</h1>
```

```
<
```

```
<p> <</p> <p> <</p> <p> </p> <
```

```
<
```