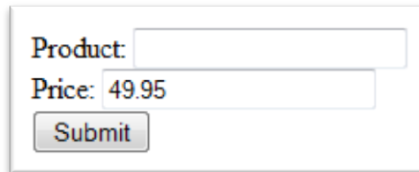


Simple Form

```
<form action="/product/update" method="post">  
  Product: <input type="text" name="product"/><br />  
  Price: <input type="text" name="price" value="49.95"/><br />  
  <input type="submit" value="Submit"/>  
</form>
```



Product:

Price:

Controller Code for Form

```
class StudentsController < ApplicationController
  def index
    ...
  end

  def edit
    if params[:id] then
      @student = Student.find(params[:id])
    else
      @student = Student.new()
    end
  end

  def update
    ...
  end
end
```

} Display
form

} Handle
post

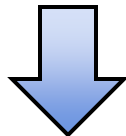
Rails Form Helpers

Describes type, provides initial values



```
<%= form_for(@student, method: :post,  
  url: {action: :update, id: @student.id})  
  do |form| %>  
  <%= form.text_field(:name) %>  
  <%= form.text_field(:birth) %>  
  <%= form.submit "Modify Student" %>  
<% end %>
```

Object representing
form



```
<form action="/student/update/4" method="post">  
  <input id="student_name" name="student[name]"  
    size="30" type="text" value="Chen" />  
  <input id="student_birth" name="student[birth]"  
    size="30" type="text" value="1990-02-04" />  
  <input name="commit" type="submit"  
    value="Modify Student" />  
</form>
```

Customize Format

```
<%= form_for(@student, method: :post,  
  url: {action: :update, id: @student.id}) do |form| %>  
  <table class="form">  
    <tr>  
      <td><%= form.label(:name, "Name:") %></td>  
      <td><%= form.text_field(:name) %></td>  
    </tr>  
    <tr>  
      <td><%= form.label(:birth, "Date of birth:") %></td>  
      <td><%= form.text_field(:birth) %></td>  
    </tr>  
    ...  
  <table>  
    <%= form.submit "Modify Student" %>  
<% end %>
```

Post Action Method

Hash with all of form data

```
def update
  @student = Student.find(params[:id])
  @student.name = params[:student][:name]
  @student.birth = params[:student][:birth]
  @student.gpa = params[:student][:gpa]
  @student.grad = params[:student][:grad]
  if @student.save then
    redirect_to(:action => :index)
  else
    render(:action => :edit)
  end
end
```

Redisplay form on error

Redirects on success

More Compact Approach

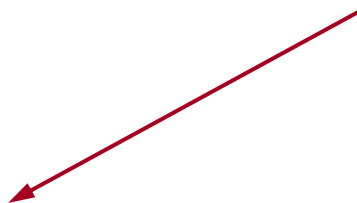
```
def update
  @student = Student.find(params[:id])
  if @student.update(params[:student]) then
    redirect_to(:action => :index)
  else
    render(:action => :edit)
  end
end
```

Security checks in Rails 4 cause update to fail

Rails 4 Pattern

```
def update
  @student = Student.find(params[:id])
  if @student.update(student_params(
    params[:student])) then
    redirect_to(:action => :index)
  else
    render(:action => :edit)
  end
end
```

Creates new object
where specified
elements allowed for
mass update



```
private
def student_params(params)
  return params.permit(:name, :birth, :gpa, :grad)
end
```

Creating New Record

```
def create
  @student = Student.new(student_params(
    params[:student])
  if @student.save() then
    redirect_to(:action => :index)
  else
    render(:action => :edit)
  end
end
```


Validation (in Model)

```
class Student < ActiveRecord::Base
  validates :birth, format: {
    with: /\d\d\d\d-\d\d-\d\d/,
    message: "must have format YYYY-MM-DD"
  }

  def validate_gpa
    if (gpa < 0) || (gpa > 4.0) then
      errors.add(:gpa, "must be between 0.0 and 4.0")
    end
  end

  validate :validate_gpa
end
```

Built-in validator

Custom validation method

Saves error info

Error Messages

```
<% @student.errors.full_messages.each do |msg| %>
  <p><%= msg %></p>
<% end %>
<%= form_for(@student, method: :post,
  url: {action => :update, id: @student.id})
  do |form| %>
  ...
  <%= form.label(:birth, "Date of birth:") %>
  <%= form.text_field(:birth) %>
  ...
<% end %>
```

File Uploads with Rails

```
<%= form_for(:student, method: ...) do |form| %>
  ...
  <%= form.file_field(:photo) %>
  ...
<% end %>
```

In form post method:

```
params[:student][:photo].read()
params[:student][:photo].original_filename
```

