

## Bernat Farrero

# In place editing with Javascript, jQuery and Rails 3

Posted on January 16th, 2011

- [Javascript](#)
- [Ruby on Rails](#)

## Introduction

In the project I'm working on I had to create a script that would allow users to update their information directly from their (show) views. From a general perspective, I think it is useful to give users the option to edit-in-place contents by just clicking them, leaving forms for when there is no other alternative (such as creating new objects). To me, an in-place editor feels more natural and usable and helps notably the user experience. The obvious fact is that most real world applications, such as Flickr or Facebox have implemented this feature in most of their interfaces.

To do that, I wanted to think of a general solution, one that could be applied to my other projects. Even a framework agnostic solution. I decided I would write it completely in Javascript and communicate with the server in a conventional RESTful way. Doing a bit of research before starting out, I ran across the project of Jan Varwig called [Rest In Place](#) which did precisely the same thing. So I examined carefully his code, fixed some stuff and extended it to support all usage cases I wanted to cover, this is how **BestInPlace** is born.

Basically, BestInPlace makes possible to tag (via HTML classes and HTML5 data\* attributes) any field that is going to be user-editable so that the script automatically converts it to an form input when the user clicks on it, with no further muss or fuss for the developer. Of course, this field can take the form of a one-line text input, a textarea for longer texts, a select dropdown that will populate with your custom collection of options or boolean sort of data that works the same way a checkbox would, and allows value customization as well. Additionally, the script will trim and sanitize all user input, display server errors in case the format is not proper, it will also allow you to provide an external handler to activate the input.

Before getting into details.

[SEE THE DEMO](#) | [SEE THE CODE](#)

## Installation

Simply copy and load the files from the folder the following JS files in your application (in the same order):

- [jquery-1.4.2.js](#)
- [jquery.purr.js](#)
- [best\\_in\\_place.js](#)

Add the following line to your onLoad block:

```
$(document).ready(function() {  
  /* Activating Best In Place */  
  jQuery(".best_in_place").best_in_place()  
});
```

## Installation of the Rails 3 Gem

Only add the following line to your application's Gemfile and run *bundle install*:

```
gem "best_in_place"
```

You still need to load the onLoad block, jQuery.js and jquery.purr.js in your application, but you can use the generator to copy (and update when necessary) the best\_in\_place.js script:

```
rails g best_in_place:setup
```

## Usage

First of all, you should write your controllers in a RESTful way and make sure they respond properly to a json request. Here you can see an example of how a standard update action should look like in a Rails app:

```
def update  
  @user = User.find(params[:id])
```

```

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

```

At the same time, if you want to perform server-side validations so that errors are displayed when users introduce invalid data, then the models should look something like this:

```

class User < ActiveRecord::Base
  validates :name,
    :length => { :minimum => 2, :maximum => 24, :message => "has invalid length"},
    :presence => { :message => "can't be blank"}
  validates :last_name,
    :length => { :minimum => 2, :maximum => 24, :message => "has invalid length"},
    :presence => { :message => "can't be blank"}
  validates :address,
    :length => { :minimum => 5, :message => "too short length"},
    :presence => { :message => "can't be blank"}
  validates :email,
    :presence => { :message => "can't be blank"},
    :format => { :with => /^[^\s]+@((?:[-a-z0-9]+\.)+[a-z]{2,})$/i,
      :message => "has wrong email format"}
  validates :zip, :numericality => true, :length => { :minimum => 5 }
end

```

Now all defined messages will be shown to the user if he introduces invalid data and the field will remain unchanged. Notice that messages will be displayed in a jquery.purr pop up. Take time to style it the way you prefer.

## Options / Parameters

Best in Place will be wrapped into an html object (div/span preferred) with the following attributes:

- `class` = "best\_in\_place"
- `data-url` = Object Path
- `data-object` = Object name
- `data-attribute` = Object field
- `data-type` = "input", "textarea", "select", "checkbox" (defaults to input)
- `data-collection` = "[[key, 'value'], [key, 'value'],...]" (a structured JSON containing all possible values in the collection in case of the select or a couple of values in case of a checkbox ["falsevalue", "truevalue"]).
- `data-activator` = "#DOM-OBJECT". Dom object that will convert the text into an input. In case it is not specified it will work by clicking on the text.

Beware: All attributes are assigned underscore strings.

If you are using the **Rails 3 Gem**, you only need to tag user in-place editable fields like this:

## best\_in\_place object, field, OPTIONS

Params:

- **object** (Mandatory): The Object parameter represents the object itself you are about to modify
- **field** (Mandatory): The field (passed as symbol) is the attribute of the Object you are going to display/edit.

Options:

- **:type** It can be only [:input, :textarea, :select, :checkbox] or if undefined it defaults to :input.
- **:collection**: In case you are using the :select type then you must specify the collection of values it takes. In case you are using the :checkbox type you can specify the two values it can take, or otherwise they will default to Yes and No.
- **:path**: URL to which the updating action will be sent. If not defined it defaults to the :object path.
- **:nil**: The nil param defines the content displayed in case no value is defined for that field. It can be something like "click me to edit". If not defined it will show "-".
- **:activator**: Is the DOM object that can activate the field. If not defined the user will making editable by clicking on it.
- **:sanitize**: True by default. If set to false it will allow users to introduce html tags to the input/textarea.

Let's see now how to create user-editable fields by some examples. We'll use a generic User show action, as I did in the [Test Application](#).

## Inputs

In regular HTML:

```
<span class='best_in_place' id='best_in_place_user_name' data-url='/users/4' data-object='user' data-attribute='name' data-type='input'>Dominic</span>
```

Using the Rails 3 gem:

```
<%= best_in_place @user, :name, :type => :input %>
<%= best_in_place @user, :name, :type => :input, :nil => "Click me to add content!" %>
<%= best_in_place @user, :name, :activator => "#activator" %>
```

## Textarea

In regular HTML:

```
<span class='best_in_place' id='best_in_place_user_description' data-url='/users/4' data-object='user' data-attribute='description' data-type='textarea'>
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus a lectus et lacus
ultrices auctor. Morbi aliquet convallis tincidunt. Praesent enim libero, iaculis at
commodo nec, fermentum a dolor. Quisque eget eros id felis lacinia faucibus feugiat et ante.
</span>
```

Using the Rails 3 gem:

```
<%= best_in_place @user, :description, :type => :textarea %>
```

Allowing html tags:

```
<%= best_in_place @user, :description, :type => :textarea, :sanitize => false %>
```

## Select dropdowns

In regular HTML:

```
<span class='best_in_place' id='best_in_place_user_country' data-url='/users/4' data-object='user' data-collection='[[1,"Spain"],[2,"Italy"],[3,"Germany"],[4,"France"]]' data-attribute='country' data-type='select'>Germany</span>
```

Using the Rails 3 gem:

```
<%= best_in_place @user, :country, :type => :select, :collection => [[1, "Spain"], [2, "Italy"], [3, "Germany"], [4, "France"]] %>
```

## Checkbox values

In regular HTML:

```
<span class='best_in_place' id='best_in_place_user_receive_email' data-url='/users/4' data-object='user' data-collection='["No thanks","Yes of course"]' data-attribute='receive_email' data-type='checkbox'>Yes of course</span>
```

Using the Rails 3 gem:

```
<%= best_in_place @user, :receive_emails, :type => :checkbox, :collection => ["No, thanks", "Yes, of course!"] %>
```

## Wrapping up

Best In Place will make easier your application interfaces by letting users modify in-line every content. At the same time, it will control user input and display errors. Let me know if there's something you would do different or don't hesitate to give me a pull request if you contribute anyhow.

Tags: [in-place-edit](#)[javascript](#)[jquery](#)[rails 3](#)[UX](#)

[28 Comments »](#)

---

## 28 Comments on “In place editing with Javascript, jQuery and Rails 3”

[Follow comments feed](#)

[Jürgen Brüder](#) said at 2:05 pm on February 8th, 2011:

Hi!



I really like your work here. I was using Jan's solution but yours is a very great improvement.

A feature I would love to see is adding relations to the :collection. Let's say a @user can have a language. Then it would be great to just do something like:

```
:select, :collection => Language.all %>
```

It could show the name of the language but save the id.

Any thoughts on that?

Nevertheless, a great gem!

Best  
Jay

[bernat](#) said at 4:12 pm on February 8th, 2011:



Thanks, but this feature is there already. The only thing is you should pass the collection doing [[id, value], [id, value], [id, value],...]

[Paul Strong](#) said at 10:59 pm on February 10th, 2011:



Have you tried running this on Rails 2? Should I just use REST in Place or do you think it would be easy enough for me to port your library to Rails 2?

Thanks

[bernat](#) said at 10:57 am on February 11th, 2011:



It should work in Rails 2 as well. The gem is just a helper though, you could create it yourself if you needed. The important thing is the script, and it isn't framework-specific, all it needs is for you to have RESTful controllers, the way I explain in this tutorial.

[Brad](#) said at 10:30 pm on March 1st, 2011:



Thank you so much for posting this! This is exactly what I was looking for. Nice work.

[Pol](#) said at 7:50 pm on March 4th, 2011:



Buscant informació de Rails em trobo amb aquesta notícia i al cap d'una estona, coi si aquest de la foto de la dreta em sembla que l'he vist a algun lloc... ah vale és de la FIB. Felicitats pel blog, he trobat coses molt útils!

[Steve](#) said at 1:54 pm on March 22nd, 2011:



Hi Bernat,

This sounds like an nice implementation of inline editing. However, still being fairly new to rails, I'm getting the following error when I try and run my server after installing your gem. I can uninstall it and everything goes back to working fine. Its essentially a photo gallery with lots of javascript and other packages. I'm running Rails 3.0.5 on Ruby 1.9.2, with jquery 1.5.1. I'll keep digging to see if I can figure it out, but thought I'd ping you to see if you had any ideas. Thanks for your contribution and efforts on this.

Here's the error:

```
/opt/dev/rails/3/pappy/config/initializers/setup_jquery.rb:1:in ``: You have a nil object when you didn't expect it! (NoMethodError)
```

My setup\_jquery.rb has:

```
Pappy::Application.config.action_view.javascript_expansions[:defaults] = ['jquery.min', 'rails']
```

which works fine normally. I suspect there is a require missing somewhere, but I'm not sure where. Any ideas?

Thanks,  
Steve

[Steve](#) said at 2:44 am on March 23rd, 2011:



A quick update that I got things working by removing my jquery initializer. I thought I needed it to replace prototype as the default javascript lib. I found and installed the 'jquery-rails' gem and that seems to have installed jquery the correct way. Thanks again for creating and sharing your code!

- Steve

[Ynes](#) said at 7:44 pm on April 1st, 2011:

Can I have the id value from the “select” to replace a div?



[durr](#) said at 1:16 pm on April 4th, 2011:

if one wanted to override one of the callbacks, what is the “right” way? Presumably, i should be able to get a handle to the BestInPlaceEditorObject, and set the loadErrorCallback function appropriately, but what scope does that editor object exist?



[Luis](#) said at 5:55 am on April 7th, 2011:

Man, this was a great tutorial. You rock!



Is there any way to implement this in a “real” checkbox, rather than using “Yes, No” collection?

[Ryan](#) said at 3:36 am on April 8th, 2011:

Being new at this stuff, I apologize, but can you clarify how to pass in relations to the :collection part. For example, I have a table of clients with name as an attribute. How would I pass in the list of client names that have been entered into the system so far? Great gem by the way!



[Alberto](#) said at 5:29 am on April 18th, 2011:

I am trying to add the jquery datepicker to the input field and combine that with your awesome best in place. Any suggestions on how to do this would be appreciated.



[aqabawe](#) said at 4:13 pm on May 2nd, 2011:

thanks man, awesome plugin.



for those who want to use this on some other page like the INDEX page just add

or whatever your model name to the partial that populates it.

[sharath](#) said at 9:18 am on May 6th, 2011:

hello Bernat,  
thanks a lot for posting these kind of tutorials ,using these idea in the project has helped me lot and any newbie can also learn easily from u r tutorials  
you rock dude!!!!!!!!!!!!



[ejangi](#) said at 6:08 am on May 14th, 2011:

This is awesome man. But, is there a way I can define an “oncomplete” callback? I need to do some post processing to the updated html...



[bernat](#) said at 12:57 pm on May 15th, 2011:

I appologize for not keeping the project updated, i’ve been craizngly busy lately. I recommend you to look at the Github README instead of the instructions above, I’ll keep there the latest version of the specifications. And also, I’ll devote some time eventually to look at the pull requests and issues people opened, thanks everyone for posting!



[Tommie Jones](#) said at 5:48 pm on June 11th, 2011:

I can get it working by embedding the html. But I cannot get the best\_in\_place method working. I keep getting a MoMethod error. Any suggestions? Also I want to trigger another javascript event after the value is entered and accepted. Any suggestions.



[hey dear](#) said at 8:24 am on June 20th, 2011:

dude you done a great job. its so simple and having magical effects...



[imohan](#) said at 4:00 am on June 22nd, 2011:

I see the message when there is an error in the data due to validations but no message if the data is saved correctly  
I see these codes where I need to add the message any example or can i extend and put something in application.js exampple will be great if possible.



```
loadSuccessCallback : function(data) {  
  this.element.html(data[this.objectName]);  
  // Binding back after being clicked  
  $(this.activator).bind('click', { editor: this }, this.clickHandler);  
},
```

thank you  
-imohan

[jeromeyers](#) said at 10:57 pm on June 30th, 2011:

The edit in place demo doesn't really work in Chrome for the multiline "User Description" text area. Before the first time you edit-in-place the label moves up and down as you move your mouse over the text, and then, after you leave the field and let it revert to simple text, and you move your mouse over the field, the content itself jumps up and down maddeningly, seeming to squirm away from all attempts to click it, like fish through castaways' fingers.

[jeromeyers](#) said at 10:59 pm on June 30th, 2011:

To add to that, it is only on one of the examples that this is happening, it is an example with a ... tag in it. This creates a break and it is when the mouse moves over this break that the squirming starts.

[Bryan Wang](#) said at 4:20 am on July 14th, 2011:

Amazingly simple and elegant! Thank you for sharing it with the public

[Bryan Wang](#) said at 9:28 am on July 15th, 2011:

Is there a plan to add advanced input types such as tinymce or markdown or textile?

[Dmitry](#) said at 8:47 am on September 6th, 2011:

Please, say how to scroll red pen over editable field?

[Pol Miro](#) said at 9:21 pm on November 29th, 2011:

Felicitats pel railcast nano! quin crack estas fet :)

[habib](#) said at 7:19 am on December 19th, 2011:

really thanx but look at this ..... irtci.com

[rpruiz](#) said at 6:49 pm on February 2nd, 2012:

Felicidades Bernat. Muy útil. Tienes alguna idea de cómo poder implementar el uso de tabs para moverse entre campos a editar? He tratado usando :html\_attrs => { :tabindex => 1 } sin éxito. Ojalá puedas comentar algo al respecto.

Gracias.



## Say something!

Name (required)	Mail (will not be published) (requ	Website
<div></div>		
<div>Submit Comment</div>		

Powered by [WP Hashcash](#)



[I'm an IT Engineer by the Facultat d'Informàtica de Barcelona \(FIB\) and a web applications developer](#)

[in Itnig. Click to see more.](#)



---

## Recent Posts

- [In place editing with Javascript, jQuery and Rails 3](#)
- [How I manage the WordPress theme with Git](#)
- [Thunderbird crashing on start up](#)
- [My idea of accessibility](#)
- [Semàntic Podcast](#)

## Archives

- [January 2011](#)
- [September 2010](#)
- [August 2010](#)
- [June 2010](#)
- [May 2010](#)
- [April 2010](#)
- [February 2010](#)
- [January 2010](#)
- [December 2009](#)
- [November 2009](#)

## Categories

- [Computer Science](#)
- [General](#)
- [Javascript](#)
- [Mac](#)
- [Photography](#)
- [Ruby on Rails](#)
- [Web design](#)
- [Wordpress](#)

## Blogroll

- [Custom Buttons](#)
- [Itnig – IT Solutions](#)
- [Jordi Romero blog](#)
- [Nigeka](#)



---

Subscribe by email:

Subscribe

[by FeedBurner](#)

[semantic podcast logo](#)



Wordpress theme by Bernat Farrero