

Persisting multiple
objects at once

About me



Przemysław Szyszka

- Ruby
- Passioned about Elixir

Problem

```
defmodule Blog.Post do
  use Ecto.Schema

  schema "products" do
    field :title, :string
    field :body, :string

    timestamps()

    has_many :comments, Blog.Comment
  end
end
```

```
defmodule Blog.Comment do
  use Ecto.Schema

  schema "comments" do
    field :body, :string

    timestamps()

    belongs_to :post, Blog.Post
  end
end
```

put_assoc

```
Repo.insert!(%Post{  
  title: "Hello",  
  body: "world",  
  comments: [  
    %Comment{body: "Excellent!"}  
  ]  
})
```

```
post = Ecto.Changeset.change(%Post{}, title: "Hello", body: "world")  
comment = Ecto.Changeset.change(%Comment{}, body: "Excellent!")  
post_with_comments = Ecto.Changeset.put_assoc(post, :comments, [comment])  
Repo.insert!(post_with_comments)
```

cast_assoc

```
defmodule Blog.Post do
  import Ecto.Changeset

  def changeset(struct, params \\ %{}) do
    struct
    |> cast(params, [:title, :body])
    |> validate_required([:title, :body])
    |> cast_assoc(:comments)
  end
end
```

```
defmodule Blog.Comment do
  import Ecto.Changeset

  def changeset(struct, params \\ %{}) do
    struct
    |> cast(params, [:body])
    |> validate_required([:body])
  end
end
```

cast_assoc

```
defmodule Blog.Post do
  import Ecto.Changeset

  def changeset(struct, params \\ %{}) do
    struct
    |> cast(params, [:title, :body])
    |> validate_required([:title, :body])
    |> cast_assoc(:comments, with: Blog.Comment.my_custom_changeset)
  end
end
```

Ecto.Multi

```
iex> Ecto.Multi.new()  
%Ecto.Multi{names: #MapSet<[]>, operations: []}
```

```
iex> Ecto.Multi.new() |> Repo.transaction()  
{:ok, %{}}
```

Ecto.Multi

```
Ecto.Multi.new()  
|> Ecto.Multi.insert(:team, team_changeset)  
|> Ecto.Multi.update(:user, user_changeset)  
|> Ecto.Multi.delete(:foo, foo_changeset)  
|> Repo.transaction
```


Ecto.Multi

```
Ecto.Multi.new()  
|> Ecto.Multi.insert(:post, post_changeset)  
|> Ecto.Multi.run(:comment, fn repo, %{post: post} ->  
  repo.insert(comment_changeset)  
end)
```

Questions?