Talking in general about proper naming

Welcome to the lesson 6 of Getting Started with Backbone.js course! In this lesson we are going to refactor our code and introduce some useful concepts to make your application much easier to maintain and scale.

As I already told you, Backbone is not a framework, but rather a library and its not opinionated. This means that its totally you responsibility to use proper naming conventions and code structuring. Until now we haven't really taken care of this which resulted in a pretty messy code that is hard to maintain.

It is important that all members of you team understand and follow **naming conventions** that are used in the project. If everyone is naming variables and functions differently, it will result in a huge mess, whereas with clear names you will quickly understand what is the purpose of some variable.
In our demo app I tried to follow naming conventions that are employed by many JS developers, however still there is one thing that I don't like.

An example

Let's make it more clear that this is a collection class:

```
Organizer.EventsCollection = Backbone.Collection.extend({
```

Change all references to it as well. Inside the router there is a duplicating line here:

```
Organizer.EventsList = new Organizer.Events();
```

Move it into the *app.js* and rename it correctly:

```
Organizer.EventsList = new Organizer.EventsCollection();
```

In the same file rename

```
Organizer.EventsList
```

to

```
Organizer.events
```

Do the same inside the router and inside the *events.js* file.

Now reload the page and check we haven’t break anything.

So once again: you may introduce our own naming conventions, just make sure to follow them accurately because otherwise you will end up with a situation where you don’t understand what you code does.