

Restangular - Get object w/id, edit object, update object

Add  projects to your  **stackoverflow** profile.

I can't seem to figure this out -

I want to do something like:

```
content = Restangular.one('contents', 2).get()
content.name = "Chunky Bacon"
content.put()
```

But this doesn't work - content doesn't have a put function, and interestingly, it doesn't have a name, either (is that because it's a "promise"?)

Now, the Restangular docs do describe updating an object, but you get that object as part of a collection:

```
content_id2 = Restangular.all("contents").getList()[2]
content_id2.name = "Chunky Bacon"
content_id2.put()
```

And this works. But! Not only have I fetched the entire list, I can't figure out how to easily get my content by it's ID; something like

```
Restangular.all("contents").getList().find(2)
```

customPUT(...) seems like the next best bet:

```
Restangular.one("contents", 2).customPUT("", {"content" : {"name" : "Chunky Bacon"} } )
```

Only when the Rails server receives it, params["content"] == {id: 2} - it seems like something on the client side is stomping that value. (Again, note, no name field). Now, I can work around it simply by doing something like:

```
Restangular.one("contents", 2).customPUT("", {"content_data" : {"name" : "Chunky Bacon"} } )
```

but it really seems like I'm just missing how you're supposed to do this.

[angularjs](#) [ruby-on-rails-4](#) [put](#) [restangular](#)

asked Sep 20 '13 at 23:31

 [Narfanator](#)
2,174 1 14 34

1 Answer

Alright, found it, thanks to [this question](#):

```
#In the beginning:
Restangular.one("content", 2).get().then(function(c){
  $scope.content = c
})
```

```
#Elsewhere:
$scope.content.name = "Chunky Bacon"
```

```
#Finally:
$scope.content.put()
```

answered Sep 20 '13 at 23:51

 [Narfanator](#)
2,174 1 14 34

i've done the same and i get still the error "Object not found" when i call the function 'put'. - [Asqan](#) Jun 15 '15 at 8:44