Lab 12.2: Modeling Splatts
IN705 Databases Three

Introduction

We decided last week that we would model the user/splatt relationship by creating a seperate bucket for each user's splatts. This means that we'll have to organise our `SplattRepository` class a little differently than our `UserRepsoitory` class, since it will have to work with a variecty of buckets.

In this lab we will see how to create splatts using our Riak data model and how to retrieve all of a user's splatts.

1 Splatt model

The splatt model is extremely simple. We just story the body, the creation timestamp, and the ID. Recall that we decided to generate uuids to use as keys,

The splatt model goes in `app/models/splatt.rb`.

file: app/model/splatt.rb

```
class Splatt < Hashie::Dash
  property :id
  property :body
  property :created_at
 end
```

2 Creating a splatt

To create a splatt and save it to the data store, we need to create a `SplattRepository` class and modify the `SplattsController` class to use it.

The `SplattRepository` class starts out a bit different than our `UserRepository` class, since the bucket names vary. We set the bucket name in the constructor.

file: app/models/splatt_ repository.rb

```
class SplattRepository

def initialize(client, user)
  @client = client
  @bucket = user.email
end
```

There is an interesting thing about the `save` method. When it's called to save a splatt for a new user, that user's bucket doesn't even exist yet. It's not created until we save the user's first splatt. With most databases we would expect this method to throw up an error in this case.

file: app/models/splatt_ repository.rb

```
def save(splatt)
  splatts = @client.bucket(@bucket)
  key = splatt.id

  unless splatts.exists?(key)
    riak_obj = splatts.new(key)
    riak_obj.data = splatt
    riak_obj.content_type = 'application/json'
    riak_obj.store
    splatt
  end

end
```

Finally we need to modify the `create` method in the `SplattsController`. In this method, we

1. set up the splatt;
2. retrieve our user;
3. create the splatt repository;
4. save the splatt.

file: app/controllers/splatts_ controller.rb

```
def create
  @splatt = Splatt.new
  @splatt.id = SecureRandom.uuid
  @splatt.created_at = Time.now
  @splatt.body = params[:body]
  client = Riak::Client.new
  user = UserRepository.new(client).find(params[:user])
  db = SplattRepository.new(client, user)

  if db.save(@splatt)
    render json: @splatt, status: :created, location: @splatt
  else
    render json: @splatt.errors, status: :unprocessable_entity
  end
end
```

3 Retrieving a user's splatts

To retrieve a user's splatts, we need to get all of the values out of a particular bucket. There's not a direct way to do that, so we have to get the keys and then get the associated values.

file: app/models/splatt_ repository.rb

```
def all
  keys = @client.bucket(@bucket).keys
  riak_list = @client.bucket(@bucket).get_many(keys)
  results = []
  riak_list.values.each do |splatt_obj|
    splatt = Splatt.new
    splatt.id = splatt_obj.data['id']
    splatt.body = splatt_obj.data['body']
    splatt.created_at = splatt_obj.data['created_at']
    results.push(splatt)
  end
  results
end
```

Now we can use this in the `SplattsController`:

file: app/controllers/users_ controller.rb

```
def splatts
  db = UserRepository.new(Riak::Client.new)
  @user = db.find(params[:id])
  db = SplattRepository.new(Riak::Client.new, @user)
  render json: db.all
end
```

Finally, don't forget to modify your route for this method:

file: config/routes.rb

```
 get 'users/splatts/:id' => 'users#splatts', :constraints => { :id => /[0-9A-Za-z\-\.\@]+/ }
```