Keeping things in check

**Goal:** In this lab, we will ensure that the data passed to our domain classes makes sense. By the end of this lab, you should have a basic understanding of grails constraints.  
  
**Instructions**  
  
We can’t just accept any data that users might hand to us. For example, the text string “asdf” should not be valid as a value of the Commentator.email field. We need a way of expressing this concept to grails. In each of your domain classes, you should see a variable declared as follows:

static constraints = {

}

This is where you declare restrictions on fields in grails. So, for example, if, in the Commentator domain class, you wanted to restrict the email field to only accept valid e-mail addresses, you would add the following line into the constraints section:

email(nullable: false, blank: false, email: true)

This tells grails that the email field cannot be null, cannot be left blank, and must be an e-mail. Grails has a number of built-in constraints, and each one of them has the syntax constraint-type:  constraint-parameter. A full list of constraints can be found at  <http://www.grails.org/doc/latest/ref/Constraints/Usage.html>.  
  
Using this guide, add the following constraints

* Post
  + title (not null, not blank, between 1 and 50 characters)
  + teaser (between 0 and 100 characters)
  + content (not null)
  + lastUpdated (not null, not blank)
  + published (not null)
* Comment
  + post (not null, not blank)
  + comment (not null, not blank)
  + dateCreated(not null, not blank)
* Commentator
  + name (not null, not blank)
  + url (null ok, blank ok, is valid url)
  + email (null ok, blank ok, is valid e-mail address)
  + comment (not null)
* Tag
  + tag (not null, between 0 and 50 characters)