

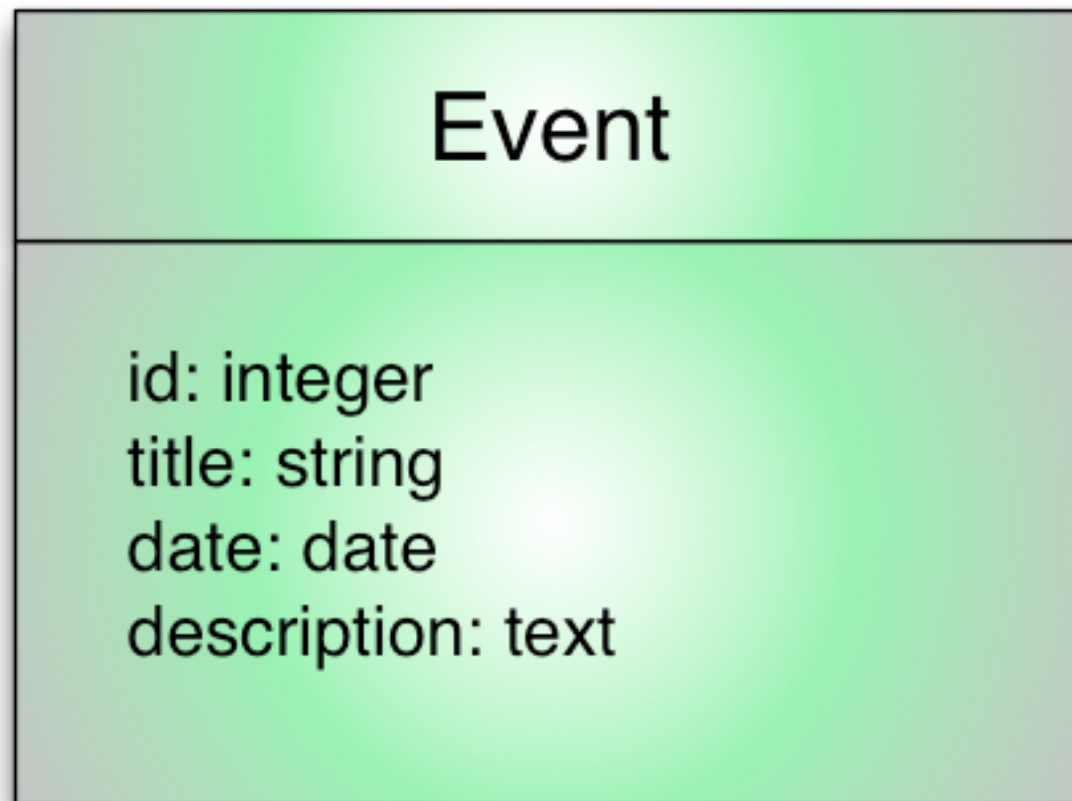
HTML5 and CSS3 for Mobile Applications

Prof. Paul Krause, University of Surrey
Capturing Associations between Classes

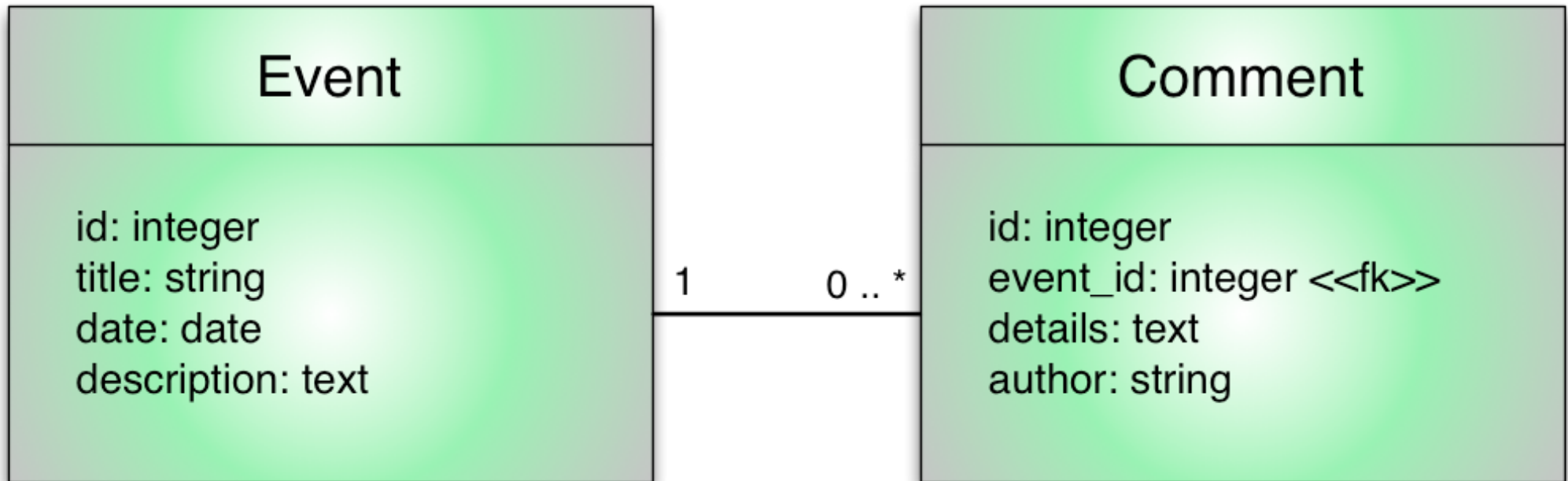
Objectives for today

- Show how easy it is to add associations between objects in a Model
- Show how we can automatically clear away orphaned dependents if we *destroy* our active objects

What we want to do



What we want to do



Three steps, and that is all!

- Generate a CRUD Scaffold for Comment
- Add a “has_many” association to our Event model
- Add a “belongs_to” association to our Comment model

At the console

```
$ rails generate scaffold Comment event_id:integer details:text author:string
```

```
$ rake db:migrate
```

models/event.rb

```
class Event < ActiveRecord::Base  
  validates_presence_of :title, :date, :description  
end
```

models/event.rb

```
class Event < ActiveRecord::Base  
  
  validates_presence_of :title, :date, :description  
  
  has_many :comments  
  
end
```


models/comment.rb

```
class Comment < ActiveRecord::Base  
  validates_presence_of :event_id, :details, :author  
  belongs_to :event  
end
```

N.B. Rails will generate a /comments path to get to the index of comments

Now let's see some serious Rails Magic!

Unwanted dependents

- Deleting an Event will leave all its associated comments in the database
- Either:
 - We provide some warning to ensure the user deletes all dependent comments before an event can be deleted
 - or
 - we simply destroy all the dependent comments when an Event is destroyed

models/event.rb

```
class Event < ActiveRecord::Base  
  
  validates_presence_of :title, :date, :description  
  
  has_many :comments, :dependent => :destroy  
  
end
```

Why “destroy” and not “delete”?

- *delete* works directly on the database
 - this is fastest
- *destroy* first loads an instance of the active record object (or can be called directly on an already loaded instance)
 - *destroy* is then called as an instance method
- the difference is important if an object has dependent associations or *before_destroy* callbacks
 - these will *not* be acted on by *delete*

What you have learnt

- Building associations in Rails is really simple
- Clearing away dependents before they get orphaned is also simple
- But don't forget to destroy your active objects, rather than just delete them!