Week 5

CSPP 52553: Web Development
Spring 2013

- Model-Backed Forms
- Partials and Layouts
- Model Validations, Part 1
- Model Queries, Part 1
- JSON APIs
- Resources
- Model Associations

Midterm Next Week

Short-answer

Multiple choice / "All that apply"

Code reading challenges

Questions are based on Github code, slides, and your expected practice at home.

Final Project

Clone Week5

```
cd ~/code
git clone git://github.com/cspp52553-spring-2013/week5.git

cd week5/lecture_code
bundle install
...etc...

open http://localhost:3000/
```

Model-Backed Forms

form_for

Partials

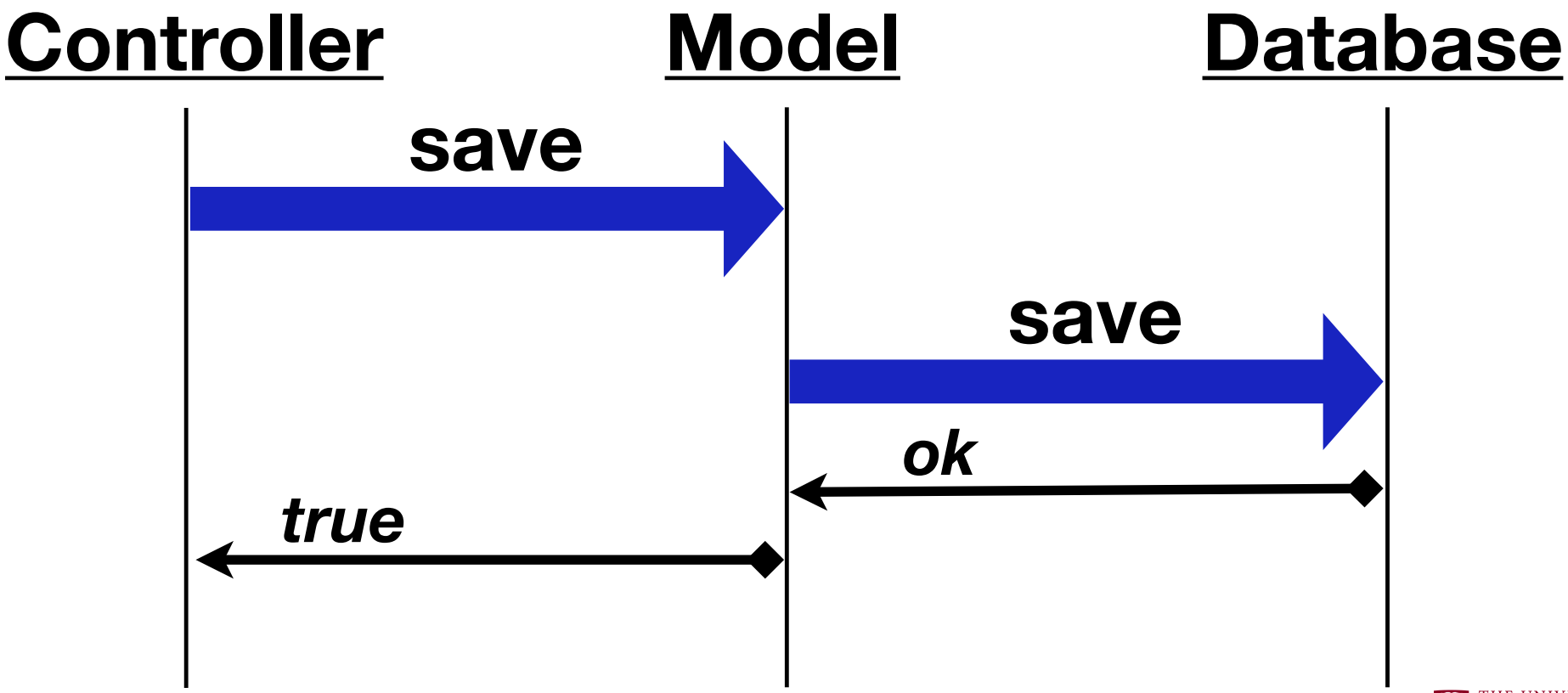
Partials provide reusable markup that can be invoked from any main template or other partial.

Filenames for partial views must start with an underscore: _*form.html.erb*

Layouts

Layouts provide reusable markup to surround every page on the entire site, every page for a specific controller, or around a given partial.

Model Validations



Model Validations

Important model methods to know:

.save
.valid?
.invalid?

These trigger the validation rules

.errors ***reports error messages after the rules have run***

Model Validations

```
class Movie < ActiveRecord::Base

  validates_presence_of :title

end
```

JSON APIs

`respond_to` **lets us choose the appropriate representation of a resource based on the requested media type.**

Model Queries, Part 1

```
Model.order("column dir")
```

dir must be asc or desc

```
teams = Team.order("city asc")
```

Model Queries, Part 1

```
Model.where(:column => value)
```

The `where` method expects a hash and returns a `relation` object.

Model Queries, Part 1

```
Model.where(:column => value)

losers = Team.where(:city => "Chicago")
my_team = Team.where(:id => 4).first
```

Model Queries, Part 1

```
Model.count
```

or

```
relation.count
```

```
Team.count  # => 6
Team.where("city" => "Chicago").count # => 2
```

Resources

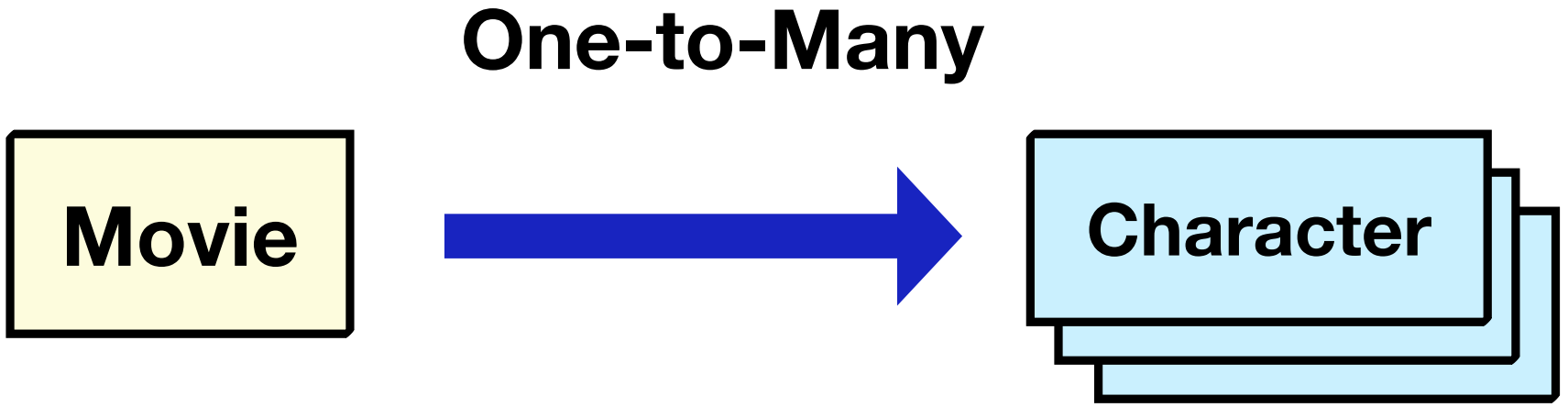
```
resources :movies
```

generates 7 routes according to RESTful principles and Rails conventions.

Model Associations

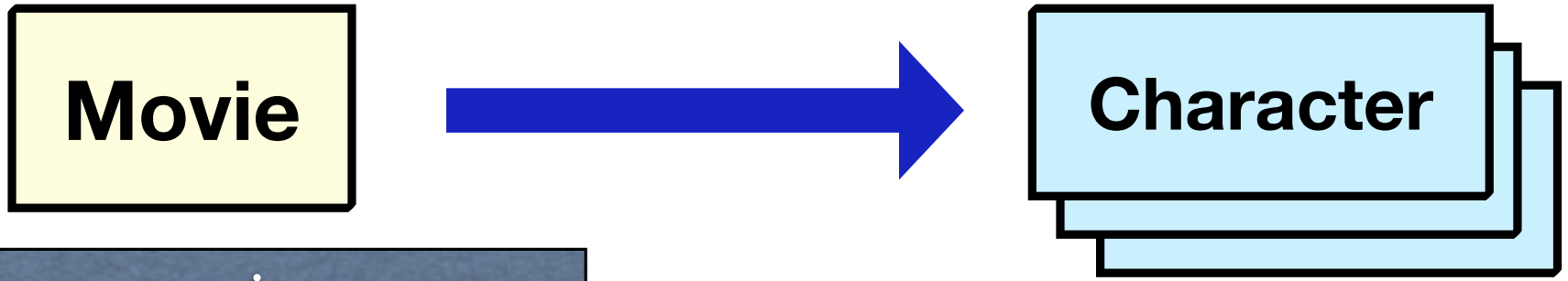
Model Associations

Model Associations



Model Associations

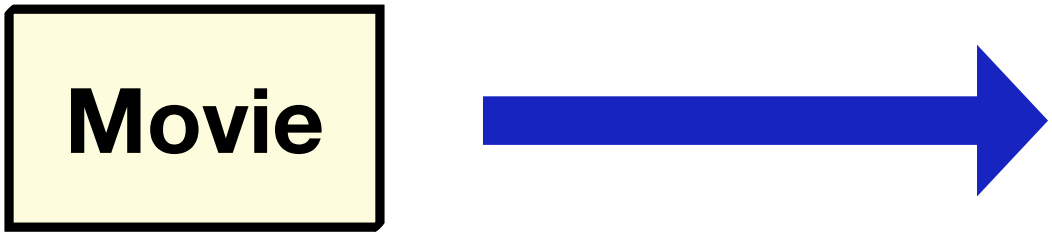
One-to-Many



movies	
id	**name**
1	Apollo 13
2	Star Wars
3	Toy Story

Model Associations

One-to-Many



movies	
id	**name**
1	Apollo 13
2	Star Wars
3	Toy Story

characters	
id	**name**
88	Luke Skywalker
89	Woody
90	Jim Lovell
91	Han Solo

Model Associations

One-to-Many

movies	
id	name
1	Apollo 13
2	Star Wars
3	Toy Story

characters		
id	name	movie_id
88	Luke Skywalker	2
89	Woody	3
90	Jim Lovell	1
91	Han Solo	2

Model Associations

How do we display the director of a given movie?

directors

id	name
1	Frank Darabont
2	Francis Ford Coppola
3	Christopher Nolan

movies

id	name	year	director_id
88	The Godfather	1972	2
89	The Dark Knight	2008	3
90	The Shawshank Redemption	1994	1
91	The Godfather: Part II	1974	2

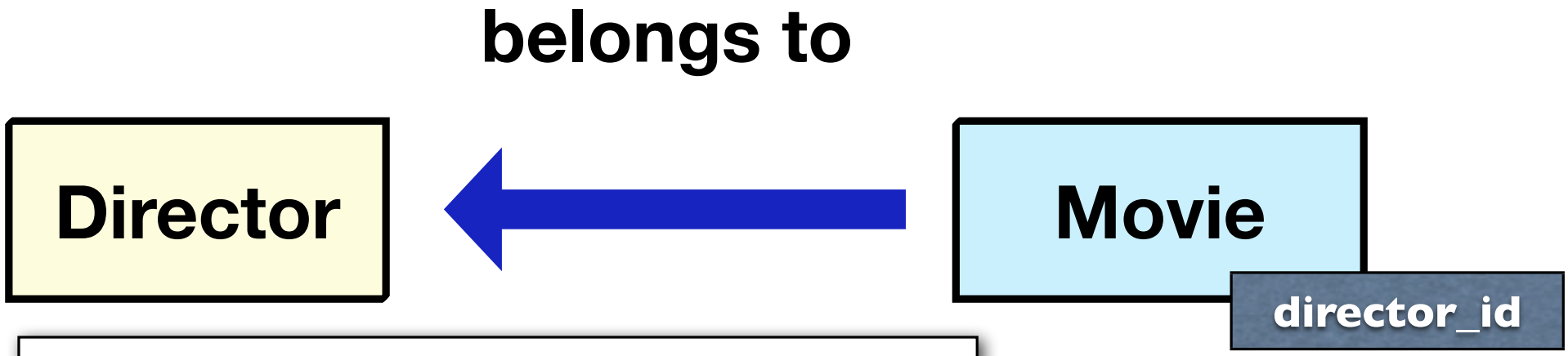
Model Associations

How do we display all of the movies by a given director?

directors	
id	name
1	Frank Darabont
2	Francis Ford Coppola
3	Christopher Nolan

movies			
id	name	year	director_id
88	The Godfather	1972	2
89	The Dark Knight	2008	3
90	The Shawshank Redemption	1994	1
91	The Godfather: Part II	1974	2

Model Associations



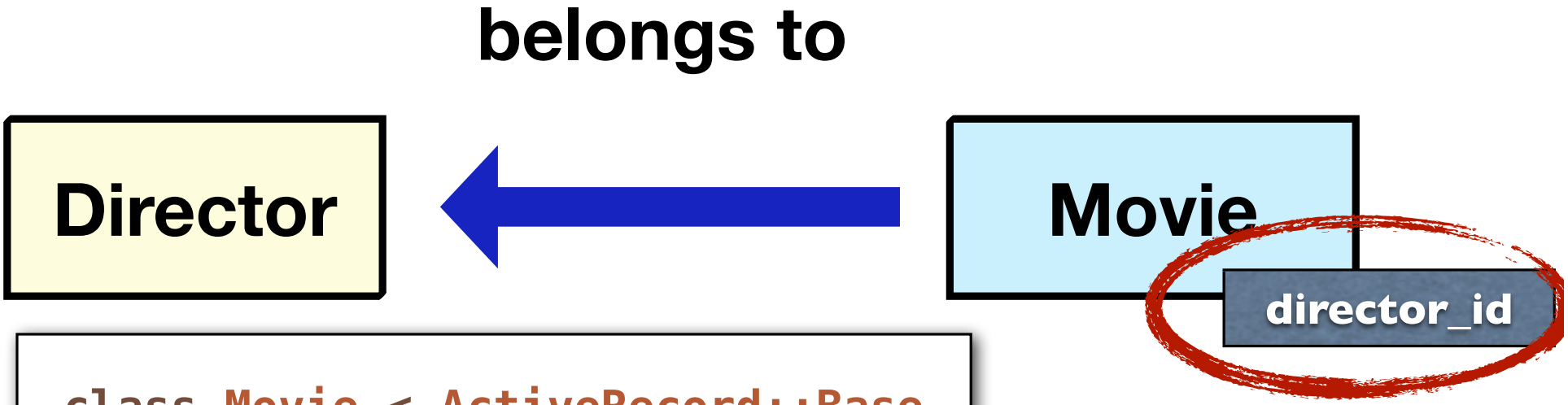

```
class Movie < ActiveRecord::Base

  belongs_to :director

end
```

x

Model Associations




```
class Movie < ActiveRecord::Base
  belongs_to :director
end
```

Model Associations

x

Model Associations

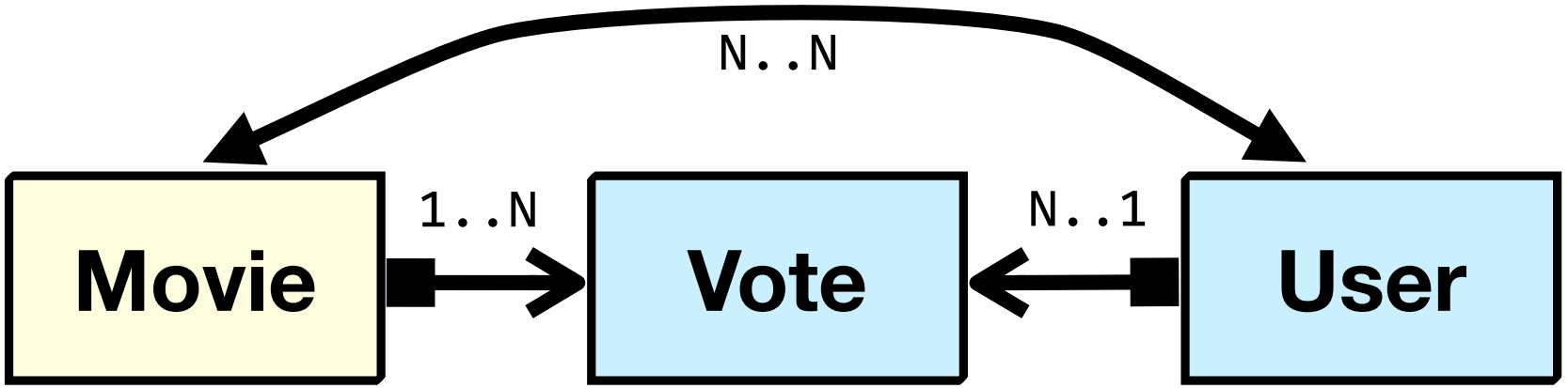

```
class Director < ActiveRecord::Base

  has_many :movies

end
```

x

Model Associations



Movies and Users are a many-to-many association.

We use two 1-to-Many associations to travel across.

Now we can say that a movie....

```
has_many :users, :through => :votes
```

x