

Ruby on Rails course

Users, scopes and ajax



User authentication

What is authentication?

Proving your identity to a system

Proving you are who you claim to be

Proving you have the right credentials

Proving you are not a bot

Proving you are not a hacker

Proving you are not a spy

User login

Log in

Email

admin@infinum.co

Password

.....

☐ Remember me

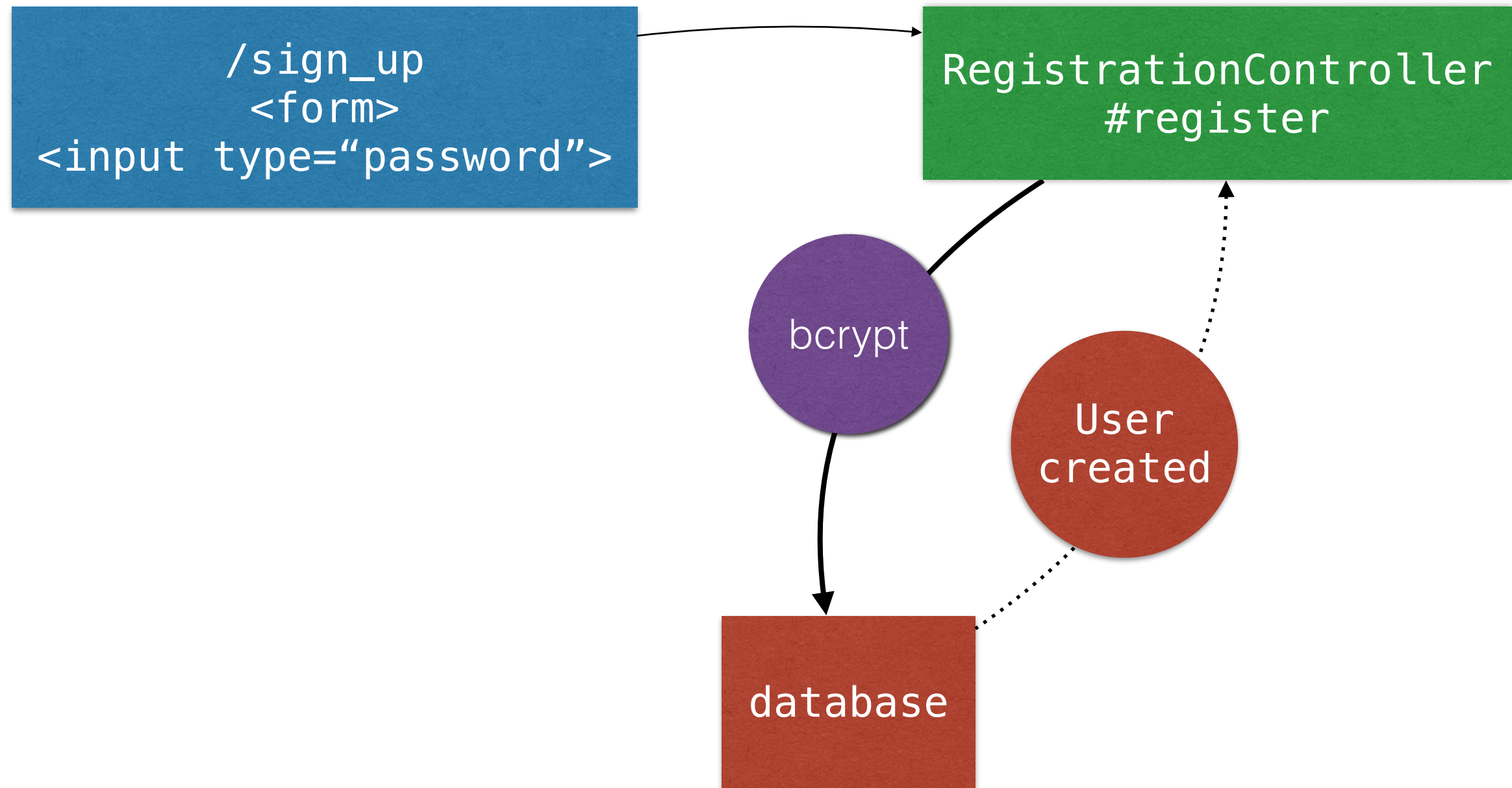
Log in

User model

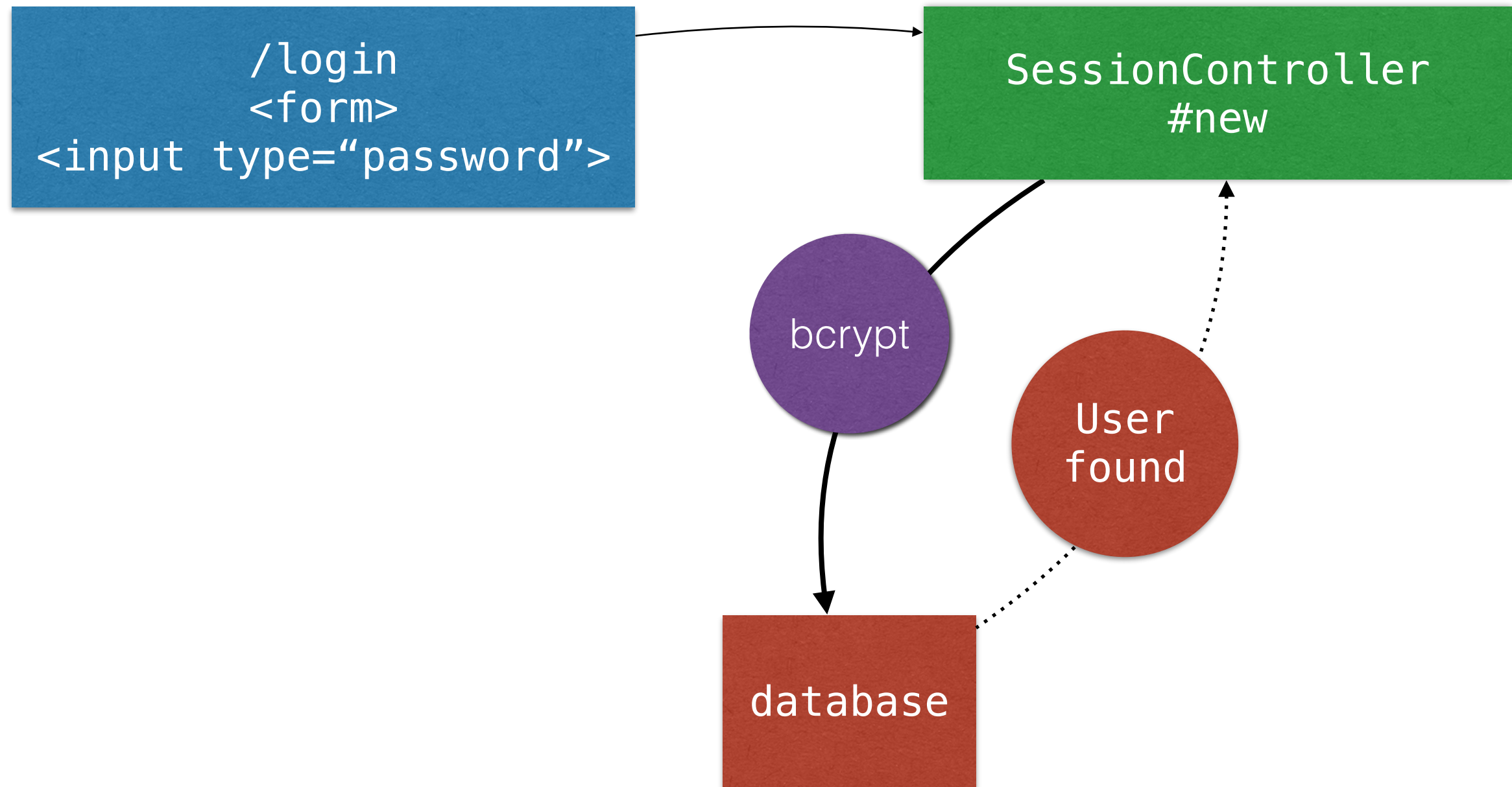
```
create_table "users" do |t|
  t.string "email", default: "", null: false
  t.string "encrypted_password", default: "", null: false
  t.string "first_name"
  t.string "last_name"
  t.datetime "created_at", null: false
  t.datetime "updated_at", null: false
end
```



User registration



User authentication



```
user.encrypted_password == bcrypt(params[:password])
```



User session

Name ▲	Value	Domain	Path	Expire...	Size	HTTP
_boatit_session	RUtrS05pU1Nqd2hRMDZZekhkR29aSzE3dldQRT...	boatit...	/	Session	669	✓

```
@user = User.find_by(id: session[:current_user_id])
```



User flow

- ✦ In each controller check if session contains :current_user_id
- ✦ if not then redirect_to login page
- ✦ else get user from database and continue normal flow



Devise

Complete User authentication solution

<https://github.com/plataformatec/devise>

Installing devise

- ✦ Add 'devise' gem to Gemfile
- ✦ rails generate devise:install
- ✦ rails generate devise User
- ✦ rake db:migrate



Configure devise model

```
# app/models/user.rb
```

```
devise :database_authenticatable, :registerable,  
       :recoverable, :rememberable, :trackable,  
       :validatable # , :confirmable
```

```
# config/routes.rb
```

```
devise_for :users
```



Using devise

```
# in controllers
```

```
before_action :authenticate_user!
```

```
# controllers and views helpers
```

```
user_signed_in?
```

```
current_user
```

Devise default views

```
rails generate devise:views
```



Flash messages

Feedback

flash ooooh

in controllers

```
flash[:notice] = "Successfully logged in."  
flash[:alert] = "You must log in to continue!"
```

in views

```
<%= flash[:notice] %>  
<%= flash[:alert] %>
```

Scopes

Writing custom queries as methods

Scopes

```
# app/models/user.rb
```

```
scope :active, -> { where(active: true) }  
scope :adult, -> { where('age > 18') }
```

```
# app/controllers/users_controller.rb
```

```
@users = User.active  
@adult_users = User.active.adult
```

Search

```
# app/models/user.rb
```

```
scope :search_by_name, -> (term) {  
  where("name like '%?%'", term)  
}
```

```
# app/controllers/users_controller.rb
```

```
@users = User.search_by_name(params[:name])
```



Lazy loading

Lazy loading

includes

```
# app/models/user.rb
```

```
has_many :posts
```

```
# app/controllers/users_controller.rb
```

```
@users = User.includes(:posts)
```

```
# app/controllers/users_controller.rb
```

```
<%= user.posts.title %>
```



Ajax

asynchronous JavaScript and XML

jquery

```
$.ajax({  
  url: 'http://localhost:3000/subreddits',  
  dataType: 'json'  
});
```



respond_to

```
def index
  @subreddits = Subreddit.all

  respond_to do |format|
    format.html { render :index }
    format.json { render json: @subreddits.to_json }
  end
end
```



remote: true

in view

```
<%= link_to 'Get subreddits with ajax',  
  subreddits_path, remote: true %>
```

in controller

```
def index
```

```
  @subreddits = Subreddit.all
```

```
  respond_to do |format|
```

```
    format.html { render :index }
```

```
    format.js { } # equivalent with above
```

```
    format.json { render json: @subreddits.to_json }
```

```
  end
```

```
end
```

