Time	~2 - 3 hours
Learning Goals	• Learn how to send and receive an email within a Rails application. • Learn how to generate and edit an Action Mailer class and mailer view. • Learn how to use an SMTP service

We have enabled users to make bookings, but how will our hosts know that a booking has been made? Let's incorporate a mailer function that notifies hosts when a booking has been made for their property, i.e. we want to create a reservation mailer that sends an email to the host when a booking is made.

For this assignment, to make things work, we'll be going to gather notes and resources across the internet. Please proceed to find a partner from your peers.

Also, we'll need a Gmail account so please go ahead and get one. You may want to sign up for a test account for personal privacy protection. **Don't forget to configure your** keys properly!

1) How does Rails handle email sending?

Our first task is to understand how Rails handles email. What are the resources? Is this guide sufficient to start with? Are there any other resources that can guide you on "how" to use? Discuss with your partner and mentors.

Good resources always cover:

1. What is the technology
2. Reason and objective to use that technology
3. How to deploy that technology
4. How to debug and test that technology

---

2) Create a branch for new development

At this point, you should know that this is for safety reasons, you should create a new branch to work out your solution. Switch to your branch to work on it.

---

3) Generate a Mailer

Now based on the resources you gathered, generate a `Reservation Mailer`. You should get something like the following in your `app/mailers/reservation_mailer.rb`:

```
class ReservationMailer < ApplicationMailer
```

```
end
```

---

4) Create the booking email

All right, at this time, we should proceed with creating an email.

Mailers are very similar to Rails controllers. They also have methods called **“actions”** and use views to structure the content. While a controller generates content like HTML to send back to the client, a Mailer creates a message to be delivered via email.

So, we now know `app/mailers/reservation_mailer.rb` is a mailer controller and `app/views/reservation_mailer/` holds the email contents. Great! Let’s move forward and create the `booking_email` method.

On the controller side, it looks something like this:

```
 def booking_email(customer, host, reservation_id)
 ...
 end
```

On the view (email contents) side, it looks something like this:

```
<!DOCTYPE html>
<html>
 <head>
   <meta content='text/html; charset=UTF-8' http-equiv='Content-Type' />
 </head>
 <body>
   <h1>Congratulations! You have received a booking from <%= @customer.name %></h1>
   <p>
```

```
    You can see details of your new reservation here: <%= @url %>
  </p>
 </body>
</html>
```

Points to discuss:

1. Since you are designing the emails, what are the details that you wish to present?
2. Is there a way to view the email contents without sending? Hint: use this gem.
3. What are the variable contents in the email, for e.g. `<%= @customer.name %>` as shown above?

---

5) Calling the Mailer

Now that we have our mailer ready, let's use them. From the MVC architecture perspective, we can see that the "mailer controller" is similar to any of our models. Hence, we can use it like a model class.

Discussion with partner:

1. Based on the example above, should the mailer methods be coded in a way of accepting params or operatives? Which case is rational?
2. Any other options other than "deliver_now"? Where can I find these resources?

---

6) Configure your Mailer in Rails

All right, we're all set. In this tutorial, we're going to use Gmail mailer service. To start, let's configure Rails to use Gmail to send an email.

Discussion among partner:

1. How can I control when to send and when not to send an email?
2. Should I send an email when I'm in development mode? Can I use letter_opener gem?
3. If I'm building a marketing subscription list, can Gmail handle it?
4. Why and when to use dedicated SMTP service like Mandrill or SendGrid?

Once you can send an email, congrats! You’re done with this assignment. You can proceed to commit the changes, raise pull request and merge back to master branch.