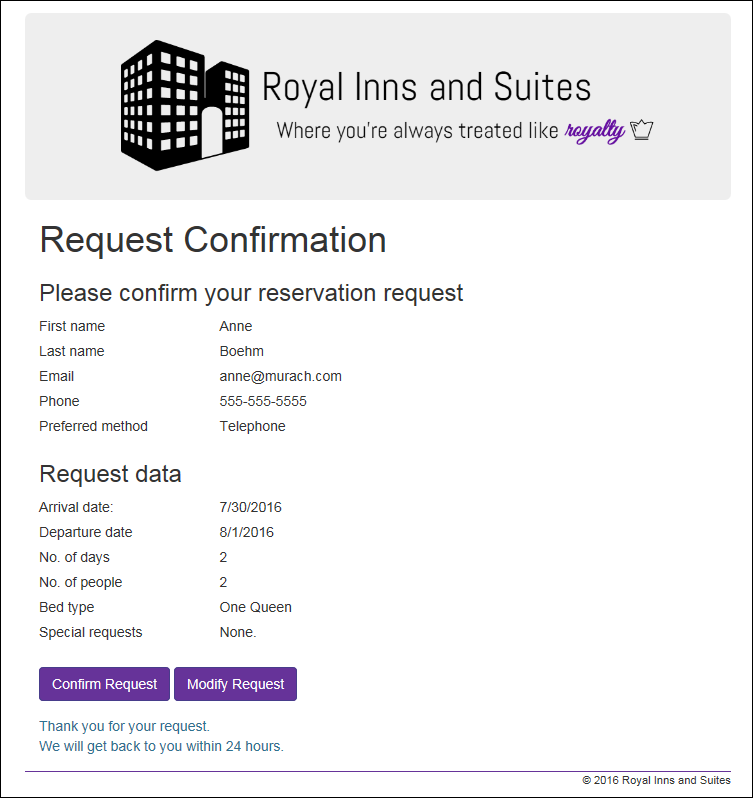
Extra 8-1 Use session state to store reservation data

In this exercise, you’ll store the data that’s entered into the Request page of the Reservation application so it can be displayed on a Confirmation page like the one that follows.



Review the new form, new class, enhanced Load event handler,   
and new method

1. Open the Site you used for Chapter 6 and Chapter 7’s assignment.
2. Create a new class file in the Models directory and paste in the code from the reservationCS.txt file provided.
3. Create a Confirm.aspx file. You will find the code for the .aspx file in a txt file provided. The code for the aspx.cs file is also located in a txt file provided.
4. Review the code for the Load event handler of the Request form (this is in one of the txt files and will need to be in your Request.aspx.cs file). If the page request isn’t a postback and a Reservation object is stored in session state (which means that it has already been created), this handler calls the DisplayReservation method that moves the data from session state to the controls on the Request page. Otherwise, the Load event handler sets the starting values for the arrival date text box and the radio buttons. Note that it sets the year in the copyright notice in the footer either way.
5. Review the code in the DispayReservation (this is in the txt file and will need to be added to the Request.aspx.cs file) method to see how it gets the Reservation object from session state and then moves the data from the Reservation object to the controls on the form.

Code the Click event handler for the Submit button of the Request form

1. Start a Click event handler for the Submit button of the Request form. It should initialize a new Reservation object and then set its properties based on the values the user entered on the form. Then, write a statement that saves the Reservation object in session state, and finish with a statement that redirects to the Confirmation page.
2. If you have any trouble writing the statements that convert the control data to properties in the Reservation object, here are some tips.

* To convert a text date to a DateTime object, use the Convert.ToDateTime method.
* To get the number of nights, you can subtract the departure date from the arrival date and then use the Days property of the resulting TimeSpan object.
* To convert a selection in the drop-down list for the number of people to an integer, use the Convert.ToInt32 method.

If you still have trouble, you can skip the conversions that you can’t figure out.

Finish the Confirmation form

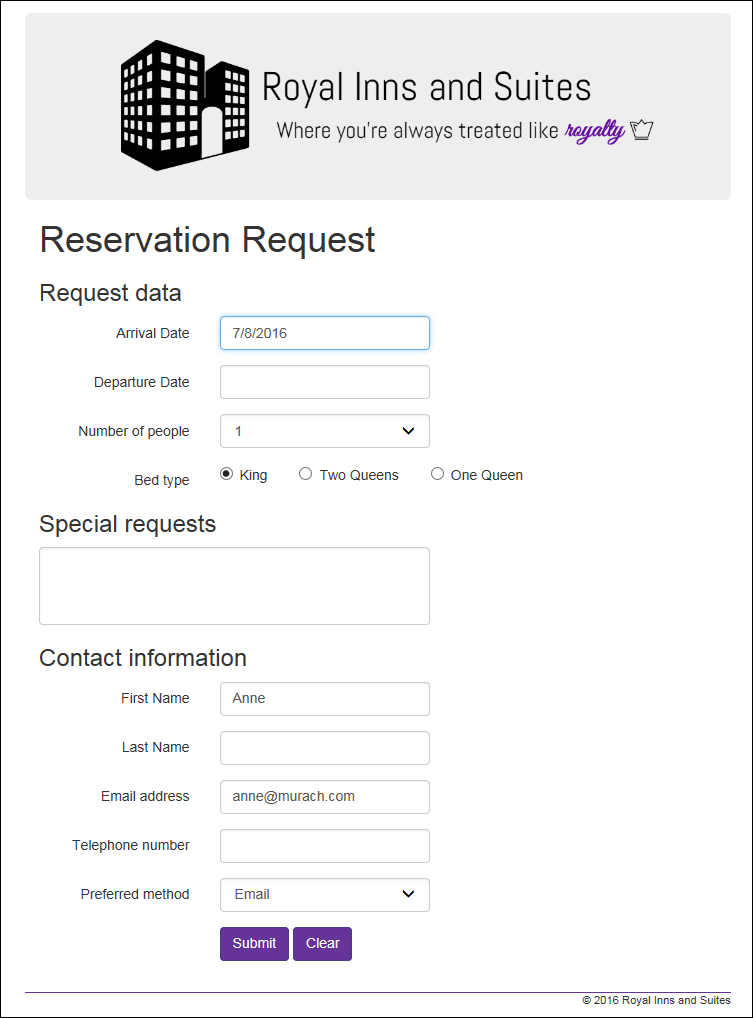
1. Review the Load event handler for the Confirmation form. This handler sets the year for the copyright in the footer and then displays reservation data in the labels on the form by calling a DisplayReservation method.
2. Code the DisplayReservation method. It should retrieve the Reservation object from session state, store it in a variable, and then get the values from the reservation object and move them to the label controls on the form. But if you skipped any of the properties in step 6 or 7, skip those in this method too.
3. Create a Click event handler for the Confirm Request button on the Confirmation form. It should display a message like the one on the form above.
4. Enhance the aspx code for the Modify Request button on the Confirmation form. It should post back to the Request form.

Test the application

1. Run the application, complete the Request form, and click the Submit button to display the Confirmation page. If the reservation data isn’t displayed correctly, fix the problems and test again.
2. Click the Modify Request button to return to the Request form. The data for the reservation should still be displayed so you can modify it and click the Submit button again. When you do that, the Conformation page should show the changed data.
3. Continue testing and fixing until you’re sure this application works.

Extra 8-2 Use cookies to store user information

In this exercise, you’ll use cookies to store the first name and email address that the user enters into the Request form of the Reservation application. Then, you’ll display them in the controls on the Request form the next time the user starts the application, as shown here.



Add the code that creates some persistent cookies and retrieves the cookies

1. If it’s not already open, open the web application that you worked on in exercise 8-1.
2. Add code to the Click event handler of the Confirm Request button on the Confirmation form that creates two persistent cookies with the user’s first name and email address and adds them to the HttpResponse object. Set the expiration date for the cookies to six months from the current date.
3. Add code to the Load event handler of the Request form that retrieves the two cookies and displays their values on the form. This code should be executed only if the page is not being posted back, a Reservation object is not found in session state, and the cookies exist.

Test the application to make sure the cookie persists

1. Run the application, complete the Request form, and submit and confirm the request. Then, close the browser window.
2. Run the application again. This time, the first name and email that were saved in cookies should be displayed in the Request form. Close the browser window again.

Add code to clear the persistent cookies

1. Add code to the Click event handler of the Clear button on the Request page that removes the two persistent cookies.
2. Run the application and see that the user name and email are still displayed on the Request form. Click on the Clear button to remove them from the form, and then close the browser window.
3. Run the application again. This time, the user name and email text boxes should be blank.