RealThinClient SDK – Lesson 2b – Using Query Parameters

Leave a reply

Now that we have seen how to Create a Web Server and how to Send Dynamically Generated Content, we are going to see how to accept Query Parameters.

We are going to make some small changes to our previous code (Code for Demo Sending Dynamically Generated Content) to accept request's query parameters.

When we coded our Dynamically Generated Content we give no option to the user to decide what Square values the system will return, so now, we are going to give the user the choice to:

1. Select the starting number of the Square values.
2. Select the ending number of the Square values.

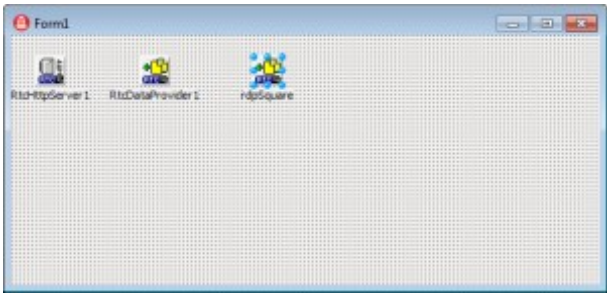
Basically we are going to:

- Open our Lesson 2 project
- Add logic to process two request's query parameters
- Check that our Server is working.

Steps.

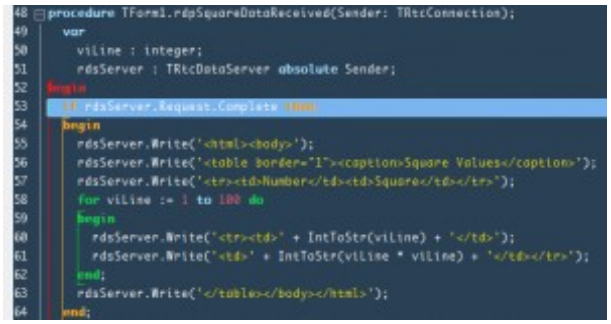
1. Open our Lesson 2 project.

We open our Lesson 2 project to continue our modifications.



Open Project

2. Edit our OnDataReceived event for RtcDataProvider component



```
procedure TForm1.rdpSquareDataReceived(Sender: TRtcConnection);
  var
    viLine : integer;
    rdsServer : TRtcDataServer absolute Sender;
begin
  if rdsServer.Request.Complete then
  begin
    rdsServer.Write('<html><body>');
    rdsServer.Write('<table border="1"><caption>Square Values</caption>');
    rdsServer.Write('<tr><td>Number</td><td>Square</td></tr>');
    for viLine := 1 to 100 do
    begin
      rdsServer.Write('<tr><td>' + IntToStr(viLine) + '</td>');
      rdsServer.Write('<td>' + IntToStr(viLine * viLine) + '</td></tr>');
    end;
    rdsServer.Write('</table></body></html>');
  end;
```

OnDataReceived Evet Code before modifications

Previously, the **URL** in our browser's address bar for the Square request was http://localhost/square.

Now, we need to pass two parameters, the starting and ending number to return Squared values. Our **URL** will look like:

http://localhost/square?start=10&end=20.

We are sending two query parameters, **start** and **end**.

We must take some steps to prevent our Content to grow and cause a denial of service or something even worst in our first web server. In theory, there's no limit in the HTTP response's size, but we are going to prevent that by checking that no more than 1,000 squared numbers are requested.

Another issue is what happens if there's no start or end query parameter. We are going to set a default value for each one of these query parameters so no error will raise when our Server is running.

However we need to inform about any of this issues to the user.

Code using **with**

```
procedure TForm1.rdpSquareDataReceived(Sender: TRtcConnection);
  var
    viLine : integer;
    viStart, viEnd : integer;
    vbStartError, vbEndError, vbRangeError : boolean;
begin
  with TRtcDataServer(Sender) do
  begin
    if Request.Complete then
    begin
      viStart := 1;
```

```
12          viEnd := 100;
13          vbStartError := True;
14          vbEndError   := True;
15          vbRangeError := True;
16
17          if Request.Query['start'] <> '' then
18             try
19                viStart := StrToInt(Request.Query['start']);
20                vbStartError := False;
21             except
22             end;
23
24          if Request.Query['end'] <> '' then
25             try
26                viEnd := StrToInt(Request.Query['end']);
27                vbEndError := False;
28             except
29             end;
30
31          if viEnd - viStart > 1000 then
32             viEnd := viStart + 100
33          else
34             vbRangeError := False;
35
36          Write('<html><body>');
37          Write('<table border="1"><caption>Square Values</caption>');
38
39          if vbStartError = True then
40             Write('<tr><td colspan="2" style="color:red;">ERROR: Wrong start
41parameter. Set to Default (1)</td></tr>');
42
43          if vbEndError = True then
44             Write('<tr><td colspan="2" style="color:red;">ERROR: Wrong end
45parameter. Set to Default (100)</td></tr>');
46
47          if vbRangeError = True then
48             Write('<tr><td colspan="2" style="color:red;">ERROR: Wrong
49Range. Set to Default (100)</td></tr>');
50
51          Write('<tr><td>Number</td><td>Square</td></tr>');
52
53          for viLine := viStart to viEnd do
54          begin
55             Write('<tr><td>' + IntToStr(viLine) + '</td>');
56             Write('<td>' + IntToStr(viLine * viLine) + '</td></tr>');
57          end;
58
59          Write('</table></body></html>');
```

```
      end;
    end;
  end;
```

Code without using **with**

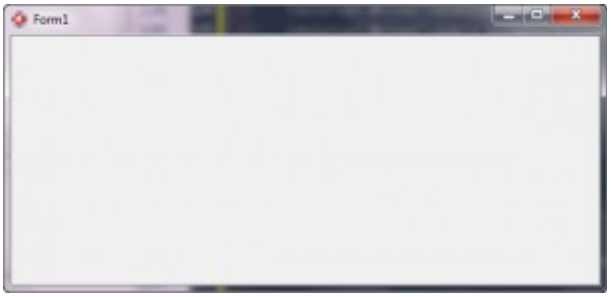
```
procedure TForm1.rdpSquareDataReceived(Sender: TRtcConnection);
  var
    viLine : integer;
    rdsServer : TRtcDataServer absolute Sender;
    viStart, viEnd : integer;
    vbStartError, vbEndError, vbRangeError : boolean;
begin
  if rdsServer.Request.Complete then
  begin
    viStart := 1;
    viEnd := 100;
    vbStartError := True;
    vbEndError   := True;
    vbRangeError := True;

    if rdsServer.Request.Query['start'] <> '' then
      try
        viStart := StrToInt(rdsServer.Request.Query['start']);
        vbStartError := False;
      except
      end;

    if rdsServer.Request.Query['end'] <> '' then
      try
        viEnd := StrToInt(rdsServer.Request.Query['end']);
        vbEndError := False;
      except
      end;

    if viEnd - viStart > 1000 then
      viEnd := viStart + 100
    else
      vbRangeError := False;

    rdsServer.Write('<html><body>');
    rdsServer.Write('<table border="1"><caption>Square
Values</caption>');

    if vbStartError = True then
      rdsServer.Write('<tr><td colspan="2" style="color:red;">ERROR:
Wrong start parameter. Set to Default (1)</td></tr>');

```

```
        if vbEndError = True then
           rdsServer.Write('<tr><td colspan="2" style="color:red;">ERROR:
43Wrong end parameter. Set to Default (100)</td></tr>');
44
45      if vbRangeError = True then
46         rdsServer.Write('<tr><td colspan="2" style="color:red;">ERROR:
47Wrong Range. Set to Default (100)</td></tr>');
48
49      rdsServer.Write('<tr><td>Number</td><td>Square</td></tr>');
50
51      for viLine := viStart to viEnd do
52      begin
53         rdsServer.Write('<tr><td>' + IntToStr(viLine) + '</td>');
54         rdsServer.Write('<td>' + IntToStr(viLine * viLine) + '</td></tr>');
55      end;
56
57      rdsServer.Write('</table></body></html>');
    end;
  end;
```

We are checking for our two query parameters (**start** and **end**), if there's no data for this parameters, we use the default values (1 for **start** and 100 for **end**). Then we check that the range (**end** minus **start**) isn't bigger than 1,000, if it is so, then we set it to 100. In every case we send an error message to the user if any of these checks failed.

3. Check that our Server is running and sending the right response.

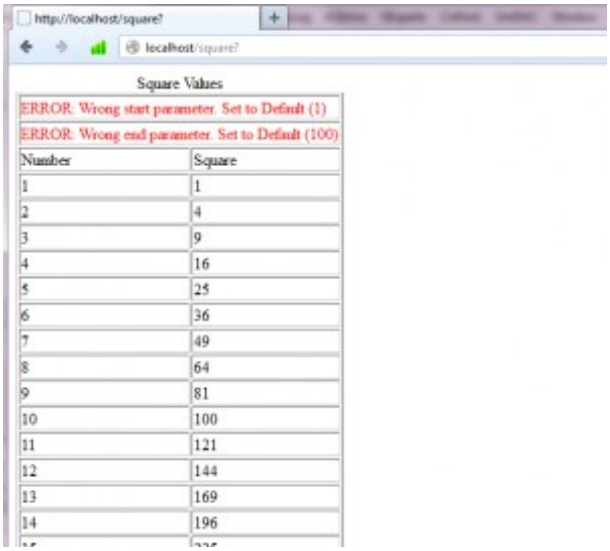
Now we compile and run our application.



Server Running

Then, open the web browser and use any of these addresses:

- http://localhost/square?start=10&end=200
- http://localhost/square
- http://localhost/square?start=-15
- http://localhost/square?start=helloworld



Server Response Example

Your server knows how to handle any of these. Try to do your tests and improve the code.