

CellRenderer.java

```
1 import java.awt.Color;
2 import java.awt.Component;
3 import java.util.ArrayList;
4
5 import javax.swing.JTable;
6 import javax.swing.table.DefaultTableCellRenderer;
7
8 /**
9  *
10  * @author Ayisha S.R. Sowkathali, Sifaben Vahora
11  *
12  *      CellRender.java
13  *
14  *      Class allows the user to click on the table cell to set appointment
15  */
16 */
17 class CellRenderer extends DefaultTableCellRenderer {
18
19     /*
20      * serialVersionUID is generated by default to be used during deserialization to
21      * verify that the sender and receiver of a serialized object have loaded
22      * classes for that object that are compatible with respect to serialization
23      */
24     private static final long serialVersionUID = 1L;
25
26     ArrayList<SetAppointment> appointmentList;
27     SetAppointment current;
28
29     public CellRenderer(ArrayList<SetAppointment> appointmentList) {
30         this.appointmentList = appointmentList;
31     }
32
33     public void setCurrent(SetAppointment current) {
34         this.current = current;
35     }
36
37     @Override
38     public Component getTableCellRendererComponent(JTable table, Object obj, boolean
isSelected, boolean hasFocus,
39         int row, int column) {
40
41         Component cell = super.getTableCellRendererComponent(table, obj, isSelected,
hasFocus, row, column);
42         cell.setBackground(Color.white);
43         Object value = table.getModel().getValueAt(row, column);
44         if (current != null && value != null) {
45             current.day = (Integer) value;
46             if (current.day != 0) {
47                 int index = appointmentList.indexOf(current);
48                 if (index >= 0) {
49                     cell.setBackground(Color.yellow);
50                 }
51             }
52         }
53     }
54 }
```

CellRenderer.java

```
52     }  
53     return cell;  
54 }  
55 }
```