

CenterPanel.java 是 AuthAnalyzer GUI 界面的核心面板，它管理 HTTP 请求/响应的显示、比较、过滤和导出等功能，基于 Swing 进行 UI 组件的布局和交互。以下是详细的功能解析：

---

## 1. 主要功能概述

- **请求/响应管理：**
    - JTable 显示 HTTP 请求/响应数据。
    - ListSelectionModel 处理用户选择的请求/响应。
    - RequestTableModel 作为 JTable 的数据模型，提供请求/响应的表格化存储。
  - **Diff 差异对比：**
    - 使用 Diff\_match\_patch 计算 HTTP 响应之间的差异，并以 HTML 形式展示。
    - JEditorPane 作为差异显示面板，JScrollPane 用于滚动查看对比结果。
  - **过滤和搜索：**
    - PlaceholderTextField 允许用户输入搜索关键字。
    - 复选框 JCheckBox 控制不同类型的请求过滤（如标记、重复、Bypass 状态等）。
    - CustomRowSorter 控制表格的排序逻辑。
  - **UI 交互：**
    - JButton 控制表格清除、导出、切换视图（单/对比）。
    - JPopupMenu 右键菜单，支持请求重复发送、删除、标记等操作。
- 

## 2. 代码结构解析

### (1) 主要成员变量

```
private final JTable table;  
private final RequestTableModel tableModel;  
private final JEditorPane diffPane;  
private final JButton changeMessageViewButton;  
private final JButton expandDiffButton;  
private final JCheckBox showDiffCheckBox;  
private final JSplitPane splitPane;
```

- table：核心 JTable 组件，显示 HTTP 请求/响应数据。
- tableModel：表格的数据模型，管理请求/响应存储。
- diffPane：用于显示请求/响应的文本差异（Diff）。
- changeMessageViewButton：控制切换 **单视图/对比视图**。
- expandDiffButton：控制差异视图的展开/折叠。
- showDiffCheckBox：是否显示请求/响应的 diff 视图。
- splitPane：管理 tablePanel 和 messageViewPanel 的分割视图。

---

### (2) 构造函数 CenterPanel(MainPanel mainPanel)

```
public CenterPanel(MainPanel mainPanel) {  
    this.mainPanel = mainPanel;  
    setLayout(new BorderLayout());  
    table = new JTable();  
}
```

```
tablePanel.setBorder(BorderFactory.createLineBorder(Color.gray));
```

- 绑定 MainPanel, 用于 UI 交互。
- BorderLayout 作为布局管理器, 支持 JPanel 组件的分区。
- 创建 JTable 并初始化 tablePanel。

---

### (3) updateDiffPane() 方法 (计算 Diff 差异)

```
public void updateDiffPane() {
    if (changeMessageViewButton.getText().equals(BUTTON_TEXT_SINGLE_VIEW) &&
        showDiffCheckBox.isSelected()) {
        String msg1 = tabbedPanel1.getCurrentMessageString();
        String msg2 = tabbedPanel2.getCurrentMessageString();
        if (msg1 == null || msg2 == null) {
            diffPane.setText(TEXT_DIFF_VIEW_DEFAULT);
        } else {
            diffPane.setText(getHTMLCenterText("Calculating differences..."));
            new Thread(new Runnable() {
                @Override
                public void run() {
                    Diff_match_patch dmp = new Diff_match_patch();
                    LinesToCharsResult a = dmp.diff_linesToChars(msg1, msg2);
                    String lineText1 = a.getChars1();
                    String lineText2 = a.getChars2();
                    List<String> lineArray = a.getLineArray();
                    LinkedList<Diff> diffs = dmp.diff_main(lineText1, lineText2, false);
                    dmp.diff_charsToLines(diffs, lineArray);
                    final String diffPaneText = getHTMLfromDiff(diffs);
                    diffPane.setText(diffPaneText);
                    SwingUtilities.invokeLater(() -> {
                        comparisonScrollPane.getVerticalScrollBar().setValue(0);
                        messageViewPanel.revalidate();
                    });
                }
            }).start();
        }
    }
}
```

#### 功能分析:

1. 获取用户当前选中的请求/响应数据:
  - msg1 = tabbedPanel1.getCurrentMessageString();
  - msg2 = tabbedPanel2.getCurrentMessageString();
2. 检查是否为空, 若为空则不计算 diff。
3. 计算差异:
  - Diff\_match\_patch 进行文本差异计算:
    1. diff\_linesToChars(): 将字符串转为简化格式, 减少计算量。

2. diff\_main(): 核心 Diff 计算。
3. diff\_charsToLines(): 转换回原始文本格式。

#### 4. 更新 UI:

- o diffPane.setText(diffPaneText); 显示 HTML 格式化的差异数据。

---

#### (4) setupTableContextMenu() (右键菜单)

```
private void setupTableContextMenu() {
    table.addMouseListener(new MouseAdapter() {
        @Override
        public void mouseReleased(MouseEvent event) {
            if (event.getButton() == MouseEvent.BUTTON3) {
                int[] rows = table.getSelectedRows();
                if (rows.length > 0) {
                    JPopupMenu contextMenu = new JPopupMenu();
                    JMenuItem deleteRowItem = new JMenuItem("Delete Row");
                    deleteRowItem.addActionListener(e -> {
                        for (int row : rows) {
                            tableModel.deleteRequestResponse(row);
                        }
                    });
                    contextMenu.add(deleteRowItem);
                    contextMenu.show(event.getComponent(), event.getX(), event.getY());
                }
            }
        }
    });
}
```

#### 功能:

- 监听鼠标右键点击 JTable。
- JPopupMenu 右键菜单提供 **删除选中行** 功能。

---

#### (5) clearTablePressed() (清空表格)

```
public void clearTablePressed() {
    clearTableButton.setIcon(loaderImageIcon);
    if (config.isRunning()) {
        config.getAnalyzerThreadExecutor().execute(() -> {
            clearTable();
            clearTableButton.setIcon(null);
        });
    } else {
        clearTable();
        clearTableButton.setIcon(null);
    }
}
```

功能：

- 异步清空表格，防止 UI 阻塞。
- 若 config.isRunning() 处于分析状态，则通过线程池执行 clearTable()。

---

### 3. 总结

#### 功能总结

功能	代码实现
HTTP 请求/响应管理	JTable + RequestTableModel
Diff 差异对比	Diff_match_patch
数据过滤/搜索	PlaceholderTextField + CustomRowSorter
UI 交互	JButton、JPopupMenu、JCheckBox
异步处理	SwingUtilities.invokeLater()、new Thread()

#### 核心逻辑

1. 请求/响应管理：JTable 显示数据，RequestTableModel 作为表格数据模型。
2. 文本对比：调用 Diff\_match\_patch，HTML 渲染 Diff 结果。
3. 用户交互：
  - 按钮控制视图切换（单视图/对比视图）。
  - 右键菜单支持删除、标记、重新发送请求等功能。
4. 优化点：
  - SwingUtilities.invokeLater() 处理 UI 线程，避免界面卡顿。
  - diff\_linesToChars() 进行文本压缩，加快 diff 计算。

这份代码是 AuthAnalyzer 插件的核心组件，主要用于管理 HTTP 请求/响应比对，提供 可视化差异分析，是安全审计和渗透测试工具的重要功能模块。