

A HTML Browser application.

browser.xml :

```
<?xml version="1.0" encoding="UTF-8"?>
<shell text="Browser example" style="SWT.SHELL_TRIM">
    <grid-layout numcolumns="3" />
    <tool-bar style="SWT.BORDER">
        <grid-data h-span="3" h-align="SWT.FILL" />
        <tool-item text="Back" SWT.Selection=
"com.sheelapps.swtмл.examples.browser.ToolBarListener" />
        <tool-item text="Forward" SWT.Selection=
"com.sheelapps.swtмл.examples.browser.ToolBarListener" />
        <tool-item text="Stop" SWT.Selection=
"com.sheelapps.swtмл.examples.browser.ToolBarListener" />
        <tool-item text="Refresh" SWT.Selection=
"com.sheelapps.swtмл.examples.browser.ToolBarListener" />
        <tool-item text="Go" SWT.Selection=
"com.sheelapps.swtмл.examples.browser.ToolBarListener" />
    </tool-bar>
    <label text="Address">
</label>
    <text id="location" style="SWT.BORDER">
<grid-data h-align="SWT.FILL" h-span="2" h-grab="true" />
</text>
    <browser id="mybrowser" >
<grid-data h-align="SWT.FILL" v-align="SWT.FILL" h-span="3" h-grab="true" v-grab="true" />
</browser>
    <label id="status">
<grid-data h-align="SWT.FILL" h-grab="true" h-span="2" />
</label>
    <progress-bar id="progress-bar">
        <grid-data h-align="SWT.END" />
    </progress-bar>
</shell>
```

MyBrowser.java :

```
package com.sheelapps.swtмл.examples.browser;

import org.eclipse.swt.SWT;
import org.eclipse.swt.browser.Browser;
import org.eclipse.swt.browser.LocationEvent;
import org.eclipse.swt.browser.LocationListener;
import org.eclipse.swt.browser.ProgressEvent;
import org.eclipse.swt.browser.ProgressListener;
import org.eclipse.swt.browser.StatusTextEvent;
import org.eclipse.swt.browser.StatusTextListener;
import org.eclipse.swt.widgets.Display;
import org.eclipse.swt.widgets.Event;
import org.eclipse.swt.widgets.Label;
import org.eclipse.swt.widgets.Listener;
import org.eclipse.swt.widgets.ProgressBar;
import org.eclipse.swt.widgets.Shell;
import org.eclipse.swt.widgets.Text;
```

```
import com.sheelapps.swtml.SWTObjectMap;
import com.sheelapps.swtml.SWTParser;
```

```
public class MyBrowser {
```

```
    public static void main(String[] args) {
        MyBrowser t = new MyBrowser();
        t.openShell("/com/sheelapps/swtml/examples/browser/browser.xml", args);
    }
```

```
    private Shell shell;
```

```
    public void openShell(String file, String[] args) {
        Display display = Display.getDefault();
```

```
        try {
            // Instance of SWTML parser
            SWTParser parser = new SWTParser();
            // load common resources
            //
```

```
            parser.doParse(MyBrowser.class.getResourceAsStream("/com/sheelapps/swtml/examples/resources.xml"));
```

```
            // Root object is Shell in XML
            shell = (Shell) parser.doParse(MyBrowser.class.getResourceAsStream(file));
            shell.open();
            shell.setSize(800,600);
            doWork();
            while (!shell.isDisposed()) {
                if (!display.readAndDispatch())
                    display.sleep();
            }
            display.dispose();
        } catch (Exception e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
```

```
}
```

```
    protected void doWork() {
        Browser browser = getBrowser();
        getLocation().addListener(SWT.DefaultSelection, new Listener() {
            public void handleEvent(Event e) {
                getBrowser().setUrl(getLocation().getText());
            }
        });
    }
```

```

    }
});

browser.addListener(new ProgressListener() {
    public void changed(ProgressEvent event) {
        if (event.total == ) return;
        int ratio = event.current * 100 / event.total;
        getProgressBar().setSelection(ratio);
    }
    public void completed(ProgressEvent event) {
        getProgressBar().setSelection();
    }
});

browser.addStatusTextListener(new StatusTextListener() {
    public void changed(StatusTextEvent event) {
        getStatus().setText(event.text);
    }
});

browser.addLocationListener(new LocationListener() {
    public void changed(LocationEvent event) {
        if (event.top) getLocation().setText(event.location);
    }
    public void changing(LocationEvent event) {
    }
});

}

public static Browser getBrowser() {

    return (Browser) SWTObjectMap.getInstance().getObjectMap().get("mybrowser");
}

public static Text getLocation() {
    return (Text)SWTObjectMap.getInstance().getObjectMap().get("location");
}

public static ProgressBar getProgressBar() {
    return (ProgressBar)SWTObjectMap.getInstance().getObjectMap().get("progress-bar");
}

public static Label getStatus() {
    return (Label)SWTObjectMap.getInstance().getObjectMap().get("status");
}

}

```

ToolBarListener.java :

```
package com.sheelapps.swtмл.examples.browser;
```

```
import org.eclipse.swt.browser.Browser;
```

```

import org.eclipse.swt.widgets.Event;
import org.eclipse.swt.widgets.Listener;
import org.eclipse.swt.widgets.Text;
import org.eclipse.swt.widgets.ToolItem;

public class ToolBarListener implements Listener {

    public void handleEvent(Event event) {
        ToolItem item = (ToolItem)event.widget;
        String string = item.getText();
        Browser browser = MyBrowser.getBrowser();

        Text location = MyBrowser.getLocation();
        if (string.equals("Back")) browser.back();
        else if (string.equals("Forward")) browser.forward();
        else if (string.equals("Stop")) browser.stop();
        else if (string.equals("Refresh")) browser.refresh();
        else if (string.equals("Go")) browser.setUrl(location.getText());
    }
}

```



