/\*

\* FirewallOptionPane.java - Firewall options panel

\* :tabSize=8:indentSize=8:noTabs=false:

\* :folding=explicit:collapseFolds=1:

\*

\* Copyright (C) 1999 Dirk Moebius

\* Portions copyright (C) 2002 Slava Pestov

\*

\* This program is free software; you can redistribute it and/or

\* modify it under the terms of the GNU General Public License

\* as published by the Free Software Foundation; either version 2

\* of the License, or any later version.

\*

\* This program is distributed in the hope that it will be useful,

\* but WITHOUT ANY WARRANTY; without even the implied warranty of

\* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the

\* GNU General Public License for more details.

\*

\* You should have received a copy of the GNU General Public License

\* along with this program; if not, write to the Free Software

\* Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.

\*/

package org.gjt.sp.jedit.options;

//{{{ Imports

import java.awt.event.\*;

import javax.swing.\*;

import org.gjt.sp.jedit.\*;

//}}}

public class FirewallOptionPane extends AbstractOptionPane {

//{{{ FirewallOptionPane constructor

public FirewallOptionPane()

{

super("firewall");

} //}}}

//{{{ \_init() method

public void \_init()

{

// checkbox

addComponent(httpEnabled = new JCheckBox(jEdit.getProperty(

"options.firewall.http.enabled")));

// proxy host

addComponent(jEdit.getProperty("options.firewall.http.host"),

httpHost = new JTextField(jEdit.getProperty("firewall.host"), 15));

// proxy port

addComponent(jEdit.getProperty("options.firewall.http.port"),

httpPort = new JTextField(jEdit.getProperty("firewall.port"), 15));

// proxy username

addComponent(jEdit.getProperty("options.firewall.http.user"),

httpUser = new JTextField(jEdit.getProperty("firewall.user"), 15));

// proxy password

addComponent(jEdit.getProperty("options.firewall.http.password"),

httpPass = new JPasswordField(jEdit.getProperty("firewall.password"), 15));

// no proxy for

addComponent(jEdit.getProperty("options.firewall.http.nonProxy"),

httpNonProxy = new JTextField(jEdit.getProperty("firewall.nonProxyHosts"), 15));

boolean enabled = jEdit.getBooleanProperty("firewall.enabled");

httpEnabled.setSelected(enabled);

httpHost.setEnabled(enabled);

httpPort.setEnabled(enabled);

httpUser.setEnabled(enabled);

httpPass.setEnabled(enabled);

httpNonProxy.setEnabled(enabled);

httpEnabled.addActionListener(new ActionHandler());

// checkbox

addComponent(socksEnabled = new JCheckBox(jEdit.getProperty(

"options.firewall.socks.enabled")));

// proxy host

addComponent(jEdit.getProperty("options.firewall.socks.host"),

socksHost = new JTextField(jEdit.getProperty("firewall.socks.host"), 15));

// proxy port

addComponent(jEdit.getProperty("options.firewall.socks.port"),

socksPort = new JTextField(jEdit.getProperty("firewall.socks.port"), 15));

enabled = jEdit.getBooleanProperty("firewall.socks.enabled");

socksEnabled.setSelected(enabled);

socksHost.setEnabled(enabled);

socksPort.setEnabled(enabled);

socksEnabled.addActionListener(new ActionHandler());

} //}}}

//{{{ \_save() method

public void \_save() {

jEdit.setBooleanProperty("firewall.enabled", httpEnabled.isSelected());

jEdit.setProperty("firewall.host", httpHost.getText());

jEdit.setProperty("firewall.port", httpPort.getText());

jEdit.setProperty("firewall.user", httpUser.getText());

jEdit.setProperty("firewall.password", new String(httpPass.getPassword()));

jEdit.setProperty("firewall.nonProxyHosts", httpNonProxy.getText());

jEdit.setBooleanProperty("firewall.socks.enabled", socksEnabled.isSelected());

jEdit.setProperty("firewall.socks.host", socksHost.getText());

jEdit.setProperty("firewall.socks.port", socksPort.getText());

} //}}}

//{{{ Private members

private JCheckBox httpEnabled;

private JTextField httpHost;

private JTextField httpPort;

private JTextField httpUser;

private JPasswordField httpPass;

private JTextField httpNonProxy;

private JCheckBox socksEnabled;

private JTextField socksHost;

private JTextField socksPort;

//}}}

//{{{ ActionHandler class

class ActionHandler implements ActionListener

{

public void actionPerformed(ActionEvent evt)

{

httpHost.setEnabled(httpEnabled.isSelected());

httpPort.setEnabled(httpEnabled.isSelected());

httpUser.setEnabled(httpEnabled.isSelected());

httpPass.setEnabled(httpEnabled.isSelected());

httpNonProxy.setEnabled(httpEnabled.isSelected());

socksHost.setEnabled(socksEnabled.isSelected());

socksPort.setEnabled(socksEnabled.isSelected());

}

}

}