Builder (IcsFileImporter)

Embora não tenha construtor tem métodos para mudar o objeto

IcsFileImporter.java

static class CalendarEditorPage implements WizardPage {

private File myFile;

private JPanel myPanel = new JPanel();

private List<CalendarEvent> myEvents;

private void setFile(File f) {

myFile = f;

}

void setEvents(List<CalendarEvent> events) {

myEvents = events;

}

List<CalendarEvent> getEvents() {

return myEvents;

}

public String getTitle() {

return

ourLocalizer.formatText("impex.ics.previewPage.title");

}

public JComponent getComponent() {

return myPanel;

}

public void setActive(AbstractWizard wizard) {

if (wizard != null) {

myPanel.removeAll();

if (myFile != null && myFile.exists() &&

myFile.canRead()) {

if (myEvents != null) {

myPanel.add(new

CalendarEditorPanel(wizard.getUIFacade(), myEvents,

null).createComponent());

return;

} else {

LOGGER.error("No events found in file {}", new

Object[]{myFile}, Collections.emptyMap(), null);

}

} else {

LOGGER.error("File {} is NOT readable", new

Object[]{myFile}, Collections.emptyMap(), null);

}

myPanel.add(new

JLabel(ourLocalizer.formatText("impex.ics.filePage.error.noEve

nts", myFile.getAbsolutePath())));

}

}

}

Facade (ImporterFromCsvFile)

ImporterFromCsvFile.java

public void run() {

File selectedFile = getFile();

BufferProject bufferProject = new BufferProject(getProject(),

getUiFacade());

GanttCSVOpen opener = new GanttCSVOpen(selectedFile,

bufferProject.getTaskManager(),

bufferProject.getHumanResourceManager(),

bufferProject.getRoleManager(),

bufferProject.getTimeUnitStack());

opener.setOptions(((GanttProject)getProject()).getGanttOptions

().getCSVOptions());

try {

List<Pair<Level, String>> errors = opener.load();

importBufferProject(getProject(), bufferProject,

BufferProjectImportKt.asImportBufferProjectApi(getUiFacade()),

myMergeResourcesOption, null);

reportErrors(errors, "CSV");

} catch (Exception e) {

getUiFacade().showErrorDialog(e);

}

}

Proxy (DialogBuilder)

DialogBuilder.java

public Dialog createDialog(Component content, Action[]

buttonActions, String title, final NotificationManager

notificationManager) {

final JDialog dlg = new JDialog(myMainFrame, true);

final DialogImpl result = new DialogImpl(dlg, myMainFrame,

notificationManager);

dlg.setTitle(title);

dlg.getContentPane().setLayout(new BorderLayout());

dlg.getContentPane().add(content, BorderLayout.CENTER);

final Commiter commiter = new Commiter();

Action cancelAction = null;

int buttonCount = 0;

if (buttonActions.length > 0) {

JPanel buttonBox = new JPanel(new GridLayout(1,

buttonActions.length, 5, 0));

for (final Action nextAction : buttonActions) {

JButton nextButton = null;

if (nextAction instanceof OkAction) {

final JButton \_btn = new JButton();

final AbstractAction \_delegate = (AbstractAction)

nextAction;

OkAction proxy = new OkAction() {

// These two steps handel the case when focus is

somewhere in text input

// and user hits Ctrl+Enter

// First we want to move focus to OK button to allow

focus listeners, if any,

// to catch focusLost event

// Second, we want it to happen before original

OkAction runs

// So we wrap original OkAction into proxy which

moves focus and schedules "later" command

// which call the original action. Between them EDT

sends out focusLost events.

final Runnable myStep2 = new Runnable() {

@Override

public void run() {

result.hide();

commiter.commit();

nextAction.actionPerformed(null);

\_delegate.removePropertyChangeListener(myDelegateListener);

}

};

final Runnable myStep1 = new Runnable() {

@Override

public void run() {

\_btn.requestFocus();

SwingUtilities.invokeLater(myStep2);

}

};

@Override

public void actionPerformed(final ActionEvent e) {

SwingUtilities.invokeLater(myStep1);

}

private void copyValues() {

for (Object key : \_delegate.getKeys()) {

putValue(key.toString(),

\_delegate.getValue(key.toString()));

}

setEnabled(\_delegate.isEnabled());

}

private PropertyChangeListener myDelegateListener =

new PropertyChangeListener() {

@Override

public void propertyChange(PropertyChangeEvent evt)

{

copyValues();

}

};

{

\_delegate.addPropertyChangeListener(myDelegateListener);

copyValues();

}

};

\_btn.setAction(proxy);

nextButton = \_btn;

if (((OkAction) nextAction).isDefault()) {

dlg.getRootPane().setDefaultButton(nextButton);

}

}