

Changing AutoConfig to False in NatTable Constructor & Adding default style and header menus using autoconfiguration meathod.

Here you will find general information about configuring NatTable and the concepts involved.

ConfigRegistry

This is a global object holding the following kinds of configuration

- Styling
- Editing
- Comparators for sorting
- Any piece of arbitrary information can be stored in this registry.

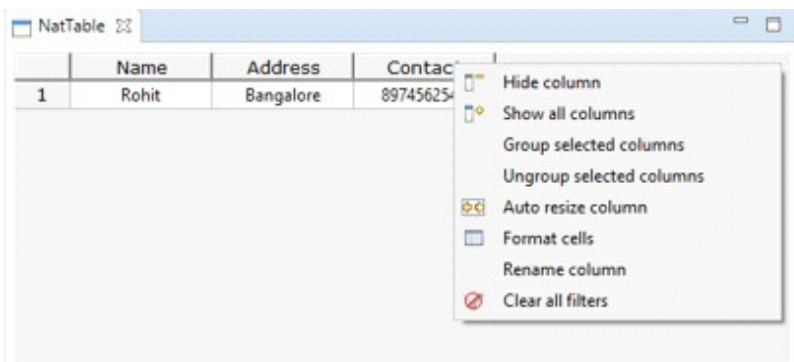
UiBindingRegistry

This is a global object holding the following kinds of configuration

- Key bindings
- Mouse bindings

```
NatTable natTable = new NatTable(composite, gridLayer, false);
GridDataFactory.fillDefaults().grab(true, true).applyTo(natTable);
natTable.addConfiguration(new DefaultNatTableStyleConfiguration());
natTable.addConfiguration(new HeaderMenuConfiguration(natTable));
natTable.configure();
```

Run the Application :



Adding Popup Menu to NatTable:

The PopupMenuBuilder is a builder in NatTable to create a menu with menu items that perform NatTable commands. It has several methods for adding such menu items and initializes and returns the menu on calling PopupMenuBuilder#build().

To create a menu with NatTable commands you need to perform the following steps:

- Create an IConfiguration for the menu by extending AbstractUiBindingConfiguration
- Create a menu using the NatTable PopupMenuBuilder helper class
- Register a PopupMenuAction binding using the created menu
- Add the IConfiguration to the NatTable instance

The following code shows the DebugMenuConfiguration that is shipped with NatTable to add debugging capability in a rendered NatTable.

```
natTable.addConfiguration(new AbstractUiBindingConfiguration() {
    private final Menu bodyMenu = new PopupMenuBuilder(natTable).withMenuItemProvider(new IMenuItemProvider() {

        @Override
        public void addMenuItem(NatTable natTable, Menu popupMenu) {
            MenuItem insertRow = new MenuItem(popupMenu, SWT.PUSH);
            insertRow.setText("Insert below");
            insertRow.setEnabled(true);

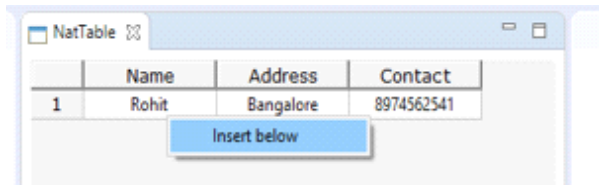
            insertRow.addSelectionListener(new SelectionAdapter() {
                @Override
                public void widgetSelected(SelectionEvent event) {
                    Contact contact = AddressbookFactory.eINSTANCE.createContact();
                    contact.setName("Sachin");
                    contact.setAddress("Mumbai");
                    contact.setContact("9874561237");

                    addressBook.getContacts().add(contact);
                    natTable.refresh();
                }
            });
        }
    }).build();

    @Override
    public void configureUiBindings(UiBindingRegistry uiBindingRegistry) {
        uiBindingRegistry.registerMouseDownBinding(
            new MouseEventMatcher(SWT.NONE, GridRegion.BODY, MouseEventMatcher.RIGHT_BUTTON),
            new PopupMenuAction(this.bodyMenu) {
                @Override
                public void run(NatTable natTable, MouseEvent event) {
                    super.run(natTable, event);
                }
            });
    }
});
natTable.configure();
```

Activate Windows
Go to Settings to activate Windows.

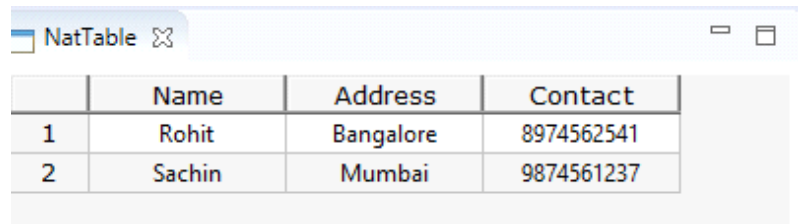
Run the Application :



The screenshot shows a window titled "NatTable" containing a table with the following data:

	Name	Address	Contact
1	Rohit	Bangalore	8974562541

Below the table is a blue button labeled "Insert below".



The screenshot shows a window titled "NatTable" containing a table with the following data:

	Name	Address	Contact
1	Rohit	Bangalore	8974562541
2	Sachin	Mumbai	9874561237

ABOUT ANCIT:

ANCIT Consulting is an Eclipse Consulting Firm located in the "Silicon Valley of Outsourcing", Bangalore. Offers professional Eclipse Support and Training for various Eclipse based Frameworks including RCP, EMF, GEF, GMF. Contact us on annamalai@ancitconsulting.com to learn more about our services.