

Mit Theme-Styling zu modern aussehenden SWT-Anwendungen

NatTable Enhanced

Mit der Version 1.1.0 des beliebten Tabellen-/Grid-/Tree-Widgets wurden diverse Fehler behoben und einige neue Features hinzugefügt. Das Besondere an diesem Release ist, dass zum ersten Mal größere Contributions aus der Community dem Projekt hinzugefügt und veröffentlicht wurden. Die größten neuen Features in diesem Release sind *ThemeConfigurations*, über die das Styling einer NatTable-Instanz an einer Stelle definiert und zur Laufzeit ausgetauscht werden kann, sowie dynamisch berechnete Summenwerte im GroupBy-Feature, Hover-Support und Layer-Kompositionen mit mehreren Viewports. Ein Überblick über die wichtigsten Neuerungen und ein Ausblick auf die zukünftige Entwicklung im NatTable-Projekt.

von Dirk Fauth

Wie oben bereits erwähnt, sind einige der großen neuen Features und Anpassungen in dieser Version der NatTable Contributions aus der Community. Dazu gehören der überarbeitete Suchdialog (Abb. 1) und die Anpassung der Behandlung des aktiven Editors. Der über CTRL+F zu öffnende Suchdialog ist jetzt an den Standardsuchdialog der Eclipse-IDE angelehnt und unterstützt erweiterte Suchfunktionen. Der aktive Editor in einer editierbaren NatTable wird nicht mehr in einem Singleton gehalten, sondern ist in der zugehörigen NatTable-Instanz als Eigenschaft zugreifbar. Dies ermöglicht die Verwendung der NatTable selbst als Editor-Control innerhalb einer NatTable. Auch die zur Laufzeit berechneten Summenwerte in Kombination mit der *groupBy*-Funktion wurden durch die Community der NatTable beigesteuert. Während die ersten beiden Neuerungen ohne weitere Konfiguration für alle Benutzer verfügbar sind, müssen die *groupBy*-Summenwerte konfiguriert werden (s. u.).

Theme Styling

Das Standardstyling der NatTable ist in der heutigen Zeit nicht mehr attraktiv. Die Stylekonfigurationen sind daher die am häufigsten angepassten Einstellungen. Bisher mussten hierfür in den verschiedensten Layern entsprechende Stylekonfigurationen angepasst werden, was schnell zu sehr unübersichtlichen Codestrukturen führte. Um das Konfigurieren der Darstellung leichter zu gestalten, wurden daher *ThemeConfigurations* eingeführt. Über *NatTable#setTheme(ThemeConfiguration)* kann das Styling einer NatTable-Instanz zur Laufzeit gesetzt werden. Dabei ist zu beachten, dass die *ThemeConfiguration* erst nach *NatTable#configure()* gesetzt wird. Ansonsten kann es passieren, dass die Einstellungen durch layerspezifische Stylekonfigurationen überschrieben werden. Da eine *ThemeConfiguration* in der Vererbungshierarchie von *IConfiguration* erbt, ist es auch möglich, eine *ThemeConfiguration* über *NatTable#addConfiguration(IConfiguration)* zu setzen. Davon wird allerdings aufgrund von Nebeneffekten mit anderen

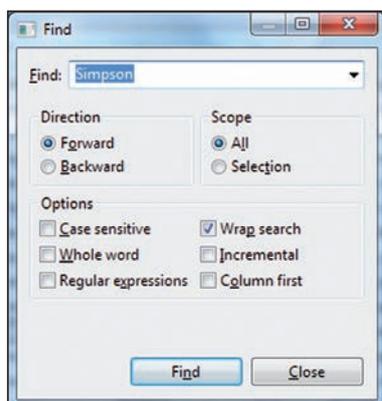


Abb. 1: Neuer NatTable-Suchdialog

	Firstname	Lastname	Age	Money	Married	Gender	BirthDay
1	Carlson [2] - (2)		51	1.620,37			
2	Homer	Carlson	46	772,35	<input type="checkbox"/>	MALE	Sat Jul 06 00:00:0...
3	Ned	Carlson	57	848,02	<input checked="" type="checkbox"/>	MALE	Sat Nov 02 00:00:...
4	Flanders [2] - (2)		26	904,96			
5	Waylon	Flanders	5	427,08	<input checked="" type="checkbox"/>	MALE	Thu Oct 29 00:00:...
6	Carl	Flanders	48	477,88	<input checked="" type="checkbox"/>	MALE	Fri Nov 25 00:00:...
7	Krabappel [1] - (...)		53	600,22			
8	Bart	Krabappel	53	600,22	<input checked="" type="checkbox"/>	MALE	Mon Jul 17 00:00:...
9	Leonard [1] - (1)		61	808,76			
10	Lisa	Leonard	61	808,76	<input checked="" type="checkbox"/>	FEMALE	Wed Jan 14 00:00:...
11	Lovejoy [2] - (2)		38	1.326,49			
12	Maggie	Lovejoy	31	347,54	<input type="checkbox"/>	FEMALE	Wed Oct 12 00:00:...
13	Marge	Lovejoy	45	978,95	<input type="checkbox"/>	FEMALE	Tue Jun 17 00:00:...
14	Simpson [1] - (1)		67	403,53			
15	Marge	Simpson	67	403,53	<input checked="" type="checkbox"/>	FEMALE	Mon Mar 31 00:00:...
16	Smithers [1] - (1)		80	635,45			
17	Carl	Smithers	80	635,45	<input checked="" type="checkbox"/>	MALE	Fri Dec 07 00:00:...
Σ			Avg: 49,30	6.299,78			

Abb. 2: DefaultNatTableThemeConfiguration

Stylekonfigurationen abgeraten. In NatTable sind drei Standard-Themes enthalten:

- *DefaultNatTableThemeConfiguration* (Abb. 2): das bekannte klassische Design
- *ModernNatTableThemeConfiguration* (Abb. 3): ein moderneres Design
- *DarkNatTableThemeConfiguration* (Abb. 4): ein auf dem modernen Design aufbauendes Dark Theme

Um eine eigene *ThemeConfiguration* zu erstellen, kann von der abstrakten Klasse *ThemeConfiguration* abgeleitet und es können die entsprechenden Methoden implementiert werden. Sinnvoller ist es allerdings, eine der bestehenden *ThemeConfigurations* zu erweitern und nur die dort definierten Attribute zu ändern, die angepasst werden sollen.

Um eine bestehende *ThemeConfiguration* zu erweitern, wurde das Konzept der *IThemeExtension* eingeführt. Dadurch können zusätzlich Konfigurationen, wie z. B. bedingte Formatierungen, einer bestehenden *ThemeConfiguration* hinzugefügt werden, ohne eine komplett neue Konfiguration erstellen zu müssen (Listing 1). Das *GroupBy*-Styling in der *GlazedLists* Extension ist z. B. über *IThemeExtensions* gelöst. Passend zu jedem Standard-Theme gibt es dort eine *IThemeExtension*, um keine Abhängigkeiten von NatTable Core zur *GlazedLists* Extension aufgrund von Styling aufzubauen.

GroupBy Enhancement

Wie bereits erwähnt, hat das *GroupBy*-Feature ein interessantes neues Feature spendiert bekommen. Die dynamische Berechnung von *GroupBy*-Summenwerten erlaubt es, zusätzliche Informationen zu einer Gruppierung anzuzeigen. Ähnlich der Verwendung des *SummaryRowLayers* muss hierfür lediglich ein *IGroupBySummaryProvider* (z. B. *SummationGroup-*

	Firstname	Lastname	Age	Money	Married	Gender	Birthday
1	Carlson [2]...		51	1.620,37			
2	Homer	Carlson	46	772,35	<input type="checkbox"/>	MALE	Sat Jul 06 00:00:...
3	Ned	Carlson	57	848,02	<input checked="" type="checkbox"/>	MALE	Sat Nov 02 00:...
4	Flanders [...]		26	904,96			
5	Waylon	Flanders	5	427,08	<input checked="" type="checkbox"/>	MALE	Thu Oct 29 00:...
6	Carl	Flanders	48	477,88	<input checked="" type="checkbox"/>	MALE	Fri Nov 25 00:0...
7	Krabappel...		53	600,22			
8	Bart	Krabappel	53	600,22	<input checked="" type="checkbox"/>	MALE	Mon Jul 17 00:...
9	Leonard [...]		61	808,76			
10	Lisa	Leonard	61	808,76	<input checked="" type="checkbox"/>	FEMALE	Wed Jan 14 00:...
11	Lovejoy [2]...		38	1.326,49			
12	Maggie	Lovejoy	31	347,54	<input type="checkbox"/>	FEMALE	Wed Oct 12 00:...
13	Marge	Lovejoy	45	978,95	<input type="checkbox"/>	FEMALE	Tue Jun 17 00:...
14	Simpson [...]		67	403,53			
15	Marge	Simpson	67	403,53	<input checked="" type="checkbox"/>	FEMALE	Mon Mar 31 00:...
16	Smithers [...]		80	635,45			
17	Carl	Smithers	80	635,45	<input checked="" type="checkbox"/>	MALE	Fri Dec 07 00:0...
Σ			Avg: 49,30	6.299,78			

Abb. 3: ModernNatTableThemeConfiguration

	Firstname	Lastname	Age	Money	Married	Gender	Birthday
1	Carlson [2]...		51	1.620,37			
2	Homer	Carlson	46	772,35	<input type="checkbox"/>	MALE	Sat Jul 06 00:00:...
3	Ned	Carlson	57	848,02	<input checked="" type="checkbox"/>	MALE	Sat Nov 02 00:...
4	Flanders [...]		26	904,96			
5	Waylon	Flanders	5	427,08	<input checked="" type="checkbox"/>	MALE	Thu Oct 29 00:...
6	Carl	Flanders	48	477,88	<input checked="" type="checkbox"/>	MALE	Fri Nov 25 00:0...
7	Krabappel...		53	600,22			
8	Bart	Krabappel	53	600,22	<input checked="" type="checkbox"/>	MALE	Mon Jul 17 00:...
9	Leonard [...]		61	808,76			
10	Lisa	Leonard	61	808,76	<input checked="" type="checkbox"/>	FEMALE	Wed Jan 14 00:...
11	Lovejoy [2]...		38	1.326,49			
12	Maggie	Lovejoy	31	347,54	<input type="checkbox"/>	FEMALE	Wed Oct 12 00:...
13	Marge	Lovejoy	45	978,95	<input type="checkbox"/>	FEMALE	Tue Jun 17 00:...
14	Simpson [...]		67	403,53			
15	Marge	Simpson	67	403,53	<input checked="" type="checkbox"/>	FEMALE	Mon Mar 31 00:...
16	Smithers [...]		80	635,45			
17	Carl	Smithers	80	635,45	<input checked="" type="checkbox"/>	MALE	Fri Dec 07 00:0...
Σ			Avg: 49,30	6.299,78			

Abb. 4: DarkNatTableThemeConfiguration

BySummaryProvider) für das Konfigurationsattribut *GroupByConfigAttributes#GROUP_BY_SUMMARY_PROVIDER* registriert werden (Listing 2). Hierfür können die neu eingeführten Labels verwendet werden:

- *GroupByDataLayer#GROUP_BY_SUMMARY*: gilt für eine ganze *GroupBy*-Zeile

Listing 1

```
class ConditionalStylingThemeExtension implements IThemeExtension {

    @Override
    public void registerStyles(IConfigRegistry configRegistry) {
        // add custom styling
        IStyle femaleStyle = new Style();
        femaleStyle.setAttributeValue(
            CellStyleAttributes.BACKGROUND_COLOR,
            GUIHelper.COLOR_YELLOW);
        femaleStyle.setAttributeValue(
            CellStyleAttributes.FOREGROUND_COLOR,
            GUIHelper.COLOR_BLACK);
        configRegistry.registerConfigAttribute(
            CellConfigAttributes.CELL_STYLE,
            femaleStyle,
            DisplayMode.NORMAL,
```

```
FEMALE_LABEL);
    }

    @Override
    public void unregisterStyles(IConfigRegistry configRegistry) {
        // unregister custom styling
        configRegistry.unregisterConfigAttribute(
            CellConfigAttributes.CELL_STYLE,
            DisplayMode.NORMAL,
            FEMALE_LABEL);
    }
    ...
    ThemeConfiguration conditionalDarkTheme = new
        DarkNatTableThemeConfiguration();
    conditionalDarkTheme.addThemeExtension(new
        ConditionalStylingThemeExtension());
    natTable.setTheme(conditionalDarkTheme);
```

Lastname > Firstname		Lastname	Age	Money
1	Carlson [16] - (8)		39	9,084.51
2	Jessica [2] - (2)		35	1,112.54
3	Jessica	Carlson	16	843.78
4	Jessica	Carlson	54	268.76
5	Lenny [5] - (5)		47	3,858.94
6	Lenny	Carlson	47	986.06
7	Lenny	Carlson	36	992.73
8	Lenny	Carlson	12	515.06
9	Lenny	Carlson	78	931.27
10	Lenny	Carlson	65	433.82
11	Lisa [3] - (3)		66	1,261.23
12	Lisa	Carlson	67	417.15
13	Lisa	Carlson	64	12.9
14	Lisa	Carlson	69	831.18
15	Maggie [1] - (1)		48	75.73
16	Maggie	Carlson	48	75.73
17	Marge [2] - (2)		7	1,025.76
18	Marge	Carlson	6	384.14
19	Marge	Carlson	9	641.62
20	Ned [1] - (1)		17	749.78
21	Ned	Carlson	17	749.78
22	Timothy [1] - (1)		14	170.92
23	Timothy	Carlson	14	170.92
24	Waylon [1] - (1)		27	829.61
25	Waylon	Carlson	27	829.61
26	Flanders [15] - (9)		59	7,194.83

Abb. 5: GroupBy-Summary-Beispiel

Stylinganpassungen

Für die vollständige Umsetzung des Theme-Stylings war es notwendig, verschiedene Anpassungen und Änderungen an bestehenden Stylingklassen vorzunehmen:

- Änderung von Paintern und Stylekonfigurationen, z. B. wurden im *ColumnGroupHeaderTextPainter* und im *TreeImagePainter* die Referenzen auf die Modellelemente entfernt. Stattdessen wird das Styling jetzt NatTable-typisch über Labels gesteuert. Benutzer, die an diesen Stellen eigene Labels in den Label-Stack schreiben, müssen darauf achten, dass durch diese Anpassung keine Änderung in der Darstellung auftritt.
- Einführung neuer Stylekonfigurationen, z. B. die Farbe der Grid-Linien oder ob Grid-Linien gezeichnet werden sollen.
- Anpassung der Implementierung des *VerticalTextPainter*. Die neue Implementierung nutzt *SWT.Transform* anstelle von gedrehten Images. Sollten bei der Benutzung Probleme auftauchen, ist es immer noch möglich, die alte Implementierung über den *VerticalTextImagePainter* zu verwenden.

Listing 2

```
configRegistry.registerConfigAttribute(
    GroupByConfigAttributes.GROUP_BY_SUMMARY_PROVIDER,
    new SummationGroupBySummaryProvider(columnPropertyAccessor),
    DisplayMode.NORMAL,
    GroupByDataLayer.GROUP_BY_COLUMN_PREFIX + 3);
```

- *GroupByDataLayer#GROUP_BY_SUMMARY_COLUMN_PREFIX + <column_index>*: gilt für die über *<column_index>* spezifizierte Spalte in einer *GroupBy*-Zeile

Zusätzlich ist es möglich, in der Gruppierungsspalte die Anzahl der Kindelemente anzeigen zu lassen. Hierfür muss ein Pattern, bestehend aus Platzhaltern, registriert werden (siehe folgende Codezeilen), das an das Ende des Gruppierungswerts angehängt wird. Der Platzhalter *{0}* wird dabei durch die Anzahl aller Kindelemente ersetzt (nur interessant bei mehrstufiger Gruppierung), während der Platzhalter *{1}* nur durch die Anzahl der direkten Kindelemente ersetzt wird.

```
configRegistry.registerConfigAttribute(
    GroupByConfigAttributes.GROUP_BY_CHILD_COUNT_PATTERN,
    "{0} - ({1})");
```

Um die Summenwerte im *GroupBy*-Feature nutzen zu können, ist es erforderlich, den *GroupByDataLayer*-Konstruktor zu nutzen, der eine *IConfigRegistry*-Instanz als Parameter erwartet. Er wurde zusammen mit dem Feature eingeführt, um Zugriff auf die Konfigurationen zu erhalten. **Abbildung 5** zeigt das *GroupBy*-Feature mit aktivierten Summenwerten.

Hover-Support

Der bislang fehlende Hover-Interaktionsmodus wurde über diverse Ergänzungen der NatTable hinzugefügt. Um spezielles Styling beim Hover über eine Zelle aktivieren zu können, wurden zwei neue *DisplayModes* eingeführt:

- *DisplayMode#HOVER*: wird gesetzt, wenn der Mauszeiger über eine Zelle bewegt wird
- *DisplayMode#SELECT_HOVER*: wird gesetzt, wenn der Mauszeiger über eine selektierte Zelle bewegt wird

Damit die neuen *DisplayModes* gesetzt und interpretiert werden, ist es erforderlich, den neu eingeführten *HoverLayer* in den Layer-Stack aufzunehmen. Neben dem Setzen des *DisplayMode#HOVER* übernimmt dieser auch das selektive Neuzeichnen der einzelnen Zellen, ohne dass ein kompletter Refresh notwendig ist.

Um Action Bindings zu ermöglichen, implementiert der interne *IMouseModeEventHandler* jetzt auch den *MouseTrackListener*. Dadurch können Aktionen auf *mouseHover*, *mouseEnter* und *mouseExit*-Events über die *UiBindingRegistry* registriert werden, wobei *mouseEnter* und *mouseExit* sich auf das NatTable-Control beziehen und nicht auf einzelne Zellen. Im Package *org.eclipse.nebula.widgets.nattable.hover.config* finden sich diverse Standardkonfigurationen, um verschiedenste Kompositionen zu unterstützen, z. B. die *ColumnHeaderResizeHoverBindings*, die die *HoverStylingAction* beim Bewegen der Maus über Zellen der *ColumnHeader*-Region registriert und sicherstellt, dass das Hover-Styling

wieder entfernt wird, wenn sich der Mauszeiger aus der *ColumnHeader*-Region und der *NatTable* bewegt.

Multiple Viewports

Um komplexere *NatTable*-Kompositionen zu ermöglichen, wurde der *ViewportLayer* erweitert. Damit ist es möglich, mehrere Viewports in einer *NatTable* einzusetzen, um mehrere unabhängig voneinander scrollbare Bereiche konfigurieren zu können (Abb. 6).

Eine detaillierte Beschreibung dieses Features würde ein eigenes Tutorial füllen. Daher an dieser Stelle nur in Kürze die notwendigen Schritte, um mehrere Viewports in eine *NatTable*-Komposition einzubinden:

- Erzeugen mehrerer *ViewportLayer*-Instanzen, die über einen *CompositeLayer* verbunden werden
- Setzen der Minimum-/Maximumwerte der *ViewportLayer*, z. B. über *ViewportLayer#setMaxColumnPosition(int)* und *ViewportLayer#setMinColumnPosition(int)*
- Definieren eines speziellen *ClientAreaAdapters*, der für die Scrollberechnungen des ersten *ViewportLayers* benötigt wird
- Setzen von benutzerdefinierten *Slidern*, um die Scrollbalken der *ViewportLayer* darzustellen; notwendig, da ein *Scrollable Composite* nur einen Scrollbalken unterstützt

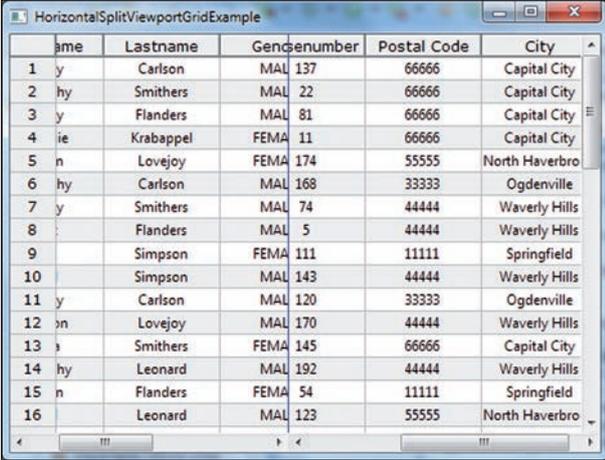
Da es aktuell noch keine Tutorials hierzu gibt, bleibt an dieser Stelle nur der Verweis auf die *NatTable*-Examples-Anwendung, die von der *NatTable*-Webseite [2] über Webstart gestartet werden kann. Alternativ finden sich die Beispiele natürlich auch im *NatTable*-Sourcecode.

Neben den hier aufgeführten Neuerungen gibt es noch eine Reihe weiterer nennenswerter Punkte. Dazu gehören neue Commands, um die Nodes einer Baumdarstellung auf einmal komplett zu- und aufzuklappen, Anpassungen an die Excel-Export-Funktionalität, neue Stylingkonfigurationen und Painter und die Lokalisierung diverser Konverter. Eine genauere Auflistung ist unter [1] zu finden.

Ausblick

In den folgenden Monaten wird der größte Kritikpunkt an *NatTable* in Angriff genommen: die spärliche Dokumentation. Mit einer Reihe von Tutorials soll diese Lücke endlich gefüllt werden. In der überarbeiteten *NatTable*-Examples-Anwendung wurden hierfür bereits zahlreiche neue Beispiele unter dem Punkt *Tutorial Examples* hinzugefügt (Abb. 7). Diese werden den Tutorials als Codebeispiele dienen und können bereits jetzt als Referenz zu Rate gezogen werden.

Die Version 1.1 wurde zum Zeitpunkt ihres Erscheinens als finale Version der 1.x-Architektur bekannt gegeben. Aufgrund diverser Rückmeldungen aus der Community und des nicht unbeträchtlichen Aufwands einer Neuimplementierung wurde dies allerdings wieder



	id	first	last	gender	postal	city
1	y	Carlson	MAL	137	66666	Capital City
2	hy	Smithers	MAL	22	66666	Capital City
3	y	Flanders	MAL	81	66666	Capital City
4	ie	Krabappel	FEMA	11	66666	Capital City
5	n	Lovejoy	FEMA	174	55555	North Haverbro
6	hy	Carlson	MAL	168	33333	Ogdenville
7	y	Smithers	MAL	74	44444	Waverly Hills
8		Flanders	MAL	5	44444	Waverly Hills
9		Simpson	FEMA	111	11111	Springfield
10		Simpson	MAL	143	44444	Waverly Hills
11	y	Carlson	MAL	120	33333	Ogdenville
12	on	Lovejoy	MAL	170	44444	Waverly Hills
13		Smithers	FEMA	145	66666	Capital City
14	hy	Leonard	MAL	192	44444	Waverly Hills
15	n	Flanders	FEMA	54	11111	Springfield
16		Leonard	MAL	123	55555	North Haverbro

Abb. 6: Split Viewport

verworfen. Stattdessen wird nun ein inkrementelles Update angestrebt, in dem einige Key-Features sukzessive eingebaut werden, wie z. B. die Unterstützung weiterer UI-Toolkits und die Einführung von Dependency Injection. Weitere Anpassungen, die die Architektur grundlegend beeinflussen, werden erst zu einem späteren Zeitpunkt in Angriff genommen, wie z. B. das Event Handling und die diversen internen Transformationen. Damit soll die Stabilität der *NatTable* gesichert werden und Benutzer, die ihre Anwendungen auf Basis von *NatTable* aufbauen, müssen diese nicht komplett neu schreiben. Dennoch muss mit diversen API-Änderungen gerechnet werden, z. B. im Bereich der Konfigurationen, um Farben oder Grafiken zu setzen. Außerdem wird durch dieses Vorgehen angestrebt, neue Beteiligte und Committer in das Projekt zu holen.

Die Roadmap, die das Vorgehen beschreibt, ist über die Homepage von *NatTable* [2] zu finden und befand sich zum Zeitpunkt, zu dem dieser Artikel entstand, noch in Arbeit.

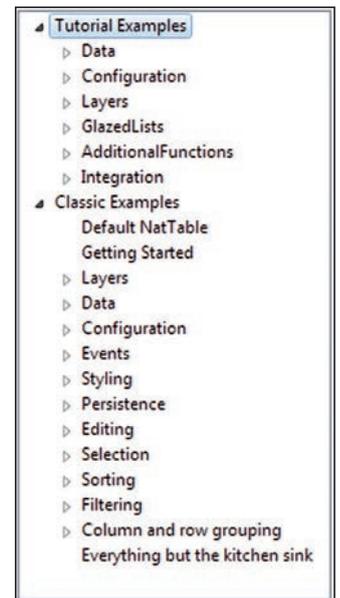


Abb. 7: NatTable-Examples-Struktur



Dirk Fauth ist Principal Consultant bei der codecentric AG in Stuttgart und seit mehreren Jahren im Bereich der Java-Entwicklung tätig. Er ist aktiver Committer und Co-Project-Lead im Nebula-NatTable-Projekt, Nebula-Committer und aktiver Contributor im Eclipse-4-Umfeld.

Links & Literatur

- [1] https://www.eclipse.org/nattable/nandn/nandn_110.php
- [2] <https://www.eclipse.org/nattable/>