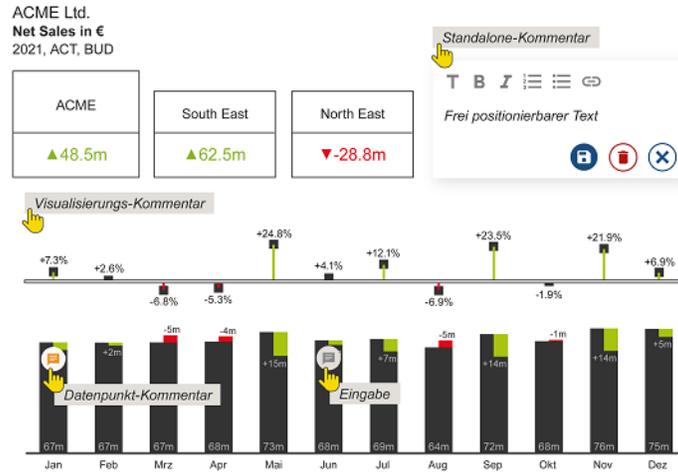


Handbuch	2
Einführung	3
Installation	4
Custom Widget	5
comments backend server	6
Installation Docker Compose	7
Installation Kubernetes	8
Active Directory Benutzer	9
Begriffserklärung	11
Datenbankstruktur Kommentare	12
Kontext	13
graphomate property sheet GPS	15
Data Tab	16
Labels Tab	17
Scaling Tab	19
Input Output Tab	20
Scripting Dokumentation	21
Admin UI	23
Comments	24
Contexts	25
Users	26
User Roles	27
User Groups	28
Authorizations	29
Keycloak	31
Interaktivität zur Laufzeit	33
Known Issues	34
Anhang	35
Changelog	36

Handbuch zur Nutzung der graphomate comments 2023.2.0 für SAP Analytics Cloud (SAC)



Version 2023.2.0 – Stand Juli 2023

<https://www.graphomate.com>

graphomate .ll



Inhaltsverzeichnis

- Einführung
- Installation
- Kontext
- graphomate property sheet GPS
- Scripting Dokumentation
- Admin UI
- Interaktivität zur Laufzeit
- Known Issues
- Anhang
- Changelog

Einführung

Kommentierung von Geschäftszahlen

Gute Visualisierungen helfen Daten besser zu verstehen und Muster oder Trends zu erkennen.

Mit unserer **graphomate suite** und den **graphomate tiles** für verschiedene BI-Tools haben wir unseren Kunden die Möglichkeit eröffnet, einen Visualisierungsstandard zu formulieren und Daten adressatengerecht und aussagekräftig zu visualisieren. Durch die Schaffung von "visuellem Kontext" können Entscheidungsträger die geschäftliche Situation einfacher und besser einordnen.

Ein weiterer Aspekt ist jedoch die Anreicherung von Berichten und Dashboards mit **textuellen Inhalten**. Man wünscht sich die Möglichkeit, erklärungsbedürftige Datenkonstellationen mit erläuterndem Text anzureichern.

Die **graphomate comments** schaffen diese Möglichkeit auf unterschiedlichen Ebenen:

1. Kommentare als eigenständiges Text-Objekt

Über eine eigene Kommentierungskomponente wird Platz vorgehalten, um erläuternden Text für eine bestimmte Datenkonstellation erfassen zu können. Die Kommentare werden mit dem Erfasser sowie dem Erstellungsdatum gespeichert und bei Wiederaufruf der Datenkonstellation wieder angezeigt. Zum Beispiel:

"Das EBIT für die gesamte Organisation ist im Monat Mai 2022 gesunken, da eine außerordentliche Abschreibung aufgrund eines Brandes in einer Lagerhalle verbucht wurde".

2. Kommentare zu einem Datenpunkt

Datenpunktkommentare beschreiben eine bestimmte Datenkonstellation in einem Diagramm oder einer Tabelle: „Die Erlöse im Monat Mai 2022 sind aufgrund der Fakturierung des Großprojekts 'Flughafen' um 20% höher.“

Die Existenz eines Kommentars wird über ein kleines Kommentar-Icon am Diagramm-Element bzw. in der Tabellenzelle angezeigt:

3. Kommentare zu einer Visualisierungskomponente

Weiterhin können Kommentare zu einer Visualisierung gespeichert werden.

Nicht ein einzelner Datenpunkt, sondern z.B. ein zeitliche Entwicklung soll als Ganzes im Titel oder im Footer der Komponente kommentiert werden. Diese werden entweder direkt vollständig oder in Form eines Icons angezeigt.

Die Eingabe der Kommentare geschieht immer über ein **identisches Eingabefeld**, das sich auf Anforderung öffnet.

Wichtig: Voraussetzung für die Erfassung von Kommentaren ist die Definition eines sog. **Kontexts**. Dieser Kontext beschreibt den Raum, der aus Dimensionen und Kennzahlen sowie weitere Restriktionen für die graphomate comments aufgespannt wird. Innerhalb dieses Raumes können dann sämtliche Ausprägungen der gewählten Dimensionen und Kennzahlen frei kommentiert werden. Die graphomate comments bieten Ihnen darüber hinaus die Möglichkeit, Nutzer über einen Kommentar per Mail zu informieren.

Im Verlauf des 2. Halbjahres 2023 wollen wir inhaltliche Berechtigungen auf Kontextebene abbilden und ein Workflow-Konzept - beispielsweise zur Genehmigung von Kommentaren - angehen.

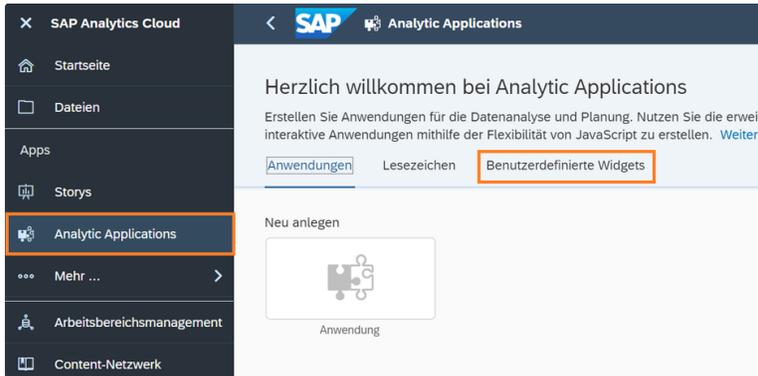
Installation

Custom Widget

In der SAP Analytics Cloud werden die graphomate extensions als sogenannte 'Benutzerdefinierte Widgets' installiert. Für die Installation müssen Sie Administrator-Rechte besitzen.

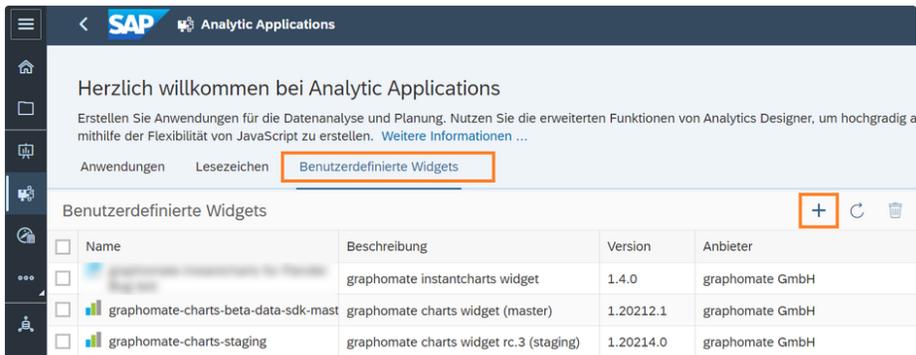
1. Schritt

Navigieren Sie zur Sidebar und wählen Sie den Eintrag "Analytic Applications"



2. Schritt

Wechseln Sie nun auf den Reiter "Benutzerdefinierte Widgets". Über das '+'-Symbol wählen Sie die von uns bereit gestellte JSON-Dateien zur Installation der graphomate widgets aus.



Die graphomate widgets können nun in Analytic Applications und Storys als 'Benutzerdefinierte Widgets' verwendet werden.

comments backend server

Installationsbundle

Es existieren zwei unterschiedliche Installationsbundles. Eines dient der Installation in einer reinen Docker-Umgebung per docker-compose. Das Andere ist zur Installation in einer Umgebung mit Kubernetes gedacht, bspw. Open Shift.

Wichtig:

Der bei der Installation erstellte **admin** Benutzer wird gleichzeitig als Service-User verwendet. Das bedeutet, er darf nicht gesperrt werden und wenn sein Passwort in Keycloak geändert wird, muss das Passwort danach in der Docker oder Kubernetes Konfiguration angepasst werden.

- [Installation Docker Compose](#)
- [Installation Kubernetes](#)

Installation Docker Compose

Unser Installationsbundle für den Kommentierungsserver besteht aus mehreren Dateien:

.env

In dieser Datei werden Umgebungsvariablen festgelegt, die definieren unter welchem SSL-Port die Anwendung laufen soll und wie die Dateien heißen, in denen das (von Ihnen für Ihre Domain bereitgestellte) SSL Zertifikat und der SSL Key liegen.

***_password.txt**

In zwei Dateien, die mit "_password.txt" enden, wird festgelegt, welche Passwörter (initial) für die Datenbank und den globalen Administrator-User verwendet werden sollen.

config.properties

Hier werden Einstellungen für unseren Kommentierungsserver festgelegt. Aktuell findet dort die Konfiguration des SMTP Mail-Servers statt, der dazu verwendet wird Benachrichtigungsemails zu versenden, wenn ein Benutzer in einem Kommentar erwähnt wurde.

init-commenting-db.sql

Die Initialisierung des Postgres-Servers mit einer neuen Datenbank und zwei Schemata ist hier definiert. Hier ist vom Benutzer nichts anzupassen.

nginx.conf

In der Konfigurationsdatei befindet sich die Definition, wie der NGINX Reverse Proxy die Anfragen gegen das Kommentierungsbackend an die einzelnen Docker Container weiterleiten soll. Hier ist in der Regel vom Benutzer nichts anzupassen.

docker-compose.yaml

Die Definition welche Container für das Kommentierungsbackend erstellt werden müssen, welche Dateien vom Host-System verwendet werden dürfen und wie das interne Netzwerk aufgebaut ist muss in der Regel nicht angepasst werden.

commenting.war

In dieser Datei liegt die Logik für das Kommentierungsbackend, sie wird automatisch in dem Tomcat-Container in Docker installiert.

Installationsvorgang

1. Installieren Sie Docker wie hier beschrieben: [📄 Install Docker Engine](#)
2. Entpacken Sie das Installationsbundle und die Dateien aus dem Installationsbundle kopieren Sie auf das Docker-Hostsystem an einen beliebigen Ort.
3. Sie fügen Ihr SSL Zertifikat mit dem Dateinamen "ssl_certificate.crt" und den dazugehörigen privaten Schlüssel mit dem Dateinamen "ssl_certificate.pem" an dem selben Ort wie in Schritt 1 ein. Bei unterschiedlichen Pfaden oder Dateinamen, muss die Datei ".env" angepasst werden.
4. Wenn sie einen anderen Port als "3333" für den Server verwenden möchten, so passen Sie die in der Datei ".env" an.
5. In der "commenting_db_password.txt" Datei definieren Sie das Passwort für die Datenbank der Userverwaltung und der Kommentarablage. Dieses wird intern an die entsprechenden Container übermittelt.
6. In der Datei "keycloak_admin_password.txt" definieren Sie das Passwort für den Administrator des Kommentarbackends. Dieser besitzt die Berechtigung Kommentare zu lesen, zu schreiben, zu bearbeiten, zu löschen und neue Benutzer zu definieren. Sie benötigen dessen Passwort also um sich initial in unsere Komponenten und die Verwaltungsoberfläche einzuloggen. Der Benutzername lautet hier "admin". Neu angelegten Benutzern können dann die Rollen "viewer", "editor" oder "admin" zugewiesen werden.
7. Ist die Konfiguration vollständig, so können Sie in der Kommandozeile in dem Ordner des Installationsbundles den Befehl "docker-compose up" ausführen. Damit werden alle benötigten Komponenten gestartet.

Installation Kubernetes

Unser Installationsbundle für den Kommentierungsserver auf Kubernetes besteht aus mehreren Dateien:

Im Folgenden gehen wir davon aus, dass Ihnen bekannt ist, wie Dockerfiles und .yaml Konfigurationsdateien in Kubernetes zu verwenden sind.

Dockerfile*

In den drei Dockerfiles befinden sich die Anweisungen, wie die drei für die Kommentierung benötigten Container erstellt werden. Die so erstellten Container müssen dann in Ihre docker-registry gepusht werden um sie in Kubernetes zu nutzen.

Ordner comments-keycloak-theme, commenting.jar, nginx.conf, keycloak-docker-entrypoint.sh

Diese Dateien werden von den Dockerfiles verwendet um die einzelnen Container zu erstellen. Eine Modifikation ist hier in der Regel nicht notwendig.

-kubernetes-.yaml

In den fünf YAML-Dateien befinden sich die Konfigurationen für Kubernetes um unseren Service starten zu können. Diese dienen als Orientierung und es können je nach Ihrer Kubernetes-Konfiguration Anpassungen nötig sein. In der **0-kubernetes-ns.yaml** ist der Namespace definiert, in dem die graphomate comments Entitäten in Kubernetes erstellt werden. Die **1-kubernetes-role.yaml** definiert die Rollen die notwendig sind, damit die graphomate services die Konfigurationen lesen dürfen. Die Datei **2-kubernetes-config.yaml** definiert unterschiedliche Konfigurationsparameter für den unseren Service. Dazu gehören die Verbindungsdaten für die Datenbanken in denen die Kommentare und die Benutzer gespeichert werden. Diese können auch in der selben Datenbank gespeichert werden. Dafür sind in PostgreSQL zwei Schemata notwendig. Bisher konnten wir auch erfolgreich MSSQL auf Azure ohne Schemata testen. Die Konfiguration dafür ist in etwas komplexer als die für PostgreSQL und kann bei Bedarf durch uns erklärt werden. Auch die Definition für den Mailserver werden hier getätigt. Dieser wird verwendet um einem Benutzer der in einem Kommentar erwähnt wurde darüber per Mail zu benachrichtigen. In der **3-kubernetes-secrets.yaml** werden Zugangsdaten für die Datenbanken, den Mailserver und den initialen Administrator gepflegt. Diese sind wie für Kubernetes üblich Base64 encodiert. In **4-kubernetes-deployment.yaml** werden die drei für unseren Service benötigten Container Konfiguriert. Diese sind in unserem Beispiel alle in einem Pod vereint und verwenden temporäre in-memory Volumes. Es steht ihnen natürlich frei, andere, beispielsweise persistente, Volumes zu verwenden und die Container für besseres Load-Balancing in unterschiedlichen Pods zu verteilen. Dafür ist allerdings im Fall von dem keycloak-Container eine relative komplexe Konfiguration und eine Anpassung der nginx.conf für den Nginx-Container notwendig, auf die wir in diesem Rahmen nicht eingehen wollen.

Installationsvorgang

Die drei Docker-Container müssen mit den von uns ausgelieferten Dockerfiles erstellt und in Ihre von kubernetes verwendete Docker-Registry gepusht werden. Nachdem die Konfigurationsparameter und Secrets angepasst wurden sollten diese in Kubernetes angewendet werden. Außer der Erstellung einer Datenbank und der Schematas übernimmt unser Service die restliche Konfiguration der Datenbank selbst, sobald der Service das erste Mal gestartet wird. Vor den Port 80 des Nginx-Containers sollten Sie bspw. einen Ingress-Controller setzen, der die SSL Terminierung übernimmt und den Port nach außen mit SSL gesichert exponiert. Weitere Ports auf unseren Service sind nicht notwendig, da der Nginx-Container hier als Reverse-Proxy fungiert. Wenn Sie den Service erfolgreich installiert haben, können Sie drei Dinge tun um dessen Funktionsfähigkeit zu überprüfen. An der Root-URL auf den Service, sollten ein JSON mit Informationen zu dem Service und der installierten Version erscheinen. <Server URL>/auth sollte die Bedienoberfläche von Keycloak öffnen, in der Sie sich mit den Daten des initialen Admin-Benutzers einloggen können, wenn Sie auf "Administration Console" klicken. Hier können Sie auch andere Ident-Provider an unseren Service anbinden. Bspw. sind hier Active Directory per SAML und externe OAuth Provider zu nennen. Für weitere Informationen hierzu konsultieren Sie bitte die Dokumentation von Keycloak. Als letztes überprüfen Sie bitte, ob unter <Server URL>/commenting/administration/ unsere Bedienoberfläche für das Backend erscheint. Hier können Sie, mit einem gültigen Benutzer (bspw. wieder der Admin), bestehende Kommentare einsehen, diese bearbeiten und auch neue erstellen. Gleichzeitig haben Sie die Möglichkeit neue Benutzer zu erstellen und diese auf Kontextkombinationen zu berechnen.

Active Directory Benutzer

Um Benutzer an unser *comments backend* anzubinden, die über Microsoft Active Directory verwaltet werden, ist die Nutzung von Active Directory Federation Services (AD FS) und SAML v2.0 notwendig. Haben Sie bereits AD FS auf Ihrem Server eingerichtet, so folgen Sie bitte folgender Anleitung, um ihr AD an die *comments* anzubinden: [Configure MS ADFS as a brokered Identity Provider in KeyCloak](#)

Die dafür zu notwendige Verwaltung von Keycloak erreichen sie unter: https://<SERVER_URL>/auth

Der unter "SETUP IDENTITY PROVIDER IN KEYCLOAK" zu findende Schritt 6 ist optional und bietet in unserem Fall die Möglichkeit bestehende Attribute in SAML auf die drei in den *comments* definierten Rollen zu mappen. Im letzten Feld sollte in unserem Fall dann eben statt "manager" entweder "viewer", "editor" oder "admin" stehen.

Wenn Sie ansonsten der Anleitung gefolgt sind, so gibt es beim Login in die *comments* nun eine weitere Option:

A screenshot of a login form titled "Sign in to your account". It contains two input fields: "Username or email" and "Password". Below the fields is a blue "Sign In" button. Underneath the button, there is a horizontal line, followed by the text "Or sign in with", and then a button labeled "ActiveDirectory".

Wenn Sie nun die untere Option auswählen, so können Sie sich (falls Sie nicht bereits eingeloggt sind) mit einem Benutzer aus dem Active Directory einloggen. Beim ersten Login fragt ein Formular den Vornamen und Nachnamen des zu importierenden Nutzers ab. Um dies zu verhindern sind weitere Mapper notwendig. Die dafür benötigte Konfiguration in der AD FS Konsole sieht wie folgt aus:

Edit Rule - email, subject, firstname, lastname

You can configure this rule to send the values of LDAP attributes as claims. Select an attribute store from which to extract LDAP attributes. Specify how the attributes will map to the outgoing claim types that will be issued from the rule.

Claim rule name:

Rule template: Send LDAP Attributes as Claims

Attribute store:

Mapping of LDAP attributes to outgoing claim types:

	LDAP Attribute (Select or type to add more)	Outgoing Claim Type (Select or type to add more)
▶	E-Mail-Addresses	E-Mail-Adresse
	SAM-Account-Name	UPN
	Given-Name	Angebener Name
	Surname	Nachname
•		

View Rule Language...

In Keycloak müssen dem SAML Identity Provider noch zwei weitere Mapper hinzugefügt werden:

ID:

Name:

Sync Mode Override:

Mapper Type:

Attribute Name:

Friendly Name:

User Attribute Name:

ID:

Name:

Sync Mode Override:

Mapper Type:

Attribute Name:

Friendly Name:

User Attribute Name:

Nach dieser Konfiguration ist nun nach Login eines Active Directory Nutzers keine weitere Dateneingabe nötig.

Automatischer Login ohne manuelle Auswahl des neuen Identity Providers

Nach dieser Konfiguration muss ein Nutzer beim Login manuell den neuen Identity Provider vom Active Directory auswählen. Um diesen Schritt zu überspringen folgen Sie bitte dem Schritt "default identity provider" in folgender Anleitung: [Server Administration G](#)

[uide](#)

Begriffserklärung

Docker

Docker ist ein Open-Source Projekt, das dazu dient Software in einer isolierten Umgebung - einem sogenannten Container - auszuführen. Das heißt, Software die innerhalb von einem Container ausgeführt wird teilt sich die Basisfunktionalität des eigentlichen Betriebssystems, ist aber ansonsten was das Dateisystem und den Arbeitsspeicher angeht von dem restlichen Betriebssystem und anderen Containern isoliert. Das bietet den Vorteil, dass Software die innerhalb eines Docker-Containers ausgeführt wird, immer in einer fest definierten Umgebung mit fest definierten und installierten Abhängigkeiten ausgeführt wird.

Der Docker-Host, der notwendig ist um Docker-Container auszuführen, kann auf jedem beliebigen Linux, Windows oder Mac OSX Server installiert werden. Da jedoch immer eine Linux "Zwischenschicht" notwendig ist, ist ein Linux Server zu empfehlen. Weitere Informationen zur Installation von Docker find Sie hier: [🌐 Install Docker Engine](#)

Docker Container

Docker bietet eine Vielzahl von vordefinierten Containern für bestimmte Anwendungszwecke. Diese Container werden oft von den Entwicklern einer Software selbst gepflegt. Beispielsweise gibt es einen offiziellen Container-Abbild von Postgres, in dem die gleichnamige Datenbank ausgeführt wird. Um einen Docker Container zu konfigurieren exponiert dieser ein vordefiniertes Set von Einstellungen, die von außen an den Container weitergegeben werden können.

Docker Compose

Mit der Beschreibungssprache "Docker Compose" kann konfiguriert werden, wie eine Anwendung, die aus mehreren Docker Containern besteht, aufgebaut ist. Hier ist beispielsweise definiert, welche Container inkl. ihrer Konfiguration für die gesamte Anwendung ausgeführt werden müssen, wie die Container miteinander in einem virtuellen Netzwerk kommunizieren, welche Dateien vom Host-System im Container zur Verfügung stehen sollen und welche Passwörter verwendet werden sollen.

Datenbankstruktur Kommentare

comment

- id: UUID;
- createdAt: Timestamp;
- changedAt: Timestamp;
- createdBy: string;
- lastModifiedBy: string;
- text: Lob;
- dashboardId: string;
- dashboardName: string;
- hostId: string;
- hostName: string;
- contexts: Foreign Key Context[];

context

- id: UUID;
- createdAt: Timestamp;
- changedAt: Timestamp;
- createdBy: string;
- lastModifiedBy: string;
- type: "DataContext" | "EnvironmentContext";
- key: string;
- value: string;
- keyText: string <optional for type DataContext>;
- valueText: string <optional for type DataContext>;

Beispiel Tabellenzeilen

comments

	id	created_by	last_modified_by	changed_at	created_at	dashboard_id	dashboard_name	host_id	host_name	text	plain_text	html_text	encode	encode
1	178051cc6744-4366-9f5e-4577a6071...	admin	admin	2022-11-21 13:42:37.827812	2022-11-21 13:42:37.827812	[null]	[null]	[null]	[null]	18046	18045	[null]	*blocks*	text

contexts

	type	id	created_by	last_modified_by	changed_at	created_at	key	value	key_text	value_text	comment_id
1	DataContext	1d79ad5c-df3f-4204-8cd3...	admin	admin	2022-11-21 13:42:37.961738	2022-11-21 13:42:37.845847	genre	Action	[null]	[null]	178051cc6744-4366...
2	DataContext	a7ec342f-78c2-4fe0-9b16...	admin	admin	2022-11-21 13:42:37.967597	2022-11-21 13:42:37.840345	Measures	imdrating	[null]	[null]	178051cc6744-4366...
3	DataContext	dc2c5940c4a-48d4-b1a...	admin	admin	2022-11-21 13:42:37.969246	2022-11-21 13:42:37.850286	name	Result	[null]	[null]	178051cc6744-4366...

Kontext

Um einen Kommentar zuordnen zu können, wird dieser mit einer frei zu definierenden Menge von sog. **Kontexten** abgelegt. Durch diese Kontexte lässt sich der Kommentar etwa einer Datenkonstellation, einer Filterkonstellation, einer bestimmten Visualisierung, einem spezifischen Dashboard oder einer beliebigen anderen Einschränkung zuordnen. Beispielsweise, dass der aktuelle Kommentar für das "Jahr 2017" im "Land Deutschland" gilt.

Ein Kontext ist immer durch einen *key* z.B. eine Dimension und einen *value* z.B. eine Ausprägung beschrieben. In den Beispielen oben wären die *keys* "Jahr" oder "Land" und die *values* "2017" bzw. "Deutschland". Die Kontexte können **statisch** im GPS festgelegt werden oder in BI-Umgebungen wie die SAP Analytics, die ein Skripting ermöglichen auch **dynamisch** über die Skriptsprache. Wird die Kommentierung in einer Visualisierung von uns genutzt, so werden für **Datenpunktkommentare** automatisch Datenkontexte für die dem Datenpunkt zugrundeliegende Datenhierarchie angelegt. Für einen **Visualisierungskommentar**, der die Visualisierung als Ganzes beschreibt, wird automatisch ein Umgebungskontext angelegt, der auf eine zufällig generierte und persistierte ID der Visualisierung verweist.

Aus der Menge der Kontexte zu einem Kommentar wird implizit eine Hierarchie gebildet, die in unserer eigenständigen Kommentarkomponente unter "*more specific comment(s)*" abgebildet wird. So wird es möglich, dass wenn als Kontext beispielsweise ausschließlich der Name des Dashboards als Umgebungskontext definiert wurde, unter dem Punkt für die weiteren Kommentare solche auftauchen, die vorher in einem anderen Kontext geschrieben wurden, der zusätzlich zu dem Namen des Dashboards beispielsweise auch noch eine Ausprägung der Dimension "Jahr" als Datenkontext festgelegt hatten und somit spezifischer als die aktuelle Menge von Kontexten festlegt sind.

Kontextarten

Umgebungskontext

Der "Environment Context" ist dazu gedacht den Kommentarraum auf Eigenschaften einzuschränken, die nicht von den Daten abhängig sind. So könnte der Kommentarraum beispielsweise auf das aktuelle Dashboard eingegrenzt werden. Ein Umgebungskontext ist auch durch eine Kombination von *key* und *value* definiert. In unserem Beispiel könnte der *key* also etwas wie "Dashboard Name" und der *value* z.B. "Sales Dashboard" sein.

Datenkontext

Der "Data Context" ist dazu gedacht den Kommentarraum auf eine gewisse Datenkonstellation einzuschränken. Er enthält zusätzlich zu den Eigenschaften *key* und *value* noch die optionalen Eigenschaften *keyText* und *valueText*. Diese dienen dazu den Anzeigenamen aus der Datenquelle (wenn denn vorhanden) für die Dimension und den Member zu pflegen, da dies die Darstellung von einem Kommentar zugehörigen Kontexten verständlicher macht. Da Dashboards und Datenlagen sehr divers und dynamisch sein können, werden diese in Umgebungen mit Skripting-Möglichkeiten meistens direkt ausgelesen und so gepflegt. Als Datenkontext könnte also beispielsweise festgelegt werden, dass der *key* "Jahr" mit dem *value* "2017" definiert ist.

Festlegungsarten von Kontexten

Statische Kontexte

Die "Static Contexts" werden im GPS festgelegt und gelten so unabhängig von beispielsweise per Skripting festgelegten Filterzuständen über die gesamte Laufzeit hinweg. Ein Static Context könnte bspw. dafür verwendet werden, den Kommentarraum auf das aktuelle Dashboard einzugrenzen. Dafür wäre der Type "EnvironmentContext" zu wählen.

Dynamische Kontexte

Die "Dynamic Contexts" werden in Umgebungen mit Skripting dort basierend dem aktuellen Zustand gesetzt. Diese werden in der Regel dafür verwendet, dynamisch den aktuellen Filterzustand auf den Daten an die Kommentare weiter zu geben, wodurch ein gepflegter Kommentar dann genau dieser Datenkonstellation zugeordnet wird. Beispielsweise kann ein Dropdown in einer Anwendung, dass das aktuelle Jahr auf der Datenquelle filtert gleichzeitig einen Datenkontext an der Kommentarkomponente setzen, der den *key* "Jahr" mit dem *value* der im Dropdown ausgewählt wurde zuordnet. Dynamische Kontexte überschreiben die Statischen Kontexte nicht.

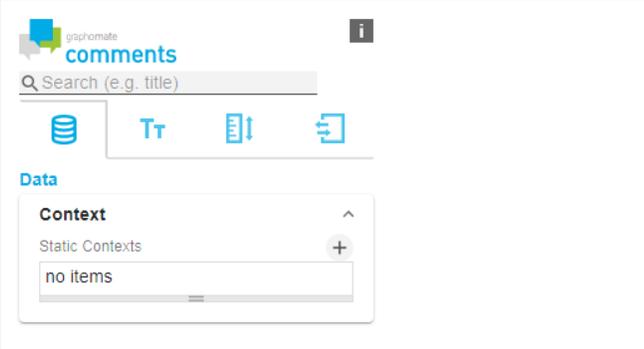
Automatische Kontexte

In der Verwendung der Kommentare in Verbindung mit unseren Visualisierungen, werden für Datenpunktkommentare die dem Datenpunkt zugehörige Datenhierarchie und für Visualisierungskommentar die ID der Visualisierung automatisch dem Kommentar zugeordnet.

graphomate property sheet GPS

- [Data Tab](#)
- [Labels Tab](#)
- [Scaling Tab](#)
- [Input Output Tab](#)

Data Tab

<p>Context</p> <p>Static Contexts</p> <p>Dem Kommentar zugehörige statische Kontexte, wie beispielsweise ein Dashboardname oder eine Dashboardseite können hier gepflegt werden.</p>	 <p>The screenshot shows a web interface for a 'comments' data tab. At the top, there is a search bar with the placeholder text 'Search (e.g. title)'. Below the search bar are three icons: a database icon, a 'Tt' icon, and a list icon. The main content area is titled 'Data' and contains a 'Context' section. This section has a sub-header 'Static Contexts' with a plus sign icon to its right. Below this, a text box displays 'no items'.</p>
--	--

Labels Tab

Information

Show Contexts

Ist diese Option aktiviert, so wird über dem Kommentar dargestellt, für welche Menge an Kontexten er aktuell gilt.

Show Authoring Information

Durch Aktivieren dieser Option, wird unter dem Kommentar dargestellt von wem er erstellt oder das letzte mal bearbeitet. Auch wird abgebildet, wann dies geschehen ist.

Show More Comments

Bei aktivierter Option, werden weitere spezifischere Kommentare unter dem "Haupt"-Kommentar dargestellt.

More Comments Expandable

Die spezifischeren Kommentare werden im Standard alle direkt dargestellt. Da es sich hierbei um eine sehr große Menge von Kommentaren handeln könnte, ist es mit dieser Option möglich, dass diese zunächst "eingeklappt" sind und erst durch einen weiteren Klick dargestellt werden.

Show Menu Edge

Zeigt eine blaue Ecke oben Rechts, durch die ein Menü geöffnet wird. In dem Menü wird der aktuell eingeloggte Benutzer dargestellt und dieser hat die Möglichkeit sich auszuloggen.

Font Family

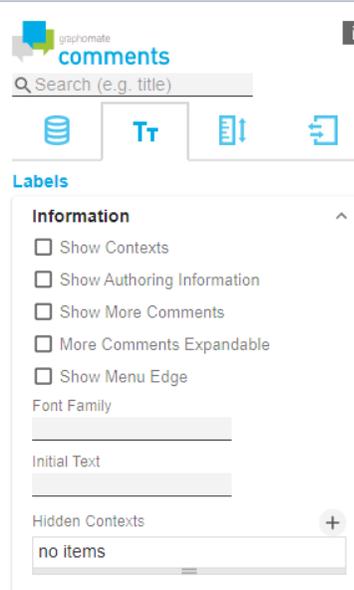
Hier kann die Schriftart definiert werden, die für den Kommentartext und auch die Schaltflächen in den Kommentaren verwendet wird.

Initial Text

Ist kein Kommentar vorhanden, so wird durch einen Hinweistext auf diesen Zustand hingewiesen. Dieser kann hier modifiziert werden. Diese Funktion kann auch verwendet werden, um eine Grundstruktur für neue Kommentare als Template anzulegen. Hier kann HTML zur Formatierung verwendet werden.

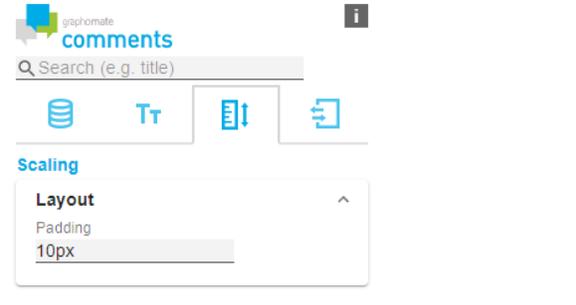
Hidden Contexts

Da nicht alle Kontexte dynamisch gesetzt werden und somit in jedem Zustand eines Dashboards identisch bleiben kann es Sinn ergeben diese auszublenden. Durch die Definition eines **Hidden Contexts** kann somit definiert werden, dass ein Kontext zwar verwendet wird um einen Kommentar einzuordnen, aber nicht in der UI dargestellt wird. Alle



Optionen sind hier optional. Dadurch ist es bspw. möglich, wenn im ausgeblendeten Kontext nur der Typ "Environment Context" definiert wird, dass alle Kontexte von diesem Typ ausgeblendet werden. Auch können alle Kontexte mit einem bestimmten Key so ausgeblendet werden.

Scaling Tab

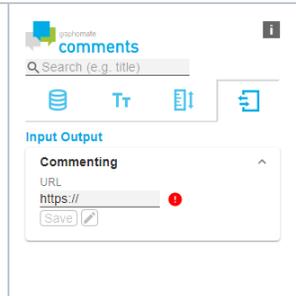
<p>Scaling</p> <p>Padding</p> <p>Das Padding definiert wieviel Abstand die Komponente zu dem äußeren Rand lassen soll.</p>	 <p>The screenshot shows a design tool interface for a component titled 'comments'. At the top, there is a search bar with the placeholder text 'Search (e.g. title)'. Below the search bar are four icons: a database icon, a text icon, a list icon, and a document icon. The 'Scaling' tab is selected, and a 'Layout' panel is open, showing 'Padding' set to '10px'.</p>
--	---

Input Output Tab

Server

Comments Server URL

Diese URL verweist auf das *graphomate comments backend* und dient zur Ablage der Kommentare und der Benutzerverwaltung. Wurde die URL einmalig gepflegt, so ist es in der Regel nicht notwendig, sie in jeder Instanz unserer comments und Dashboard erneut zu pflegen. Bei Klick auf "save" wird überprüft, ob der Server korrekt antwortet. Ist dies der Fall, so wird es durch einen grünen "Haken" symbolisiert. Gleichzeitig erscheint ein Verweis auf die Verwaltungsoberfläche der Backends.



Scripting Dokumentation

Die graphomate comments integrieren sich nahtlos in die SAP Analytics Cloud. So wird auch das Scripting im Application Designer der SAP Analytics Cloud unterstützt.

Die im PropertySheet dargestellten Eigenschaften, lassen sich per Scripting von der Komponente abrufen und setzen. Daneben gibt es noch Methoden, die der Vereinfachung des setzens von Kontexten für Kommentare dienen:

createDataContext(key: string, value: string): DynamicContext;

Über diese Methode kann ein Datenkontext erstellt werden, der dann über die Methode **setDynamicContexts**, der ein Array aus Kontexten übergeben wird, gesetzt werden kann. Ein Datenkontext beinhaltet in der Regel eine Kombination aus Dimension und Member. Also bspw. ist der key "year" und der value "2017".

createDataContextWithText(key: string, value: string, keyText: string, valueText, string): DynamicContext;

Unterscheidet Ihre Datenquelle zwischen IDs für Dimensionen und Membern und deren Anzeigetext, so kann diese Methode wie createDataContext verwendet werden, mit der zusätzlichen Möglichkeit auch die Anzeigetexte zu setzen. Bpsw. ist hier der key "CAL_YEAR" und der keyText "Year".

createEnvironmentContext(key: string, value: string): DynamicContext;

Ein Umgebungskontext kann dazu genutzt werden um den Kommentarraum auf bspw. eine bestimmte Umgebung oder ein Dashboard einzuschränken. Hierbei könnte dann key etwas wie "Dashboard Name" sein und value dazu "Global Sales 2017".

addOrReplaceDataContext(key: string, value: string): void;

Diese Methode funktioniert grundsätzliche die die createDataContext, nur dass diese den hier definierten Kontext direkt dem Widget hinzufügt oder abändert und somit der web über setDynamicContexts nicht notwendig ist.

addOrReplaceDataContextWithText(key: string, value: string, keyText: string, valueText: string): void;

Hier definierte Werte verhalten von den Parametern wie die der Methode createDataContextWithText und werden wie in addOrReplaceDataContext direkt hinzugefügt oder abgeändert.

addOrReplaceEnvironmentContext(key: string, value: string): void;

Die Parameter verhalten sich analog zu createEnvironmentContext. Dieser wird jedoch direkt hinzugefügt oder abgeändert.

removeDataContext(key: string): void;

Entfernt den Datenkontext mit dem key aus dem Kontextraum des Kommentarwidgets.

removeEnvironmentContext(key: string): void;

Entfernt den Umgebungskontext mit dem key aus dem Kontextraum des Kommentarwigets.

setSelectedData(selectedData: SelectionContext[]): void; (ab Version 1.20224.1)

Die in einer graphomate visualisierung per klick selektierten Daten können so dem comments widget zur Generierung von Datenkontexten übergeben werden.

setDynamicContexts(dynamicContexts: DynamicContext[]): void; (ab Version 1.20231.0)

Kontexte, die über die **create*Context** Methoden erstellt wurden, können hier verwendet werden um die gesamten dynamischen Kontexte neu zu setzen. Der alte Zustand wird hierbei überschrieben.

addOrReplaceContexts(contexts: DynamicContext[]): void; (ab Version 1.20231.0)

Mehrere Kontexte gleichzeitig zu setzen führt bei Verwendung von **addOrReplace*Context** zu Unstimmigkeiten. Diese Methode schafft hier Abhilfe, denn sie ermöglicht mehrere, per **create*Context** erstellte Kontexte, parallel an das Widget zu übergeben.

removeContexts(contexts: DynamicContext[]): void; (ab Version 1.20231.0)

Um mehrere Kontexte gleichzeitig zu entfernen, sollte statt einzelnen Aufrufen von **remove*Context**, diese Methode verwendet werden um mit **create*Context** erstellte Kontexte dieser Methode zu übergeben. Diese werden dann aus dem Zustand des Widgets

entfernt. In der create*Context Methode kann der key oder der value auch als leerer String gesetzt werden. Dann werden alle Kontexte auf die nur der definierte key oder value zutrifft entfernt.

Beispiele

Um je nach Auswahl eines Wertes in einem DropDown-Feld einen Datenkontext zu setzen, kann folgender Code angepasst werden:

```
1 var countryMember = Country_Dropdown.getSelectedKey();
2 if (countryMember === "None") {
3     Chart_1.getDataSource().setDimensionFilter("Country_Region", []);
4     graphomate_comments_1.removeDataContext("Country_Region");
5 } else {
6     Chart_1.getDataSource().setDimensionFilter("Country_Region", countryMember);
7     graphomate_comments_1.addOrReplaceDataContext("Country_Region", countryMember);
8 }
```

Folgender Code setzt die Informationen über einen in einem graphomate chart selektierten Datenpunkt als Kontexte für das comments widget:

```
1 graphomate_comments_1.setSelectedData(graphomate_charts_1.getSelectedData()); // ab Version 1.20224.1
```

Mehrere Kontexte gleichzeitig können wie folgt abgeändert oder hinzugefügt werden:

```
1 var regionContext = graphomate_comments_1.createDataContext("region", "North");
2 var countryContext = graphomate_comments_1.createDataContext("country", "Germany");
3 graphomate_comments_1.addOrReplaceContexts([regionContext, countryContext]);
```

Um die oben hinzugefügten Kontexte gleichzeitig wieder zu entfernen, kann wie folgt vorgegangen werden:

```
1 var regionContext = graphomate_comments_1.createDataContext("region", "North");
2 var countryContext = graphomate_comments_1.createDataContext("country", "Germany");
3 graphomate_comments_1.removeContexts([regionContext, countryContext]);
```

Sollen die beiden Kontexte von oben entfernt werden, unabhängig davon auf welchen Wert sie gesetzt sind, so können Leerstrings verwendet werden:

```
1 var regionContext = graphomate_comments_1.createDataContext("region", "");
2 var countryContext = graphomate_comments_1.createDataContext("country", "");
3 graphomate_comments_1.addOrReplaceContexts([regionContext, countryContext]);
```

Um alle hinzugefügten dynamischen Kontexte neu zu setzen, als auch die bisher gesetzten zu entfernen, gehen wir wie folgt vor:

```
1 // zunächst fügen wir nur zur Verdeutlichung ein Kontext für das Jahr hinzu
2 graphomate_comments_1.addOrReplaceDataContext("year", "2023");
3 // danach setzen wir dynamische Kontexte für Region und Land
4 var regionContext = graphomate_comments_1.createDataContext("region", "North");
5 var countryContext = graphomate_comments_1.createDataContext("country", "Germany");
6 // durch die Verwendung von setDynamicContexts, statt addOrReplaceContexts, ist der Kontext für das Jahr dann
7 graphomate_comments_1.setDynamicContexts([regionContext, countryContext]);
```

Admin UI

Die Admin UI ist eine Weboberfläche, die mit dem graphomate comments server backend ausgeliefert wird. Diese kann über die URL des Servers mit dem Zusatz "/commenting/administration/" erreicht werden. Alternativ befindet sich unter dem Eingabefeld für die Server URL in dem Komponenten ein Button mit der Beschriftung "ADMIN", der auch zu dieser Oberfläche führt.

Sie dient sowohl als Übersicht und Exportmöglichkeit für die bestehenden Kommentare, als auch dazu administrative Aufgaben in Bezug auf Userverwaltung und Autorisierung auszuführen.

Ein Benutzer ohne die Rolle "admin", sieht ausschließlich den Punkt "Comments" und hat so die Möglichkeit einer Übersicht der für ihn freigegebenen Kommentare.

Ist jedoch die Rolle "admin" für ihn vergeben, so erscheinen die zusätzlichen Menüpunkte "Contexts", "Users", "User Roles", "User Groups" und "Authorizations".

Eine Erklärung der einzelnen Unterpunkte folgt auf den nächsten Unterseiten.

Comments

Jeder Nutzer, auch die ohne die Benutzerrolle "admin", hat hier die Möglichkeit eine Übersicht über bereits bestehende Kommentare zu erlangen. Dargestellt werden dabei ausschließlich solche Kommentare, zu dessen Ansicht der Benutzer berechtigt ist.

Durch einen Klick auf einen Spaltenkopf, können die Kommentare nach dieser Eigenschaft sortiert werden.

Menüleiste

Die Menüleiste unter der Überschrift "Comments" bietet folgende Optionen:

1. **Columns:** Hier kann die Sichtbarkeit von Tabellenspalten eingeschränkt und erweitert werden. Zusätzlich zu den im Standard dargestellten Spalten kann noch die Kommentar-ID und der Zeitpunkt und Benutzer der letzten Änderung dargestellt werden.
2. **Filters:** Die Liste der dargestellten Kommentare kann hier eingeschränkt werden. Aktuell verfügbar ist eine Einschränkung auf "Autor", "Bearbeiter", "Erstellungszeitpunkt" und "Änderungszeitpunkt".
3. **Export:** Zwei Möglichkeiten des Exports werden hier angeboten. Das Drucken der aktuellen Seite oder der Export als CSV. Der CSV-Export enthält alle, auch die nicht dargestellten, Kommentare und Tabellenspalten.
4. **Refresh:** Durch Klick auf diesen Button werden neue Daten vom Server geladen.
5. **Add:** Die Möglichkeit zur "manuellen" Kommentarerstellung bietet sich hier. Da Kontexte hier auch manuell definiert werden müssen, dient dieses Feature eher zu Testzwecken als zum produktiven Einsatz.

Tabellenspalten

1. **ID:** Im Standard nicht sichtbar wird hier die ID des Kommentares dargestellt. Es handelt sich hierbei um eine zufällig generierte UUID.
2. **Creation:** Der Zeitpunkt der Erstellung des Kommentares.
3. **Changed:** Im Standard nicht sichtbar, der Zeitpunkt der letzten Änderung.
4. **Text:** Der Text des Kommentares ohne Formatierung.
5. **Creator:** Der Benutzername des Erstellers des Kommentares.
6. **Changed:** Im Standard nicht sichtbar, der Name des Benutzers, der die letzte Änderung vorgenommen hat.
7. **Contexts:** Die Liste der dem Kommentar zugehörigen Kontexte.
8. **Actions:** Buttons zum Bearbeiten und Löschen eines Kommentares.

Bearbeiten

Nach dem Klick auf den Button zum Bearbeiten, öffnet sich für den Kommentar der aus den Komponenten bekannte Editor. Durch Klick auf das Symbol der Diskette, werden Änderungen abgespeichert.

Seitenauswahl

Unten rechts befindet sich ein Menü mit dem sich die Anzahl der dargestellten Kommentare pro Seite und die aktuelle Seite einstellen lassen.

Contexts

Benutzer mit der Rolle "admin" sehen hier eine Übersicht der bestehenden Kontexte. Löschen lassen sich diese nur, wenn dem Kontext kein Kommentar mehr zugewiesen ist.

Um Kontexte bereits zur Definition von Autorisierungen zur Verfügung zu haben, bevor diese durch die Erstellung eines Kommentares automatisch mit erstellt werden, können diese durch ein Klick auf "ADD" vorab angelegt werden.

Users

Bereits angelegte oder durch den Import aus einer externen Userverwaltung definierte Benutzer werden hier dargestellt.

Durch Klick auf den "Bearbeiten Stift", kann ein Benutzer bearbeitet werden. Textfelder können durch einen Klick auf den Text bearbeitet werden. Für Rollen und Gruppen bietet sich die Möglichkeit bereits hinzugefügte Rollen durch das  zu entfernen oder durch das "Add Dropdown" neue hinzuzufügen.

Mittels Klick auf "(+) Add" kann ein neuer Benutzer angelegt werden. Verpflichtend sind die Felder "Username" und "Initial Password". Das initiale Passwort muss vom Benutzer beim ersten Login abgeändert werden. Eine E-Mail-Adresse ist notwendig, um das Feature der automatischen Mail-Benachrichtigung zu nutzen, sobald ein User in einem Kommentar erwähnt wird. Rollen und Gruppen sind dann notwendig, wenn Berechtigungen darauf basieren.

Ein Klick auf den "(+) JSON" Button öffnet ein Fenster in dem mehrere Benutzer gesammelt angelegt werden können. Das dafür notwendige JSON-Format soll im folgenden Beispiel verdeutlicht werden:

```
1  [
2    {
3      "username": "Meier",
4      "password": "secretPassword"
5    },
6    {
7      "username": "Michaelsen",
8      "password": "nobodyKnows",
9      "firstName": "Bernd",
10     "lastName": "Michaelsen",
11     "email": "bernd.michaelsen@example.com",
12     "roles": ["viewer", "editor"],
13     "groups": ["CEO"]
14   }
15 ]
```

User Roles

Benutzerrollen werden hier dargestellt und können über ein Klick auf "(+) ADD" hinzugefügt werden. Dafür ist lediglich ein frei wählbarer Name notwendig. Diese Benutzerrollen können dann auf der "User"-Seite Benutzern zugeordnet und in den Autorisierungen verwendet werden, um basierend auf den Rollen den Benutzern dieser Rolle Rechte zuzuweisen.

User Groups

Benutzergruppen werden hier dargestellt. Durch einen Klick auf "(+) ADD" lässt sich eine neue Gruppe mit einem frei definierbaren Namen hinzufügen. Diese Gruppen können auf der "User"-Seite Benutzern zugewiesen und dann in Autorisierungen verwendet werden, um Benutzern einer Gruppe Rechte zuzuweisen. Um ganze Benutzerrollen einer Gruppe zuzuweisen muss aktuell noch die UI von Keycloak selbst verwendet werden.

Authorizations

Unter dem Punkt "Authorizations" in der Backend-Admin-UI kann definiert werden, welche Benutzer, Benutzerrollen und Benutzergruppen berechtigt sind auf eine Kontextkombination zu Kommentieren. Diese Einstellungen stehen ausschließlich Benutzern mit der Rolle "admin" zur Verfügung.

Initiale Konfiguration

Im Auslieferungszustand der graphomate comments sind automatisch drei Benutzerrollen angelegt: "admin", "editor" und "viewer". Unter dem Punkt "Authorizations" sind dazu die passenden Regeln bereits vordefiniert. Die erste Regel verbietet zunächst allen Benutzern jegliche Interaktion mit den Kommentaren, so dass die folgenden Regeln eine sogenannte *Whitelist* abbilden.

Die zweite Regel definiert, dass Benutzer mit der Rolle "admin" oder "editor" alle Kommentare sehen, erstellen, bearbeiten und löschen dürfen.

Als letzte Regel wird definiert, dass Benutzer mit der Rolle "viewer" alle Kommentare nur sehen dürfen.

Diese Konfigurationen können Sie natürlich beliebig für ihre Szenarien anpassen.

Autorisierung hinzufügen

Durch einen Klick auf den + ADD Button, in der Menüleiste über der Liste bestehender Autorisierungen, kann eine neue Autorisierung hinzugefügt werden. Vorab sei bereits erwähnt, dass wenn für eine Eigenschaft der Autorisierung keine Optionen ausgewählt werden, die Autorisierung für alle Ausprägungen dieser Untereinstellung gilt.

Als Sonderfall gilt hier die Einstellung der Benutzer, Benutzerrollen und Benutzergruppen. Erst wenn alle drei leer gelassen werden, gilt die Regel für alle Benutzer.

Eine Autorisierung besteht aus mehreren Untereinstellungen, welche beim Hinzufügen einer neuen Autorisierung und beim Bearbeiten einer Bestehenden konfiguriert werden können. Beim erstellen einer Autorisierung können mehrere Punkte durch Selektion ihrer Checkboxen einzeln, oder insgesamt durch die obere Checkbox neben dem Titel ausgewählt werden. Durch Klick auf die Pfeile werden sie entweder der Autorisierung zugewiesen (dann befinden sie sich auf der rechten Seite), oder wieder abgewählt (und somit wieder auf der linken Seite). In der Bearbeiten-Ansicht einer bestehenden Autorisierung können gewählte Untereinstellungen durch ein Klick auf das  am rechten Rand entfernt und neue durch das Dropdown-Feld hinzugefügt werden. In beiden Fällen steht eine Suche zur Verfügung.

Autorisierungseigenschaften

Name

Der frei definierbare Name der Autorisierung dient bspw. dazu die Funktionalität zusammenzufassen.

Type

Die Auswahl zwischen "Allow" und "Deny" definiert, ob durch diese Autorisierungsregel die gewählten Aktionen für eine Kontextkombination erlaubt oder verboten werden. Hierdurch lassen sich durch Kombination Regeln nach dem Schema "alles außer" und "ausschließlich für" definieren.

Actions

Die Aktionen die bezogen auf Kommentare ausgeführt werden können sind in vier Freigaben unterteilt:

1. "view": Benutzer die dafür freigegeben sind können bestehende Kommentare lesen.
2. "create": Ermöglicht Benutzern neue Kommentare anzulegen.
3. "edit": Bereits bestehende Kommentare können bearbeitet werden.
4. "delete": Kommentare dürfen gelöscht werden.

Werden keine Aktionen ausgewählt, so gilt diese Freigabe für alle Aktionen. Der Unterschied dazu, alle Aktionen auszuwählen liegt darin, dass ohne eine Auswahl Aktionen die in späteren Versionen hinzugefügt werden könnten auch direkt freigegeben sind.

Contexts

In diesem Bereich wird definiert, für welche Kontextkombination die Autorisierung angewendet werden soll. Die Darstellung beginnt mit einem "env" für EnvironmentContext oder einem "dat" für DataContext. Darauf folgt vor dem "=" der key des Kontexts und dahinter der *value*.

Kontexte die hier ausgewählt werden gelten grundsätzlich auch für spezifischere Kontextkombinationen. Das bedeutet, dass sobald ein Kommentar mindestens die gewählten Kontexte enthält, die Regel dafür greift. Nehmen wir also bspw. an, ein Benutzer ist für die Kontextkombination "Region=Central" und "Country=Germany" freigegeben, so kann dieser auf Kommentaren für die Kontextkombination Central+Germany und "Year=2017" zugreifen. Dadurch untersagt ist der Zugriff hingegen für einen Kommentar der bspw. für Central+Austria gilt.

Werden keine Kontexte ausgewählt, so gilt die Regel für alle Kommentare.

Users

Einzelbenutzer, für die eine Autorisierung greift, können hier ausgewählt werden.

Wird kein Benutzer, keine Benutzerrolle und keine Benutzergruppe ausgewählt, so gilt diese Regel für alle Benutzer.

Roles

Die Autorisierung kann hier allen Usern in einer oder mehreren Benutzerrollen zugeordnet werden.

Groups

Analog zu den "Roles" können hier ganze Benutzergruppen ausgewählt werden.

Reihenfolge

In der Spalte "Order" auf der Übersichtsseite der Autorisierungen kann durch Klick auf die Pfeile die Reihenfolge der Regeln geändert werden. Die Regeln werden sequenziell, beginnend mit dem niedrigsten Index, abgearbeitet. Das bedeutet, dass, wenn wie in der initialen Konfiguration zunächst ein Verbot für alles definiert wird, dieses Verbot durch einzelne Regeln mit einem höheren Index wieder abgeschwächt werden kann. Die Textuell verfasste initiale Konfiguration lautet also:

Verbiете zunächst allen Benutzern jegliche Interaktion mit Kommentaren. Erlaube jedoch "view", "create", "edit" und "delete" für Benutzer mit der Rolle "admin" oder "editor". Im letzten Schritt erlaube das Ansehen von Kommentaren für Benutzer mit der Rolle "viewer" das Betrachten von jeglichen Kommentaren.

Noch ein hypothetisches Beispiel: Würde am Ende der Liste von Autorisierungen mit der höchsten Order eine Regel stehen, die alles verbietet, so wäre alle vorherigen Freigaben nichtig.

Keycloak

Wird der Server-URL ein "/auth" angehängen, so öffnet sich die UI von Keycloak, in der alle Einstellungen zur Benutzerverwaltung vorgenommen werden können. Dies inkludiert auch Einstellungen die in der restlichen Admin UI als übersichtlichere Abkürzung vorgenommen werden können.

Auf der Startseite befindet sich ein Verweis auf die "Administration Console" und die "Documentation". Nach Klick auf "Administration Console" wird der Benutzer, sofern er nicht bereits eingeloggt ist, aufgefordert sich zu Authentifizieren. Ein Benutzer mit der Rolle "admin" hat hier volle Berechtigungen.

Vorwiegend relevante Einstellungen seien im Folgenden kurz umrissen. Für eine allumfassende Dokumentation seien Sie auf die Keycloak eigene Dokumentation verwiesen.

Realm Settings / Master / Login

Features wie das Erinnern an einen Benutzer nach Browser Neustart und die "Forgot password" Funktion können hier aktiviert werden. Für Letzteres ist unter dem Punkt **Email** die Konfiguration eines SMTP-Hosts notwendig.

Roles

Analog zu den Rolleneinstellungen in der Admin UI können hier Rollen erstellt werden. Unter **Default Roles** kann definiert werden, welche Standardrollen einem neuen Benutzer zugewiesen werden sollen. So kann beispielsweise erreicht werden, dass neue Benutzer immer alle Kommentare sehen dürfen, in dem automatisch die Rolle "viewer" zugewiesen wird.

Identity Providers

Externe Benutzerverwaltungen wie ein Active Directory über SAML oder OpenID Connect können hier angebunden werden. Rollen und Gruppen lassen sich hier von dem externen System auf interne Rollen und Gruppen mappen.

User Federation

Unter diesem Menüpunkt es ist möglich Benutzer von einem externen System zu importieren. Beispielsweise kann hier ein Active Directory über LDAP angebunden werden. Der Unterschied zu den "Identity Providers" ist, dass hier die Benutzer nicht erst beim ersten Login importiert werden, sondern direkt in der Userliste zu Verfügung stehen. Rollen und Gruppen lassen sich auch hier von dem externen System auf interne Rollen und Gruppen mappen.

Authentication / Password Policy

Voraussetzungen für sichere Passwörter lassen sich hier definieren.

Groups

Analog zu den Gruppeneinstellungen in der Admin UI lassen sich hier Gruppen anlegen und editieren. Zusätzlich bietet sich hier die Möglichkeit ganze Benutzerrollen unter **Role Mappings** einer Gruppe zuzuweisen und so eine Sammlung von Rollen in einer Gruppe zu vereinen, die dann gemeinsam eine Berechtigung über die Authorizations erhalten. Unter **Default Groups** kann definiert werden, dass neuen Benutzern automatisch Gruppen zugewiesen werden.

Users

Analog zu den Benutzereinstellungen in der Admin UI können hier Benutzer angelegt und bearbeitet werden. Unter **Credentials** kann das Passwort eines Benutzers zurückgesetzt werden. Der Schalter "Temporary" sorgt dafür, dass der Benutzer dieses Passwort beim nächsten Login wieder ändern muss.

Interaktivität zur Laufzeit

Die graphomate comments lassen sich umfangreich zur Laufzeit bedienen. Vorausgesetzt, Sie haben bereits die URL zu dem comments Backend in unserem GPS gepflegt, erscheint beim Laden des comments widgets zunächst ein Feld für den initialen Login. Sind sie bereits eingeloggt, so erscheint dieses Fenster nicht mehr. Dort können Sie sich mit dem initialen Administrator oder, wenn bereits gepflegt, jedem beliebigen Benutzer in unserem Backend anmelden. War der Login erfolgreich, so erscheint, wenn sie noch keinen Kontext für den Kommentar gesetzt haben, ein Hinweis, dass zunächst ein Kontext definiert werden muss. Ein Kommentar muss immer in einen Kontext geschrieben werden. Die Definition des Kontextes kann im GPS und per Scripting geschehen.

Existiert noch kein Kommentar für den aktuellen Kontext, so erscheint ein Hinweistext, der auf diesen Umstand hinweist. Durch Klick auf den Button mit der Beschriftung "Create" öffnet sich ein Editor um einen Kommentar zu schreiben. Gleiches gilt für den Falls, dass schon ein Kommentar existiert. Durch Klick auf den "Edit"-Button öffnet sich wieder der Editor und der Kommentar kann bearbeitet werden. Der Editor bietet in seiner Menüleiste neben Formatierungsmöglichkeiten für den Text die Möglichkeit Hyperlinks in den Text einzubetten. Wird im Text ein anderer Benutzer durch das Zeichen "@" und direkt dahinter dem Benutzernamen erwähnt, so erhält dieser bei korrekter Mail-Konfiguration im Backend eine E-Mail über den Kommentar. Sobald mit dem "@" mindestens zwei Buchstaben geschrieben wurden, werden die verfügbaren Benutzer mit passendem Namen vorgeschlagen. Ist der Editor geöffnet, so befinden sich unterhalb des Texteingabefeldes Buttons um den Kommentar abzuspeichern, zu löschen oder die Eingabe abubrechen.

Die blaue oben rechts platzierte "Ecke" öffnet bei Klick ein Menü in dem der aktuelle Benutzer dargestellt wird und bietet die Möglichkeit sich auszuloggen.

Unter dem aktuellen Kommentar wird dargestellt, ob es spezifischere Kommentare zu dem aktuellen Kontext gibt. Spezifischer ist ein Kommentar, wenn er zu der aktuell definierten Menge an Kontexten noch weitere einschränkende Kontexte enthält. So werden beispielsweise in einem Kommentar-Widget, das den Kontext 2017 aber keinen Monat enthält, unter spezifischeren Kommentaren diejenigen dargestellt, die für einen spezifischen Monat im Jahr 2017 gepflegt wurden. Per Klick auf den Hinweis auf spezifischere Kommentare, öffnet sich die Liste weiterer Kommentare. Diese sind danach sortiert, wie viele zusätzliche Kontexte sie enthalten. Unter dem jeweiligen Kommentar wird dargestellt, welche zusätzlichen Kontexte für den Kommentar gelten. Durch Klick auf einen zusätzlichen Kontext wird dieser in den Filter der aktuellen Kontextmenge aufgenommen. Werden die aktuellen Kontexte dargestellt (Konfiguration im GPS), so kann dieser Filter durch Klick auf den zusätzlichen Kontext in der Liste wieder entfernt werden.

Known Issues

Derzeit sind keine Known Issues bekannt.

Changelog

Version 2023.2.X

Frontend

- New editor options "Title" and "Hyperlink"
- Enable and disable features based on authorizations
- "Are your sure?" dialog for deleting comments
- Scrolling fixed
- Base font size configurable
- UI improvements

Backend

- Database scheme normalized to unique contexts
- Routes secured on backend side based on authorizations by contexts
- Visible comments filtered by authorizations by contexts
- Definition of user roles in Admin UI
- Definition of user groups in Admin UI
- Assingment of user groups to user in Admin UI
- Definition of "Authorizations by Contexts" in AdminUI
- User JSON import in Admin UI
- Backend UI improvements

Version 2023.1.X

- UI improvements
- Performance improvements
- Configurable visibility of contexts
- More specific comments display configurable
- Font family configurable
- Initial text for unset comment configurable
- User edge hidable
- Editor text formatting as title
- Editor text with embedded hyperlink
- Compatibility with data point and visualisation comments improved

Backend

- CSV Export with better formatting and more information
- Filter von Author and Date fixed
- Pagination showing strange result lists fixed