Draft Handbuch Berechtigungen in KIX18

Berechtigungsebenen und -arten

Berechtigungsebenen

Das KIX18 Berechtigungskonzept unterscheidet zwischen verschiedene "Zugriffsebenen" die einander bedingen:

- Ressourcenebene steuert Zugriffe auf Objektsammlungen
  - Sonderfall ist eine Sammlung mit einem Element ("/tickets/123" definiert eine Sammlung bestehend aus genau einem Objekt)
- Objektebene steuert Zugriffe auf konkrete Objekte die durch Eigenschaften der Objekte beschrieben werden
  - z.B. Tickets in bestimmten Teams oder Tickets zu bestimmten Kundenorganisationen
- Ebene der Objekteigenschaften steuer Zugriffe auf Objekteigenschaften
  - z.B. Ticket.State, Ticket.AccountedTime, Ticket.Priority

Zwischen den Ebenen bestehen Zusammenhänge, d.h.

- ohne Zugriff auf Ressource, kein Zugriff auf Objekt
- ohne Zugriff auf Objekt, kein Zugriff auf Eigenschaft

Vorhandene Ressourcen, Objekte und Eigenschaften

Die verfügbaren Ressourcen, Objekte und deren Eigenschaften können der API-Dokumentationentnommen werden.

Berechtigungsarten

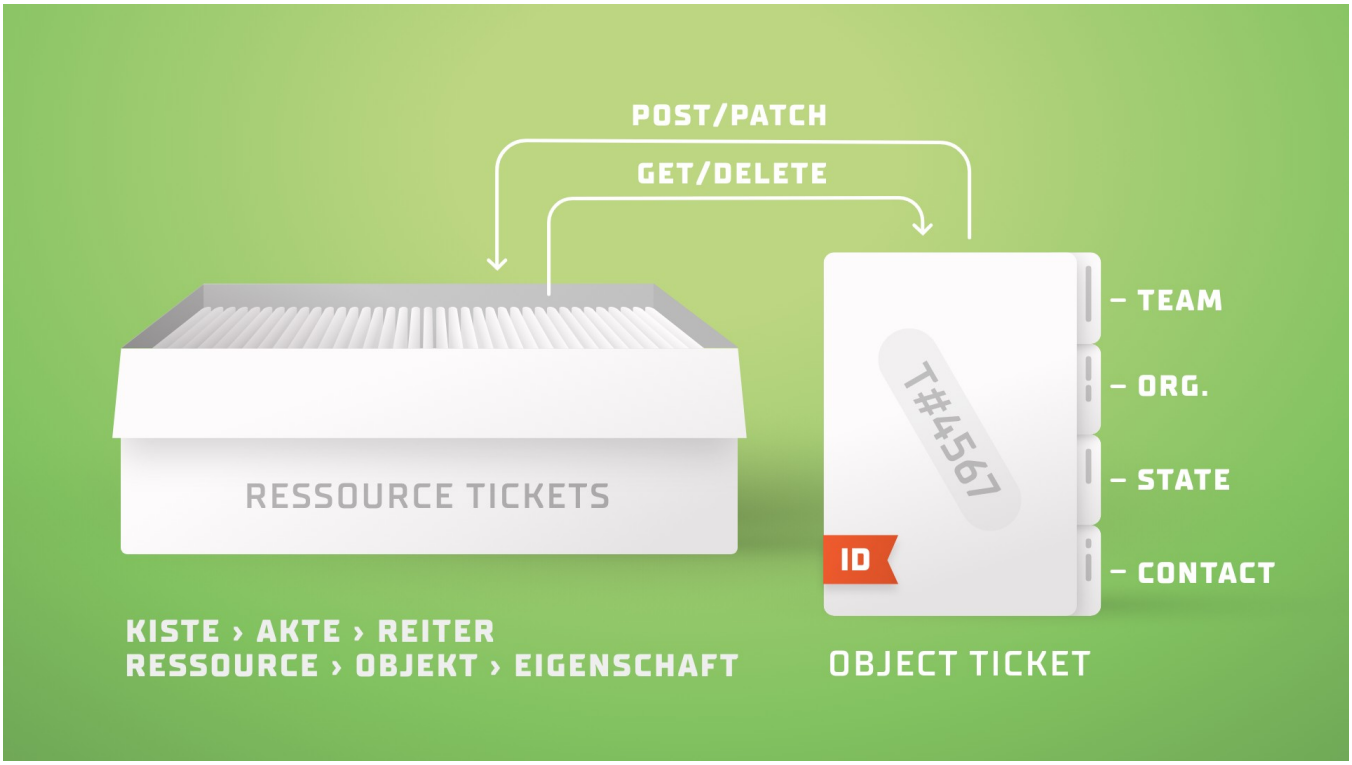
Dabei gibt es folgende Formen der Zugriffe (aka "Berechtigungsarten")

- C: CREATE - entspricht einem HTTP-POST
- R: READ - entspricht einem HTTP-GET
- U: UPDATE - entspricht einem HTTP-PATCH
- D: DELETE - entspricht einem HTTP-DELETE
- -: nix
- Deny: Entzug aller anderweitig zugestandenen Berechtigungen auf Einträge
  - relevant in Bezug auf Kombination von Rollen)
  - kann nicht überschrieben werden

Für leichtere Lesbarkeit wird die Notation "CRUDX" verwendet. Das kann z.B. so aussehen

```
"-R---" # für nur Lesen
"CR---" # für Erstellen und Lesen
"CRU--" # für Erstellen, Lesen und Ändern
"---D-" # für Löschen
"-----" # für keinen Zugriff (der durch andere Rollen wieder erweitert werden kann)
"----X" # für absolut keinen Zugriff (der nicht durch andere Rollen aufgehoben werden kann)
```

Die Aktenanalogie



	Berechtigungen				
Ebenen	**Create**	**Read**	**Update**	**Delete**	**Deny**
1: Ressourcen	Erstellen in Ressource *Ich kann neue Akten anlegen.*	Lesen in Ressource *Ich kann Akten lesen*	Ändern in Ressource *Ich kann Akten ändern*	Löschen in Ressource *Ich kann Akten löschen.*	jeglichen Zugriff auf Ressource entziehen *Ich darf absolut gar nichts mit Akten.*
2: Objekte	Objekt darf wie übermittelt erstellt werden *Ich darf DIESE Akte anlegen*	bestehende Objekte lesen *Ich darf DIESE Akte lesen.*	Objekte darf wie übermittelt geändert werden *Ich darf DIESE Akte ändern.*	bestehende Objekte löschen *Ich darf DIESE Akte löschen.*	jeglichen Zugriff auf ein Objekt sperren *Ich darf absolut gar nichts mit DIESER Akte.*
3: Eigenschaften	erlaubt Setzen von bestimmten Eigenschaften bei Erstellung *Ich darf eine Akte mit DIESEM Kapitel anlegen*	erlaubt Lesen bestimmter Eigenschaften *Ich darf DIESE Eigenschaft der Akte lesen.*	erlaubt Ändern bestimmter Eigenschaften *Ich darf DIESE Eigenschaft an der Akte setzen.*	n.a.	jeglichen Zugriff auf eine Eigenschaft sperren *Ich darf absolut gar nichts mit dieser Eigenschaft.*

Subressourcen

(1) Berechtigungen auf Subressourcen entsprechen den erteilten Berechtigungen auf übergeordneten Ressourcen, sofern keine abweichende Aussage getroffen wird.

(2) Es gilt weiterhin, dass Berechtigungen auf Subressourcen nicht weiter gefasst sein dürfen als Berechtigungen auf übergeordnete Ressourcen. Besteht bspw. kein Updaterecht auf Ressource "/system", kann auch kein Update-Recht auf Ressource "/system/automation" erteilt werden.

Identifizierende Subressourcen (Artikel zu Tickets)

Berechtigungen auf identifizierende Subressourcen stellen eine kleine Besonderheit dar. Um diese beschreiben zu können wird der Stern-Platzhalter wie in **"/ticket/*/articles"** verwendet. Der Stern hat die gleiche Wirkung wie bspw. im Dateisystem und bedeutet hier "alle Elemente direkt unter /tickets/".

(3) Berechtigungen auf identifizierenden Subressourcen hängen sowohl von grundlegenden Berechtigungen auf die Ressource als auch von den konkreten Berechtigungen auf das übergeordnete, identifizierte Objekt ab.

Das Anlegen eines Artikels an Ticket 123 erfordert also

- Create- und Update-Recht auf Ressource "/tickets"
  UND
- Update-Recht auf Objekt "/tickets/123"
  UND
- Create-Recht auf Ressource "/tickets/*/articles" (im Beispiel mit * == 123)

Objektberechtigungen

Wie bereits erwähnt erlauben Objektberechtigung die Prüfung des Objekts auf bestimmte Eigenschaften. Dabei definiert die HTTP-Methode welche Datenmenge zur Prüfung verwendet wird.

Bei CREATE/UPDATE erfolgt die Prüfung auf Basis der übermittelten, (neu) zu setzenden Eigenschaften ("in der HTTP-Anfrage POST~~/PATCH~~").

Bei READ/DELETE werden die im Backend/der DB bestehenden Eigenschaften verwendet.

Es kann dabei auf versch. Eigenschaften geprüft werden. Diese Prüfungen können UND-verknüpft werden mittels "&&". Alternative Optionen ("ODER"-Verknüpfungen) können durch mehrere Berechtigungen abgebildet werden.

Weiterhin kann gegen absolute Werte oder Eigenschaften des ausführenden Nutzers mittels des Platzhalters "$CurrentUser" geprüft werden.

Zulässige Vergleichsoperatoren sind "LT", "LTE", "GT", "GTE", "CONTAINS", "LIKE", "IN", "STARTSWITH", "ENDSWITH", "EQ", "NE". Die Vergleichsoperatoren können mittels "!" negiert werden.

Beispiele Objekteigenschaftsprüfungen

```
# ALLOW full access to ticket
# WHERE title contains "Security"
#   AND priority-ID is 3
Object | /tickets/*{Ticket.Title CONTAINS "Security" && Ticket.PriorityID LT 3}  |  CRUD-

# ALLOW to read tickets
# WHERE SLA is not 5
#   AND team/ueue is not 1,2 or 3
Object | /tickets/*{Ticket.SLAID NE 5 && Ticket.QueueID !IN [1,2,3]}            |  -R----

# ALLOW to read tickets
# WHERE title LIKE "*something*"
Object | /tickets/*{Ticket.Title LIKE "*something*"}                           |  -R----

# DO NOT ALLOW ACCESS to tickets
# WHERE contactID differs from current users contact id AND
#   AND tickets organization is not current users primary organisation
Object | /tickets/*{Ticket.ContactID NE $CurrentUser.Contact.ID && Ticket.OrganisationID NE $CurrentUser.
Contact.PrimaryOrganisationID} |  ------

# DO NOT ALLOW ACCESS to articles
# WHERE customer visible flag is not set
Object | /tickets/*/articles/*{Article.CustomerVisible NE 1}                   |  ------
```

Eigenschaftsberechtigungen

Die optionale Angabe von Eigenschaftsberechtigungen erlaubt die Beschränkung des Zugriffs auf Objektattribute. Um diese angeben zu können wird Kenntnis der vorhandenen Attribute benötigt. Im Hinblick auf dynamische Felder können diese in versch. Umgebungen variieren.

Die verfügbaren Ressourcen, Objekte und deren Eigenschaften können der API-Dokumentation entnommen werden.

Eine konkrete Anwendung erfolgt in Rolle "Customer", die die im Self Service Portal darstellbaren Ticketeigenschaften einschränkt in dem eine White-List definiert wird.

Beispiel Eigenschaftsberechtigungen

```
# LIMIT shown ticket attributes to the mentioned (line break for improved readability)
Property | /tickets/*{Ticket.[TicketNumber,Age,Articles,Changed,
   ContactID,Created,CreateTimeUnix,DynamicFields,
   OrganisationID,PriorityID,QueueID,StateID,TypeID
]} |  -R---
```

Rollen

Rollen werden dadurch definiert, dass sie neben allg. Kopfdaten wie Name, Gültigkeit, Kommentar etc. eine Menge aus Berechtigungen beinhalten (1:N-Beziehung, 1 Rolle N Berechtigungen).

Nutzer können mehreren Rollen zugeordnet werden, die resultierenden Berechtigungen sind dabei die Vereinigungsmenge aller einzelnen Rollenberechtigungen. Ein "Deny" gewinnt dabei immer und wirkt schwerer.

```
    Role1  | resource | /resource/xyz/abc | -R---
  + Role2  | resource | /resource/xyz/abc | -----
  + Role3  | resource | /resource/xyz/abc | C----
  = Result | resource | /resource/xyz/abc | CR---

    Role1  | resource | /resource/xyz/abc | -R---
  + Role2  | resource | /resource/xyz/abc | CRUD-
  + Role3  | resource | /resource/xyz/abc | ----X
  = Result | resource | /resource/xyz/abc | ----X
```

Beispiele

Beispiel 1 - Team Red

Szenario: Agenten in Rolle "Team Red" sollen nur Tickets in "Team Red" und "Servicedesk" sehen.

Lösungsskizze:

Die Vorgaberolle "Ticket Agent" kann nicht benutzt werden, da darin Zugriffsrechte auf alle Tickets enthalten sind. Es wird also eine Art "Basisberechtigung" benötigt die alles bereitstellt was man zur Ticketbearbeitung benötigt außer Zugriffe auf konkrete Teams/Queues (Object-/Template-Action-IDs können variieren).

Rollendefinition "Basic Ticket Agent"

```
Resource | /contacts                              | -R---
Resource | /links                                 | CRUD-
Resource | /organisations                         | -R---
Resource | /system/automation                     | -RU--
Resource | /system/automation/*                   | -----
Resource | /system/automation/macros              | --U--
Resource | /system/automation/macros/*            | -----
Resource | /system/communication                  | -R---
Resource | /system/communication/*                | -----
Resource | /system/communication/channels         | -R---
Resource | /system/communication/sendertypes      | -R---
Resource | /system/communication/systemaddresses | -R---
Resource | /system/objectactions                  | -R---
Resource | /system/objectactions/*                | -----
Resource | /system/objectactions/2                | -R---
Resource | /system/objectactions/3                | -R---
Resource | /system/objectactions/4                | -R---
Resource | /system/objectactions/6                | -R---
Resource | /system/slas                           | -R---
Resource | /system/templates                      | -R---
Resource | /system/templates/*                    | -----
Resource | /system/templates/4                    | -R---
Resource | /system/templates/5                    | -R---
Resource | /system/templates/7                    | -R---
Resource | /system/textmodules                    | -R---
Resource | /system/ticket                         | -R---
Resource | /system/ticket/*                       | -----
Resource | /system/ticket/locks                   | -R---
Resource | /system/ticket/priorities              | -R---
Resource | /system/ticket/states                  | -R---
Resource | /system/ticket/types                   | -R---
Resource | /system/ticket/queues                  | -R---   # specific to Basic Ticket Agent (allow accessing
queues)
Resource | /system/ticket/queues/*                | -----   # specific to Basic Ticket Agent (prevent accessing
specific queues)
Resource | /tickets                               | CRUD-   # specific to Basic Ticket Agent (allow anything on
tickets)
Object   | /tickets/*{}                           | -----   # specific to Basic Ticket Agent (prohibit anything
on certain tickets)
```

Im folgenden wird angenommen dass Team "Servicedesk" die ID 2 hat und "Team Red" die ID 4. Dadurch definiert sich die Rolle "Team Red" wie folgt:

Rollendefinition "Team Red"

```
Resource | /system/ticket/queues/2                | -R---   # specific to Team Red (allow knowledge of queue
"Servicedesk", with ID 2)
Resource | /system/ticket/queues/4                | -R---   # specific to Team Red (allow knowledge of queue
"Team Red" with ID 4)
Object   | /tickets/*{Ticket.QueueID IN [2,4]}   | CRUD-   # specific to Team Red (allow anything with ID 2 or
4)
```

Anwendung Rolle "Team Red"

Zur Anwendung werden nun dem Agenten "Rodney Red" die Rollen "Agent User", "Basic Ticket Agent" und "Team Red Only" zugeordnet.

Beispiel 2 - SECRET Customer

Szenario: ein Kunde dessen Daten (und Tickets) nicht alle Agenten einsehen dürfen bzw. ein Kunde für den nicht alle Agenten arbeiten dürfen (aka "Mandant").

Lösungsskizze:

"Ticket Agent" wäre prinzipiell OK, müsste aber eingeschränkt werden. Da wir jedoch auch eine Rolle benötigen für Nutzer die für SECRET tätig sein dürfen, behalten wir diese Rolle bei und kopieren sie auf eine neue Rolle "Ticket Agent ohne SECRET". Im folgenden wird angenommen, dass die Organisation "SECRET" die ID 2 hat. Dadurch definiert sich die Rolle "Ticket Agent ohne SECRET" wie folgt (Object-/Template-Action-IDs können variieren). Die ergänzenden Berechtigungen sind mit Kommentar versehen.

Rollendefinition "Ticket Agent ohne SECRET"

```
Resource | /contacts                                          | -R---
Resource | /contacts/*{Contact.PrimaryOrganisationID NE 2}   | -R--- # specific (limit access to contacts not
belonging to Org wit ID 2)
Resource | /links                                             | CRUD-
Resource | /organisations                                     | -R---
Resource | /organisations/*{Organisation.Number NE "SECRET"} | -R--- # specific (limit access to org. with
other Number than "SECRET")
Resource | /system/automation                                 | -RU--
Resource | /system/automation/*                               | -----
Resource | /system/automation/macros                          | --U--
Resource | /system/automation/macros/*                        | -----
Resource | /system/communication                              | -R---
Resource | /system/communication/*                            | -----
Resource | /system/communication/channels                     | -R---
Resource | /system/communication/sendertypes                  | -R---
Resource | /system/communication/systemaddresses              | -R---
Resource | /system/objectactions                              | -R---
Resource | /system/objectactions/*                            | -----
Resource | /system/objectactions/2                            | -R---
Resource | /system/objectactions/3                            | -R---
Resource | /system/objectactions/4                            | -R---
Resource | /system/objectactions/6                            | -R---
Resource | /system/slas                                       | -R---
Resource | /system/templates                                  | -R---
Resource | /system/templates/*                                | -----
Resource | /system/templates/4                                | -R---
Resource | /system/templates/5                                | -R---
Resource | /system/templates/7                                | -R---
Resource | /system/textmodules                                | -R---
Resource | /system/ticket                                     | -R---
Resource | /system/ticket/*{Ticket.OrganisationID EQ 2}       | ----- # specific (prohibit access to tickets of
Org 2)
Resource | /system/ticket/locks                               | -R---
Resource | /system/ticket/priorities                          | -R---
Resource | /system/ticket/states                              | -R---
Resource | /system/ticket/types                               | -R---
```

Auswirkungen Berechtigungen in GUI

Ressourcenberechtigungen sind erforderlich um bestimmte Bereiche der GUI oder Zugriffe überhaupt zu bekommen, z.B. wird mind. Read auf Ressource "/tickets" benötigt um das Ticket-Modul zur Verfügung zu haben - analog gilt das für FAQ- ("/faq"), Asset- ("/cmdb") etc.

Die Zugehörigkeit von Nutzern zu Rollen ist Voraussetzung für die Nutzung von konfigurierbare Ticket-/Artikelaktionen oder Ticketvorlagen.

Die Berechtigungen und angezeigten Inhalte eines Kundennutzers im SSP werden durch die Vorgaberolle "Customer" definiert. Der Zugriff auf Objekte des Assetmanagements wird weiterhin im Backend durch die Konfiguration des Zuordnungsmappings (SysConfig "AssignedConfigItemsMapping") definiert und ergänzt das Berechtigungssystem.

Sonstiges Tipps

Für Pflege von Rollen kann auch https://github.com/cape-it/kix18sync und Libre Office Calc, MS Excel verwendet werden (CSV, UTF-8 codiert, Semikolon-separiert, Formate Beispiel-CSV beachten).

Verwendung Skript kix18.ManageRoles.pl

```
# Alle Rollen einer KIX-Umgebung exportieren
[vagrant@localhost kix18sync]$ ./bin/kix18.ManageRoles.pl --config ./config/kix18.ManageRoles.cfg --url
https://t2020112690000644-api.kix.cloud --d /tmp

# Rollen importieren
[vagrant@localhost kix18sync]$ ./bin/kix18.ManageRoles.pl --config ./config/kix18.ManageRoles.cfg --url
https://t2020112690000644-api.kix.cloud --dir import --f ./sample/RoleData_Sample.csv --verbose 2
```

⚠ Die Rolle des Nutzers welcher für das Skript verwendet wird, sollten NICHT im CSV-File verändert werden. Das hat derzeit den Effekt, dass der Ast auf dem man sitzt abgesägt wird.