

## 1. Qu'est-ce que JQL ?

Le langage Jira Query Language (JQL) est un langage de requête puissant utilisé dans Jira pour rechercher et filtrer les tickets. Il permet de trouver des problèmes en fonction de critères spécifiques comme le statut, le type de problème, l'affectation, la priorité, et bien plus encore.

## 2. Structure de base d'une requête JQL

Une requête JQL est généralement constituée de trois éléments :

- **Champ** : le nom de l'attribut du ticket (par exemple status, assignee, project, etc.).
- **Opérateur** : l'action à effectuer sur le champ (par exemple =, !=, >, <, IN, etc.).
- **Valeur** : la valeur recherchée (par exemple, un nom de statut ou une date spécifique).

## Exemple :

assignee = "Jean Dupont" AND status = "En cours"

Cela recherche tous les tickets assignés à "Jean Dupont" et ayant un statut "En cours".

## 3. Opérateurs de JQL

Voici quelques opérateurs courants utilisés dans JQL :

= : égal à.  
!= : différent de.  
> : supérieur à.  
< : inférieur à.  
>= : supérieur ou égal à.  
<= : inférieur ou égal à.  
IN : pour spécifier une liste de valeurs.  
NOT IN : pour spécifier que la valeur ne doit pas figurer dans une liste.  
~ : recherche par texte partiel (utile pour les chaînes de caractères).  
IS : pour vérifier si un champ a une valeur spécifique ou est vide.

## 4. Exemples de requêtes JQL

### a. Filtrer par projet et statut

Cette requête permet de récupérer tous les tickets du projet "PROJ" qui sont dans un statut "Ouvert" :

```
project = PROJ AND status = "Ouvert"
```

### b. Filtrer par utilisateur affecté et priorité

Pour afficher tous les tickets assignés à "Marie" et ayant une priorité "Haute" :

```
assignee = "Marie" AND priority = "Haute"
```

### **c. Filtrer par date de création**

Pour trouver tous les tickets créés après le 1er janvier 2025 :

```
created > "2025-01-01"
```

### **d. Rechercher des tickets sans affectation**

Pour obtenir tous les tickets qui ne sont pas encore assignés :

```
assignee IS EMPTY
```

### e. Filtrer par étiquettes

Si vous voulez trouver tous les tickets étiquetés avec "bug" :

```
labels IN ("bug")
```

## 5. Requêtes avancées avec des fonctions

### a. Rechercher des tickets en fonction de l'âge

Pour afficher tous les tickets qui ont plus de 5 jours :

```
created <= -5d
```

Cela renvoie les tickets créés il y a plus de 5 jours.



### **b. Rechercher des tickets en fonction de l'historique des transitions**

Vous pouvez aussi chercher des tickets qui ont changé de statut récemment. Par exemple, tous les tickets qui ont été fermés dans les 7 derniers jours :

```
status changed to "Fermé" after -7d
```

### **c. Utilisation de ORDER BY**

Il est possible de trier les résultats d'une requête JQL avec ORDER BY. Par exemple, trier les tickets par date de création la plus récente :

```
project = "PROJ" ORDER BY created DESC
```

### 6. Gestion des erreurs courantes

- **Erreur de syntaxe** : Vérifiez que tous les champs, opérateurs et valeurs sont correctement formatés. Assurez-vous également de ne pas oublier les parenthèses lorsque vous utilisez plusieurs critères.
- **Champ inexistant** : Assurez-vous que le champ utilisé existe dans votre instance Jira. Si nécessaire, vérifiez les noms des champs personnalisés dans votre système.
- **Valeur incorrecte** : Certaines valeurs, comme les noms d'utilisateurs ou les projets, sont sensibles à la casse. Assurez-vous de respecter la casse exacte.



## 7. Conseils pratiques

Utilisez des alias pour simplifier les requêtes dans des projets complexes. Par exemple, au lieu de `assignee = "Jean Dupont"`, vous pouvez faire `assignee = jdupont`. Expérimentez avec les fonctions temporelles comme `-1w` pour la semaine dernière, `-2d` pour deux jours avant, etc.

Utilisez des parenthèses pour regrouper des conditions et clarifier la logique (comme dans les expressions booléennes complexes).

### 1. Attributs de base

- **assignee** : La personne à qui le ticket est assigné.
  - Exemple : `assignee = "Jean Dupont"`
- **reporter** : La personne qui a créé le ticket.
  - Exemple : `reporter = "Marie"`
- **status** : Le statut actuel du ticket (ex : `"Ouvert"`, `"En cours"`, `"Fermé"`).

- Exemple : status = "En cours"
- **priority** : La priorité du ticket (ex : "Haute", "Moyenne", "Basse").
  - Exemple : priority = "Haute"
- **created** : La date et l'heure de création du ticket.
  - Exemple : created >= "2025-01-01"
- **updated** : La dernière date de mise à jour du ticket.
  - Exemple : updated >= -1w (tickets mis à jour dans la dernière semaine)
- **due** : La date d'échéance du ticket.
  - Exemple : due <= "2025-03-31"
- **resolved** : La date à laquelle le ticket a été résolu.
  - Exemple : resolved <= -3d (résolu dans les 3 derniers jours)
- **summary** : Le titre ou le résumé du ticket.
  - Exemple : summary ~ "problème de connexion"
- **description** : La description détaillée du ticket.

- Exemple : description ~ "erreur d'affichage"
- **labels** : Les étiquettes ou mots-clés associés au ticket.
  - Exemple : labels IN ("bug", "urgent")
- **assignee** : La personne à qui le ticket est affecté.
  - Exemple : assignee = "Jean Dupont"

## 2. Attributs avancés

- **project** : Le projet auquel appartient le ticket.
  - Exemple : project = "PROJ123"
- **issueType** : Le type de ticket (ex : "Bug", "Tâche", "Histoire").
  - Exemple : issueType = "Bug"
- **component** : Le composant (partie ou module du produit) lié au ticket.
  - Exemple : component = "Frontend"
- **fixVersion** : La version dans laquelle le ticket sera corrigé ou résolu.
  - Exemple : fixVersion = "1.0"

- **affectsVersion** : La version qui est affectée par le problème.
  - Exemple : affectsVersion = "2.0"
- **environment** : L'environnement dans lequel le problème a été observé (par exemple, "Production", "Test").
  - Exemple : environment = "Production"
- **securityLevel** : Le niveau de sécurité du ticket.
  - Exemple : securityLevel = "Confidentiel"
- **due** : La date d'échéance du ticket.
  - Exemple : due = "2025-03-20"
- **priority** : La priorité d'un ticket.
  - Exemple : priority = "Moyenne"

## 3. Attributs relatifs aux relations

- **epic** : L'Epic auquel un ticket est lié (utilisé dans les frameworks Agile).
  - Exemple : epic = "EPIC-123"
- **parent** : Le ticket parent (si applicable).
  - Exemple : parent = "PROJ-123"
- **subtask** : Le ticket est une sous-tâche (ou "sub-task").
  - Exemple : issuetype = "Sub-task"
- **linkedIssues** : Les tickets liés à un autre ticket (par exemple, "Bloque", "Duplication").
  - Exemple : linkedIssues = "PROJ-456"
- **sprint** : Le sprint auquel le ticket appartient (si Jira Agile est utilisé).
  - Exemple : sprint = "Sprint 2"

## 4. Attributs relatifs à l'historique

- **status changed** : Recherche de tickets qui ont changé de statut à un moment donné.

Exemple : status changed to "Fermé" after "2025-03-01"

- **comment** : Recherche de tickets qui contiennent un certain texte dans les commentaires.

Exemple : comment ~ "amélioration nécessaire"

- **worklogAuthor** : Recherche de tickets où un certain utilisateur a ajouté un temps de travail.

Exemple : worklogAuthor = "Jean Dupont"

## 5. Attributs liés aux projets et aux versions

- **version** : Recherche des tickets affectés ou liés à une version spécifique.

Exemple : fixVersion = "1.0"

- **project** : Recherche des tickets dans un projet spécifique.

Exemple : project = "PROJ"

## 6. Attributs temporaires et relatifs aux dates

- **created** : La date de création du ticket.

Exemple : created >= "2025-03-01"

- **updated** : La date de dernière mise à jour du ticket.

Exemple : updated >= -1w

- **due** : La date d'échéance.

Exemple : due = "2025-03-25"

- **resolved** : La date où le ticket a été résolu.

Exemple : resolved <= -2d (résolu il y a deux jours)

### 7. Attributs personnalisés

Jira permet également la création de **champs personnalisés** (custom fields), qui peuvent être utilisés dans les requêtes JQL si votre instance Jira a des champs spécifiques. Par exemple :

- **customfield\_10001** : Un champ personnalisé, comme une priorité métier ou un champ texte.
  - Exemple : customfield\_10001 = "Important"