



Ressources Kuzzle

I. Structure générale des requêtes

Kuzzle fonctionne via une API JSON unifiée. Chaque requête suit une structure de base comprenant :

```
{
  "controller": "nom_du_controlleur",
  "action": "nom_de_l'action",
  "index": "nom_de_l_index",
  "collection": "nom_de_la_collection",
  "body": {
    // Données à envoyer ou critères de recherche
  }
}
```

Les opérations principales sont les suivantes :

Créer un document :

```
await kuzzle.document.create('maBase', 'maCollection', {
  title: 'Nouveau document',
  content: 'Contenu ici'
});
```

Modifier un document existant :

```
await kuzzle.document.update('maBase', 'maCollection', 'id_du_document',
{
  content: 'Contenu modifié'
});
```

Remplacer complètement un document :

```
await kuzzle.document.replace('maBase', 'maCollection', 'id_du_document',
{
  title: 'Titre remplacé',
  content: 'Contenu remplacé'
});
```

Récupérer un document par son ID :

```
const document = await kuzzle.document.get('maBase', 'maCollection', 'id_
du_document');
console.log(document);
```

Rechercher des documents par critères :

```
const results = await kuzzle.document.search('maBase', 'maCollection', {
  query: {
    match: {
      title: 'exemple'
    }
  }
});
```

Exemples d'utilisation de filtres Elasticsearch :

Filtre "match" (correspondance partielle sur un champ) :

```
query: {
  match: {
    nom: 'Dupont'
  }
}
```

Filtre "term" (correspondance exacte) :

```
query: {
  term: {
    statut: 'actif'
  }
}
```

```
}  
}
```

Filtre "range" (plage de valeurs numériques ou dates) :

```
query: {  
  range: {  
    age: {  
      gte: 18,  
      lte: 30  
    }  
  }  
}
```

Filtre "bool" (combinaison de conditions) :

```
query: {  
  bool: {  
    must: [  
      { match: { ville: 'Lille' } },  
      { range: { age: { gte: 25 } } }  
    ],  
    must_not: [  
      { term: { statut: 'inactif' } }  
    ]  
  }  
}
```

II. Authentification avec Kuzzle

Kuzzle prend en charge plusieurs stratégies d'authentification (local, OAuth, etc.). La stratégie par défaut est "local" (nom d'utilisateur + mot de passe).

Connexion (login) :

```
const jwt = await kuzzle.auth.login('local', {  
  username: 'monUser',
```

```
password: 'monMotDePasse'  
});
```

La réponse contient un **JWT (JSON Web Token)** qui doit être stocké et utilisé pour les prochaines requêtes.

Utilisation du JWT :

```
kuzzle.jwt = monJwt;
```

Ou bien automatiquement au prochain login via WebSocket.

Vérification d'authentification (vérifier si un utilisateur est connecté)

```
const user = await kuzzle.auth.getCurrentUser();  
console.log(user);
```

Récupérer un utilisateur et ses rôles

Pour récupérer un utilisateur par son identifiant :

```
const user = await kuzzle.security.getUser('kuid:monUtilisateur');  
console.log(user);
```

Pour obtenir ses rôles :

```
const policies = user.policies;  
console.log(policies);
```

Chaque politique contient les rôles et permissions associées.

III. Exemples de requêtes utiles

Lister tous les documents d'une collection :

```
const results = await kuzzle.document.search('maBase', 'maCollection', {  
  query: {  
    match_all: {}  
  }  
});
```

```
}  
});
```

Mettre à jour un document :

```
await kuzzle.document.update('maBase', 'maCollection', 'id_du_document',  
{  
  field: 'nouvelle valeur'  
});
```

Supprimer un document :

```
await kuzzle.document.delete('maBase', 'maCollection', 'id_du_document');
```

IV. Système d'abonnements en temps réel

Kuzzle permet de s'abonner à des événements en temps réel sur une collection de documents. Lorsqu'un document est créé, modifié ou supprimé, une notification est automatiquement envoyée aux abonnés.

Exemple d'abonnement à la création de documents :

```
await kuzzle.realtime.subscribe('maBase', 'maCollection', {}, notification =>  
{  
  console.log('Nouveau document :', notification);  
});
```

Exemple avec un filtre spécifique :

```
await kuzzle.realtime.subscribe('maBase', 'maCollection', {  
  equals: { statut: 'urgent' }  
, notification => {  
  console.log('Document urgent ajouté ou modifié :', notification);  
});
```

L'abonnement retourne un identifiant qui peut être utilisé pour s'unsubscribe :

```
const roomId = await kuzzle.realtime.subscribe(...);  
await kuzzle.realtime.unsubscribe(roomId);
```

Ces fonctionnalités sont particulièrement utiles pour créer des applications réactives (ex : tableaux de bord, chats, systèmes de suivi en temps réel).