

TP API Facebook – Lucas MASTIER

Requêtes CRUD

// Utilisateurs

```
db.createCollection("utilisateurs", {
  validator: {
    $jsonSchema: {
      bsonType: "object",
      required: ["email", "nom", "prenom", "adresse"],
      properties: {
        email: {
          bsonType: "string",
          description: "Adresse email unique de l'utilisateur"
        },
        nom: {
          bsonType: "string",
          description: "Nom de l'utilisateur"
        },
        prenom: {
          bsonType: "string",
          description: "Prénom de l'utilisateur"
        },
        adresse: {
          bsonType: "string",
          description: "Adresse complète de l'utilisateur"
        }
      }
    }
  },
  // Autres champs selon les besoins
})
```

```

});

// Événements
db.createCollection("evenements", {
  validator: {
    $jsonSchema: {
      bsonType: "object",
      required: ["nom", "dateDebut", "dateFin", "lieu"],
      properties: {
        nom: {
          bsonType: "string",
          description: "Nom de l'événement"
        },
        dateDebut: {
          bsonType: "date",
          description: "Date de début de l'événement"
        },
        dateFin: {
          bsonType: "date",
          description: "Date de fin de l'événement"
        },
        lieu: {
          bsonType: "string",
          description: "Lieu de l'événement"
        }
      }
    }
  },
  // Autres champs selon les besoins
});

```

```
// Groupes
db.createCollection("groupes", {
  validator: {
    $jsonSchema: {
      bsonType: "object",
      required: ["nom", "type", "autoriserPublication", "autoriserCreationEvenement"],
      properties: {
        nom: {
          bsonType: "string",
          description: "Nom du groupe"
        },
        type: {
          bsonType: "string",
          enum: ["public", "prive", "secret"],
          description: "Type de groupe (public, privé, secret)"
        },
        autoriserPublication: {
          bsonType: "bool",
          description: "Autoriser les membres à publier dans le groupe"
        },
        autoriserCreationEvenement: {
          bsonType: "bool",
          description: "Autoriser les membres à créer des événements dans le groupe"
        }
      }
    }
  },
  // Autres champs selon les besoins
});
```

```
// Fils de discussions
```

```

db.createCollection("filsDeDiscussions", {
  validator: {
    $jsonSchema: {
      bsonType: "object",
      required: ["contenu", "groupeId", "evenementId"],
      properties: {
        contenu: {
          bsonType: "string",
          description: "Contenu du fil de discussion"
        },
        groupeId: {
          bsonType: "objectId",
          description: "ID du groupe associé (peut être null)"
        },
        evenementId: {
          bsonType: "objectId",
          description: "ID de l'événement associé (peut être null)"
        }
      }
    }
  }
});

```

// Albums photo

```

db.createCollection("albumsPhoto", {
  validator: {
    $jsonSchema: {
      bsonType: "object",
      required: ["evenementId"],
      properties: {

```

```
    evenementId: {  
      bsonType: "objectId",  
      description: "ID de l'événement associé"  
    }  
    // Autres champs selon les besoins  
  }  
}  
});
```

// Sondages

```
db.createCollection("sondages", {  
  validator: {  
    $jsonSchema: {  
      bsonType: "object",  
      required: ["evenementId"],  
      properties: {  
        evenementId: {  
          bsonType: "objectId",  
          description: "ID de l'événement associé"  
        }  
        // Autres champs selon les besoins  
      }  
    }  
  }  
});
```

// Billetterie

```
db.createCollection("billetterie", {  
  validator: {  
    $jsonSchema: {
```

```
bsonType: "object",
required: ["eventId", "nom", "montant", "quantiteLimitee"],
properties: {
  eventId: {
    bsonType: "objectId",
    description: "ID de l'événement associé"
  },
  nom: {
    bsonType: "string",
    description: "Nom du type de billet"
  },
  montant: {
    bsonType: "double",
    description: "Montant du billet"
  },
  quantiteLimitee: {
    bsonType: "int",
    description: "Quantité limitée de billets"
  }
}
// Autres champs selon les besoins
}
}
}
});
```

```
// Requêtes CRUD MongoDB
```

```
// 1. Utilisateurs
```

```
// Créer un utilisateur
```

```
// POST /utilisateurs
```

```
db.utilisateurs.insertOne({  
  email: "john.doe@example.com",  
  nom: "Doe",  
  prenom: "John",  
  adresse: "123 Main St"  
  // Autres champs selon les besoins  
});
```

```
// Lire un utilisateur
```

```
// GET /utilisateurs/:email
```

```
db.utilisateurs.findOne({ email: "john.doe@example.com" });
```

```
// Mettre à jour un utilisateur
```

```
// PUT /utilisateurs/:email
```

```
db.utilisateurs.updateOne({ email: "john.doe@example.com" }, { $set: { nom: "Smith" } });
```

```
// Supprimer un utilisateur
```

```
// DELETE /utilisateurs/:email
```

```
db.utilisateurs.deleteOne({ email: "john.doe@example.com" });
```

```
// 2. Événements
```

```
// Créer un événement
```

```
// POST /evenements
```

```
db.evenements.insertOne({  
  nom: "Soirée Concert",  
  dateDebut: ISODate("2023-12-01T20:00:00Z"),  
  dateFin: ISODate("2023-12-01T23:59:59Z"),  
  lieu: "Salle de concert XYZ"  
  // Autres champs selon les besoins  
});
```

// Lire un événement

// GET /evenements/:nom

db.evenements.findOne({ nom: "Soirée Concert" });

// Mettre à jour un événement

// PUT /evenements/:nom

db.evenements.updateOne({ nom: "Soirée Concert" }, { \$set: { lieu: "Nouvelle Salle" } });

// Supprimer un événement

// DELETE /evenements/:nom

db.evenements.deleteOne({ nom: "Soirée Concert" });

