

```
const express = require("express")
const app = express()
const {MongoClient} = require('mongodb');
const url = 'mongodb://localhost:27017';
const dbName = 'tester';
let db
MongoClient.connect(url, function(err, client) { console.log("Connexion réussie avec Mongo");
db = client.db(dbName);
});
```

```
async function connectToDatabase() {
  try {
    const client =await MongoClient.connect(url);
    console.log("Connexion réussie");
    db=client.db (dbName);
  } catch (err) {
    console.error("Erreur de connexion à la base de données:", err);
    process.exit(1);
  }
}
```

```
async function startServer() {
  await connectToDatabase();
  app.listen(5000, () => {
    console.log('REST API via ExpressJS');
  });
}
startServer();
```

```
app.get('/api', async (req, res) => {
  if (!db) {
    return res.status(500).json({ error: "Connexion à la base de données non établie."});
  }
  try {
```

```

const docs= await db.collection('datas').find({}).toArray();
res.status(200).json(docs);
} catch (err) {
console.error("Erreur lors de la récupération des formations:", err);
return res.status(500).json({error: "Erreur lors de la récupération des données."});
}
});

```

///// recherche par email

```

app.get('/api/:email', async (req, res) => {
  if (!db) {
    return res.status(500).json({ error: "Connexion à la base de données non établie."});
  }
  try {
    const email=req.params.email;
    const client= await db.collection('datas').find({email}).toArray();
    res.status(200).json(client);
  } catch (err) {
    console.error("Erreur lors de la récupération des client:", err);
    return res.status(500).json({error: "Erreur lors de la récupération des données."});
  }
});

```

////obloge

```

app.use(express.json())
app.post('/api', async (req, res)=>{
  const formationData=req.body;
  console.log("log:", req.body);
  const formation = await db.collection('datas').insertone(formationData)
  console.log("resultat de insertone:", formation);
  res.status(200).json(formation);
});

```

```

app.delete('/client/:email', async (req, res) => {
  if (!db) {
    return res.status(500).json({ error: "Connexion à la base de données "})
  }
  try {
    const email = req.params.email;
    const client = await db.collection('datas').deleteOne({email});
    res.status(200).json(client);
  } catch (err) {
    console.error("Erreur lors de la récupération des clients:", err);
    return res.status(500).json({ error: "Erreur lors de la récupération "})
  });

  //////////////////////////////////////

//      const express = require("express");
// const app = express();
// const { MongoClient } = require('mongodb');

// const url = 'mongodb://localhost:27017';
// const dbName = 'tester';
// let db;

// async function connectToDatabase() {
//   try {
//     const client = await MongoClient.connect(url);
//     console.log("Connexion réussie à MongoDB");
//     db = client.db(dbName);
//   } catch (err) {
//     console.error("Erreur de connexion à la base de données:", err);
//     process.exit(1);
//   }
// }

// async function startServer() {
//   await connectToDatabase();
//   app.listen(5000, () => {

```

```
// console.log('Serveur démarré sur le port 5000');
// });
// }

// startServer();

// // Middleware pour parser le JSON
// app.use(express.json());

// // GET toutes les données
// app.get('/api', async (req, res) => {
//   if (!db) {
//     return res.status(500).json({ error: "Connexion à la base de données non établie." });
//   }
//   try {
//     const docs = await db.collection('datas').find({}).toArray();
//     res.status(200).json(docs);
//   } catch (err) {
//     console.error("Erreur lors de la récupération des données:", err);
//     res.status(500).json({ error: "Erreur lors de la récupération des données." });
//   }
// });

// // Recherche par email
// app.get('/api/:email', async (req, res) => {
//   if (!db) {
//     return res.status(500).json({ error: "Connexion à la base de données non établie." });
//   }
//   try {
//     const email = req.params.email;
//     const client = await db.collection('datas').find({ email }).toArray();
//     if (client.length === 0) {
//       return res.status(404).json({ message: "Aucun client trouvé avec cet email." });
//     }
//   }
// });
```

```

//    res.status(200).json(client);
//  } catch (err) {
//    console.error("Erreur lors de la récupération du client:", err);
//    res.status(500).json({ error: "Erreur serveur." });
//  }
// });

// // Ajouter un nouveau client
// app.post('/api', async (req, res) => {
//   if (!db) {
//     return res.status(500).json({ error: "Connexion à la base de données non
établie." });
//   }
//   try {
//     const clientData = req.body;
//     const result = await db.collection('datas').insertOne(clientData);
//     res.status(201).json({ message: "Client ajouté", result: result.ops[0] });
//   } catch (err) {
//     console.error("Erreur lors de l'ajout du client:", err);
//     res.status(500).json({ error: "Erreur lors de l'ajout des données." });
//   }
// });

// // Supprimer un client par email
// app.delete('/api/:email', async (req, res) => {
//   if (!db) {
//     return res.status(500).json({ error: "Connexion à la base de données non
établie." });
//   }
//   try {
//     const email = req.params.email;
//     const result = await db.collection('datas').deleteOne({ email });
//     if (result.deletedCount === 0) {
//       return res.status(404).json({ message: "Client non trouvé." });
//     }
//     res.status(200).json({ message: "Client supprimé avec succès" });
//   }

```

```

//   } catch (err) {
//       console.error("Erreur lors de la suppression du client:", err);
//       res.status(500).json({ error: "Erreur serveur." });
//   }
// });
+++++
const express = require("express")
const app = express()
const {MongoClient} = require('mongodb');
const url = 'mongodb://localhost:27017';
const dbName = 'tester';
let db
MongoClient.connect(url, function(err, client) { console.log("Connexion réussi avec
Mongo");
db = client.db(dbName);
});

async function connectToDatabase() {
  try {
    const client =await MongoClient.connect(url);
    console.log("Connexion réussie");
    db=client.db (dbName);
  } catch (err) {
    console.error("Erreur de connexion à la base de données:", err);
    process.exit(1);
  }
}

async function startServer() {
  await connectToDatabase();
  app.listen(5000, () => {
    console.log('REST API via ExpressJS');
  });
}
startServer();

```

```

app.get('/api', async (req, res) => {
  if (!db) {
    return res.status(500).json({ error: "Connexion à la base de données non établie."});
  }
  try {
    const docs= await db.collection('datas').find({}).toArray();
    res.status(200).json(docs);
  } catch (err) {
    console.error("Erreur lors de la récupération des formations:", err);
    return res.status(500).json({error: "Erreur lors de la récupération des données."});
  }
});

```

///// recherche par email

```

app.get('/api/:email', async (req, res) => {
  if (!db) {
    return res.status(500).json({ error: "Connexion à la base de données non établie."});
  }
  try {
    const email=req.params.email;
    const client= await db.collection('datas').find({email}).toArray();
    res.status(200).json(client);
  } catch (err) {
    console.error("Erreur lors de la récupération des client:", err);
    return res.status(500).json({error: "Erreur lors de la récupération des données."});
  }
});

```

////obloge

```

app.use(express.json())
app.post('/api', async (req, res)=>{
  const formationData=req.body;
  console.log("log:", req.body);

```

```
const formation = await db.collection('datas').insertone(formationData)
console.log("resultat de insertone:", formation);
res.status(200).json(formation);
});
```

```
app.delete('/client/:email', async (req, res) => {
  if (!db) {
    return res.status(500).json({ error: "Connexion à la base de données "})
  }
  try {
    const email = req.params.email;
    const client = await db.collection('datas').deleteOne({email});
    res.status(200).json(client);
  } catch (err) {
    console.error("Erreur lors de la récupération des clients:", err);
    return res.status(500).json({ error: "Erreur lors de la récupération "})
  });
});
```

```
////////////////////////////////////
//      const express = require("express");
// const app = express();
// const { MongoClient } = require('mongodb');
```

```
// const url = 'mongodb://localhost:27017';
// const dbName = 'tester';
// let db;
```

```
// async function connectToDatabase() {
//   try {
//     const client = await MongoClient.connect(url);
//     console.log("Connexion réussie à MongoDB");
//     db = client.db(dbName);
//   } catch (err) {
//     console.error("Erreur de connexion à la base de données:", err);
//     process.exit(1);
//   }
// }
```



```

// }
//}

// async function startServer() {
//   await connectToDatabase();
//   app.listen(5000, () => {
//     console.log('Serveur démarré sur le port 5000');
//   });
//}

// startServer();

// // Middleware pour parser le JSON
// app.use(express.json());

// // GET toutes les données
// app.get('/api', async (req, res) => {
//   if (!db) {
//     return res.status(500).json({ error: "Connexion à la base de données non établie." });
//   }
//   try {
//     const docs = await db.collection('datas').find({}).toArray();
//     res.status(200).json(docs);
//   } catch (err) {
//     console.error("Erreur lors de la récupération des données:", err);
//     res.status(500).json({ error: "Erreur lors de la récupération des données." });
//   }
// });

// // Recherche par email
// app.get('/api/:email', async (req, res) => {
//   if (!db) {
//     return res.status(500).json({ error: "Connexion à la base de données non établie." });
//   }

```

```
// try {
//   const email = req.params.email;
//   const client = await db.collection('datas').find({ email }).toArray();
//   if (client.length === 0) {
//     return res.status(404).json({ message: "Aucun client trouvé avec cet email."
//   });
//   }
//   res.status(200).json(client);
// } catch (err) {
//   console.error("Erreur lors de la récupération du client:", err);
//   res.status(500).json({ error: "Erreur serveur." });
// }
// });
```

```
// // Ajouter un nouveau client
// app.post('/api', async (req, res) => {
//   if (!db) {
//     return res.status(500).json({ error: "Connexion à la base de données non
// établie." });
//   }
//   try {
//     const clientData = req.body;
//     const result = await db.collection('datas').insertOne(clientData);
//     res.status(201).json({ message: "Client ajouté", result: result.ops[0] });
//   } catch (err) {
//     console.error("Erreur lors de l'ajout du client:", err);
//     res.status(500).json({ error: "Erreur lors de l'ajout des données." });
//   }
// });
```

```
// // Supprimer un client par email
// app.delete('/api/:email', async (req, res) => {
//   if (!db) {
//     return res.status(500).json({ error: "Connexion à la base de données non
// établie." });
//   }
//   try {
```

```
//    const email = req.params.email;
//    const result = await db.collection('datas').deleteOne({ email });
//    if (result.deletedCount === 0) {
//        return res.status(404).json({ message: "Client non trouvé." });
//    }
//    res.status(200).json({ message: "Client supprimé avec succès" });
// } catch (err) {
//    console.error("Erreur lors de la suppression du client:", err);
//    res.status(500).json({ error: "Erreur serveur." });
// }
// };
```