

app.js

```
const express=require("express");
```

```
const restapi=require("../src/models/rest");
```

```
app.use(express.json());
```

```
const app=express();
```

```
const port=process.env.PORT || 3000;
```

```
const a=[];
```

```
app.post("/res",async(req,res)=>{
```

```
  try{
```

```
    const restdirectories=new restapi(req.body);
```

```
    console.log(req.body);
```

```
    const insertrest=await restdirectories.save();
```

```
    res.status(201).send(insertrest);
```

```
  }
```

```
  catch(e) {
```

```
    res.status(400).send(e);
```

```
  }
```

```
function incomingreqthings(req, res, next) {/* this is for incoming request*/
```

```
  console.log(`Req arrived- [{req.method}] ${req.path}`);
```

```
  next();
```

```
}
```

```
function b(req, res, next) {/*Validator for newqueries*/
```

```
  const {firstName, lastName, hobby} = req.body;
```

```
  if (!firstName || !lastName || !hobby) {
```

```
    return res.status(400).json({  
      error: 'Must contain input data'
```

```
    });
```

```
  }
```

```
  next();
```

```
}
```

```
app.use(incomingreqthings);
```

```

app.post('/user', b, (req, res) => { /* it adds a new client*/
  const newacc = String(a.length + 1);
  const newac = {
    id: newacc,
    firstName: req.body.firstName,
    lastName: req.body.lastName,
    hobby: req.body.hobby
  };
  a.push(newac);
  res.status(201).json(newac);
});

app.get('/users', (req, res) => {
  res.status(200).json(a);
});

app.get("/rest", async(req, res) => {
  try{

    const _id = req.params.id;

    const getres = await restapi.find({}).sort({"id": "1",
      "firstName": "Anshika",
      "lastName": "Agarwal",
      "hobby": "Teaching"});

    app.put('/user/:id', b, (req, res) => { /*It changes the required things for the clients*/
      const newacc = req.params.id;
      const newcursor = p.findIndex(u => u.id === userId);
      if (newcursor === -1) {
        return res.status(404).json({ error: `account is not valid` });
      }

      p[newcursor] = {
        id: userId,
        firstName: req.body.firstName,
        lastName: req.body.lastName,
        hobby: req.body.hobby
      };
      res.status(200).json(userList[userIndex]);
    });

    app.delete("/rest/:id", async(req, res) => { /* This will delete the input things*/
      try{

        const getres = await restapi.findByIdAndDelete(req.params.id);

        res.send(getres);
      }
    }
  }
});

```

```
}catch(e){  
  res.status(500).send(e);  
}  
})  
  
app.listen(PORT, () => {  
  console.log(`hoisted at http://localhost:${PORT}`);  
});
```

Github Link:

[Hitman-97/clientapi.js: this includes all the required files for building a project](#)