

```

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

const app = express();
app.use(express.json());

const port = process.env.PORT || 3000;
const a = [];

function incomingreqthings(req, res, next) { /*Midlleware*/
  console.log(`Req arrived- [${req.method}] ${req.path}`);
  next();
}

function b(req, res, next) { /*Validator*/
  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("/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);
  }
});

app.post('/user', b, (req, res) => { /*To add user*/
  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);
});

```

```

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

app.get("/rest", async (req, res) => { /*A get request*/
  try {
    const getres = await restapi.find({}).sort({
      id: 1,
      firstName: 1,
      lastName: 1,
      hobby: 1
    });
    res.status(200).json(getres);
  } catch (e) {
    res.status(500).send(e);
  }
});

/* To update things*/
app.put('/user/:id', b, (req, res) => {
  const userId = req.params.id;
  const userIndex = a.findIndex(u => u.id === userId);
  if (userIndex === -1) {
    return res.status(404).json({ error: `Account is not valid` });
  }
  a[userIndex] = {
    id: userId,
    firstName: req.body.firstName,
    lastName: req.body.lastName,
    hobby: req.body.hobby
  };
  res.status(200).json(a[userIndex]);
});

/* To delete something */
app.delete("/rest/:id", async (req, res) => {
  try {
    const getres = await restapi.findByIdAndDelete(req.params.id);
    res.send(getres);
  } catch (e) {
    res.status(500).send(e);
  }
});

app.listen(port, () => {
  console.log(`Server is running at http://localhost:${port}`);
});

```