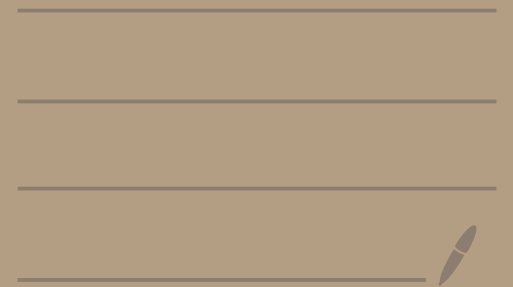


Written Code



```
const express = require("express");
const app = express();
const PORT = 8080;

app.get("/major/:major/coursesInfo", (req, res) => {
  let major = req.params.major;
  let course = req.query.course;
  if (!major && course)
    res.status(400).send("error");
  else
    res.send(major + " " + course);
});

app.listen(PORT, () => { });

module.exports = { }
```

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

function readFile() {
  fs.readFile("file", "encoding");
}

function readFilePromise() {
  let f = fs.promises.readFile("file", "encoding");
  f.then()
    .then()
    .catch();
}

async function readFileAsync() {
  try {
    let f = await fs.promises.readFile("file", "encoding");
    console.log(f);
  }
  catch (err) {
    console.log(err);
  }
}
```

```

CREATE TABLE tableName (
  col1 INTEGER PRIMARY KEY,
  col2 TEXT NOT NULL,
  col3 BLOB,
  ;
);

```

```

INSERT INTO tableName (col1, col2, ...)
VALUES (val1, val2, ...);

```

```

SELECT *
FROM t1
JOIN t2 ON t1.fk = t2.pk
WHERE IN (...) AND LIKE

```

```

UPDATE table
SET col1=val
WHERE cond

```

```

DELETE FROM table
WHERE cond

```

```

let db = await getDBConnection(dbFileName);
let sql = fs.readFile(sqlFileName, "utf-8");
const result = await db.exec(sql);

db.close();

```

```

const test = require("node:test");
const assert = require("node:assert");

```

```

test.describe(" ~", () => {
  test.it(" ~", () => {
    assert.strictEqual(expect, real);
  });
});

```

```

const request = require("supertest");
const app = require("../app");

```

```

jest.mock(" ~");

```

```

describe(" ~", () => {
  test(" ~", () => {
    jest.mockResolvedValue( ~ );
    const res = await request(app).get("path");
    expect(res).toBe( ~ );
  });
});

```