

连接池

1. 缓存多个连接，用户查询时直接使用，不用频繁打开关闭连接

```
const pool = mysql.createPool(cfg);
pool.query(sql, value, (err, results) => {
  if (err) reject(err);
  else resolve(results);
})
```

分页显示

1. 分页数据查询

```
const count =
  await query('SELECT count(*) as count FROM open_course')
    .then(results => results[0].count)
const total = Math.ceil(count / size); // 总页数
const first = page !== 1; // 是否有首页
const last = page !== total; // 是否有最后一页
const prev = page > 1; // 是否有上一页
const next = page < total; // 是否有下一页
```

2. 提取partial

纯后端解决方案，需要刷新页面，体验欠佳，改为前端方案会提升

数据库整合之Sequelize: <http://docs.sequelizejs.com/>

1. 概述：基于Promise的ORM（Object Relation Mapping），支持多种数据库，事务，关联
2. 安装：**npm i sequelize mysql2 -S**
3. 测试：

```
const Sequelize = require('sequelize');
// 建立连接
const sequelize = new Sequelize('kkb', 'kaikeba_admin', 'admin', {
  host: 'localhost',
  dialect: 'mysql', // 方言设置
  pool: {max: 5, acquire: 30000, idle: 10000}, // 连接池
  timestamps: false
});
// 定义模型
const User = sequelize.define('user', { // 字段定义
  firstName: {type: Sequelize.STRING(20), allowNull: true},
  lastName: Sequelize.STRING(20),
```

```

    age: Sequelize.INTEGER,
  });

  // 同步数据库, FORCE如果为true则会删除已存在同名表
  User.sync({force: true}).then(() => {
    // 插入若干测试数据
    return User.create({
      firstName: 'Tom',
      lastName: 'Cruise'
    })
  }).then(() => {
    // 查询前面插入数据
    User.findAll().then(users => {
      console.log(users);
    })
  })
})

```

4. 模型单文件编写及动态导入

~编写模型

```

module.exports = (sequelize, Types) => {
  const OpenCourse = sequelize.define('OpenCourse', {
    name: Types.STRING(50),
    description: Types.STRING(100),
    time: Types.DATE,
    count: Types.INTEGER,
  }, {
    tableName: 'open_course', // 自己命名表名
    timestamps: false, // 禁止seq自动添加createdAt, updatedAt
  })
  OpenCourse.sync();
  return OpenCourse;
}

```

~导入导出index.js

```

const fs = require('fs');
const path = require('path');
// 要导出的对象
const db = {Sequelize, sequelize};
// 动态导入模型
// 读取当前目录中所有文件名
fs.readdirSync(__dirname)
  .filter(file => (file !== 'index.js' && file !== 'db.js'))
  .forEach(file => { // 从文件中导入模型
    const model = sequelize.import(path.join(__dirname, file));
    db[model.name] = model;
  });

module.exports = db;

```

作业：

1. 复习与吸收
2. 预习multer上传文件处理
3. 预习Express中Restful API的编写，及跨域问题的解决
4. 预习express-session、cookie-parser的使用