

mongoDB®

把数据转化成 JS 对象
以便我们可以用 JS 代码
来读写 mongodb 的数据

动手用 Mongoose

<http://mongoosejs.com/docs/index.html>

express

mongoose



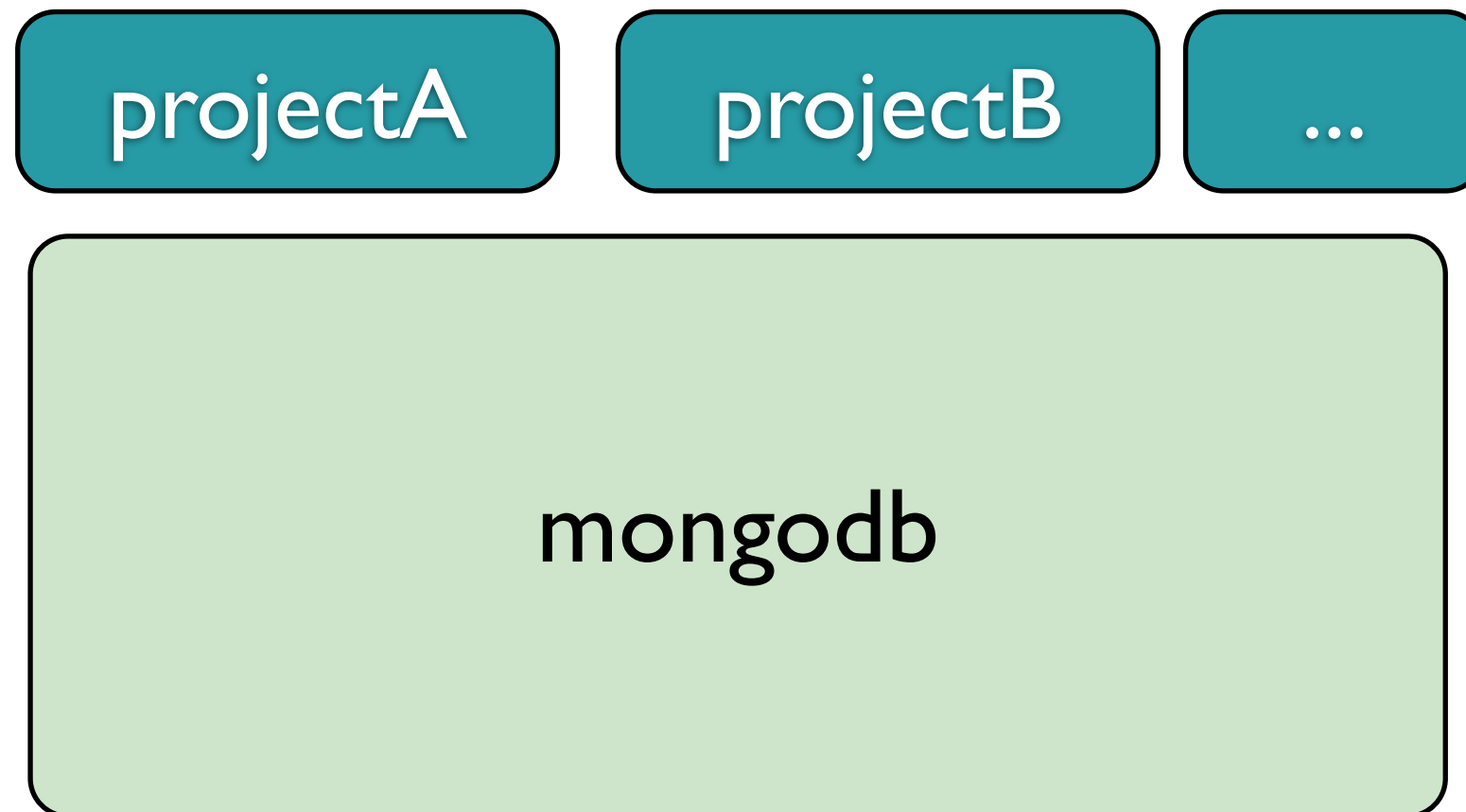
第一步，确保 mongod 启动
也就是确保 mongo shell 可以用

```
mongod --dbpath ~/data/db
```

第二步，JS 代码中连接 mongodb

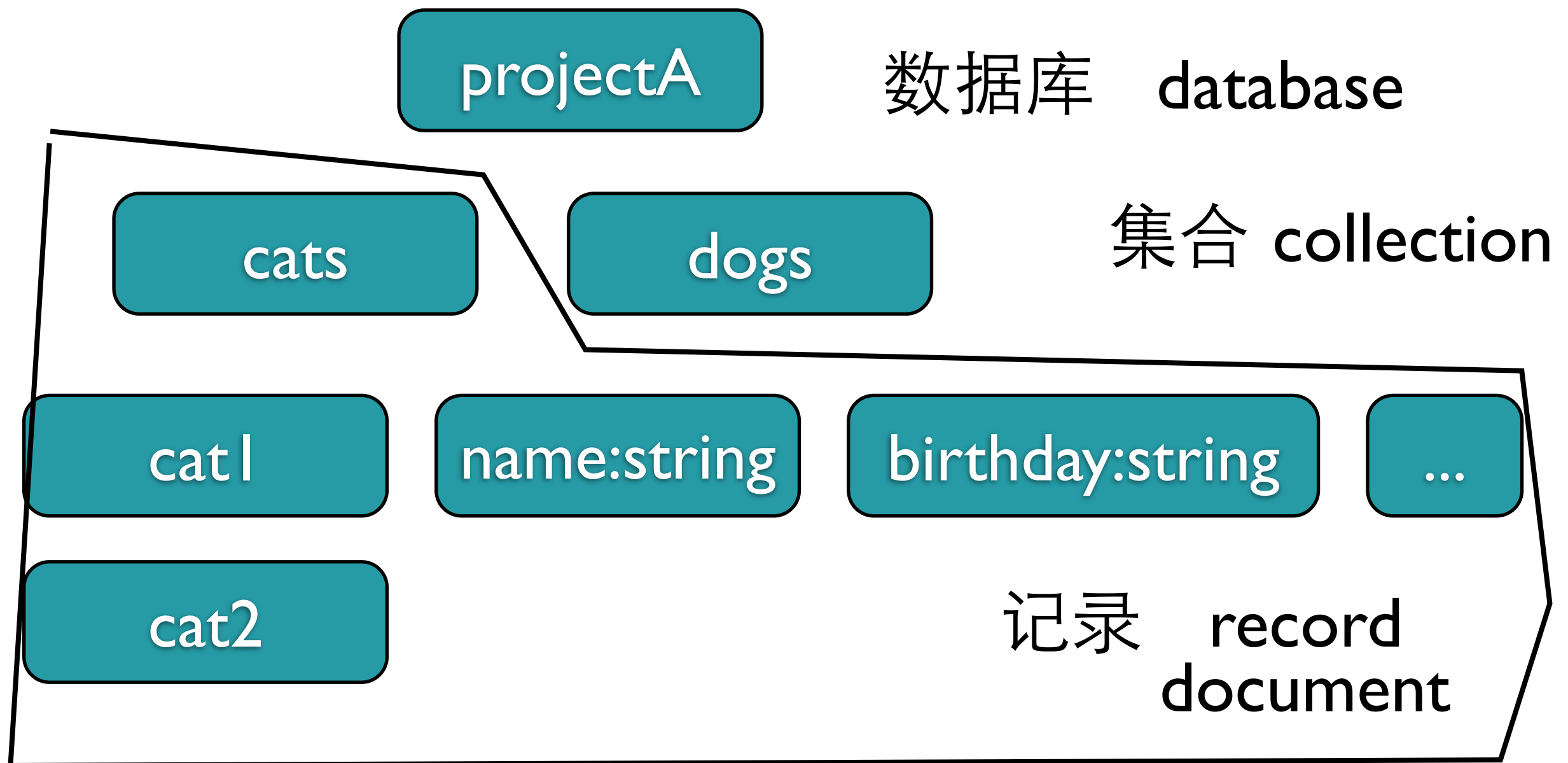
`mongodb://localhost:27017/peterdb`

第二步，JS 代码中连接 mongodb



每个项目只需要一个数据库就够了
因为一个数据库中 can 创建多个 collection 集合

第二步，JS 代码中连接 mongodb



第三步， 建立数据 Schema

Schema 概要

描述数据结构

```
mongoose.schema({  
  name: string,  
  weight: num  
})
```


第三步，建立数据 Schema

Schema 概要

描述记录（文档）结构
规定

这个记录中有几个字段（ field ）
每个字段名字（ name, birthday ...）
已经字段类型（ string, num ）

第三步，了解 Model 模型的概念

model MVC 之中的 M

Model View Controller

模型 视图 控制器

model 通常会写成一个类
存放所有和数据直接相关的代码

第三步，了解 Model 模型的概念

```
var cat = mongoose.model('cat', catSchema);
```

第一个参数：模型对应的集合名称的单数

第二个参数：一条数据记录的结构

<http://mongoosejs.com/docs/models.html>

第四步，学会用 JS 代码来增删改查

CRUD

CRUD

Create

Read

Update

Delete

Create

```
var userSchema = mongoose.Schema({  
  name: String,  
  password: String,  
  age: String  
})
```

```
var peter = new user({ name: 'pppeter', password: '111', ag  
// 成功构建一条数据记录  
peter.save()
```

Read

```
user.find().exec(function(err, users) {  
  // 异步执行  
  console.log(users);  
});
```

update

```
user.findById({_id: '57ecbcd4d95146106254fe0'}, function(err, user) {  
  user.name = 'kkkkk'  
  user.save(function(err){  
    console.log('更新了! ')  
  });  
});
```


delete

```
User.findById({_id: '57ecbcd4d95146106254fe0'}, function(err, user) {  
  user.remove(function(err){  
    console.log('删除了! ')  
    User.find().exec(function(err, users) {  
      // 异步执行  
      console.log(users);  
    });  
  });  
});
```

路由

app.get('/')
app.post('/')

mongodb

remove()
find()
update
save()