构建 REST API

本节课程会以数据库中的 `posts` 集合为例，构建 REST API 来创建、读取、更新、删除（CRUD） `posts` 集合中的记录，将实现如下一系列 API:

```
GET /posts
POST /posts
PUT /posts/:post_id
GET /posts/:post_id
DELETE /posts/:post_id
```

打开 `routes.js` 文件，测试 API 可以删除了，删除代码:

```
app.get('/api', function(req, res) {
  res.json({message: 'get request!'})
});

app.post('/api', function(req, res) {
  console.log(req.body)
  res.json({message: 'post request!'})
});
```

获取所有文章

打开 `routes.js` 文件，文件内容如下：

```
var express = require('express');
var Post = require('./models/post');

module.exports = function(app) {
  app.get('/posts', function(req, res) {
    Post.find().sort({'createdAt': -1}).exec(
      if (err) return res.status(500).json({e
      res.json({ posts: posts })
    });
  });
}
```

注意：若想把错误信息发送给客户端，必须指定非 `2xx` 范围内的状态码，要不然客户端使用 `axios` 请求 API 的时候，捕捉不到该错误。

参考接口文档：

- res.status()
- Model#find()

- Query#sort()
- Query#exec()

创建新文章

打开 routes.js 文件，添加代码：

```
module.exports = function(app) {
  ...
  app.post('/posts', function(req, res) {
    if (req.body.title === '') return res.sta
    var post = new Post();
    for (prop in req.body) {
      post[prop] = req.body[prop];
    }
    post.save(function(err) {
      if (err) return res.status(500).json({e
      res.json({
        message: '文章创建成功了！'
      });
    });
  });
}
```

Model#save()

读取文章

打开 routes.js 文件，添加代码：

```
module.exports = function(app) {
  ...
  app.get('/posts/:post_id', function(req, re
```

```
    Post.findById({_id: req.params.post_id},
      if (err) return res.status(500).json({e
      res.json({ post: post })
    })
  });
}
```

Model#findById()

更新文章

打开 `routes.js` 文件，添加代码：

```
module.exports = function(app) {
  ...
  app.put('/posts/:post_id', function(req, re
    if (req.body.title === '') return res.sta
    Post.findById({_id: req.params.post_id},
      if (err) return res.status(500).json({e
      for (prop in req.body) {
        post[prop] = req.body[prop];
      }
      post.save(function(err) {
        if (err) return res.status(500).json(
        res.json({
          message: '文章更新成功了！'
        });
      });
    });
  });
}
```

删除文章

打开 routes.js 文件，添加一行代码：

```
module.exports = function(app) {
  ...
  app.delete('/posts/:post_id', function(req,
    Post.findById({_id: req.params.post_id},
      if (err) return res.status(500).json({e
      post.remove(function(err){
        if (err) return res.status(500).json(
        res.json({ message: '文章已经删除了！'
      });
    });
  });
}
```

Model#remove()

参考文档：

- HTTP 响应状态码
- Twitter API 响应状态码 （翻墙才能看哦）
- 如何设计 RESTful API
- 基于 Node 和 Express 4 构建 RESTful API