

全部课程 (/courses/) / Node.js+Redis实现漂流瓶 (/courses/450) / Node.js+Redis实现漂流瓶

在线实验，请到PC端体验

## 一、实验介绍

### 1.1 实验内容

本课程使用 Node.js+Redis 实现一个简单的漂流瓶应用，主要学习 Node.js 和 Redis 的使用和结合。项目中，我们使用redis ([https://github.com/NodeRedis/node\\_redis.git](https://github.com/NodeRedis/node_redis.git)) 这个 npm 包来操作 Redis 数据库。同时使用 Express 提供 Web 服务。

### 1.2 实验知识点

- Node.js 基础
- Redis

### 1.3 实验环境

- Node.js 6.x

### 1.4 适合人群

本课程难度一般，属于初级级别课程，适合具有 JavaScript 基础的用户，学习 Node.js 与 Redis 的结合使用。

## 二、开发准备

开发之前我们先来了解一下 Redis，Redis 是一个高性能Key-Value存储系统。Redis数据库存储在内存中（也可以存在硬盘中作数据持久化），所以速度非常快，适合处理系统缓存等。Redis还能实现很多实用的功能，比如消息队列、排行榜应用等。

Redis支持的数据类型主要有：string（字符串），hash（哈希表），list（列表），set（集合），sorted set（有序集合）。

打开虚拟机（Ubuntu）终端，输入如下命令启动Redis服务：

```
$ sudo service redis-server start
```

然后输入如下命令即可进入Redis命令行客户端，Redis默认运行在6379端口：

```
$ redis-cli
127.0.0.1:6379>
```

存储一个字符串，存储 “k => v” 这样一个键值对：

```
127.0.0.1:6379> set k v
OK
```

获取键“k”的值：

```
127.0.0.1:6379> get k
"v"
```

删除键值：

```
127.0.0.1:6379> del k
(integer) 1
```

这就是存储/读取/删除字符串的基本方法

动手实践是学习 IT 技术最有效的方式！

开始实验

存储/读取一个哈希表：

```
127.0.0.1:6379> hset user name shiyanlou
OK
127.0.0.1:6379> hget user name
"shiyanlou"
127.0.0.1:6379> hgetall user
1) "name"
2) "shiyanlou"
```

这里存储的是这样一个哈希表：

```
{
  "user": {
    "name": "shiyanlou"
  }
}
```

hget 命令每次只能存储一个值，比如上面的“name => shiyanlou”，我们可以用 hmset 命令一次设置多个值：

```
127.0.0.1:6379> hmset user2 name shiyanlou2 address chengdu
OK
127.0.0.1:6379> hget user2 name
"shiyanlou2"
127.0.0.1:6379> hmget user2 name address
1) "shiyanlou2"
2) "chengdu"
```

这里存储的是这样一个哈希表：

```
{
  "user2": {
    "name": "shiyanlou2",
    "addr": "chengdu"
  }
}
```

删除哈希表的值：

```
127.0.0.1:6379> hdel user name
(integer) 1
```

Redis不同于其他数据库，Redis不能自定义数据名称，默认情况下Redis会分配16个数据库，分别为0~15号数据库，默认情况下会进入0号数据库。切换数据库（进入1号数据库）：

```
127.0.0.1:6379> select 1
OK
127.0.0.1:6379[1]>
```

## 三、实验步骤

### 3.1 Node.js 操作 Redis

在这个项目中，我们使用Express作为web服务，创建Express App：

```
$ express drift
$ cd drift & npm install
```

在 drift 下创建 models 文件夹，然后在 models 文件夹中创建 redis.js 文件，关于操作Redis的代码就写在这个文件中。

我们使用redis ([https://github.com/NodeRedis/node\\_redis.git](https://github.com/NodeRedis/node_redis.git))这个包来操作Redis，首先安装包，在drift文件夹下执行如下命令：

```
$ npm install redis --save
```

redis 简单用法：

动手实践是学习 IT 技术最有效的方式！

开始实验

```
// 引入包
var redis = require('redis');

// 连接数据库
var client = redis.createClient();

// 选择数据库
// 默认会使用 0 号数据库
client.select(1, function() {
  // 存储哈希值
  client.hmset('user', { 'name': 'shiyancelou', 'addr': 'chengdu' }, function() {
    // 释放连接
    client.quit();
  });
});
```

我们还需要一个生成唯一ID的包：`node-uuid` (<https://github.com/broofa/node-uuid.git>)，因为我们需要给每个漂流瓶分配一个ID，安装包：

```
npm install node-uuid --save
```

写代码之前，我们先说说这个简单版的漂流瓶吧。我们要实现的漂流瓶包含的数据很简单，存储为哈希表，如下所示：

```
{
  "34fvdsf432": {
    "owner": "shiyancelou",
    "type": "male",
    "content": "hello world",
    "time": "1445659045635"
  },
  "jytfvdsf43": {
    "owner": "shiyancelou2",
    "type": "female",
    "content": "hello world haha",
    "time": "1445659047635"
  }
}
```

每个漂流瓶都有一个唯一ID，然后有一个扔瓶子的人（owner），类型（type）决定扔给男生还是女生，以及内容（content）和扔瓶子的时间（time）。

我们把扔给男生的瓶子，保存到Redis的0号数据库，把扔给女生的瓶子保存到Redis的1号数据库。打开 `redis.js`，输入如下代码（其实代码很少，注释很详细）：

```
'use strict';

// 引入包
var redis = require('redis');
var uuid = require('node-uuid');

// 瓶子类型
var type = { male: 0, female: 1 };

/**
 * 扔漂流瓶
 *
 * @param {obj} bottle 漂流瓶对象
 * @param {function} callback 回调函数
 */
function throwBottle(bottle, callback) {
  // 创建 redis 连接
  var client = redis.createClient();
  // 生成漂流瓶 id
  var bottleId = uuid.v4();
  // 漂流瓶的创建时间
  bottle.time = bottle.time || Date.now();

  // 根据漂流瓶类型选择数据库
  // male 类型漂流瓶保存到 0 号数据库
  // female 类型漂流瓶保存到 1 号数据库
  client.select(type[bottle.type], function() {
    // 使用 hash 类型保存漂流瓶对象
    client.hmset(bottleId, bottle, function(err, result) {
      // 扔瓶子失败
      if (err) {
        return callback({ code: 0, msg: '过会儿再试试吧~' });
      }
      // 设置漂流瓶生存期
      // 86400 秒，即一天
      // 若一天后没有人捡到这个瓶子
      // 就会被 Redis 删除
      client.expire(bottleId, 86400, function() {
        // 释放连接
        client.quit();
      });
      // 返回结果
      callback({ code: 1, msg: '你成功扔出了一个漂流瓶~' });
    });
  });
}

/**
 * 捡漂流瓶
 *
 * @param {obj} info 捡瓶子的用户对象
 * @param {function} callback 回调函数
 */
function pickBottle(info, callback) {
  // 创建 redis 连接
  var client = redis.createClient();

  // 根据用户性别选择数据库
  // male 选择 0 号数据库
  // female 选择 1 号数据库
  client.select(type[info.type], function() {
    // 从 redis 数据库中随机取出一个瓶子
    client.randomkey(function(err, bottleId) {
      // 出错了
      if (err) {
        return callback({ code: 0, msg: '你捞到一个海星~' });
      }
      // 没有取到时
      // 返回海星
      if (!bottleId) {
        return callback({ code: 1, msg: '你捞到一个海星~' });
      }

      // 取到了瓶子 动手实践是学习 IT 技术最有效的方式!
      client.hgetall(bottleId, function(err, bottle) {
```

开始实验

```
// 读取瓶子内容出错
if (err) {
    return callback({ code: 0, msg: '这个瓶子破损了。。。' });
}

// 读取瓶子内容成功
// 从 redis 数据库删除此瓶子
client.del(bottleId, function() {
    // 释放连接
    client.quit();
});

// 返回结果
callback({ code: 1, msg: bottle });
});
});
}

// 导出扔瓶子方法
exports.throw = function(bottle, callback) {
    throwBottle(bottle, function (result) {
        callback(result);
    });
};

// 导出捡瓶子方法
exports.pick = function(info, callback) {
    // 20% 概率捡到海星
    if (Math.random() <= 0.2) {
        return callback({ code: 1, msg: "你捞到一个海星~" });
    }
    pickBottle(info, function (result) {
        callback(result);
    });
};
};
```

## 3.2 实现扔瓶子和捞瓶子的URL路由

打开 drift/routes/ 文件夹下的 index.js 文件，添加两个简单的Express路由函数，代码如下：

```
var express = require('express');
var router = express.Router();
// 导入 redis.js
var redis = require('../models/redis');

/* GET home page. */
router.get('/', function(req, res, next) {
  res.render('index', { title: 'Express' });
});

// 扔瓶子
router.post('/throw', function(req, res, next) {
  // 获取 post 数据
  var body = req.body;
  var owner = body.owner;
  var type = body.type;
  var content = body.content;

  // owner type content 都必须填
  if (!(owner && type && content)) {
    return res.json({ code: 0, msg: '信息不完整' });
  }
  // 类型必须填正确
  if (type && ['male', 'female'].indexOf(type) === -1) {
    return res.json({ code: 0, msg: '类型错误' });
  }
  // 调用扔瓶子的方法
  // 返回 json 数据
  redis.throw(body, function(result) {
    res.json(result);
  });
});

// 捡瓶子
router.get('/pick', function(req, res, next) {
  // 获取 get 参数
  var user = req.query.user;
  var type = req.query.type;

  // 必须填捞瓶子的人的用户名
  if (!user) {
    return res.json({ code: 0, msg: '信息不完整' });
  }
  // 用户性别必须填正确
  if (!type || ['male', 'female'].indexOf(type) === -1) {
    return res.json({ code: 0, msg: '类型错误' });
  }
  // 调用捞瓶子的方法
  // 返回 json 数据
  redis.pick(req.query, function(result) {
    res.json(result);
  });
});

module.exports = router;
```

### 3.3 展示页面

功能都实现了，那我们来写个页面吧。为了快速搭建页面，我们使用Bootstrap这个CSS框架，在 drift 文件夹下运行一下命令，安装Bootstrap和jQuery：

```
$ npm install bootstrap jquery --save
```

然后复制jQuery和bootstrap到public文件夹下：

```
$ mkdir public/bootstrap
$ cp node_modules/bootstrap/dist/* public/bootstrap -r
$ cp node_modules/jquery/dist/jquery.min.js public/javascripts/
```

打开 drift/views/ 文件夹下的 layout.jade 文件，Express默认使用的jade模板，jade这里就不介绍了，可以去官网看看，使用方法很简单，输入如下代码：

动手实践是学习 IT 技术最有效的方式！

开始实验

```
doctype html
html
  head
    meta(charset='utf-8')
    meta(http-equiv='X-UA-Compatible' content='IE=edge')
    meta(name='viewport' content='width=device-width, initial-scale=1')
    title 漂流瓶
    link(rel='stylesheet', href='/bootstrap/css/bootstrap.min.css')
  body
    block content
      script(src='/javascripts/jquery.min.js')
      script(src='/bootstrap/js/bootstrap.min.js')
    block js
```

在 layout.jade 文件中引入了bootstrap和jQuery然后打开 index.jade 文件，输入如下代码：

```

extends layout

block content
  .container
    h1 漂流瓶
    br
    br
    .btn-group(role='group')
      button.btn.btn-primary.pick(type='button') 捞一下
      button.btn.btn-success.throw(type='button') 扔一个
    br
    br
    .show-msg
    .show-bottle.alert.alert-success
      div 捞到的瓶子:
        br
        .bottle-user 用户:
          span
        .bottle-content 内容:
          span
    #pick-modal.modal.fade(role='dialog')
      .modal-dialog(role='document')
        .modal-content
          .modal-header
            h4.modal-title 填写个人信息
          .modal-body
            form.form-horizontal
              .form-group
                label.col-md-2.control-label 用户名
                .col-md-10
                  input.form-control(name='user' type='text')
              .form-group
                label.col-md-2.control-label 性别
                .col-md-10
                  label.radio-inline
                    input(type='radio' name='type' value='male')
                    span 男
                  label.radio-inline
                    input(type='radio' name='type' value='female')
                    span 女
          .modal-footer
            button.btn.btn-default(type='button' data-dismiss='modal') 取消
            button.btn.btn-primary(type='button') 确定
    #throw-modal.modal.fade(role='dialog')
      .modal-dialog(role='document')
        .modal-content
          .modal-header
            h4.modal-title 填写瓶子信息
          .modal-body
            form.form-horizontal
              .form-group
                label.col-md-2.control-label 用户名
                .col-md-10
                  input.form-control(name='owner' type='text')
              .form-group
                label.col-md-2.control-label 类型
                .col-md-10
                  label.radio-inline
                    input(type='radio' name='type' value='male')
                    span 扔给男生
                  label.radio-inline
                    input(type='radio' name='type' value='female')
                    span 扔给女生
              .form-group
                label.col-md-2.control-label 内容
                .col-md-10
                  input.form-control(name='content' type='text')
          .modal-footer
            button.btn.btn-default(type='button' data-dismiss='modal') 取消
            button.btn.btn-primary(type='button') 确定
  block js
    script(src='/javascripts/script.js')

```

在这个文件中，编写了扔瓶子和捞瓶子的按钮，以及两个modal，一个用于显示扔瓶子的表单，一个用于显示捞瓶子的表单，下面编写在文件末尾引入的 script.js JavaScript文件。

在 drift/public/javascripts/ 文件夹下新建 script.js 文件，输入如下代码：

动手实践是学习 IT 技术最有效的方式！

开始实验



```

'use strict';

// 捞瓶子
$('.pick').on('click', function() {
  var modal = $('#pick-modal');

  modal.modal('show');
  $('.btn-primary', modal).off('click').on('click', function() {
    var user = $('form', modal).serialize();

    $.getJSON('/pick', user, function(data) {
      if (typeof data.msg === 'object') {
        // 捞到了一个瓶子
        // 显示瓶子信息
        $('.bottle-user span').text(data.msg.owner);
        $('.bottle-content span').text(data.msg.content);
      } else {
        // 没有捞到瓶子
        // 显示提示信息
        showMsg(data);
      }
      modal.modal('hide');
    });
  });
});

// 扔瓶子
$('.throw').on('click', function() {
  var modal = $('#throw-modal');

  modal.modal('show');
  $('.btn-primary', modal).off('click').on('click', function() {
    var bottle = $('form', modal).serialize();

    $.post('/throw', bottle, function(data) {
      // 显示提示信息
      showMsg(data);
      modal.modal('hide');
    });
  });
});

/**
 * 显示提示信息
 *
 * @param {object} data 数据对象
 */
function showMsg(data) {
  // 只显示简单信息
  if (typeof data.msg === 'object') {
    return false;
  }

  var dataClass = data.code === 0 ? 'danger' : 'success';
  var box = $('<div class="alert alert-' + dataClass + '" role="alert"></div>');

  box.text(data.msg);
  $('.show-msg').html(box);

  // 5 秒后信息自动消失
  setTimeout(function() {
    $('.show-msg').html('');
  }, 5000);
}

```

### 3.4 运行项目

在 drift 文件夹下运行一下命令启动项目：

```
$ npm start
```

打开浏览器，访问：<http://localhost:3000/>，点击按钮扔瓶子，然后可以捞瓶子。

开始实验

## 四、总结

一个简单的漂流瓶就实现完成了，由于没有注册/登录用户功能，所以每次扔瓶子或者捞瓶子都需要填用户名。小伙伴们可以自行扩展更丰富的功能。

可以将作业写入实验报告并发布，与本课程的同学及老师交流讨论。

### 课程教师



**forever**  
共发布过8门课程

[查看老师的所有课程 > \(/teacher/45\)](#)

### 前置课程

[Redis基础教程 \(/courses/106\)](#)

[Node.js 教程 \(/courses/44\)](#)

### 进阶课程

[Node.js 经典项目实战 \(/courses/455\)](#)



## 动手做实验，轻松学IT



公司 <http://weibo.com/shiyanlou2013>

[关于我们 \(/aboutus\)](#)  
[联系我们 \(/contact\)](#)  
[加入我们 \(http://www.simplecloud.cn/jobs.html\)](http://www.simplecloud.cn/jobs.html)  
[技术博客 \(https://blog.shiyanlou.com\)](https://blog.shiyanlou.com)

### 服务

[企业版 \(/saas\)](#)  
[实战训练营 \(/bootcamp/\)](#)  
[会员服务 \(/vip\)](#)  
[实验报告 \(/courses/reports\)](#)  
[常见问题 \(/questions/?tag=%E5%B8%B8%E8%A7%81%E9%97%AE%E9%A2%98\)](#)  
[隐私条款 \(/privacy\)](#)

### 合作

[我要投稿 \(/contribute\)](#)  
[教师合作 \(/labs\)](#)  
[高校合作 \(/edu/\)](#)  
[友情链接 \(/friends\)](#)  
[开发者 \(/developer\)](#)

### 学习路径

[Python学习路径 \(/paths/python\)](#)  
[Linux学习路径 \(/paths/linuxdev\)](#)  
[大数据学习路径 \(/paths/bigdata\)](#)  
[Java学习路径 \(/paths/java\)](#)  
[PHP学习路径 \(/paths/php\)](#)  
[全部 \(/paths/\)](#)