

高级 Web 技术 2016 春
期末项目文档

YourMap

出品 —— 高

秦海峰 13302010079
陈晨光 13302010033
汤定一 13302010077
雷丽暇 13302010104
仝嘉文 13302010085

目录

1. 项目设计

1.1 设计框架

1.2 关键功能点

2. 团队分工

2.1 分工安排

2.2 过程管理

1. 项目设计

1.1 设计框架

前端使用 AngularJS 的 MVC 框架部署，完成 View 和 Controller 分离，并在测试前端时使用 Model 管理测试样例数据。Controller 管理前端页面跳转逻辑、与后台交互的请求逻辑等业务逻辑，View 管理各个页面渲染。

前后端业务分离，后台使用 Spring-Struts-Hibernate 框架部署，其中 Spring 负责集成管理，在 applicationContext.xml 中使用 java bean 封装数据库 driver、数据库 Dao、各个 action handler 以及 service 层的对象，提供了控制反转的条件；Struts 管理 action，负责接收和处理前台请求 url，在 struts.xml 中把 url 和对应的处理类挂钩起来；Hibernate 提供了 EntityDAO 对数据库的通用访问接口，添加 Service 层，根据业务需求设计数据库访问和存储的调用 EntityDAO 的代码。

1.2 关键功能点

AngularJS 使用百度地图 API

创建地图实例之后，

```
var map = new BMap.Map("container");
```

在前端 AngularJS 代码中调用百度地图的 JavaScript 开发 API，使用定位控件、平移缩放控

件、比例尺控件和自定义控件等地图控件，与地图 UI 交互、实现相关的业务需求。

定位控件：

```
var opts3 = {offset: new BMap.Size(5, 1150), enableAutoLocation: true}  
//定位控件位置  
var geolocationControl = new BMap.GeolocationControl(opts3); // 设置定位  
控件 start
```

平移缩放控件：

```
var opts1 = {offset: new BMap.Size(10, 1120)}  
map.addControl(new BMap.NavigationControl(opts1));
```

比例尺控件：

```
var opts2 = {offset: new BMap.Size(10, 1120)}  
var scaleControl = new BMap.ScaleControl(opts2);
```

自定义控件：

```
// 定义一个控件类,即 function  
function lookSightDetail() {  
    this.defaultAnchor = BMAP_ANCHOR_TOP_LEFT;  
    this.defaultOffset = new BMap.Size(50, 20);  
}  
// 创建控件  
var myDetailCtrl = new lookSightDetail();  
// 添加到地图当中  
map.addControl(myDetailCtrl);
```

使用覆盖物 Overlay 接口标识指定区域，并使用标注 Marker 操作百度地图的景点，

```
var point = new BMap.Point(121.48, 31.22);  
var marker = new BMap.Marker(point);  
var infoWindow = new BMap.InfoWindow(sContent);  
map.addOverlay(marker);
```

用户点评数据的上传和发布

相比传统的 Web1.0 中，网络数据由官方的机构作为服务端发布，普通用户只能作为客户端访问数据，Web2.0 是一个连接了众多普通用户创建的内容而构建的网络。Web2.0 支持用户上传数据，本应用支持用户评价，对景点打分，上传图片、视频和模型等不同类型的文件，前端界面收集用户发布的这些数据，使用 AngularJS 的 \$http 对象将数据发送给后台，后台根据请求 url，查找 Struts.xml 定位到 action handler 类，接收数据，并存储文件到云端服务器。当用户在前端请求数据时，后端以 JSON 的形式将数据返回给前端，前台 Controller 接收数据，Html View 将数据渲染成页面效果。

使用 Three.js 的 3D 景观展示

本应用中对于复旦大学和世博园中国馆添加了景观 3D 模型，3D 模型的搭建包括材质、贴图、光线方面的处理。首先在 maya 中建模，然后用 github 上 three.js 项目中的 exporter 把 maya 中建好的中国馆模型导出成 json 文件，因为 maya 中的材质与 three.js 中的材质不

通用，要在 three.js 中通过代码设置材质。在 three.js 中用 THREE.JSONLoader 导入 JSON 文件。中国馆的主体部分采用 Phong 材质可以体现金属的光泽，玻璃部分采用 Lambert 材质并设置为透明。光线采用平行光颜色白色。相机移动方式为 Orbit，可以用左键控制镜头角度，右键控制相机位置，中键控制镜头远近。maya 与 three.js 的三维空间不同，需要把从 maya 导入的模型沿 y 轴旋转-90 度，沿 x 轴旋转 90 度。

基于内容的推荐算法

为了给用户推荐他喜欢的景点，本应用使用了基于内容的推荐算法。n 个景观，m 个用户，建立一个景观-用户的 $n \times m$ 评分矩阵，其中每一个入口就是用户对该景点的“喜欢程度”，包括足迹、心愿单和收藏。评分矩阵为 sightMatrix：

```
//所有景点的特征集合矩阵，所有入口初始化为 0
```

```
sightMatrix = new int[sSize][uSize];
```

如果用户对该景点添加了足迹，则在该入口加 1，添加了心愿单和收藏的景点，则在该入口加 2，不同类型的“喜欢程度”对应的具体分数值是我们组自己定的，可以修改。完成了每个入口的计算之后，该矩阵作为 item-based 的推荐基础，每个景观的特征向量为该评分矩阵中该景观的向量。对用户返回的推荐结果为，该用户没有足迹的景观中，与该用户已有足迹的景观最类似的前几个景观，返回结果最多不超过 3 个。景观之间的相似程度由景观特征向量的余弦夹角值表示。具体代码可以详见后端 util 包中的 Recommend.java。

```
public double computeSimilarity(int a, int b) {
    double similarity = -1;
    int[] v1 = sightMatrix[a];
    int[] v2 = sightMatrix[b];
    int x = 0;
    int v1LenSqr = 0;
    int v2LenSqr = 0;
    for (int i=0; i<uSize; i++) {
        x += v1[i]*v2[i];
        v1LenSqr += v1[i]*v1[i];
        v2LenSqr += v2[i]*v2[i];
    }
    similarity = x / (Math.sqrt(v1LenSqr) * Math.sqrt(v2LenSqr));
    return similarity;
}
```

基于 GitHub 的第三方登录

具体步骤为

- (1) 创建访问第三方应用（GitHub）登陆页面的入口
- (2) 用户在第三方应用上登陆完成后，第三方应用返回 code，
- (3) 本应用再给第三方发送带有 code 的登陆请求
- (4) code 验证成功后，第三方应用返回 access token
- (5) 本应用将包含 access token 的登陆请求发送给第三方
- (6) access token 验证成功后，第三方应用返回登陆用户的相关数据，并跳回本应用指定

的页面

2. 团队分工

2.1 分工安排

百度地图 API 使用、景观相关页面前台逻辑、搜索历史实现 ----- 秦海峰

用户管理相关页面前台逻辑、交互设计、界面优化调整 ----- 雷丽暇

Service 层访问数据库接口设计与实现、Action Handler 设计与实现、第三方登录实现 ----
-- 陈晨光

数据收集、数据库管理、高级推荐算法调研和实现、文档编写、测试 ----- 仝嘉文
第三方登录调研、3D 模型开发部署 ----- 汤定一

2.2 过程管理

PJ 布置-5月中旬 ----- 需求理解、前后端接口定义、前端页面设计

5月中旬-5月下旬 ----- 数据库搭建、百度地图 API 调用

5月中旬-6月中旬 ----- 具体的各个前后端交互功能实现

6月中旬-6月下旬 ----- 推荐算法实现、3D 模型搭建、景观素材库搭建

6月29日凌晨 ----- 界面优化调整、测试与调整、文档编写