



链滴

java 递归树

作者: [liangliang](#)

原文链接: <https://ld246.com/article/1546926351824>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

环境

使用 springboot + [MyBatis-Plus](#) (简化mybatis开发) + lombok (注解getter、setter)

实体类Dto

```
@Data
@TableName("xxx表名")
public class Organize extends Model<Organize> {

    @TableId(type = IdType.AUTO)
    private String id;

    private String code;

    private String name;

    private String uporgId;

    @TableField(exist = false)
    private List<Organize> child;

    /** 指定主键 */
    @Override
    protected Serializable pkVal() {
        return this.id;
    }
}
```

controller 省略

service实现

```
@Service
public class OrganizeServiceImpl implements OrganizeService {

    @Autowired
    @SuppressWarnings("all")
    private OrganizeMapper organizeMapper;

    @Override
    public List<Organize> tree(Organize organize) {
        long startTime = System.currentTimeMillis();

        // 查出全部 获取根节点 list
        List<Organize> organizes = organizeMapper.selectList(null);
        List<Organize> childList = new ArrayList<Organize>();
    }
}
```

```

for (Organize o : organizes) {
    if (StringUtils.isNullOrEmpty(o.getUporgId())) {
        childList.add(o);
    }
}

// 开始递归
List<Organize> resultTree = setChildren(childList, organizes);

long endTime = System.currentTimeMillis();
//输出程序运行时间
System.out.println("程序运行时间: " + (endTime - startTime) + "ms");
return resultTree;
}

public List<Organize> setChildren(List<Organize> resultTree, List<Organize> allList) {

    for (int i = 0; i < resultTree.size(); i++) {
        Organize item = resultTree.get(i);

        // 从全部数据中查询上级节点为当前节点id 的数据 作为当前节点的子节点
        List<Organize> resultChildren = new ArrayList<Organize>();
        for (Organize o : allList) {
            if (null != o.getUporgId() && o.getUporgId().equals(item.getId())) {
                resultChildren.add(o);
            }
        }

        if (!resultChildren.isEmpty()) {
            // 递归
            resultChildren = setChildren(resultChildren, allList);
            // 设置子集
            item.setChild(resultChildren);
            resultTree.set(i, item);
        }
    }
    return resultTree;
}
}

```