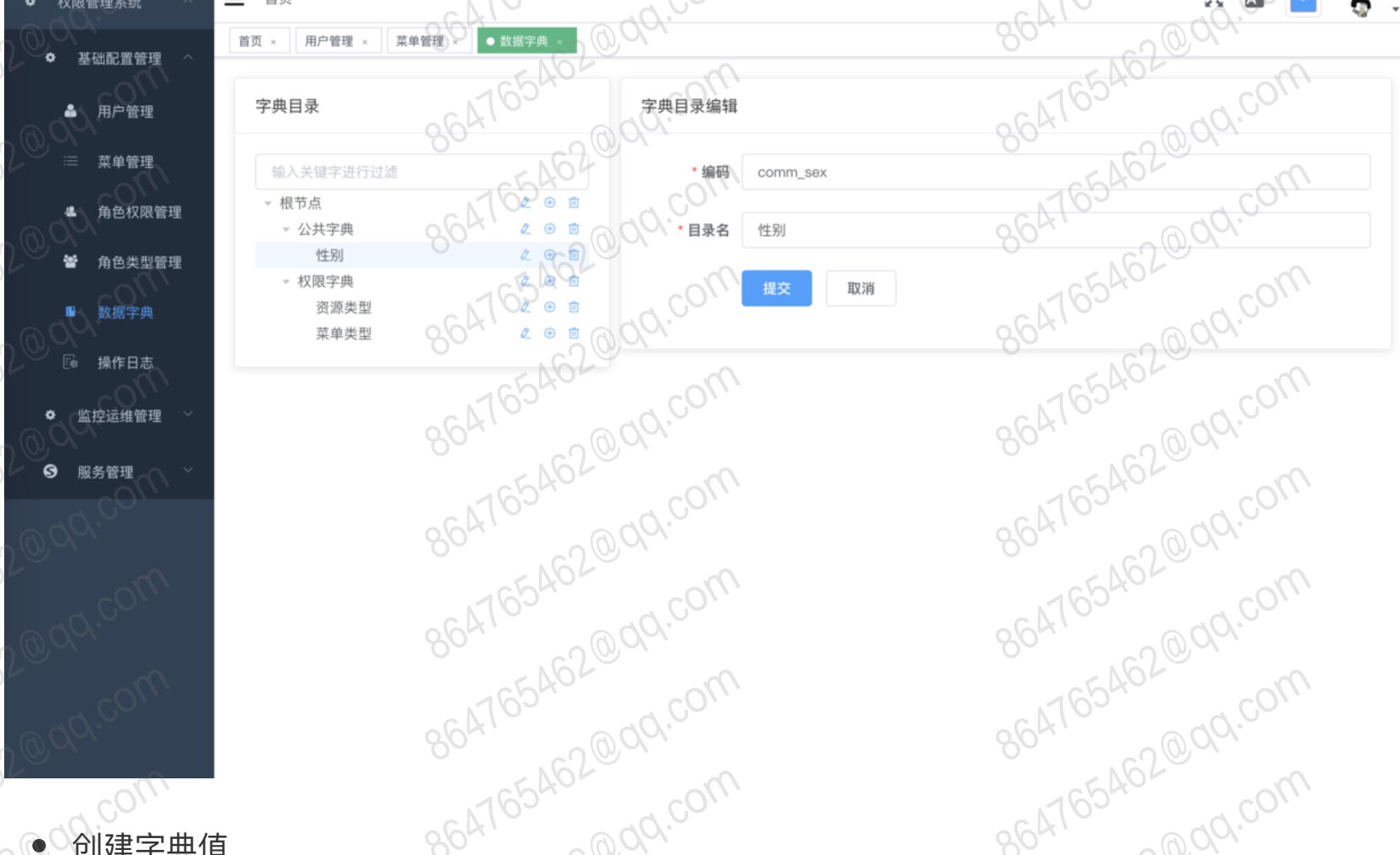


创建字典

- 创建目录 (comm_sex，数据聚合或前端显示要用到)



- 创建字典值



前端页面怎么引用数据字典

- 导入依赖

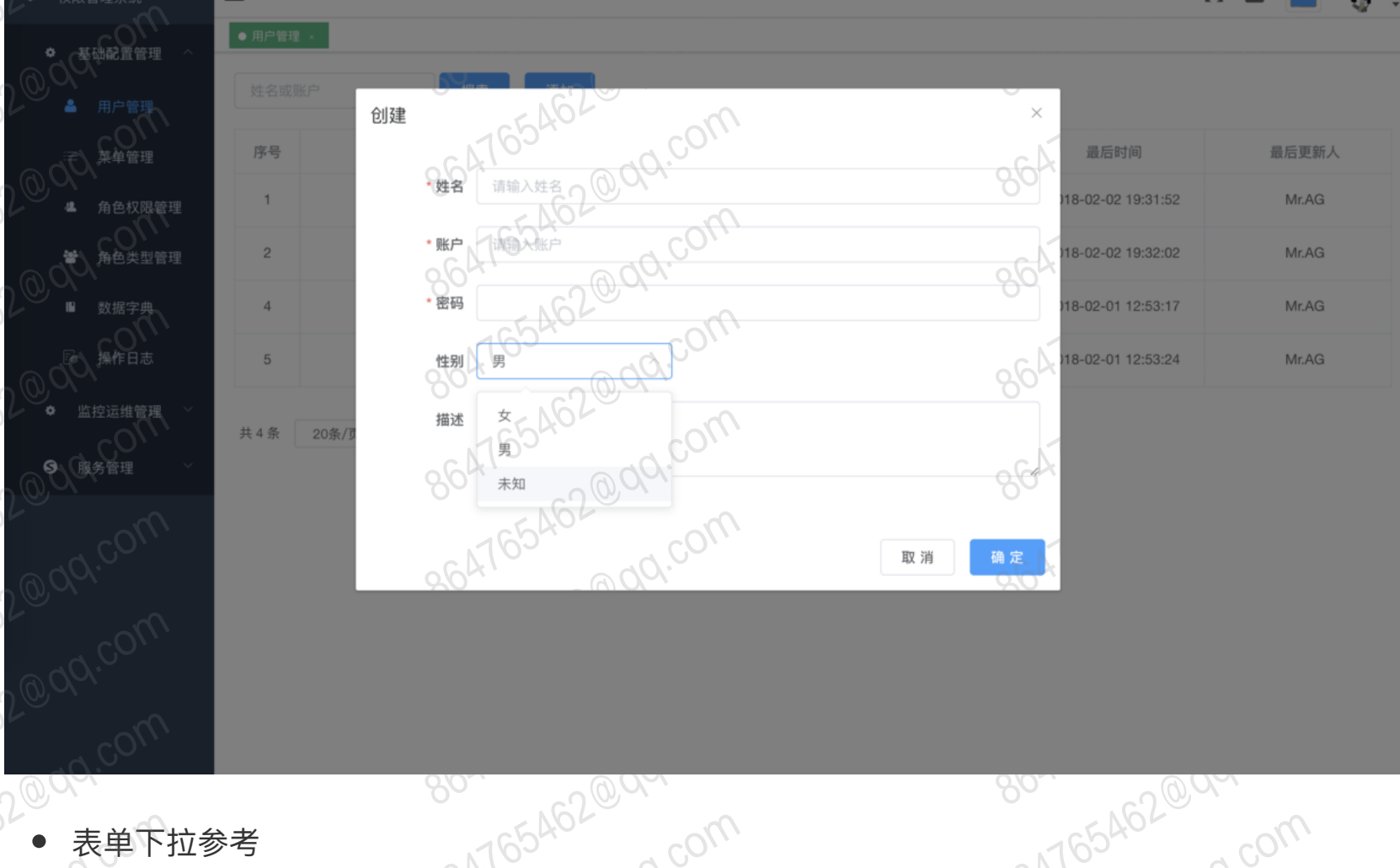
```
import { getTypeValue } from '@api/dict/dictValue/index';
```

- 获取字典

```
// comm_sex为字典目录值
getTypeValue('comm_sex').then(response => {
  this.sexOptions = response.data.rows;
});
```

用户模块性别示例(前端代码路

劲：/src/views/admin/user/index.vue)



- 表单下拉参考

```
<el-form-item label="性别">
  <el-select class="filter-item" v-model="form.sex" placeholder="请选择">
    <el-option v-for="item in sexOptions" :key="item.id"
      :label="item.labelDefault" :value="item.value"> </el-option>
  </el-select>
</el-form-item>
```

- 属性配置

```
sexOptions: [],
dialogFormVisible: false,
dialogStatus: '',
userManager_btn_edit: false,
userManager_btn_del: false,
userManager_btn_add: false,
textMap: {
  update: '编辑',
  create: '创建'
},
tableKey: 0
},
created() {
  this.getList();
  this.userManager_btn_edit = this.elements['userManager:btn_edit'];
  this.userManager_btn_del = this.elements['userManager:btn_del'];
  this.userManager_btn_add = this.elements['userManager:btn_add'];
  getTypeValue('comm_sex').then(response => {
    this.sexOptions = response.data.rows;
  });
}
```

后端数据聚合

前提说明：该框架主要聚合的数据类型必须为List，可以解决大多数分页数据属性展示问题

- 引入数据聚合框架(代码地址：https://gitee.com/geek_qi/AG-Merge)

```
@EnableAceMerge
```

- 加入聚合配置

```
# 跨服务数据合并
merge:
  enabled: true
  guavaCacheNumMaxSize: 1000 # 缓存总个数
  guavaCacheRefreshWriteTime: 10 # 缓存自动刷新时间，单位分钟
  guavaCacheRefreshThreadPoolSize: 10 # 缓存异步刷新线程数大小
  aop:
    enabled: true # 是否开启自动切面数据聚合
```

- 配置feign调用

特别说明：聚合调用的返回结果一定为Map<String,String>，且参数只有一个String。

```
@Service
public class DictService {

  @Autowired
  private DictValueBiz dictValueBiz;

  public Map<String,String> getDictValues(String code){
    Example example = new Example(DictValue.class);
    example.createCriteria().andLike("code",code+"%");
    List<DictValue> dictValues =
      this.dictValueBiz.selectByExample(example);
    Map<String, String> result = dictValues.stream().collect(
      Collectors.toMap(DictValue::getValue,
        DictValue::getLabelDefault));
    return result;
  }
}
```

- 配置实体关联

```
@Table(name = "base_user")
@AceAudit
public class User {

  @Id
  @GeneratedValue(generator = "UUID")
  private String id;

  private String username;
  // ..... 省略其他属性
  // 加入聚合属性，配置上边的feign调用
  @MergeField(key = "comm_sex", feign = DictService.class, method =
    "getDictValues")
  private String sex;
}
```

- 聚合两种方式

- 手动聚合（参考：UserBiz）

```
@Autowired
private MergeCore mergeCore;

@Override
public List<User> selectByExample(Object example) {
  List<User> users = super.selectByExample(example);
  try {
    mergeCore.mergeResult(User.class,users);
  } catch (Exception e) {
    e.printStackTrace();
  }finally {
    return users;
  }
}
```

- aop自动聚合

```
// 加入MergeResult即可
@MergeResult
@Override
public List<Element> selectByExample(Object example) {
  return super.selectByExample(example);
}
```