使用案例

前端

前端通用导出功能示例详情可以参看[前端通用导出说明文档]

后端

所有的导出都是统一的controller入口，具体区分是哪种业务导出，通过传参确定。

通过bex导出数据（数据通过调用bex获取）

- 通过bex导出数据：通过apiParam的api参数区分不同业务，机构导出对应的api为queryOrgInfoList，人员导出对应的api为queryUserInfoList。
- 需导出数据通过传参columns列信息进行匹配，将匹配成功的导出。

发送请求POST /file/export

请求参数：

```
{
    "fileType": "pdf",
    "fileName": "sampleFile",
    "columns": [
        {
            "prop": "orgCode",
            "label": "机构代码",
            "width": "0"
        },
        {
            "prop": "orgName",
            "label": "机构名称",
            "width": "0"
        },
        {
            "prop": "userNo",
            "label": "用户编号",
            "width": "0"
        },
        {
            "prop": "userName",
            "label": "用户名",
            "width": "0"
        },
        {
            "prop": "userStatus",
            "label": "用户状态",
            "width": "0"
        },
        {
            "prop": "idType",
            "label": "证件类型",
```

```
            "width": "0",
            "transType": "dict",
            "transParam": {
                "dictCode": "ID_TYPE",
                "valueKey": "dictItem",
                "labelKey": "dictItem,dictItemName"
            }
        },
        {
            "prop": "idCode",
            "label": "证件编码",
            "width": "0"
        }
    ],
    "apiParam": {
        "api": "queryUserInfoList",
        "queryParams": {
            "sex": "0"
        },
        "pageParam": {
            "pageNum": "1",
            "pageSize": "10"
        }
    },
    "reportParam": {
        "paper": {
            "size": "A4",
            "direction": "horizontal",
            "align": "center",
            "marginLeft": "20",
            "marginRight": "20",
          "marginTop": "20",
            "marginBottom": "20"
        }
    }
}
```

注意：参数apiParam.api 与 bex 的 api 对应.

返回文件流

需导出数据通过传参 进行导出

发送请求 POST /file/export

请求参数:

```
{
    "fileType": "xlsx",
    "fileName": "sampleFile",
    "columns": [
        {
            "prop": "userName",
            "label": "用户名",
            "width": "0"
        },
        {
            "prop": "sex",
```

```
            "label": "性别",
            "width": "0",
            "dict": "sex"
        }
    ],
    "data": [
        [
            "json",
            "man"
        ],
        [
            "jack",
            "man"
        ],
        [
            "marry",
            "female"
        ]
    ]
}
```

excel模板导出（目前只支持excel格式模板）

发送请求 POST /file/export

请求参数：

```
{
    "fileType":"xlsx",
    "dataConfigId":"userInfoList",
    "apiParam": {
        "api": "queryUserInfoList",
        "queryParams": {
            "sex": "0"
        },
        "pageParam": {
            "pageNum": "1",
            "pageSize": "10"
        }
    }
}
```

此功能根据请求参数dataConfigId寻找对应xml配置，从配置信息中读取模板excel，将数据写入后导出

xml配置需放在项目src/main/resources 文件夹底下，xml后缀为-export.xml

xml配置示例如下：

```
<?xml version="1.0" encoding="UTF-8"?>
<data>
    <!-- 导出配置配置 -->
    <export id="userInfoList" name="用户信息列表" templateFile="test.xlsx">
        <column property="userName" name="用户名" order="1"/>
        <column property="sex" name="性别" order="2"/>
        <column property="salary" name="薪水" order="3"/>
        <column property="address" name="住址" order="4"/>
```

```
        <column property="postNo" name="邮编" order="5"/>
        <column property="idCard" name="身份证" order="6"/>
        <column property="mobile" name="手机号" order="7"/>
        <column property="fax" name="传真" order="8"/>
        <column property="uid" name="uid" order="9"/>
    </export>
</data>
```

配置说明：

配置项	是否必填	配置说明
id	Y	xml配置id 唯一标志
name	Y	导出的文件名，替换前端参数fileName
templateFile	Y	使用的模板文件名（文件带后缀）
column	Y	对应的导出文件的字段名称（property为实际字段名，name为模板表头名，order为改字段在模板中的表头顺序）

excel模板文件存放路径可通过koca.export.templateLocation配置

excel模板文件支持配置通配符路径：默认classpath*:templates/**，表示resources底下tempates文件夹及其路径下

华表导出

发送请求 POST /file/export

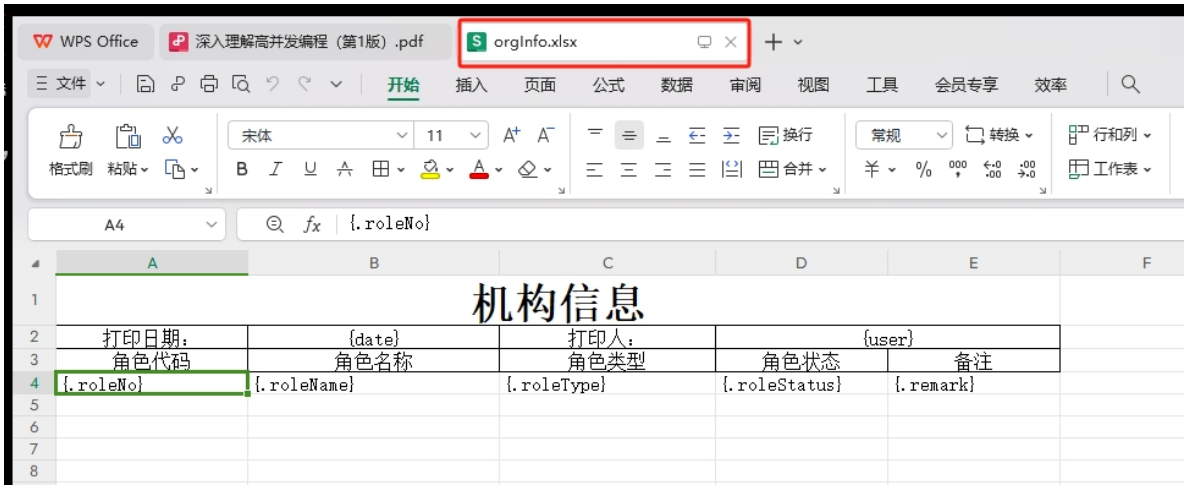
请求参数：

```
{
    "fileType": "xlsx",
    "fileName": "sampleFile",
    "columns": [
        {
            "prop": "roleNo",
            "label": "角色代码",
            "width": "0"
        },
        {
            "prop": "roleName",
            "label": "角色名称",
            "width": "0"
        },
        {
            "prop": "roleType",
            "label": "角色类型",
            "width": "0"
        },
        {
            "prop": "roleStatus",
            "label": "角色状态",
```

```
            "width": "0"
        },
        {
            "prop": "remark",
            "label": "备注",
            "width": "0"
        }
    ],
    "apiParam": {
        "api": "role-query-page",
        "queryParam": {
            "sex": "0"
        },
        "pageParam": {
            "pageNum": "1",
            "pageSize": "10"
        }
    },
    "reportParam": {
        "paper": {
            "size": "A4",
            "direction": "vertical",
            "align": "center",
            "marginLeft": "20",
            "marginRight": "20",
            "marginTop": "20",
            "marginBottom": "20"
        }
    },
        "templateParam":{
            "templateId":"orgInfo"
        }
    }
}
```

templateId就是华表的模板名称,模板里的变量对应入参的columns列。模板规则：

模板注意 用{} 来表示你要用的变量 如果本来就有"{","}" 特殊字符 用"{","}"代替
{} 代表普通变量 {.} 代表是list的变量



数据翻译

以数据字典证件类型为例：

```
{
            "prop": "idType",
            "label": "证件类型",
            "width": "0",
            "transType": "dict",
            "transParam": {
                "dictCode": "ID_TYPE",
                "valueKey": "dictItem",
                "labelKey": "dictItem,dictItemName"
            }
        }
```

翻译类型：**transType**（data,dict,service,request,cache）可以扩展类型为other1,other2,other3,other4

翻译参数：transParam

字典代码（dictCode），数据值主键（valueKey），数据文本主键（labelKey支持翻译多个key的值，比如 'key1,key2,key3'），数据文本分隔符（separatorlabelKey存在多个key时生效）等。

翻译其他类型参数都不一样，依据前端说明实现。

组件默认实现koca数据字典，实现代码如下：

```
/**
 * 数据字典翻译实现
 */
public class DictTransProvider implements TransProvider {

    @Override
    public List<Map<String, Object>> getTransInfos(List<Map<String, Object>>
dataList, String columnName,
        Object transParam) {
        //TODO 实现批量翻译
    }
     @Override
    public String getTransType() {
        return TRANS_TYPE_DICT;
    }
```

业务部门的数据翻译根据自己的需要自行实现TransProvider接口

getTransType()方法返回的类型：

```
/**
     * data.
     */
    String TRANS_TYPE_DATA = "data";

    /**
     * dict.
     */
    String TRANS_TYPE_DICT = "dict";

    /**
     * service.
     */
```

```
    String TRANS_TYPE_SERVICE = "service";

    /**
     * request.
     */
    String TRANS_TYPE_REQUEST = "request";

    /**
     * cache.
     */
    String TRANS_TYPE_CACHE = "cache";

    /**
     * other1.
     */
    String TRANS_TYPE_OTHER1 = "other1";

    /**
     * other2.
     */
    String TRANS_TYPE_OTHER2 = "other2";

    /**
     * other3.
     */
    String TRANS_TYPE_OTHER3 = "other3";

    /**
     * other4.
     */
    String TRANS_TYPE_OTHER4 = "other4";
```

文件导出类型扩展实现

需要实现ExportHandler接口，比如要实现txt类型导出，实现结构如下：

```
/**
 * txt导出
 */
@Component
public class CsvExportHandler implements ExportHandler {

    @Override
    public File exportFile(String fileName, List<Column> columnList, Paper
paper, List<Map<String, Object>> data,
        TemplateParam templateParam) {
        //TODO 导出txt逻辑
        }

    @Override
    public String getType() {
        return FILE_TYPE_TXT;
    }
}
```

业务日志接口

业务导出需要日志留痕，自行实现此接口

```
/**
 * 业务日志实现
 *
 * @author xiaoht
 * @since V4.1.0 2023-2-9
 */
public class BizExportLogImpl implements BizExportLog {

    @Override
    public void addLogInfo(BizLogInfo bizLogInfo) {
        //TODO 写入业务日志
    }
}
```

水印签章接口

pdf导出需要水印，签章的自行实现接口

```
/**
 * 水印签章配置实现
 *
 * @author xiaoht
 * @since V4.1.0 2023-2-22
 */
public class WaterMarkSignatureImpl implements WaterMarkSignature {

    @Override
    public String getWaterMark() {
        return "xiaoht";
    }

    @Override
    public String getSignature() {
        return "";
    }

    @Override
    public String getLister(ExportRequestParam requestParam) {
        return "xiaoht";
    }
}
```

导出excel生成下拉框

通过如下配置，进行下拉框选项配置：

```
koca:
  export:
    xlsx:
      dropdownboxinfos: # 设置下拉框
       - columnPropertyName: sex # 下拉框对应的列名
         values: # 下拉框值
            - male
            - female
```