

统一组件使用说明

目前统一组件全部存放于 `common` 模块 `com.ceair.*.base` 包下

以下示例以【shopping-service】项目为例演示

关于枚举

核心类：

- `BaseEnum` - 枚举的顶层接口
- `EnumDictionaryManager` - 枚举字典管理容器

使用要求

1. 所有的枚举类需要实现 `BaseEnum` 接口
2. 如果当前枚举数据需要传给前端（如：下拉列表的数据）时：
 1. 通过定义 `EnumDictionaryManager` Bean 来开启此功能

```
@Configuration
public class EnumDictionaryManagerConfig {

    @Bean
    public EnumDictionaryManager enumDictionaryManager() {
        return new EnumDictionaryManager();
    }
}
```

2. 通过参数 `enum.dictionary.manager.scanner.package` 配置枚举字典扫描的范围

```
-
Denum.dictionary.manager.scanner.package=com.ceair.shopping-service.*.enums
```

3. 枚举类通过实现 `BaseEnum.isLoadDictionary()` 方法来决定是否需要被 `EnumDictionaryManager` 加载

关于响应类

核心类

- `ResultWrapper`

使用要求

1. 所有 `Controller` 的出参均为 `ResultWrapper` 的包装类，并且需要明确指定泛型数据的类型，如：`ResultWrapper<String>`、`ResultWrapper<Long>`
2. `ResultWrapper` 类中提供类常用的静态方法，如：
 1. `ResultWrapper.success(T data)`
 2. `ResultWrapper.failure(String message)`
 3. `ResultWrapper.failure(BaseEnum failureEnum)`
3. 常规业务判断是否成功通过调用 `ResultWrapper.isSuccess()` 方法
4. 更安全的调用方式可通过 `ResponseUtils` 类完成，提供静态方法
 1. `ResponseUtils.process(ResultWrapper<T> result)` - 获取响应数据，如果 `ResultWrapper` 为空或者 `!ResultWrapper.isSuccess()` 会抛出异常，否则返回响应数据
 2. `ResponseUtils.getData(ResultWrapper<T> result)` - 获取响应数据
 3. `ResponseUtils.hasData(ResultWrapper<T> result)` - 判断数据是否为空
 4. `ResponseUtils.isSuccess(ResultWrapper<T> result)` - 判断是否成功

关于异常

核心类

1. `BaseBusinessException` - 业务异常
2. `BaseSystemException`
3. `DefaultGlobalExceptionHandler` - 全局异常处理

使用要求

1. 常规业务无需单独 `try catch` 处理异常，统一交给全局异常处理器进行处理
2. 所有抛出的异常都需要定义消息枚举，可在 `BusinessErrorEnum` 中定义，也可自行实现 `BaseEnum` 去定义
3. 通过继承 `DefaultGlobalExceptionHandler` 开启全局异常处理，自定义的异常处理可在该类中定义，如：

```
@ControllerAdvice(annotations = {RestController.class, Controller.class})
@ResponseBody
public class GlobalExceptionHandler extends DefaultGlobalExceptionHandler {
}
```

4. 提供类常用的静态方法：`ifThrow(boolean condition, BaseEnum errorType, String... params)`

```
Integer exist = baseMapper.checkNameExistByParentId(request.getParentId(), request.getType(), request.getName(), request.getId());
BaseBusinessException.ifThrow(Objects.nonNull(exist), BusinessErrorEnum.CATEGORY_NAME_EXIST);
```

