# 编码规范总述：

**JSP页面：**

页面布局以及CSS样式

样式以及页面布局参见样式文件以及样例

页面控件：

各功能数据创建页面，采用jsp form表单提交数据，form表单中控件设计遵循：

* 页面中js逻辑部分需要写行级中文注释，概要说明js代码含义及用途；

页面JS校验：

* 页面中涉及到填写提交的地方都需要包含js校验。Js框架采用jQuery开发框架
* Js代码编写在jsp页面中。
* 必填项，页面提交时需要做非空校验；
* 页面上所有控件提交时做长度类型校验，例如：数值型字段提交时校验提交数据是否为数字，字段长度是否超过数据库最大长度，1个中文在Oracle中占位3个长度；

## Java编码规范及开发要求

* + - * 详见《JAVA编码规范》
      * 下班前提交SVN代码，上班后获取最新代码，锁定后进行修改
* 涉及修改、删除操作需要记录event\_log表

记录Event\_log的方法：

Insert into event\_log (

CREATE\_TIME,

ACTION\_USER,

ACTION\_ORG\_CODE,

ACTION\_DATE,

INDEX\_VALUE,

ACTION\_NAME,

ACTION\_TYPE,

DETAILS\_1,

DETAILS\_2,

DETAILS\_3,

DETAILS\_4,

DETAILS\_5 )

Values(

Sysdate，

当前操作人ID，

当前操作人组织代码，

Sysdate，

修改时，此处用被修改数据的ID或主键；删除时，此处用被删除数据的业务名称（如：企业名称、船名、航次、IMO号），

当前请求的controller 方法名，

操作类型（insert,delete,update,select），

Details1 到details5 用于存储修改前的数据详情，以及删除前的数据详情，每个details字段长度为2000，插入时需要考虑将要插入的数据长度是否超长，当数据过长时需要在插入前将数据拆分，分别插入到1至5个details中

)

* 涉及两条以上 增、删、改的 业务，需要用事务控制。
* 数据中涉及到codelibrary表 的字段，需要按给出 codeLibrary中的Type 查询出数据，并在页面上循环展示
* 页面跳转：新增，修改，删除操作成功时跳转至当前功能列表页；新增，修改操作失败时页面不跳转，提示用户失败原因
* 修改SVN中已存在的代码文件时，需要获取最新版本后锁定再进行修改