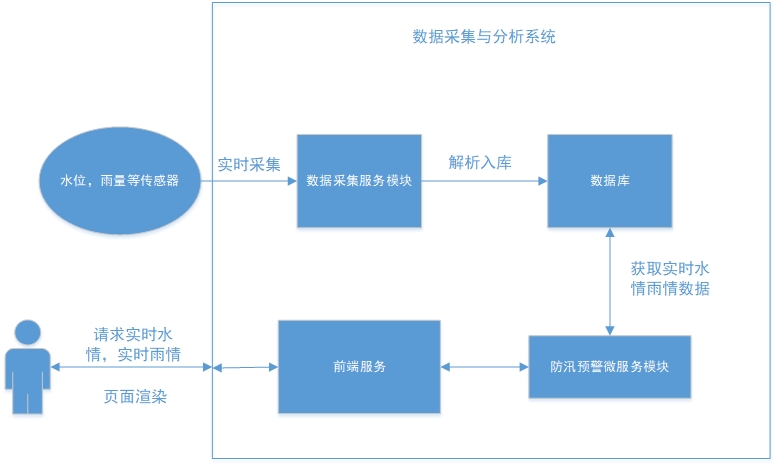
# 实时数据

## 1.1水文报表

### 1.1.1功能描述

用列表的方式展示水位站点的实时水情以及雨位站点的实时雨情。

### 1.1.2流程图



说明：

1. 数据采集服务模块会实时的接收水位、雨量等传感器发送的数据，该模块根据相应的协议进行解析，得到真实数据。
2. 解析完数据之后，需要先将实时表里的数据插入到历史表中，然后再将实时表里的数据清空。
3. 将解析完的实时数据插入到事实表中。
4. 用户点击防汛报表菜单，前端服务模块调用防汛预警微服务模块接口从数据库获取实时数据。
5. 防汛预警微服务模块将请求的数据返回前端服务，前端服务进行渲染，最后展示在页面上。

### 1.1.3接口设计

|  |  |
| --- | --- |
| 描述 | 获取实时水情和实时雨情 |
| 请求类型 | POST |
| 请求路径 | /floodcontrol-earlywarning/realTimeData |
| 请求参数 | 无 |
| 返回参数 | ｛  “code”:,  “message”:,  “success”:  “data”:{  “realTimeWater”:[{  “stationName”:””,  “time:””,  “waterLevel”:””,  “waterdepth:””,  “warningValue”:””,  “alarmValue:””},....],  “realTimePondingPoint”:[{  “stationName”:””,  “time:”:””,  “pondingPointLevel”:””,  “alarmValue”:””},....],  “realTimeRain”:[{  “stationName”:””,  “time”:””,  “rainfall”:””},......]  }  ｝ |
|  |  |

### 1.1.4数据库设计

表名：realtimedata

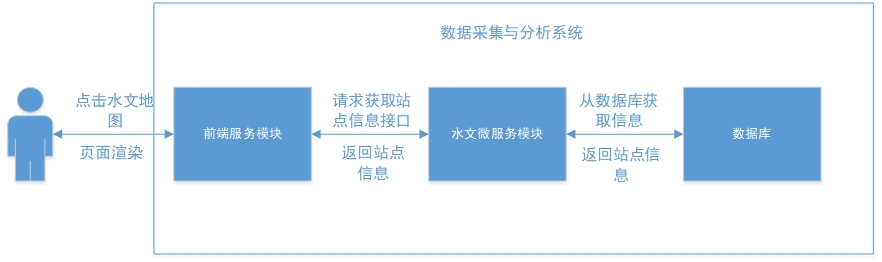
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 大小 | 是否null | 键 | 注释 |
| stationId | varchar | 50 | 是 |  | 站点id |
| time | datatime | 0 | 是 |  | 采集时间 |
| element | varchar | 50 | 是 |  | 采集参数 |
| value | double | 7 | 是 |  | 参数值 |

## 1.2水文地图

### 1.2.1功能描述

在二维地图上展示所有站点的位置和基本信息。基本信息包括水位、雨量、站点的地址、所属流域以及关联乡村等信息。另外可以根据点位名称进行查询。

### 1.2.2流程图



说明

1. 用户点击水文地图菜单，前端服务模块调用水文微服务模块的获取站点信息接口。
2. 水文微服务模块接到请求后，向数据库查询站点信息，并返回前端服务模块。
3. 前端服务模块利用经纬度进行渲染。展示在二维地图上。

### 1.2.3接口设计

|  |  |
| --- | --- |
| 描述 | 获取所有站点 |
| 请求类型 | POST |
| 请求路径 | /floodcontrol-earlywarning/stations |
| 请求参数 | 无 |
| 返回参数 | ｛  “code”:,  “message”:,  “success”:  “data”:{  “stations”:[{  “stationId”:””,  “stationName”:””,  “address”:””,  “valleyName”:””,  “code\_region”:””,  “stationType”:””,  “latitude ”:””,  “longitude”:””,  “value”:””,  }  ,{}.....]  }  ｝ |
|  |  |

### 1.2.4数据库设计

表名：station

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 大小 | 是否null | 键 | 注释 |
| stationId | varchar | 50 | 否 | 主键 | 站点id |
| stationName | varchar | 50 | 是 |  | 站点名称 |
| address | varchar | 200 | 是 |  | 站点地址 |
| valleyId | int | 7 | 是 |  | 所属流域 |
| code\_region | varchar | 50 | 是 |  | 所属区域 |
| stationType | Int | 7 | 是 |  | 站点类型 |
| latitude | varchar | 50 | 是 |  | 经度 |
| longitude | varchar | 50 | 是 |  | 纬度 |

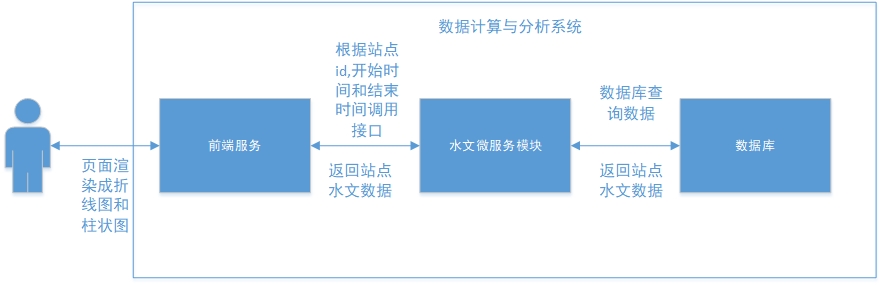
# 历史数据

## 2.1水位时段图

### 2.1.1功能描述

可以选择某个站点，自定义开始时间和结束时间查询该站水位信息，并以折线图和统计表两种形式展示，支持导出和打印。

### 2.1.2流程图



说明

1. 用户点击水文时段图菜单，选择某个站点，选择开始时间和结束时间。
2. 前端服务调用水文微服务模块的相关接口。
3. 水文微服务查询数据库，返回水文数据。
4. 前端服务进行页面渲染，分别以柱状图和折线图展现给用户。

### 2.1.3接口设计

|  |  |
| --- | --- |
| 描述 | 获取站点的水位信息 |
| 请求类型 | POST |
| 请求路径 | /floodcontrol-earlywarning/stationInfo |
| 请求参数 | {  “stationId”:””,  “startTime”:””,  “endTime”:””  } |
| 返回参数 | ｛  “code”:,  “message”:,  “success”:  “data”:{  “stationInfo”:[{  “stationId”:””,  “stationName”:””,  “waterLevel”:””}  ,{}.....]  }  ｝ |
|  |  |

### 2.1.4数据库设计

表名：stationdata

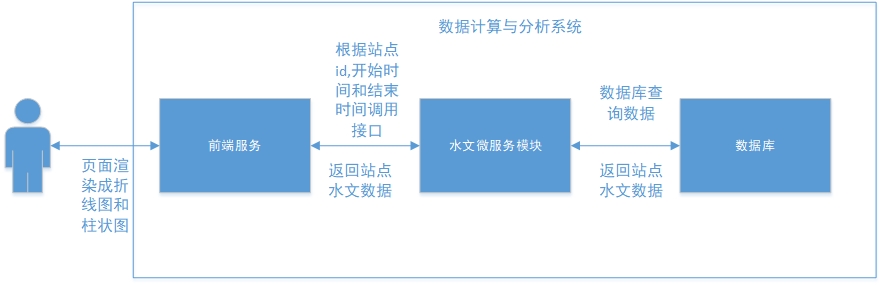
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 大小 | 是否null | 键 | 注释 |
| stationId | varchar | 50 | 是 |  | 站点id |
| time | datatime | 0 | 是 |  | 采集时间 |
| element | varchar | 50 | 是 |  | 采集参数 |
| value | double | 7 | 是 |  | 参数值 |

## 2.2雨量时段图

### 2.2.1功能描述

可以选择某个站点，自定义开始时间和结束时间，查询该站雨量信息，并以折线图和统计表两种形式展示，支持导出和打印。

### 2.2.2流程图



说明

1. 用户点击水文时段图菜单，选择某个站点，选择开始时间和结束时间。
2. 前端服务调用水文微服务模块的相关接口。
3. 水文微服务查询数据库，返回水文数据。
4. 前端服务进行页面渲染，分别以柱状图和折线图展现给用户。

### 2.2.3接口设计

|  |  |
| --- | --- |
| 描述 | 获取站点的水位信息 |
| 请求类型 | POST |
| 请求路径 | /floodcontrol-earlywarning/stationInfo |
| 请求参数 | {  “stationId”:””,  “startTime”:””,  “endTime”:””  } |
| 返回参数 | ｛  “code”:,  “message”:,  “success”:  “data”:{  “stationInfo”:[{  “stationId”:””,  “stationName”:””,  “rainfall”:””}  ,{}.....]  }  ｝ |
|  |  |

### 2.2.4数据库设计

表名：stationdata

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 大小 | 是否null | 键 | 注释 |
| stationId | varchar | 50 | 是 |  | 站点id |
| time | datatime | 0 | 是 |  | 采集时间 |
| element | varchar | 50 | 是 |  | 采集参数 |
| value | double | 7 | 是 |  | 参数值 |

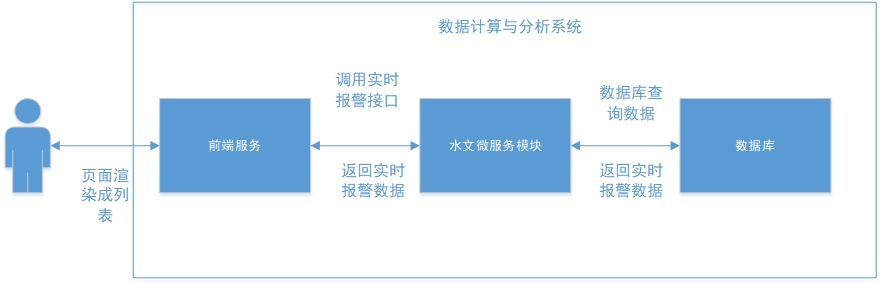
# 报警信息

## 3.1水文实时报警

### 3.1.1功能描述

可实时显示报警的站点、报警时间、报警类型、报警数值和报警描述，并可进行排序查看。

### 3.1.2流程图



说明：

1. 用户点击实时报警菜单，前端服务调用水文微服务模块的报警接口。
2. 水文微服务查询数据库获取相关报警数据，返回前端服务。
3. 前端服务对报警数据进行渲染，列表展示。

### 3.1.3接口设计

|  |  |
| --- | --- |
| 描述 | 获取实时报警信息 |
| 请求类型 | POST |
| 请求路径 | /floodcontrol-earlywarning/realTime-alarmInfo |
| 请求参数 | 无 |
| 返回参数 | ｛  “code”:,  “message”:,  “success”:  “data”:{  “alarmInfo”:[{  “stationId”:””,  “stationName”:””,  “alarmTime”:””,  “alarmType”:””,  “alarmValue”:””,  “alarmDesc”:””}  ,{}.....]  }  ｝ |
|  |  |

### 3.1.4数据库设计

表名：alarmdata

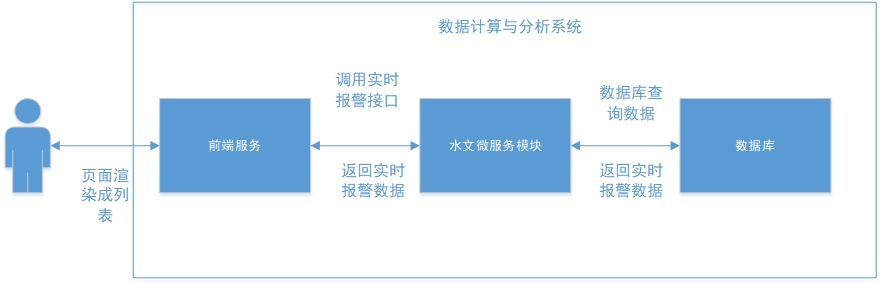
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 大小 | 是否null | 键 | 注释 |
| id | int | 11 | 否 | 主键 | Id |
| stationId | varchar | 50 | 是 |  | 站点id |
| alarmTime | datatime | 0 | 是 |  | 报警时间 |
| alarmType | int | 4 | 是 |  | 报警类型 |
| alarmValue | double | 7 | 是 |  | 报警值 |
| alarmDesc | varchar | 200 | 是 |  | 报警描述 |
| alarmState | int | 4 | 是 |  | 报警状态 |

## 3.2水文历史预警

### 3.2.1功能描述

可查询水文站点在自定义的时间段内历史报警数据，包括报警时间、报警名称、报警数值、报警描述和报警状态。

### 3.2.2流程图



说明：

1. 用户点击水文历史预警菜单，前端服务根据站点id,开始时间和结束时间调用水文微服务模块的相关接口。
2. 水文微服务根据条件查询数据库，返回给前端服务。
3. 前端服务进行对数据进行页面渲染，列表展示给用户。

### 3.2.3接口设计

|  |  |
| --- | --- |
| 描述 | 获取实时报警信息 |
| 请求类型 | POST |
| 请求路径 | /floodcontrol-earlywarning/alarmInfo |
| 请求参数 | 无 |
| 返回参数 | ｛  “code”:,  “message”:,  “success”:  “data”:{  “alarmInfo”:[{  “stationId”:””,  “stationName”:””,  “alarmTime”:””,  “alarmType”:””,  “alarmValue”:””,  “alarmDesc”:””,  ,{}.....]  }  ｝ |
|  |  |

### 3.2.4数据库设计

表名：alarmdata

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 大小 | 是否null | 键 | 注释 |
| id | int | 11 | 否 | 主键 | Id |
| stationId | varchar | 50 | 是 |  | 站点id |
| alarmTime | datatime | 0 | 是 |  | 报警时间 |
| alarmType | int | 4 | 是 |  | 报警类型 |
| alarmValue | double | 7 | 是 |  | 报警值 |
| alarmDesc | varchar | 200 | 是 |  | 报警描述 |
| alarmState | int | 4 | 是 |  | 报警状态 |

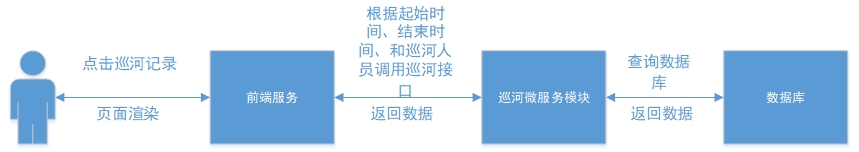
# 巡河模块

## 4.1巡河记录

### 4.1.1功能描述

可以根据区域选择站点，根据时间段、巡河人员和状态来查找对应的巡河记录。

### 4.1.2流程图



### 4.1.3接口设计

|  |  |
| --- | --- |
| 描述 | 获取巡河记录 |
| 请求类型 | POST |
| 请求路径 | /patrol-river/patrolriver |
| 请求参数 | 无 |
| 返回参数 | ｛  “code”:,  “message”:,  “success”:  “data”:{  “patrolriver”:[{  “regionId”:””,  “startTime”:””,  “endTime”:””,  “patrolTime”:””,  “patrolMileage”:””,  “directorId”:””,  “patrolQuestion”:””,  “trajectory”:””}  ,{}.....]  }  ｝ |
|  |  |

### 4.1.4数据库设计

表名：patrolriver

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 大小 | 是否null | 键 | 注释 |
| id | int | 11 | 否 | 主键 | Id |
| regionId | varchar | 11 | 是 |  | 区域id |
| startTime | datatime | 0 | 是 |  | 开始时间 |
| endTime | datatime | 0 | 是 |  | 结束时间 |
| patrolTime | varchar | 50 | 是 |  | 巡河时长 |
| patrolMileage | varchar | 50 | 是 |  | 巡河里程 |
| directorId | int | 4 | 是 |  | 巡河人员id |
| patrolQuestion | int | 4 | 是 |  | 巡河问题个数 |
| trajectory | varchar | 200 | 是 |  | 巡河轨迹 |

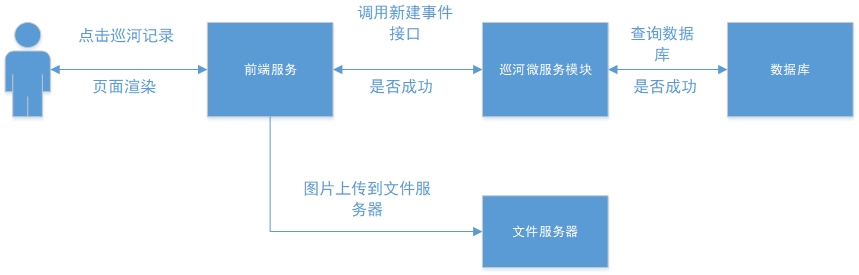
# 事件管理

## 5.1新建事件

### 5.1.1功能描述

新建一个巡河事件。

### 5.1.2流程图



说明：

1. 用户点击新建巡河事件按钮，跳出新建事件页面，填写相应的信息，上传图片后点击上报。
2. 前端服务调用巡河微服务新增事件接口，持久化到数据库。
3. 巡河微服务返回添加成功消息。

### 5.1.3接口设计

|  |  |
| --- | --- |
| 描述 | 获取巡河记录 |
| 请求类型 | POST |
| 请求路径 | /patrol-river/addEvent |
| 请求参数 | {  “questionType”:””,  “relationRiver”:””,  “reportTime”:””,  “regionId”:””,  “reportPersonId”:””,  “reportPersonPhone”:””,  “address”:””,  “fileName”:””,  “fileUrl”:””  } |
| 返回参数 | ｛  “code”:,  “message”:,  “success”:  “data”:{  }  ｝ |
|  |  |

### 5.1.4数据库设计

表名：event

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 大小 | 是否null | 键 | 注释 |
| id | int | 11 | 否 | 主键 | Id |
| questionType | int | 11 | 是 |  | 问题类型 |
| relationRiver | int | 11 | 是 |  | 相关河道 |
| reportTime | datatime | 0 | 是 |  | 上报时间 |
| regionId | varchar | 50 | 是 |  | 行政区划 |
| reportPersonId | varchar | 50 | 是 |  | 上报人id |
| reportPersonPhone | varchar | 50 | 是 |  | 上报人联系方式 |
| address | varchar | 200 | 是 |  | 问题地址 |
| questionDesc | varchar | 200 | 是 |  | 问题描述 |
| state | int | 11 | 是 |  | 状态 |
| handlerId | int | 11 | 是 |  | 处理人id |
| approver | int | 11 | 是 |  | 审批人id |

表名：file

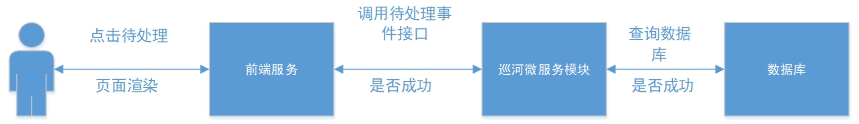
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 大小 | 是否null | 键 | 注释 |
| id | int | 11 | 否 | 主键 | Id |
| fileName | varchar | 100 | 是 |  | 文件名 |
| fileUrl | varchar | 100 | 是 |  | 文件路径 |
| relationId | int | 11 | 是 |  | 关联id |

## 5.2待处理

### 5.2.1功能描述

将需要待处理的问题事件以列表的形式展示，对问题事件进行处理，包括交办和结案。

### 5.2.2流程图



说明：

1. 用户点击待处理菜单，前端服务调用巡河微服务模块中的待处理接口。
2. 巡河微服务根据条件查询数据库，数据库返回相应的数据。
3. 巡河微服务将数据返回给前端服务，前端服务进行页面渲染，列表展示。

### 5.2.3接口设计

|  |  |
| --- | --- |
| 描述 | 获取待处理事件列表 |
| 请求类型 | POST |
| 请求路径 | /patrol-river/eventState |
| 请求参数 | {  “state”:1,  } |
| 返回参数 | ｛  “code”:,  “message”:,  “success”:  “data”:{  “event”:[{  “questionType”:””,  “relationRiver”:””,  “reportTime”:””,  “regionId”:””,  “reportPersonId”:””,  “reportPersonPhone”:””,  “address”:””  },....]  }  ｝ |
|  |  |

### 5.2.4数据库设计

表名：event

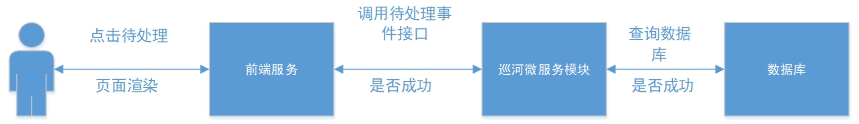
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 大小 | 是否null | 键 | 注释 |
| id | int | 11 | 否 | 主键 | Id |
| questionType | int | 11 | 是 |  | 问题类型 |
| relationRiver | int | 11 | 是 |  | 相关河道 |
| reportTime | datatime | 0 | 是 |  | 上报时间 |
| regionId | varchar | 50 | 是 |  | 行政区划 |
| reportPersonId | varchar | 50 | 是 |  | 上报人id |
| reportPersonPhone | varchar | 50 | 是 |  | 上报人联系方式 |
| address | varchar | 200 | 是 |  | 问题地址 |
| questionDesc | varchar | 200 | 是 |  | 问题描述 |
| state | int | 11 | 是 |  | 状态 |
| handlerId | int | 11 | 是 |  | 处理人id |
| approver | int | 11 | 是 |  | 审批人id |

## 5.3已交办

### 5.3.1功能描述

将已交办的问题事件列表展示。

### 5.3.2流程图



说明：

1.用户点击待处理菜单，前端服务调用巡河微服务模块中的已交办接口。

2.巡河微服务根据条件查询数据库，数据库返回相应的数据。

3.巡河微服务将数据返回给前端服务，前端服务进行页面渲染，列表展示。

### 5.3.3接口设计

|  |  |
| --- | --- |
| 描述 | 获取待处理事件列表 |
| 请求类型 | POST |
| 请求路径 | /patrol-river/eventState |
| 请求参数 | {  “state”:2,  } |
| 返回参数 | ｛  “code”:,  “message”:,  “success”:  “data”:{  “event”:[{  “questionType”:””,  “relationRiver”:””,  “reportTime”:””,  “regionId”:””,  “reportPersonId”:””,  “reportPersonPhone”:””,  “address”:””  },....]  }  ｝ |
|  |  |

### 5.3.4数据库设计

表名：event

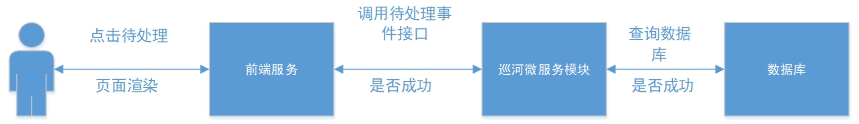
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 大小 | 是否null | 键 | 注释 |
| id | int | 11 | 否 | 主键 | Id |
| questionType | int | 11 | 是 |  | 问题类型 |
| relationRiver | int | 11 | 是 |  | 相关河道 |
| reportTime | datatime | 0 | 是 |  | 上报时间 |
| regionId | varchar | 50 | 是 |  | 行政区划 |
| reportPersonId | varchar | 50 | 是 |  | 上报人id |
| reportPersonPhone | varchar | 50 | 是 |  | 上报人联系方式 |
| address | varchar | 200 | 是 |  | 问题地址 |
| questionDesc | varchar | 200 | 是 |  | 问题描述 |
| state | int | 11 | 是 |  | 状态 |
| handlerId | int | 11 | 是 |  | 处理人id |
| approver | int | 11 | 是 |  | 审批人id |

## 5.4已结案

### 5.4.1功能描述

将已结案的列表展示。

### 5.4.2流程图



说明：

1.用户点击待处理菜单，前端服务调用巡河微服务模块中的已结案接口。

2.巡河微服务根据条件查询数据库，数据库返回相应的数据。

3.巡河微服务将数据返回给前端服务，前端服务进行页面渲染，列表展示。

### 5.4.3接口设计

|  |  |
| --- | --- |
| 描述 | 获取待处理事件列表 |
| 请求类型 | POST |
| 请求路径 | /patrol-river/eventState |
| 请求参数 | {  “state”:3,  } |
| 返回参数 | ｛  “code”:,  “message”:,  “success”:  “data”:{  “event”:[{  “questionType”:””,  “relationRiver”:””,  “reportTime”:””,  “regionId”:””,  “reportPersonId”:””,  “reportPersonPhone”:””,  “address”:””  },....]  }  ｝ |
|  |  |

### 5.4.4数据库设计

表名：event

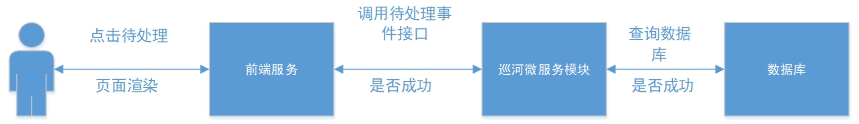
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 大小 | 是否null | 键 | 注释 |
| id | int | 11 | 否 | 主键 | Id |
| questionType | int | 11 | 是 |  | 问题类型 |
| relationRiver | int | 11 | 是 |  | 相关河道 |
| reportTime | datatime | 0 | 是 |  | 上报时间 |
| regionId | varchar | 50 | 是 |  | 行政区划 |
| reportPersonId | varchar | 50 | 是 |  | 上报人id |
| reportPersonPhone | varchar | 50 | 是 |  | 上报人联系方式 |
| address | varchar | 200 | 是 |  | 问题地址 |
| questionDesc | varchar | 200 | 是 |  | 问题描述 |
| state | int | 11 | 是 |  | 状态 |
| handlerId | int | 11 | 是 |  | 处理人id |
| approver | int | 11 | 是 |  | 审批人id |

## 5.5全部事件

### 5.5.1功能描述

将跟用户用户相关的全部事件进行列表展示。

### 5.5.2流程图



说明：

1.用户点击待处理菜单，前端服务调用巡河微服务模块中的全部事件接口。

2.巡河微服务根据条件查询数据库，数据库返回相应的数据。

3.巡河微服务将数据返回给前端服务，前端服务进行页面渲染，列表展示。

### 5.5.3接口设计

|  |  |
| --- | --- |
| 描述 | 获取待处理事件列表 |
| 请求类型 | POST |
| 请求路径 | /patrol-river/eventState |
| 请求参数 | {  “userId”:3,  } |
| 返回参数 | ｛  “code”:,  “message”:,  “success”:  “data”:{  “event”:[{  “questionType”:””,  “relationRiver”:””,  “reportTime”:””,  “regionId”:””,  “reportPersonId”:””,  “reportPersonPhone”:””,  “address”:””  },....]  }  ｝ |
|  |  |

### 5.5.4数据库设计

表名：event

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 大小 | 是否null | 键 | 注释 |
| id | int | 11 | 否 | 主键 | Id |
| questionType | int | 11 | 是 |  | 问题类型 |
| relationRiver | int | 11 | 是 |  | 相关河道 |
| reportTime | datatime | 0 | 是 |  | 上报时间 |
| regionId | varchar | 50 | 是 |  | 行政区划 |
| reportPersonId | varchar | 50 | 是 |  | 上报人id |
| reportPersonPhone | varchar | 50 | 是 |  | 上报人联系方式 |
| address | varchar | 200 | 是 |  | 问题地址 |
| questionDesc | varchar | 200 | 是 |  | 问题描述 |
| state | int | 11 | 是 |  | 状态 |
| handlerId | int | 11 | 是 |  | 处理人id |
| approver | int | 11 | 是 |  | 审批人id |