8月review

Created and last modified by 王文磊 on Sep 09, 2021

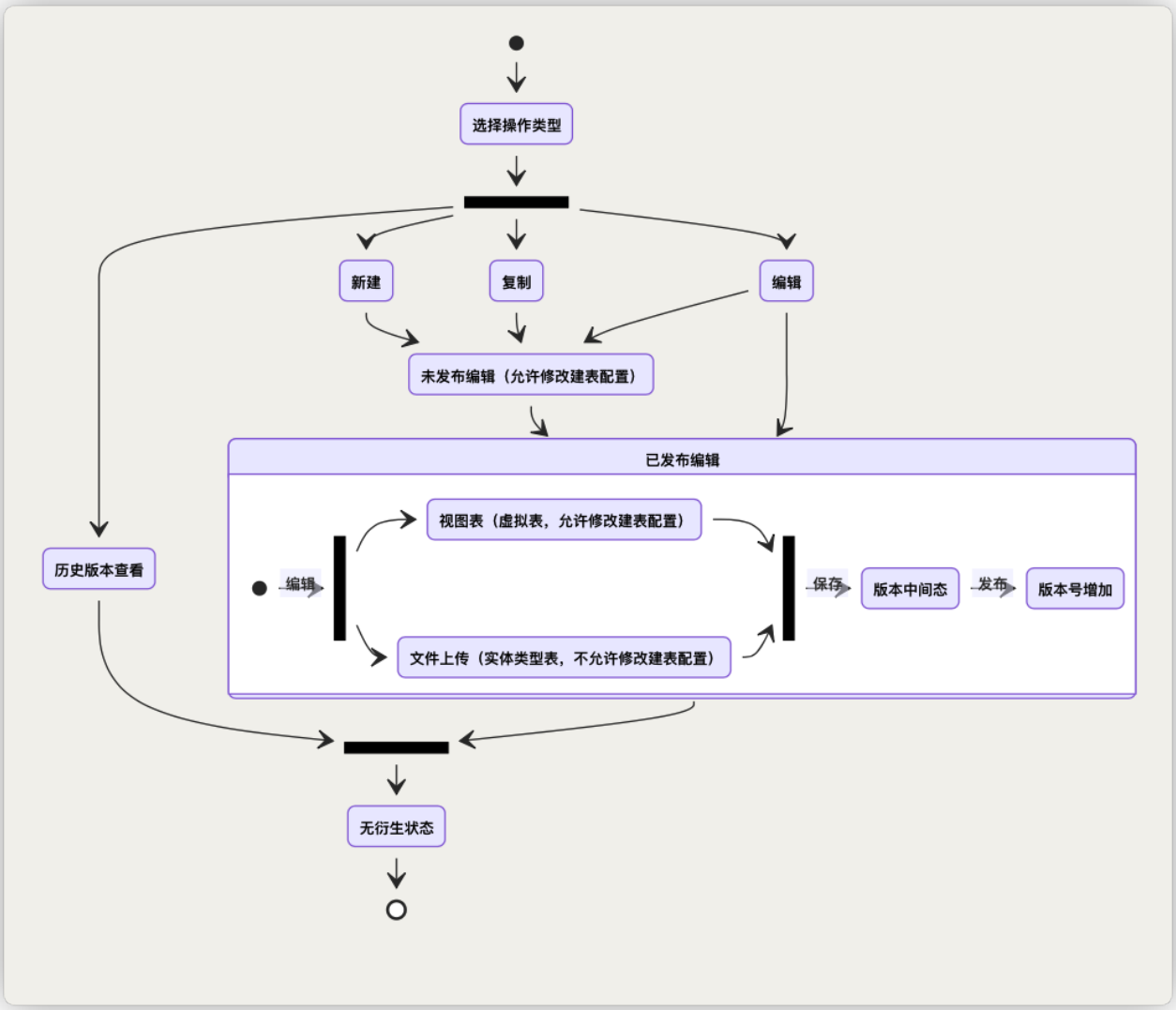
需求总览

需求名称	时间	概述
数据权限-权限管理-用户权限	8.2-8.5	需求上线调整及优化：wiki
分析图表-分析配置抽屉	8.6-8.19	拆解业务流程 -> 映射到代码逻辑 -> 开发 -> 自测 -> 联调 -> 提测 -> 调整
数据开发模块(视图表&文件上传)	8.20-8.31	开发 -> 自测 -> 联调 -> 提测 -> 调整 -> 上线

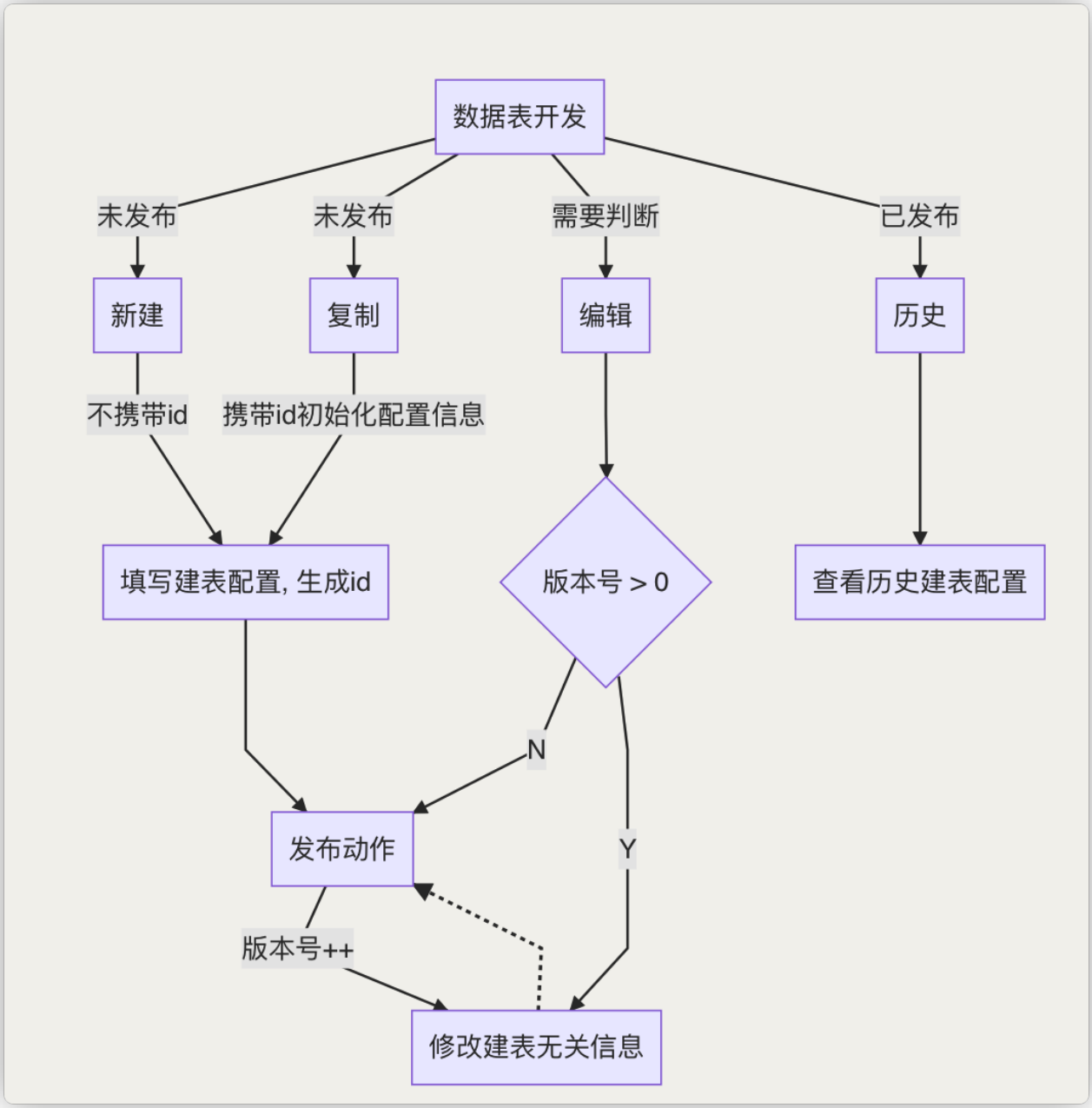
问题总结

一、视图表&文件上传

- **状态梳理：**



- **业务流程：**



- 文件上传处理，file（blob）对象 -> 二进制流文件, readAsBinaryString() -> body（fail） ? Format （success），content-type：multipart/form-data

  Upload组件api

```
<Field label="选择文件" labelClass="w6em" className="w300">
  <Upload
    name="file"
    accept=".txt, .csv, .xlsx"
    beforeUpload={this.saveFile}
    onRemove={this.removeFile}
    fileList={fileList}
  >
    <A>点击选择上传文件</A>
  </Upload>
</Field>
```

  保存文件：

```
// 保存file对象至缓存
@action.bound saveFile(file) {
  this.uploadModel.set('file', file);
  this.uploadModel.set('name', file.name);
}
```

```
saveFile = (e) => {
  const { store } = this.props;
  this.setState({
    fileList: [e],
  });
  store.saveFile(e);
  return false;
}
```

移除文件：

```
removeFile = () => {
  const { store } = this.props;
  // 控制组件样式
  this.setState({
    fileList: [],
  });
  // 修改请求参数
  store.saveFile(undefined);
}
```

提交：

```
// 文件上传处理
@action.bound handleUpload = async () => {
  if (!this.UploadFile) {
    Notification.error('请选择一个文件！');
    return;
  }
  this.uploading = true;
  const formdata = new window.FormData();
  formdata.append('input', this.UploadFile);
  formdata.append('partitionContentList', this.UploadModel.partitions.map(item => item.value));
  formdata.append('method', this.UploadModel.type);
  formdata.append('profile', this.UploadModel.profile);
  formdata.append('fileName', this.UploadModel.name);
  formdata.append('id', this.Id);
  Request.post('/dpa/api/development/data/v1/upload', {
    data: formdata,
    formData: true,
    timeout: 3600000,
  })
    .then((res) => {
      this.uploading = false;
      if (res && res.data) {
        Notification.success(res.data);
      }
    })
    .catch(() => {
      this.uploading = false;
    });
}
```

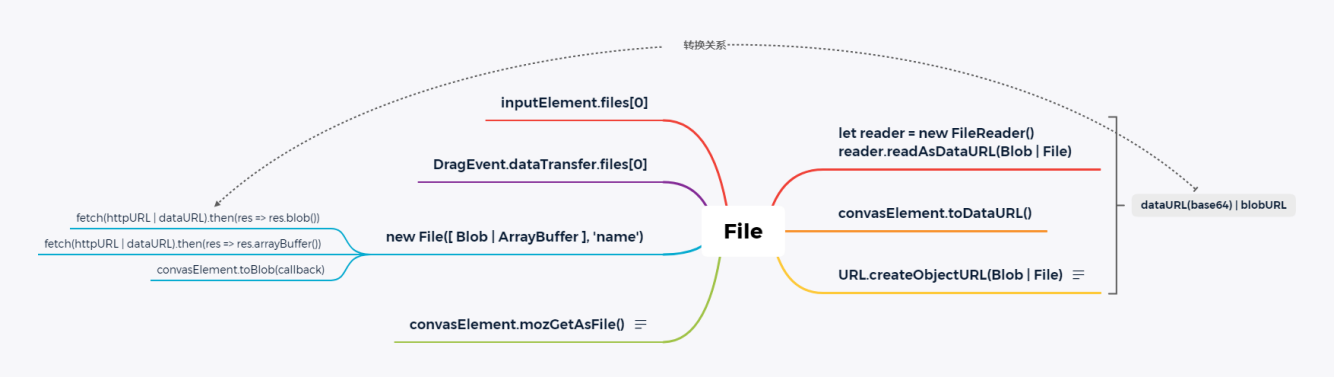
Request：

```
/**
 * 请求体设置钩子
 *
 * 由于本钩子实际上最终生成了数据形式，为了方便业务操作
 * 本钩子置于所有请求请中间件之后
 */
Request.unuse.contentTypeRequestHook = Request.useAfterPrivateHooks('send', function contentTypeRequestHook(next) {
  const { req } = this;

  if (req.formData) {
    req.body = req.data;
  } else if (/^get$/i.test(req.method)) {
    req.url = appendUrl(req.url, stringify(req.data));
  } else if (req.formData) {
    req.body = req.data;
  } else if (req.form) {
    req.headers.set('Content-Type', 'application/x-www-form-urlencoded;charset=UTF-8');
    req.body = stringify(req.data);
  } else {
    req.headers.set('Content-Type', 'application/json;charset=UTF-8');
    req.body = JSON.stringify(req.data);
  }

  req.credentials = req.credentials || 'same-origin';

  next();
});
```



基于xhr 2提供的FormData接口生成FormData对象，使用append（k，v）添加表单项，此时v可以为file对象（该对象基于Blob，扩展了支持用户上传文件的功能，只存储文件的相关信息，并不包含实质内容，但可通过使用Blob的api进行读取处理，如FileReader）。将FormData对象作为xhr请求的body时，请求内容格式content-type会被自动设置为 multipart / form-data，此时文件随表单以数据流的格式上传。

二、分析图表

- key不固定导致的重复render失去焦点问题

  老代码：

```
export default observer(function Filters({ data, setValue, getFieldDecorator, getFieldsValue, departmentList, getUserPermTypeList, userPermTypeList } = {}) {
  const { conditions, type, open_perm, perm_conditions } = data || {};
  const isValidConditions = Array.isArray(conditions) && conditions.length > 0;
  if (!isValidConditions) {
    addFilter('conditions', data, setValue)
  }
  return <div>
    {
      isValidConditions ? conditions.map((item, index) => {
      let placeholder = '';
      if (!item._id) {
        item._id = generateUUID();
      }
      if (type === 3) {
        placeholder = '请输入指标信息';
      } else if (type === 2) {
        placeholder = '请输入时间信息';
      } else if (type === 1) {
        placeholder = '请输入维度信息';
      }
      return (
        <div key={index} style={{ marginBottom: 5 }}>
          {
            index > 0 ?
              <FormItem style={{ display: 'inline-block', marginBottom: 0 }}>
                {getFieldDecorator(`logical-${item._id}`, {
                  initialValue: item.logical_opt || 'and',
                  rules: [{ required: true, message: '请选择' }],
                })(
                  <Select
                    key={`select-${index}`}
                    style={{ width: 80, display: 'inline-block' }}
                    dropdownMatchSelectWidth={false}
                    onSelect={value => changeFilter(value, item._id, 'logical_opt', 'conditions', data, setValue)}
                    placeholder="请选择"
                  >
                    <Option key="and" value="and">and</Option>
                    <Option key="or" value="or">or</Option>
                  </Select>
                )}
              </FormItem> : null
          }
```

- 原生表格（table）列宽度问题

老代码：

```
return (
  <table className={styles.tableContainer}>
    <thead className={styles.tableHead} style={{ display: rowRevert ? 'none' : 'table-header-group' }} re
      <tr key="theadTr">
        { columns.map((item, idx) => {
          let classUpVal = 'off';
          let classDownVal = 'off';
          if (item.key === sortValue.field || item.title === sortValue.field) {
            if (sortValue.order === 'ascend') {
              classUpVal = 'on';
            } else if (sortValue.order === 'descend') {
              classDownVal = 'on';
            }
          }
          return (
            <th key={idx}>
              <span style={{ whiteSpace: 'pre' }}>
                <span dangerouslySetInnerHTML={{ __html: item.title }} />
                <div className={styles.columnSorter} style={{ display: canOrder ? 'inline-block' : 'none'
                  <Icon type="caret-up" className={`${styles.up} ${styles[classUpVal]}`} onClick={() => th
                  <Icon type="caret-down" className={`${styles.down} ${styles[classDownVal]}`} onClick={()
                </div>
              </span>
            </th>
          );
        })}
      </tr>
    </thead>
    {
      dataSource && dataSource.length > 0 ? (
        <tbody
          key="tableBody"
          onScroll={this.onBodyScroll}
          className={styles.tableBody}
          ref={(ref) => { this.tableBody = ref; }}
        >
          {
            dataSource.map((item, idx) => (
              <Tr
                idx={idx}
                key={`tbodyTr${idx}`}
                columns={columns}
                rowRevert={rowRevert}
                options={options}
                showModal={this.showModal}
                dataSource={dataSource}
                tableProps={tableProps}
              />
            ))
          }
        </tbody>
      ) : (
        <tbody style={{ display: 'block' }}><tr style={{ display: 'block' }}><td className={styles.noData}
      )
    }
  </table>
```

```
.tableContainer {
  border-collapse: separate;
  border-spacing: 0;
  border-radius: 4px;
  overflow: hidden;
  color: rgba(0,0,0,.65);
  /* border: 1px solid #e9e9e9; */
  width: 100%;
  white-space:nowrap;
}

.tableHead, .tableBody {
  text-align: left;
}

.tableHead {
  border-top: 1px solid #e9e9e9;
}

.tableBody {
  overflow: hidden;
}

.tableHead td, .tableHead th, .tableBody td, .tableBody th  {
  padding: 10px 8px;
  font-size: 12px;
  color: #666;
}

.tableBody>tr>td {
  border-bottom: 1px solid #eee;
}

.tableConFix .tableBody>tr>td {
  white-space: nowrap;
  text-overflow: ellipsis;
  overflow: hidden;
}

/* .tableBody>tr:last-child>td { */
  /* border-bottom: 0px; */
/* } */

.tableHead>tr>td:last-child, .tableBody>tr>td:last-child {
  border-right: 0px;
}

.tableHead>tr>th {
  background: #FAFAFA;
  font-weight: 500;
  transition: background .3s ease;
  text-align: left;
  color: rgba(0,0,0,.85);
  padding: 9px 16px;
  font-size: 14px;
```

修改后：

```
const cols = [...group_bys, ...metrics];
let isCustom = false;       You, 2 days ago • 找回原Slice组件
let computedWidth = 0;
cols.forEach((item) => {
  if (item && item.width) {
    computedWidth += item.width;
  } else if (document.getElementsByName(`head_${item.key}`)[0]) {
    computedWidth += document.getElementsByName(`head_${item.key}`)[0].offsetWidth;
  }
  if (item && item.widthType === 'custom' && item.width) {
    isCustom = true;
  }
});
return (
  <table className={styles.tableContainer} style={{ width: computedWidth > 0 && isCustom ? computedWidth : '100%', whiteSpace: isCustom ? 'normal' : 'nowrap' }}>
    <thead className={styles.tableHead} style={{ display: rowRevert ? 'none' : 'table-header-group' }} ref={(ref) => { this.tbHead = ref; }} id="tableTheadMenu">
      <tr key="theadTr">
        {columns.map((item, idx) => {
          let classUpVal = 'off';
          let classDownVal = 'off';
          if (item.key === sortValue.field || item.title === sortValue.field) {
            if (sortValue.order === 'ascend') {
              classUpVal = 'on';
            } else if (sortValue.order === 'descend') {
              classDownVal = 'on';
            }
          }
          return (
            <th key={idx} style={{ width: cols[idx] && cols[idx].widthType === 'custom' && cols[idx].width ? cols[idx].width : 'unset', textAlign: cols[idx] ? cols[idx].align : 'left' }} name={`head_${item.key}`}>
              <span>
                <span dangerouslySetInnerHTML={{ __html: item.title }} />
                <div className={styles.columnSorter} style={{ display: canOrder ? 'inline-block' : 'none' }}>
                  <Icon type="caret-up" className={`${styles.up} ${styles[classUpVal]}`} onClick={() => this.changeSort(item, 'ascend')} />
                  <Icon type="caret-down" className={`${styles.down} ${styles[classDownVal]}`} onClick={() => this.changeSort(item, 'descend')} />
                </div>
              </span>
            </th>
          );
        })}
      </tr>
    </thead>
```

Like Be the first to like this