

# 岗位职责及介绍

大家好，我是工业物联网信息能力中心的毛浪，我的岗位是前端UI工程师。

前端UI这个岗位细分的话，它其实包含两个方面：一是UI设计，二是前端开发。

- 总的来说，前端UI最终实现web或移动互联网网页视觉的呈现与用户交互，视觉呈现就是我们点开一个网站或APP所看到的所有东西，包括颜色，布局，特效动画、各种组件等等。用户交互就是，比如看到一个按钮组件、表单组件你可以点击，系统能给予你相应回馈，能增删改查一系列信息或数据的。
- 职责细分来讲
  - UI设计（或称界面设计）是指对软件的人机交互、操作逻辑、界面美观的整体设计。我们日常生活中所用到的：手机、电脑、电视、车载系统、iPad、ATM机、工业中控系统……只要是带有电子屏幕的显示设备，都需要UI设计。<sup>[1]</sup>。好的UI设计不仅是让软件变得有个性有品位，还要让软件的操作变得舒适简单、自由，充分体现软件的定位和特点。
  - 前端开发是创建WEB页面或APP等前端界面呈现给用户的过程，通过HTML，CSS及JavaScript以及衍生出来的各种技术、框架、解决方案，来实现互联网产品的用户界面视觉静态呈现和动态交互功能。简单来说对于用户界面的最终实现UI设计定义了蓝图，前端开发负责编码实现。<sup>[2]</sup>。可以说前端是最贴近用户的程序员，前端的能力就是能让产品从90分进化到100分，甚至更好。

大概介绍了一下前端UI这个岗位是做什么的，接下来我想与大家分享一下，近期结项项目中的几个案例实现。

首先，项目介绍

## 项目介绍

项目名称：web自由组态

项目概述：web自由组态的实现能使用户通过类似“搭积木”的简单方式来完成自己所需要的软件功能，而不需要编写计算机程序，它可将自动化过程和设备中采集的各种信息以图形化等更易于理解的方式进行显示，也能对信息执行必要的分析，发出指令等。主要为可视化数据监控屏的应用。

下面这几张图就是典型可视化数据监控屏应用场景：也能完美适应pc端和移动端屏幕

图片1

图片2

web自由组态本质上就是实现各种数据可视化应用场景搭建的工具软件，例如图片上的数据、图表、表单等用户均可通过自定义编辑实现。拿图表组件来举例，类似于制图软件，用户通过从组件库中拖拽出对应图表比如柱状图，通过设置柱状图的宽高、位置、最大最小刻度、主次刻度等分、刻度线及柱体颜色等等样式定制最终达成目标效果。

# 项目案例

接下来我将介绍项目开发中遇到的几个案例

## 案例1：数据组件的定制化使用实现

- 案例分析：
  - 案例需求：实现例如折线图、柱状图、仪表盘、环形图、饼图、天气API等数据组件的动态化定制。
  - 竞品实现方案：对比繁易组态、BY组态，对于数据组件的实现单纯采用静态图片代替，简单来说就是拖拽出来的数据组件只是一张固定死的图片。这就会产生一下几个问题：
    - 1、在组件大小变换方面：图片缩放会导致失真模糊，竞品1采取禁止数据组件进行大小变换，竞品2不做任何处理，且天气API数据组件的最终渲染大小与用户定义不符。
    - 2、在实时渲染方面：静态图片并不能满足更随用户设置而实时变化渲染效果。
  - 项目实施方案：引入第三方库Echarts（响应式数据可视化框架）进行功能模块开发。
    - 实施方案难点：底层原理上不能将Echarts渲染出的数据图形纳入Konvajs生成的图形容器中，致使数据图形无法渲染和控制，组件拖拽、变换、导航操作等问题成为实现难点。
    - 难点问题解决：数据组件渲染实现使用CSS定位计算及层叠顺序，数据组件控制实现使用Konva间接图形对象控制。
- 案例实现：
  - 渲染实现：
    - 相对定位：对元素设置相对定位后可以通过设置垂直或水平位置，让这个元素“相对于”它的起点进行移动，值得注意的是在使用相对定位时，无论是否进行移动，元素仍然占据原来的空间。所以相对定位一般结合绝对定位来使用，对设置了绝对定位的元素的父元素使用相对定位。
    - 绝对定位：设置为绝对定位的元素框从文档流完全删除，元素不保留原来的空间，并相对于其包含块进行定位。由于绝对定位的元素与文档流无关，所以它们可以覆盖页面上的其它元素。于是将元素引入三维坐标，通过定义元素在z轴上位置来控制元素的层叠堆放顺序。
    - 层叠：设置元素的层叠索引，基于同一个层叠上下文比较，可以看出元素#4,元素#5,元素#6都是元素#3的子元素，这三个子元素同属于一个层叠上下文，它们进行比较。然后它们和元素#3作为一个整体与元素#3的兄弟元素进行比较。所以可以看出尽管元素#4的z-index(层叠索引)为6，但是他还是在元素#1的下面。
  - 控制实现：
    - 在组件基础数据结构中加入Konva矩形图形对象，只设置矩形宽高数据以实现隐藏效果。动态添加echarts图表容器标签，并初始化位置及大小，通过监听隐形矩形对象的位置变化、图形变换来间接改变echarts图表容器位置及大小，最终实现跟随用户设置实时渲染数据组件。

## 案例2：组件拖拽及变换限制实现

- 案例分析：

- 案例需求：实现适用于各类型组件的拖拽（位移）和变换（旋转变换、缩放变换）限制功能。
- 案例描述：可以看到PPT上的两幅动图，简单来说所谓拖拽及变换限制就是组件位置不可超出可视区域，组件变换例如旋转，缩放也不可超出可视区域。
- 竞品实现方案：竞品1对于超出可视区域的组件未做处理，竞品2对组件超出可视区域后能正常显示。对比分析，竞品1的方案组件不能正常显示，当组件完全超出可视区域将造成组件丢失。竞品2的方案设置组件编辑时能超出可视区域，于场景正式发布后只显示可视区域内的内容，是一种良好的解决方案。
- 项目实施方案：项目一期采用限制组件不可超出可视区范围的方案，二期则考虑朝竞品2的方向优化。
  - 实施方案难点：方案难点在于处理多个组件同时拖拽变换边缘点数据计算问题。
  - 难点问题解决：计算组件位移及变换后的正盒子数据。

- 案例实现：

已知旋转盒的(x,y)坐标和旋转角度rad；计算拖拽变换后正位盒的四个顶点的坐标值，取四个点坐标值中最小的x, y作为正盒子的定位坐标点，最大x与最小x之间的差值作为正盒子的宽，高同理。调用Konva转换器类Transformer的边界限制函数判断正位盒的x,y坐标值是否为负，横纵坐标分别加宽高是否大于画布宽高，若是即超出可视区范围，返回原盒子坐标及大小。

### 函数介绍

1. 勾股定理：计算斜边长： $a^2 + b^2 = c^2$ ；
2. Math.atan2函数：atan2() 返回从原点(0,0)到(x,y)点的线段与x轴正方向之间的平面角度(弧度值)
3. 三角函数及其诱导公式： $\cos a = \text{邻边/斜边}$ ； $\sin a = \text{对边/斜边}$ ； $\cos(\pi/2 + a) = -\sin a$ ； $\sin(\pi/2 + a) = \cos a$ ；

```
function getCorner(pivotX, pivotY, diffX, diffY, angle) {
  const distance = Math.sqrt(diffX * diffX + diffY * diffY);
  angle += Math.atan2(diffY, diffX);

  const x = pivotX + distance * Math.cos(angle);
  const y = pivotY + distance * Math.sin(angle);
  return { x: x, y: y };
}

function getClientRect(rotatedBox) {
  const { x, y, width, height } = rotatedBox;
  const rad = rotatedBox.rotation;

  const p1 = getCorner(x, y, 0, 0, rad);
  const p2 = getCorner(x, y, width, 0, rad);
  const p3 = getCorner(x, y, width, height, rad);
  const p4 = getCorner(x, y, 0, height, rad);

  const minX = Math.min(p1.x, p2.x, p3.x, p4.x);
  const minY = Math.min(p1.y, p2.y, p3.y, p4.y);
  const maxX = Math.max(p1.x, p2.x, p3.x, p4.x);
  const maxY = Math.max(p1.y, p2.y, p3.y, p4.y);

  return {
    x: minX,
    y: minY,
    width: maxX - minX,
    height: maxY - minY,
  };
}

shape.setAbsolutePosition(newAbsPos);
```

## 个人理解与思考

从学生到职场人的转变不仅是角色身份的转变，更是思维心态的转变，今天我主要想谈谈思维的转变。常见的学生思维：追求准备充分不善于抓住机会、习惯等待安排而不主动。

1、不要等到万事俱备只欠东风，因为东风永远不会来。（勇于面对挑战，保持学习状态）

- 在学生时代，对能力的考核往往使用考试的方式，而对于考试我们往往做好了充分的准备去应对。实际上，在职场中难免会遇到一些自己不擅长的工作，作为职场小白很多时候总觉得要准备好了之后才能接手一些项目，以至于第一反应就是先拒绝，拒绝这个摆在眼前的机会。
- 为什么要说这是一个机会呢？是因为边学边做才是职场人的常态，在职场，不会给我们整块的时间让我们学好一门课再进行考试，很多事情都需要我们边做边学。
- 比如对于组态这个项目，我其实是没有学过它主要技术Konvajs的。当时我老师让我参与组态项目的开发的时候，我其实也说我不会Konvajs这门技术，也没做过类似的项目，潜

意思就是说我可能胜任不了，但总的来说老师的意思就是学，边做边学，我就这么被动上岗了，然事实证明，我确实将这个项目做下来了，在前辈的支持下不光学到了新的技术，还积累了做项目的经验，可以说这个过程中我收获良多。又比如作为一个不善言辞的前端码农，我其实极不擅长公众演讲，但同时也明白，无论是以后面对领导时的工作汇报，还是面对客户时产品介绍，都逃不开公众演讲。

- 总之，在这个产品更迭飞快，事事瞬息万变的时代，要学会主动学习，对不擅长的事，要勇于抓住机会先做，在做事的过程中总结经验教训，再不断优化，从而慢慢改进到完美的效果，不要等到万事俱备只欠东风，只因此时东风已然离你而去。

## 2、事事待人教，步步待安排（主动积极面对工作，善于提问，团队合作）

- 学生时代，一直是学校安排什么阶段学习什么知识，老师也无条件、无回报的主动传道受业解惑，我们只需顺应学校安排，学好老师传授的知识，最终取得良好的成绩就可以了。以至于进入职场后，我们习惯了等人来教来安排，或理所应当的被教被安排，比如负责的项目领导不过问就不主动推进，或者是领导安排一步做一步，又比如遇到问题前辈不教便不去讨教，或是一个问题不经自我思考，只想平白获取答案都是不正确的。
- 再比如组态这个项目前期我一直做着前辈分配给我的任务，这些任务简单且重复性大，做的多了让我感觉意义不大且无收获。后来对项目逐渐熟练上手后，我开始主动向前辈询问项目还有哪些功能模块还没有开发，开始主动向前辈讨要任务，并开始接触一些难度更高，业务更复杂的任务。在做这些任务的过程中，遇到了不少问题，但我习惯性的死磕，查资料查的天荒地老，最后与前辈一讨论，才豁然开朗。
- 有时，问题的答案在努力过后确实得不到解决，就要学会主动请教，自己不主动请教，没人会知道或理所当然的来为你解答。在不断遇到问题、分析问题和解决问题的过程中，才能真正收获到知识，积累到经验。以至于后来我甚至开始站在用户的角度出发，针对项目的一些功能开发提出我自己的观点，并与前辈讨论，一些想法荣幸的得到了前辈们的认可，从而应用到了实际。

## 总结

岗位职责：正确全面的理解自己的岗位职责，有助于减少工作的冲突与摩擦，有利于我们工作的顺利进展。

项目：web自由组态项目顺利结项，开发过程中，许多功能的实现需打破固有思维，才能获得解决之道。

思维：想快速适应职场生活，我们就要学会合作而不是自己“闭门造车”，要学会主动面对工作而不是“等安排”，要适应边学边做的职场节奏，而非把工作当成考试，非要做好万全准备才能出场。

感恩：入职以来，无论是工作上还是生活上，感恩英威腾给予了我颇多成长与温暖。也借此机会向领导、老师和部门同事表达感谢，感谢你们理解和帮助才使我得以快速成长。

致此，我的分享结束，感谢聆听，欢迎各位评委的提问。谢谢。

1. 1、项目前期参与产品构思，提出产品界面规划，把握产品最终界面实现效果。  
2、项目过程中收集和分析用户对于GUI的需求，根据产品原型图进行界面视觉设计和交互逻辑设计，对页面进行优化，使用户操作更趋于人性化。  
3、后期结合用户体验，优化完善设计，制定可行的产品品质提升方案。  
4、日常维护现有的应用产品，积累通用类软件或互联网应用产品的人机交互方面的理解和认识，具有敏锐的用户体验观察力，富有创新精神。↩
2. 1、项目前期参与项目会议、定义整体产品体验和技术方案。  
2、项目过程中负责前端编码技术规范制定，文档撰写及实现界面交互(即代码输出)，负责前端页面技术难点攻克及不同平台的兼容性调试，包括主流PC浏览器及手机浏览器的兼容。  
3、后期负责持续优化前端体验和页面响应速度，负责整体web前端用户体验及改进工作。  
4、日常维护和升级现有网站项目，快速定位并修复现有网站缺陷。也负责对Web与移动互联网前沿技术的持续研究与新技术创新。↩