**1.JS技术框架选型**

DOM操作+event操作【通用】：jQuery，zepto等（React和Vue是直接操作DOM，所以不需要DOM选择）

模块化开发加载器【通用】：requireJS，seaJS，LabJS等（webpack中已经包含模块化）

js模板【通用】：mustache，handelbars，doT ，underscore templte等

基础函数库【通用】：underscore或者公司自己内部的基础函数库（ajax，日期处理，类继承，常用正则，cookie处理，前端统计埋点等常用功能）

统一UI组件库【通用】：输入框，弹框，混动条，日历，多选框等常用UI组件（用公司内部统一的UI库，或者找jquery的ui组件库都行；ant design【vue和react组件库，业内通用】）

**2.CSS框架和预处理器选型**

css框架：BootStrap，Foundation等

css预处理器：sass，less等

**3.项目目录结构搭建，目录中最好包含“原始开发环境，转义环境，压缩环境”这三个文件夹**

src：作为本地开发的文件夹，里面分各个子文件夹来写页面的额业务

build：作为转以后的文件夹，因为src中可能用了jsx，es6等语法，需要通过babel转义成es5才能被绝大部分浏览器所理解,

build中存放的就是能在浏览器中直接访问的文件

dest：该文件夹是在转义好的build的文件的基础上，把里面的数据进行压缩后的文件

备注:因为现在转义工具比较多，好多语言都可以转化为js，所以添加一个build文件夹还是挺有必要的。如果大家都用正常的es5的语法来写，就没必要有build文件夹

浏览器如何访问压缩和未压缩的文件：

在url后面添加一个debug=1，表示访问未压缩文件，以便于前端调试

如何实现压缩和未压缩文件之前的切换(一般只有css 、js、html模板)**：**

方案1：找到页面的所有css和js链接，把连接中的dest路径改为build路径即可。（纯前端方案，缺点是debug=1的时候数据会加载2遍，但是因为是本地调试，所以也无所谓）

方案2：服务端判断url是否带有debug=1，有的话，设置一个服务端的htmPath变量（名字随便取），html中的js和css文件链接通过服务端变量htmPath来获取，例如{{htmlPath}}/libs/jquery.js;

**4.前端打包工具选择**

gulp 、grunt 、webpack：

grunt和gulp:做差不多的事情，都是文件的压缩，合并，内容处理等。Grunt功能比gulp强大许多。

webpack：模块化的打包工具，可以转义文件，模块化加载html，css，js；压缩html，css，js文件。

Node中间层作用：主要是为了服务端渲染，保证用户体验良好和SEO（没有很苛刻的加载体验要求或者SEO要求，就不需要在中间层添加Node）