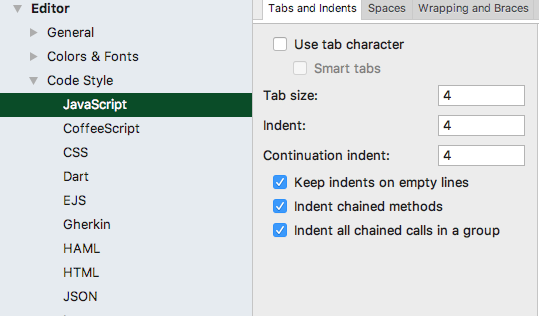
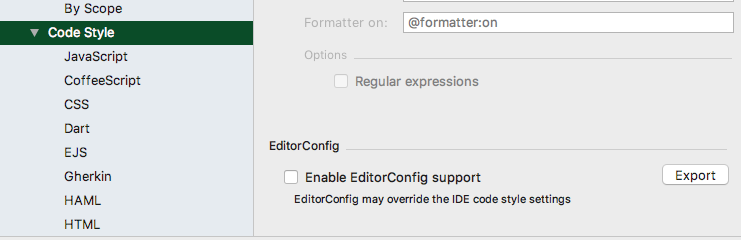
**Webstorm**

* Webstorm 缩进4格

1、将javascript的代码风格Tabsize改为4格



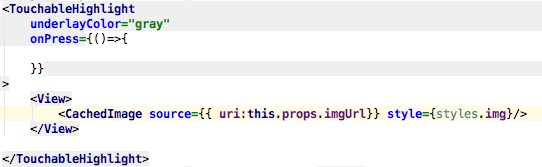
2、取消勾选IED默认风格复选框



**react-native**

* **TouchableHighlight 使用**

1. 自能有一个根子组件，且必须为RN原生组件，所以非原生组建应放在View中
2. 必须使用onPress方法，否则没有反馈



* RN中使用position:’absolute’ 后元素不会再占用任何空间
* **XMLHTTPRequest网络请求**

注：Fetch方法无法获取上传下载进度，因此用XMLHTTPRequest,xhr使用中服务器必须给出资源总大小(设置http头信息)，才能在客户端计算出下载百分比，服务端通过*unescape*(encodeURIComponent(data)).**length**方法获取文本字节数

RN代码：

httpRequest= (url,callback)=>｛  
 **var** xhr = **new** XMLHttpRequest();  
 xhr.open(**'POST'**,url,**true**);  
 xhr.setRequestHeader(**"Content-Type"**,**"text/html;charset=UTF-8"**)  
   
 xhr.timeout = 4000;  
 xhr.responseType = **'text'**;  
 xhr.timeout = (e)=>{  
 alert(**"网络缓慢,请求失败！"**);  
 }

*//下载进度*  
 **let** percent = 0  
 xhr.**onprogress** = (e)=>{ *//下载时50ms调用一次*  
 **if**(e.**lengthComputable**){ //长度是否可以计算,e.total已知才为true  
 percent = e.**loaded**/e.**total**;  
 callback(percent.toFixed(2),**null**) *//传递下载进度* }**else**{  
 }  
 }  
 xhr.onerror = (err)=>{}

//上传进度，参数同上面下载进度

xhr.upload.onprogress = (e)=>{}   
 xhr.**onreadystatechange** = (e) => {  
 **if** (xhr.**readyState** !== 4) {  
 **return**;  
 }  
 *//数据全部获取成功* **if** (xhr.**status** === 200) {  
 *//下载完成后传递下载结果*  
 **let** response = JSON.parse(xhr.**responseText**)  
 callback(1,response);  
 } **else** {  
 alert(**"请求失败 稍后再试！"**);  
 }  
 };  
 xhr.send();  
}  
***module***.exports = {httpRequest}

server代码：

**function** *getBook*(req,res){  
 *//验证登陆状态  
   
 //查询数据库* safetyBookModal  
 .find({**id**:req.**params**[**'id'**]},(err,result)=>{  
 **if**(err){  
 **let** myerr = JSON.stringify({**"code"**:501})  
 res.**end**(myerr)  
 **return**;  
 }  
 **let** data = {**"result"** : result}  
 data = JSON.stringify(data)  
 res.setHeader(**'content-type'**,**'text/plain'**);  
res.setHeader(**"ContentLength"**,*unescape*(encodeURIComponent(data)).**length**) *//文件总大小，必须发送给哭护短才能监听下载进度* res.**end**(data);  
 })  
   
}