

WebStorm配置代码模版

在开发中，经常会编写一些重复的代码，那么如何减少编写代码的重复和无趣呢？---使用WebStorm配置代码模版。

一：配置有两种方式(以mac配置为例)：

一：

1-：选择Preferences;

2-：搜索框输入Templates;

3-：点击File and Code Templates

4-：选中JavaScript File

5-：删除里面的代码，把以下代码复制进去(此代码可以自定义)：

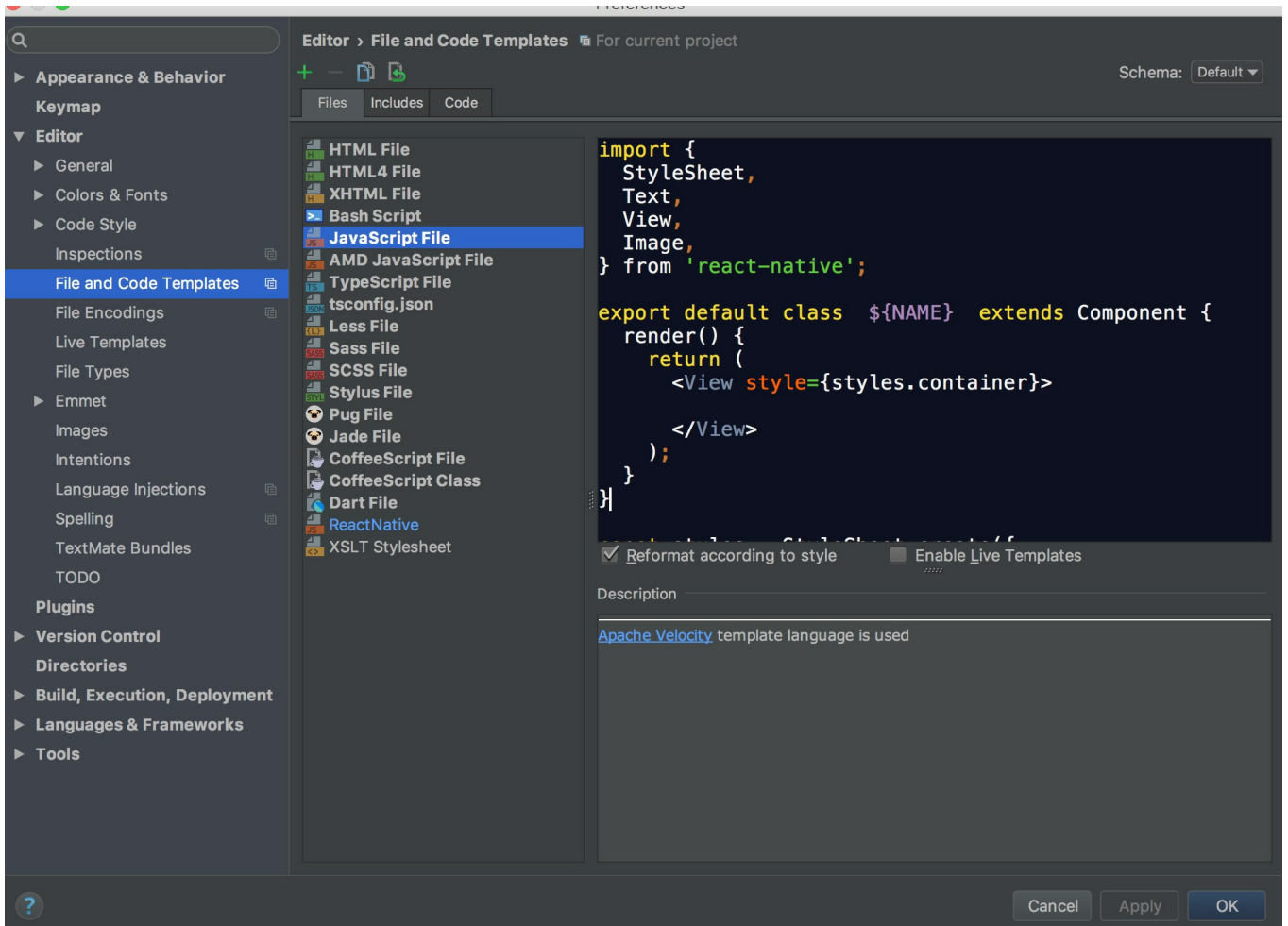
```
import React, { Component } from 'react';
import {
  StyleSheet,
  Text,
  View,
  Image,
} from 'react-native';

export default class ${NAME} extends Component {
  render() {
    return (
      <View style={styles.container}>

      </View>
    );
  }
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
  },
});
```

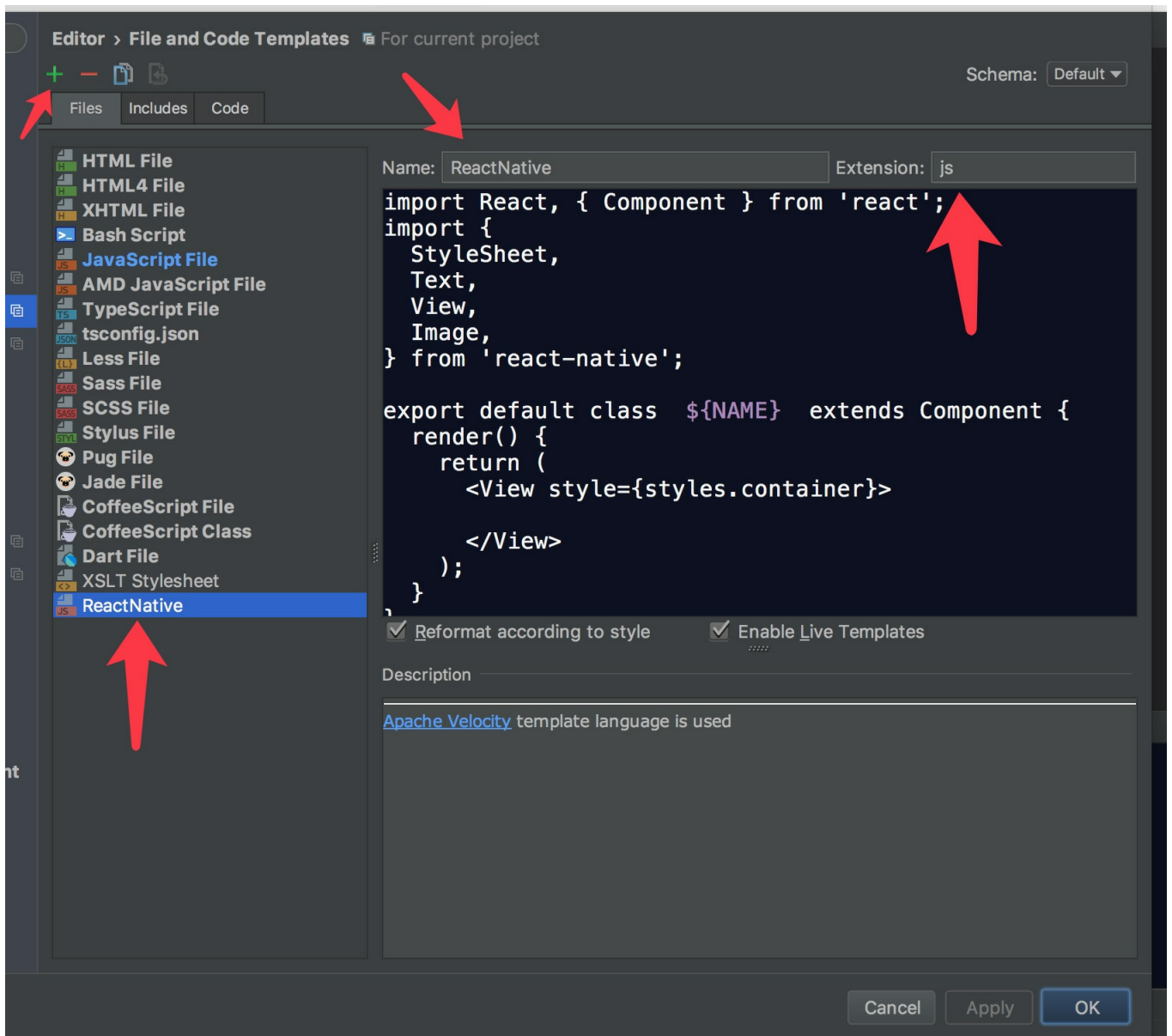
6-：点击Apply



二：

- 1-：选择Preferences;
- 2-：搜索框输入Templates;
- 3-：点击File and Code Templates
- 4-：点击左上角“+”号按钮

示例图如下：



注意点：

- 1-：注意新建文件的名称，本例子中取名为:React-Native;
- 2-：注意Extension的输入框填写为js;

提示：可根据不同的情况定义不同的代码模版，方便使用与更新。

三：使用方式：

分为两种，与上面配置方式——对应：

一：

- 1-：点击New，选择Java Script File；
- 2-：输入新建文件的名称，并点击OK；
- 3-：查看新建文件内容为填写的模版内容

二：

1-：点击New，选择React-Native文件类型(与使用第二中配置方式时新建的名字一一对应)；

2-：输入新建文件的名字，并点击OK；

3-：查看新建文件内容为填写的模版内容

三：附件(代码模版)

1-：

```
/**
 * Created by ${USER} on ${DATE}
 */

import React, { Component } from 'react';
import {
  StyleSheet,
  Text,
  View,
  Image,
} from 'react-native';

export default class ${NAME} extends Component {
  static navigationOptions = ({navigation, screenProps}) => ({
    headerTitle: '${NAME}',
  });

  constructor(props) {
    super(props);
    this.state = {
    };
  }

  render() {
    return (
      <View style={styles.container}>
        </View>
    );
  }
}

const styles = StyleSheet.create({
  container: {
```

```
        flex: 1,  
      },  
    });
```