# Node Pic\_code使用文档

## 准备工作：

1. 将middleware\_Pic\_code.js文件放入handle文件夹中，作为中间件
2. 将pic\_code文件夹（验证码底图库）放入images文件夹中

文件夹里面的数字命名的图片为验证码的底图

water.png为生成验证码的时候的水印图片

## 使用方法：

先参照github的README <https://github.com/TopuNet/Pic-code>

在要使用的路由文件中先引用此中间件

var pic\_code = require('../handle/middleware\_Pic\_code.js');

获取图片使用示例：

router.post("/pic\_code", pic\_code.Pic\_code, function (req, res) {

res.send(req.pic\_code\_result);

});

说明：

请求成功req.pic\_code\_result内容如下:

|  |  |
| --- | --- |
| **参数** | **说明** |
| error | 错误代码  ERROR - 失败，SUCCESS - 成功 |
| Y | 拼图位置纵坐标 |
| img1 | 小块拼图路径 |
| img2 | 大底图路径 |

请求失败req.pic\_code\_result内容如下:

|  |  |
| --- | --- |
| **参数** | **说明** |
| error | 错误代码  ERROR - 失败，SUCCESS - 成功 |
| msg | 失败原因 |

验证是否正确使用示例：

router.post("/valid\_pic\_code", function (req, res) {

var dix\_long = req.body["dix\_long"]; //横坐标

var valid\_range = req.body["valid\_range"]; //容错率

var result = {}; //验证结果

result['dix\_long'] = dix\_long;

result['valid\_range'] = valid\_range;

var session = req.session.pic\_code\_validcode;

if (session - valid\_range <= dix\_long && dix\_long <= session + valid\_range) {

result['error'] = 'SUCCESS';

} else {

result['error'] = 'ERROR';

}

res.send(result);

});

Ps:

Session名称：pic\_code\_validcode