

Bootstrap前端框架：快速开发 Web App 和站点的工具包

基于HTML、CSS、JavaScript 开发的简洁、直观、强悍的前端开发框架，使得 Web 开发更加快捷

Bootstrap中包含了丰富的Web组件，根据这些组件，可以快速的搭建一个漂亮、功能完备的网站。其中包括以下组件：

下拉菜单、按钮组、按钮下拉菜单、导航、[导航条](#)、[路径导航](#)、分页、排版、缩略图、警告对话框、进度条、媒体对象等

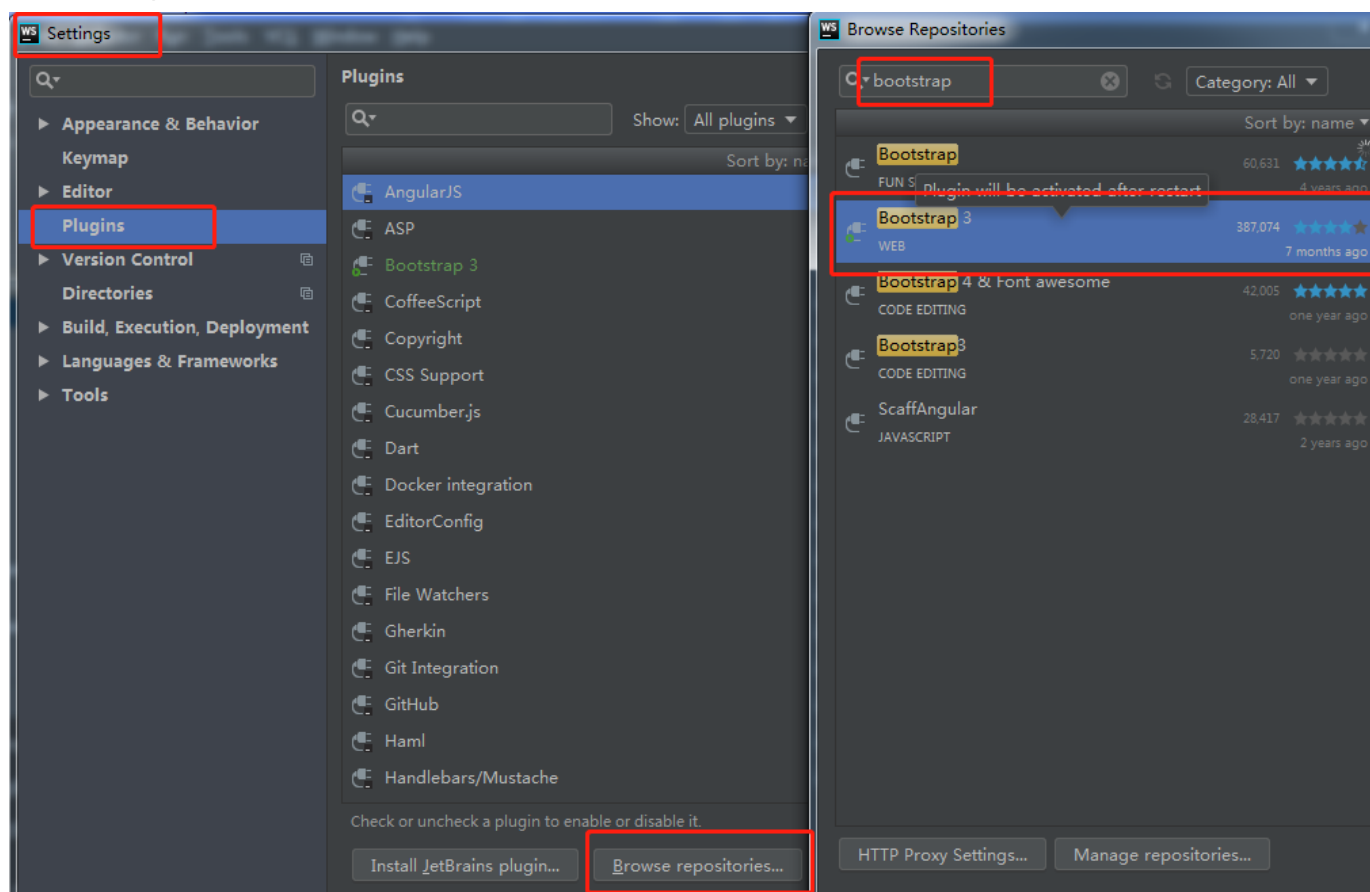
Bootstrap自带了13个jQuery插件，这些插件为Bootstrap中的组件赋予了“生命”。其中包括：

模式对话框、标签页、滚动条、弹出框等。

1、安装：npm i bootstrap@3

a、在项目的node_modules目录下找到bootstrap，在bootstrap里找到dist将dist这个目录复制到项目的public目录下

b、安装bootstrap插件：settings----->plugins----->Browse----->输入bootstrap----->选bootstrap3



2、在nodeJS中的使用：

使用的地方hbs文件中，通过Ctrl + J 来激活bootstrap插件

在hbs文件中激活后输入：**bs3-** 就可以出各种定义好的Web组件 如：**bs3-n** 就可以看到所需要的导航条

然后修改成自己所需要的即可使用。

multer文件上传

multer是一个中间件

- 1、安装：npm i multer -S
- 2、在需要使用的JS文件中引入：const multer = require('multer');
- 3、指定文件上传成功后保存的目录、获取上传文件名及扩展名

```
const storage = multer.diskStorage({  
  destination: function (req, file, cb) { // 存储目录  
    cb(null, 'public/images');  
  },  
  filename: function (req, file, cb) {  
    let extname = '';  
    switch (file.mimetype) {  
      case 'image/jpeg': extname = '.jpg'; break;  
      case 'image/png': extname = '.png'; break;  
      case 'image/gif': extname = '.gif'; break;  
    }  
    cb(null, Date.now() + extname);  
  }  
})
```

- 4、指定上传文件大小及上传文件类型

```
const upload = multer({  
  storage,  
  limits: { fileSize: 2 * 1024 * 1024 }, // 最大2M  
  fileFilter: function (req, file, cb) {  
    // 判断文件是否合法，合法则处理，不合法则拒绝  
    if (file.mimetype === 'image/gif' ||  
        file.mimetype === 'image/jpeg' ||  
        file.mimetype === 'image/png') {
```

```

    // 接收文件

    cb(null, true);

  } else {

    cb(new Error('请上传图片格式'), false);

  }

}

})

```

5、在用户提交上传请求后的处理

```

router.post('/gkk',upload.single('file'), async (req, res, next)=> {

  req.body.file = req.file ? req.file.filename:""; //判断是否有上传文件，如果就有名称

  const title = '公开课';

  try {

    const result = await query('INSERT INTO gkk SET ?', req.body);

    const message = result.affectedRows > 0 ? '新增成功' : '新增失败';

  }catch (err) {

    next(err);

  }

  res.render('admin/result',{

    layout: 'layout-admin' ,

    message

  });

});

```

表单数据验证 **express-validator**

1、安装：npm i -S express-validator

2、在需要使用的JS文件中引入：const {body, validationResult} = require('express-validator/check');

body:请求的数据

validationResult:校验的结果

3、获取表单效验并生成效验结果信息

```

const validations = [

  body('name').not().isEmpty().withMessage('公开课名称必填'),

```

```
body('description').not().isEmpty().withMessage('公开课描述信息必填'),
```

```
body('time').not().isEmpty().withMessage('公开课描述信息必填'),
```

//这里我更新后把这个约束禁了，否则影响正常更新了，如果觉得影响逻辑，可以将新增和更新拆开写

```
// .isAfter(new Date().toString()).withMessage('截止日期必须晚于当前时间'),
```

```
]
```

4、在路由中使用validations验证策略

```
router.post('/gkk',[upload.single('file'), ...validations], async (req, res, next)=> {
```

```
})
```

5、在路由中，效验所有错误并获取错误信息，返回的是一个数组

```
const errors = validationResult(req).formatWith(({msg})=>msg);
```