

乐字节教育高级架构课程

正所谓“授人以鱼不如授人以渔”，你们想要的 **Java 学习资料** 来啦！

不管你是学生，还是已经步入职场的同行，希望你们都要珍惜眼前的学习机会，奋斗没有终点，知识永不过时。

扫描下方二维码即可领取



乐字节晓啡

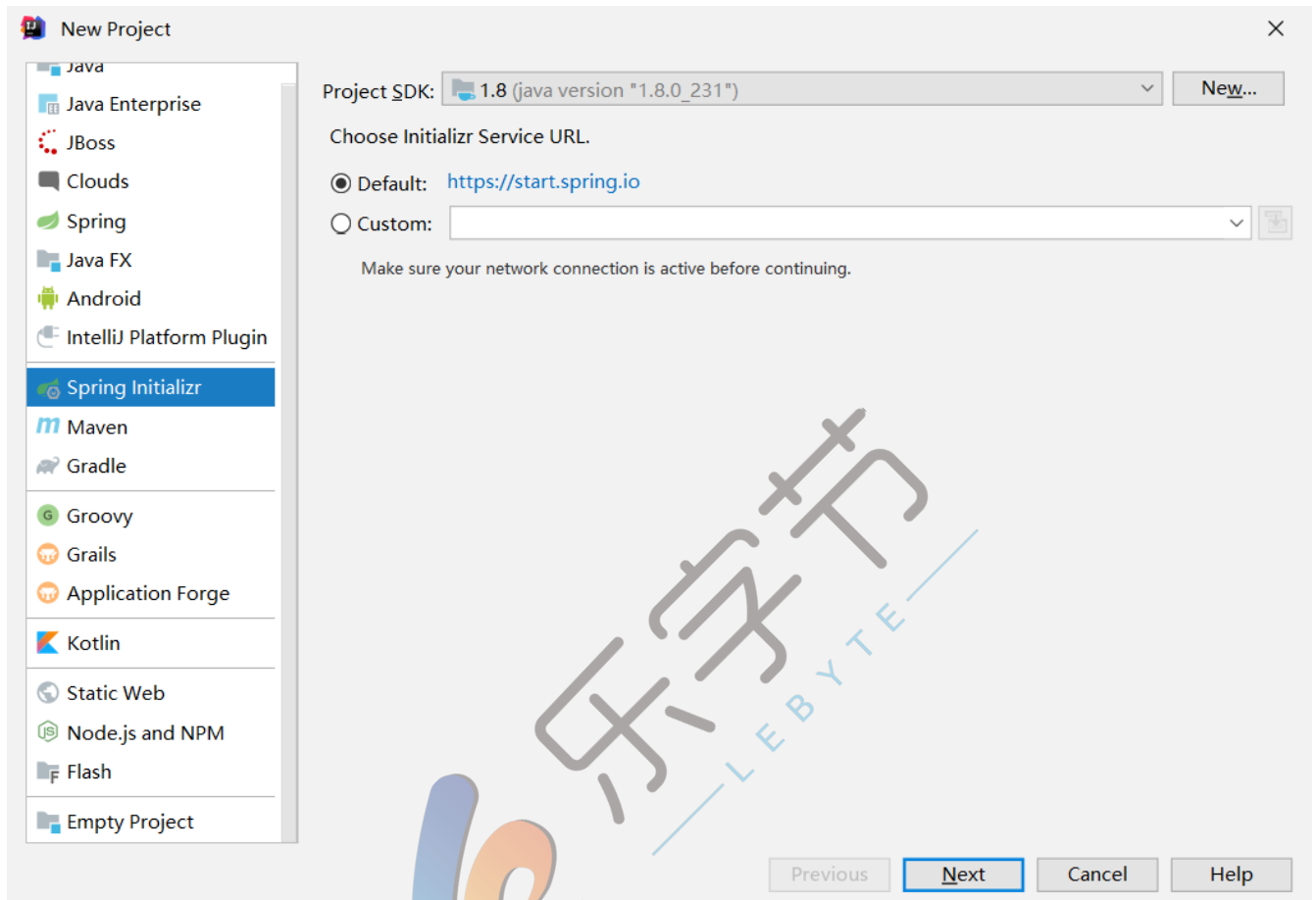



乐字节官方交流群



完整Demo

创建项目



 New Project ✕

Project Metadata

Group:

com.xxxx

Artifact:

uploaddemo

Type:

Maven Project (Generate a Maven based project archive.)

Language:

Java

Packaging:

Jar

Java Version:

8

Version:

0.0.1-SNAPSHOT

Name:

uploaddemo

Description:

Demo project for Spring Boot

Package:

com.xxxx.uploaddemo

Previous

Next

Cancel

Help

The screenshot shows the 'New Project' wizard in Spring Boot. The 'Web' dependency is selected under the 'Spring Web' category. The 'Selected Dependencies' panel on the right shows 'Web' and 'Spring Web'. The 'Next' button is highlighted.

添加依赖

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
  http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>2.2.2.RELEASE</version>
    <relativePath/> <!-- lookup parent from repository -->
  </parent>
  <groupId>com.xxxx</groupId>
  <artifactId>uploaddemo</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>uploaddemo</name>
  <description>Demo project for Spring Boot</description>

  <properties>
    <java.version>1.8</java.version>
  </properties>

  <dependencies>
    <!--web组件-->
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-web</artifactId>
    </dependency>
    <!--七牛云SDK-->
    <dependency>
      <groupId>com.qiniu</groupId>
      <artifactId>qiniu-java-sdk</artifactId>
      <version>[7.2.0, 7.2.99]</version>
    </dependency>
  </dependencies>
</project>
```

application.properties配置文件

```
#单个文件的最大上限
spring.servlet.multipart.max-file-size = 100MB
#单个请求的文件总大小上限
spring.servlet.multipart.max-request-size=1000MB
```

Upload.html前端上传页面

```
<!DOCTYPE html>
<html>
<head>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>oss文件上传</title>
</head>
<body>

<form method="post" action="/upload"
      enctype="multipart/form-data">
  <input type="file" name="file"/>
  <br/>
  <input type="submit" value="提交"/>
</form>

</body>
</html>
```

UploadService 上传服务类

UploadService.java

```
package com.xxxx.uploaddemo.service;

import java.io.InputStream;

/**
 * 上传服务
 */
public interface UploadService {
    /**
     * 上传服务
     * @param inputStream
     * @param fileName
     * @return
     */
    String upload(InputStream inputStream,String fileName);
}
```

UploadServiceImpl.java

```
package com.xxxx.uploaddemo.service.impl;

import com.qiniu.common.QiniuException;
import com.qiniu.http.Response;
import com.qiniu.storage.Configuration;
import com.qiniu.storage.Region;
import com.qiniu.storage.UploadManager;
import com.qiniu.util.Auth;
import com.xxxx.uploaddemo.service.UploadService;
import org.springframework.stereotype.Service;

import java.io.InputStream;
```

```
/**
```

```

* 上传服务实现类
*
* @author zhoubin
* @since 1.0.0
*/
@Service
public class UploadServiceImpl implements UploadService {

    @Override
    public String upload(InputStream inputStream, String fileName) {
        //构造一个带指定 Region 对象的配置类
        Configuration cfg = new Configuration(Region.region0());
        //...其他参数参考类注释
        UploadManager uploadManager = new UploadManager(cfg);
        //...生成上传凭证, 然后准备上传
        String accessKey = "CWApvs72aLOpYzpOhZ_fm7QSbx1-tsaXXw5nVuQ0";
        String secretKey = "yywbB12AvRWjYcpcnM-ehMmw5e-zxZD4U_h1V0wX";
        String bucket = "shop-test-demo01";
        //默认不指定key的情况下, 以文件内容的hash值作为文件名
        String key = fileName;
        String result = null;
        try {
            Auth auth = Auth.create(accessKey, secretKey);
            String upToken = auth.uploadToken(bucket);
            try {
                Response response = uploadManager.put(inputStream, key, upToken, null, null);
                //解析上传成功的结果
                if (response.statusCode == 200) {
                    result = "http://q63yxp74h.bkt.clouddn.com/" + fileName;
                }
            } catch (QiniuException ex) {
                Response r = ex.response;
                System.err.println(r.toString());
                try {
                    System.err.println(r.bodyString());
                } catch (QiniuException ex2) {
                    //ignore
                }
            }
        } catch (Exception ex) {
            //ignore
        }
        return result;
    }
}

```

UploadController类

```

package com.xxxx.uploaddemo.controller;

import com.xxxx.uploaddemo.service.UploadService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;

```

```
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.ResponseBody;
import org.springframework.web.multipart.MultipartFile;

import java.io.IOException;
import java.time.LocalDateTime;
import java.time.format.DateTimeFormatter;

/**
 * 上传Controller
 *
 * @author zhoubin
 * @since 1.0.0
 */
@Controller
public class UploadController {

    @Autowired
    private UploadService uploadService;

    @RequestMapping("upload")
    @ResponseBody
    public String upload(MultipartFile file) throws IOException {
        String filename = file.getOriginalFilename();
        String date =
DateTimeFormatter.ofPattern("yyyy/MM/dd/").format(LocalDateTime.now());
        filename =
date+System.currentTimeMillis()+filename.substring(filename.lastIndexOf("."));
        return uploadService.upload(file.getInputStream(), filename);
    }
}
```

测试

选择文件 1.jpg

提交



外链域名: [保存为默认域名](#)[↑ 上传文件](#)[🗑 删除文件](#)[🔄 刷新](#)

<input type="checkbox"/>	文件名	文件类型	存储类型	文件大小	最后更新	操作
<input type="checkbox"/>	2020/02/19/1582096153287.jpg	image/jpeg	标准存储	2.03 MB	2020-02-19 15:09:12	详情 更多

到底了

<http://q5xt67goc.bkt.clouddn.com/2020/02/19/1582096153287.jpg>

通过这个链接可以进行下载

这边只是使用流的方式简单实现文件上传，具体的其他功能可以在SDK文档中查阅

