

优极限

“极限教育，挑战极限”

www.yjxxt.com

极限教育，挑战极限。优极限是一个让 95% 的学生年薪过 18 万的岗前培训公司，让我们的学员具备优秀的互联网技术和职业素养，勇攀高薪，挑战极限。公司位于上海浦东，拥有两大校区，共万余平。累计培训学员超 3 万名。我们的训练营就业平均月薪 19000，最高年薪 50 万。

核心理念：让学员学会学习，拥有解决问题的能力，拿到高薪职场的钥匙。

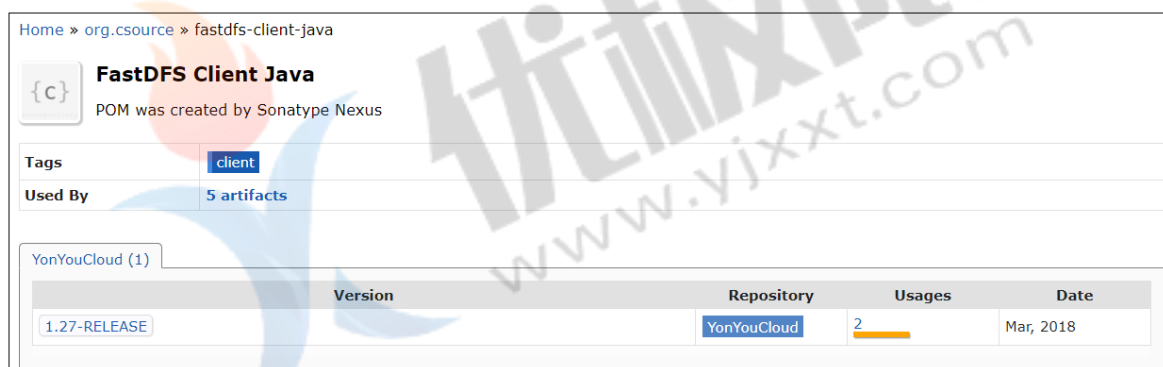
项目驱动式团队协作、一对一服务、前瞻性思维、教练式培养模型-培养你成为就业明星。首创的老学员项目联盟给学员充分的项目、技术支撑，利用优极限平台这根杠杆，不断挑战极限，勇攀高薪，开挂人生。

扫码关注优极限微信公众号：

（获取最新技术相关资讯及更多源码笔记）



FastDFS 的作者余庆先生已经为我们开发好了 Java 对应的 SDK。这里需要解释一下：作者余庆并没有及时更新最新的 Java SDK 至 Maven 中央仓库，目前中央仓库最新版仍旧是 1.27 版。所以我们需要通过 Github：<https://github.com/happyfish100/fastdfs-client-java> 下载项目源码，再通过命令 `mvn clean install` 编译打包导入 Maven 本地仓库使用。

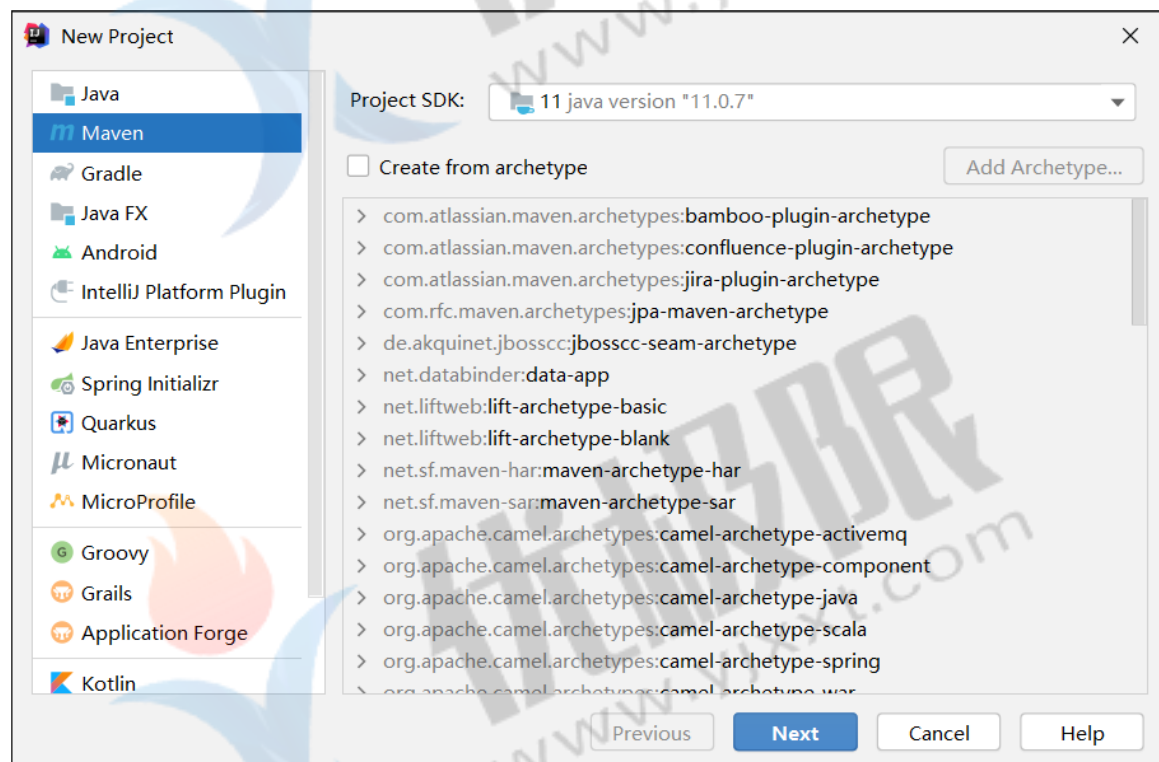


接下来我们通过 Java API 操作 FastDFS 实现文件的上传、下载、替换、删除、查询元数据、查询详情等功能。

文中案例已同步至：

- Github: <https://github.com/imrhelloworld/fastdfs-java>
- Gitee: <https://gitee.com/imrhelloworld/fastdfs-java>

创建项目



New Project

Name: fastdfs-java

Location: D:\IdeaProjects\example\fastdfs-java

▼ Artifact Coordinates

GroupId: org.example
The name of the artifact group, usually a company domain

ArtifactId: fastdfs-java
The name of the artifact within the group, usually a project name

Version: 1.0-SNAPSHOT

Previous Finish Cancel Help

添加依赖

在项目的 pom.xml 中添加以下依赖。因为我们需要一些常用工具包和单元测试，所以需要引入它们。

```
<!-- fastdfs java client -->
<dependency>
  <groupId>org.csource</groupId>
  <artifactId>fastdfs-client-java</artifactId>
  <version>1.29-SNAPSHOT</version>
</dependency>
<!-- apache commons lang3 工具包 -->
<dependency>
  <groupId>org.apache.commons</groupId>
  <artifactId>commons-lang3</artifactId>
  <version>3.11</version>
</dependency>
<!-- junit 单元测试 -->
<dependency>
  <groupId>junit</groupId>
  <artifactId>junit</artifactId>
  <version>4.13</version>
  <scope>test</scope>
</dependency>
```

编写配置文件

fdfs_client.conf

```
# 超时时间
connect_timeout = 10
network_timeout = 30
# 编码字符集
charset = UTF-8
# tracker 服务器 HTTP 协议下暴露的端口
http_tracker_http_port = 8080
# tracker 服务器的 IP 和端口
tracker_server = 192.168.10.101:22122
```

工具类

```
package org.example.client;

import org.apache.commons.lang3.StringUtils;
import org.csource.common.MyException;
import org.csource.common.NameValuePair;
import org.csource.fastdfs.*;

import java.io.*;

/**
 * FastDFS 分布式文件系统 Java 客户端工具类
 * 具体功能：文件上传、下载、替换、删除、查询文件元数据、查看文件详情
 */
public class FastDFSClient {

    // 获取配置文件地址
    private static final String CONF_FILENAME = Thread.currentThread()
        .getContextClassLoader().getResource("").getPath() +
        "fdfs_client.conf";

    // Storage 存储服务器客户端
    private static StorageClient storageClient = null;

    static {
        try {
            // 加载配置文件
            ClientGlobal.init(CONF_FILENAME);
            // 初始化 Tracker 客户端
            TrackerClient trackerClient = new
            TrackerClient(ClientGlobal.g_tracker_group);
            // 初始化 Tracker 服务端
            TrackerServer trackerServer = trackerClient.getTrackerServer();
            // 初始化 Storage 服务端
            StorageServer storageServer =
            trackerClient.getStoreStorage(trackerServer);
            // 初始化 Storage 客户端
            storageClient = new StorageClient(trackerServer, storageServer);
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

```

    } catch (MyException e) {
        e.printStackTrace();
    }
}

/**
 * 文件上传
 *
 * @param inputStream 上传的文件的字节输入流
 * @param fileName 上传的文件的原始名
 * @return
 */
public static String[] uploadFile(InputStream inputStream, String fileName)
{
    try {
        // 准备字节数组
        byte[] fileBuff = null;
        // 文件元数据
        NameValuePair[] metaList = null;
        if (inputStream != null) {
            // 查看文件的长度
            int len = inputStream.available();
            // 初始化元数据数组
            metaList = new NameValuePair[2];
            // 第一组元数据，文件的原始名称
            metaList[0] = new NameValuePair("file_name", fileName);
            // 第二组元数据，文件的长度
            metaList[1] = new NameValuePair("file_length",
String.valueOf(len));
            // 创建对应长度的字节数组
            fileBuff = new byte[len];
            // 将输入流中的字节内容，读到字节数组中
            inputStream.read(fileBuff);
        }
        /*
        上传文件。
        参数含义：要上传的文件的内容（使用字节数组传递），上传的文件的类型（扩展名），
元数据
        */
        String[] fileids = storageClient.upload_file(fileBuff,
getFileExt(fileName), metaList);
        return fileids;
    } catch (IOException e) {
        e.printStackTrace();
    } catch (MyException e) {
        e.printStackTrace();
    }
    return null;
}

/**
 * 文件上传
 *
 * @param file 上传的文件
 * @param fileName 上传的文件的原始名
 * @return
 */
public static String[] uploadFile(File file, String fileName) {

```

```

        try (FileInputStream fis = new FileInputStream(file)) {
            return uploadFile(fis, fileName);
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
        return null;
    }

    /**
     * 获取文件后缀名（不带点）
     *
     * @param fileName
     * @return 如: "jpg" or ""
     */
    private static String getFileExt(String fileName) {
        if (StringUtils.isBlank(fileName) || !fileName.contains(".")) {
            return "";
        }
        return fileName.substring(fileName.lastIndexOf(".") + 1); // 不带最后的点
    }

    /**
     * 获取文件详情
     *
     * @param groupName    组/卷名，默认值: group1
     * @param remoteFileName 文件名，例
     * 如: "M00/00/00/wKgKZl9tkTCAJAanAADhacZ_RF0495.jpg"
     * @return 文件详情
     */
    public static FileInfo getFileInfo(String groupName, String remoteFileName)
    {
        try {
            return storageClient.get_file_info(groupName == null ? "group1" :
            groupName, remoteFileName);
        } catch (IOException e) {
            e.printStackTrace();
        } catch (MyException e) {
            e.printStackTrace();
        }
        return null;
    }

    /**
     * 获取元数据
     *
     * @param groupName    组/卷名，默认值: group1
     * @param remoteFileName 文件名，例
     * 如: "M00/00/00/wKgKZl9tkTCAJAanAADhacZ_RF0495.jpg"
     * @return 文件的元数据数组
     */
    public static NameValuePair[] getMetaData(String groupName, String
    remoteFileName) {
        try {
            // 根据组名和文件名通过 storage 客户端获取文件的元数据数组
            return storageClient.get_metadata(groupName == null ? "group1" :
            groupName, remoteFileName);
        }
    }

```

```

    } catch (IOException e) {
        e.printStackTrace();
    } catch (MyException e) {
        e.printStackTrace();
    }
    return null;
}

/**
 * 文件下载
 *
 * @param groupName      组/卷名, 默认值: group1
 * @param remoteFileName 文件名, 例
如: "M00/00/00/wKgKZl9tkTCAJAanAADhacZ_RF0495.jpg"
 * @return 文件的字节输入流
 */
public static InputStream downloadFile(String groupName, String
remoteFileName) {
    try {
        // 根据组名和文件名通过 storage 客户端获取文件的字节数组
        byte[] bytes = storageClient.download_file(groupName == null ?
"group1" : groupName, remoteFileName);
        // 返回字节流对象
        InputStream inputStream = new ByteArrayInputStream(bytes);
        return inputStream;
    } catch (IOException e) {
        e.printStackTrace();
    } catch (MyException e) {
        e.printStackTrace();
    }
    return null;
}

/**
 * 文件删除
 *
 * @param groupName      组/卷名, 默认值: group1
 * @param remoteFileName 文件名, 例
如: "M00/00/00/wKgKZl9tkTCAJAanAADhacZ_RF0495.jpg"
 * @return 0为成功, 非0为失败
 */
public static int deleteFile(String groupName, String remoteFileName) {
    int result = -1;
    try {
        // 根据组名和文件名通过 storage 客户端删除文件
        result = storageClient.delete_file(groupName == null ? "group1" :
groupName, remoteFileName);
    } catch (IOException e) {
        e.printStackTrace();
    } catch (MyException e) {
        e.printStackTrace();
    }
    return result;
}

/**
 * 修改一个已经存在的文件
 *

```



```

    * @param oldGroupName 旧组名
    * @param oldFileName 旧文件名
    * @param file 新文件
    * @param fileName 新文件名
    * @return
    */
    public static String[] modifyFile(String oldGroupName, String oldFileName,
    File file, String fileName) {
        // 先上传
        String[] fileids = uploadFile(file, fileName);
        if (fileids == null) {
            return null;
        }
        // 再删除
        int delResult = deleteFile(oldGroupName, oldFileName);
        if (delResult != 0) {
            return null;
        }
        return fileids;
    }
}

```

测试

文件上传

```

// 文件上传
@Test
public void testUploadFile() {
    String[] fileids = FastDFSClient.uploadFile(new File("D:/china.jpg"),
    "china.jpg");
    for (String fileid : fileids) {
        System.out.println("fileid = " + fileid);
    }
}

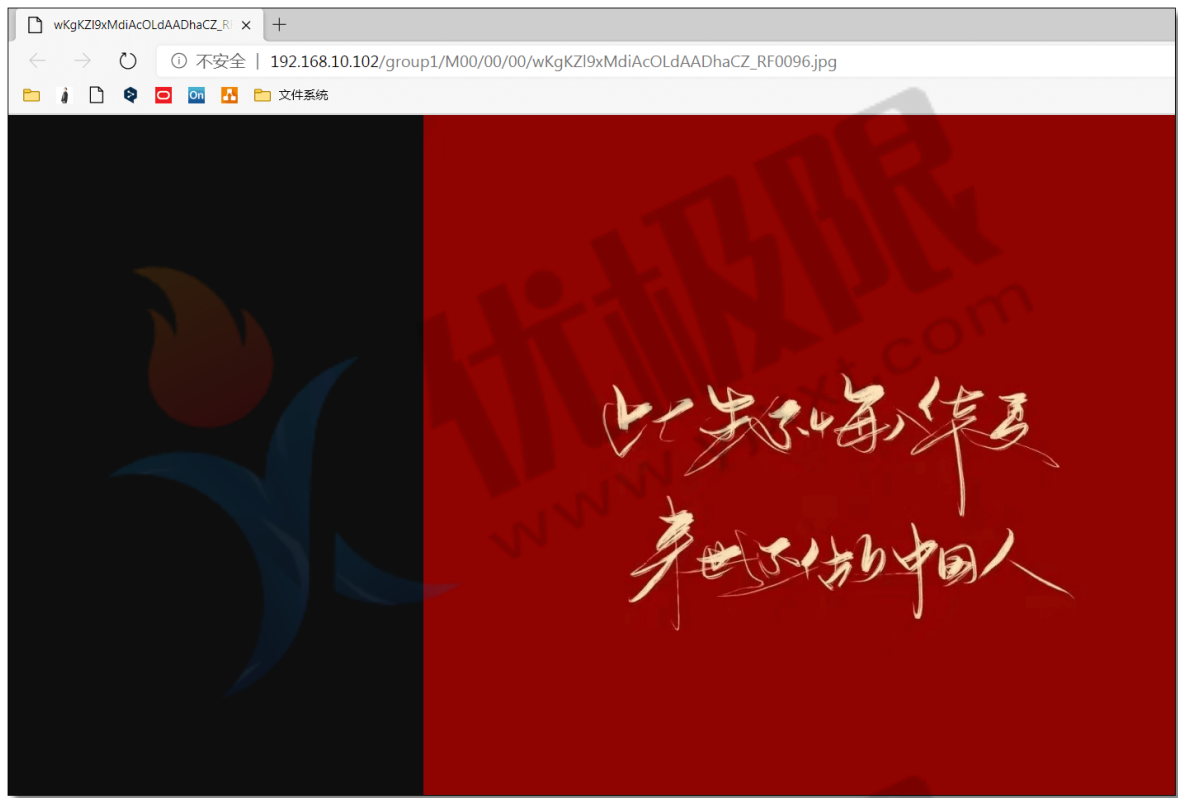
```

返回值:

```

fileid = group1
fileid = M00/00/00/wKgKZl9xMdiACOLdAADhacZ_RF0096.jpg

```

文件详情

```
// 查看文件详情
@Test
public void testGetFileInfo() {
    FileInfo fileInfo = FastDFSClient.getFileInfo("group1",
        "M00/00/00/wKgKZl9xMdiAcOLdAADhaCZ_RF0096.jpg");
    System.out.println("fileInfo = " + fileInfo);
}
```

返回值:

```
fileInfo = fetch_from_server = false, file_type = 1, source_ip_addr =
192.168.10.102, file_size = 57704, create_timestamp = 2020-09-28 08:44:08, crc32
= 645874781
```

文件元数据

```
// 获取文件数据
@Test
public void testGetMetaData() {
    NameValuePair[] metaDatas = FastDFSClient.getMetaData("group1",
        "M00/00/00/wKgKZ19xMdiACOLdAADhacZ_RF0096.jpg");
    for (NameValuePair metaData : metaDatas) {
        System.out.println(metaData.getName() + "---" + metaData.getValue());
    }
}
```

返回值:

```
file_length---57704
file_name---china.jpg
```

文件下载

```
// 文件下载
@Test
public void testDownloadFile() {
    InputStream is = FastDFSClient.downloadFile("group1",
        "M00/00/00/wKgKZ19xMdiACOLdAADhacZ_RF0096.jpg");
    try (FileOutputStream fos = new
        FileOutputStream("D:/wKgKZ19xMdiACOLdAADhacZ_RF0096.jpg")) {
        int len = 0;
        byte[] bytes = new byte[1024];
        while ((len = is.read(bytes)) != -1) {
            fos.write(bytes, 0, len);
            fos.flush();
        }
    } catch (FileNotFoundException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

文件删除

```
// 文件删除
@Test
public void testDeleteFile() {
    int result = FastDFSClient.deleteFile("group1",
        "M00/00/00/wKgKZ19xMdiACOLdAADhacZ_RF0096.jpg");
    System.out.println("result = " + result);
}
```

返回值:

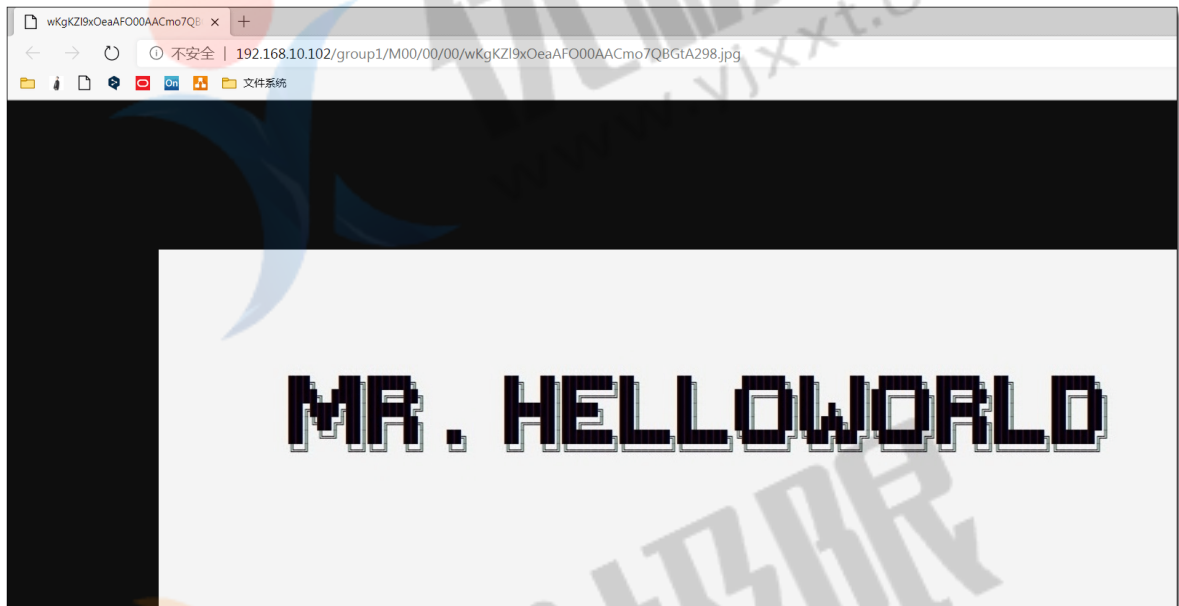
```
result = 0
```

文件替换

```
// 文件替换
@Test
public void testModifyFile() {
    String[] fileids = FastDFSClient.modifyFile("group1",
        "M00/00/00/wKgKZl9xOS2ASdu8AADhacZ_RF0898.jpg",
        new File("D:/mhw.jpg"),
        "mhw.jpg");
    for (String fileid : fileids) {
        System.out.println("fileid = " + fileid);
    }
}
```

返回值:

```
fileid = group1
fileid = M00/00/00/wKgKZl9xOeaAF000AACmo7QBgtA298.jpg
```



至此 Java 客户端操作 FastDFS 实现文件上传下载替换删除等操作就到这里, 下一篇我们带大家搭建 FastDFS 的集群环境, 多 Tracker 多 Storage 然后通过 Nginx 代理。