

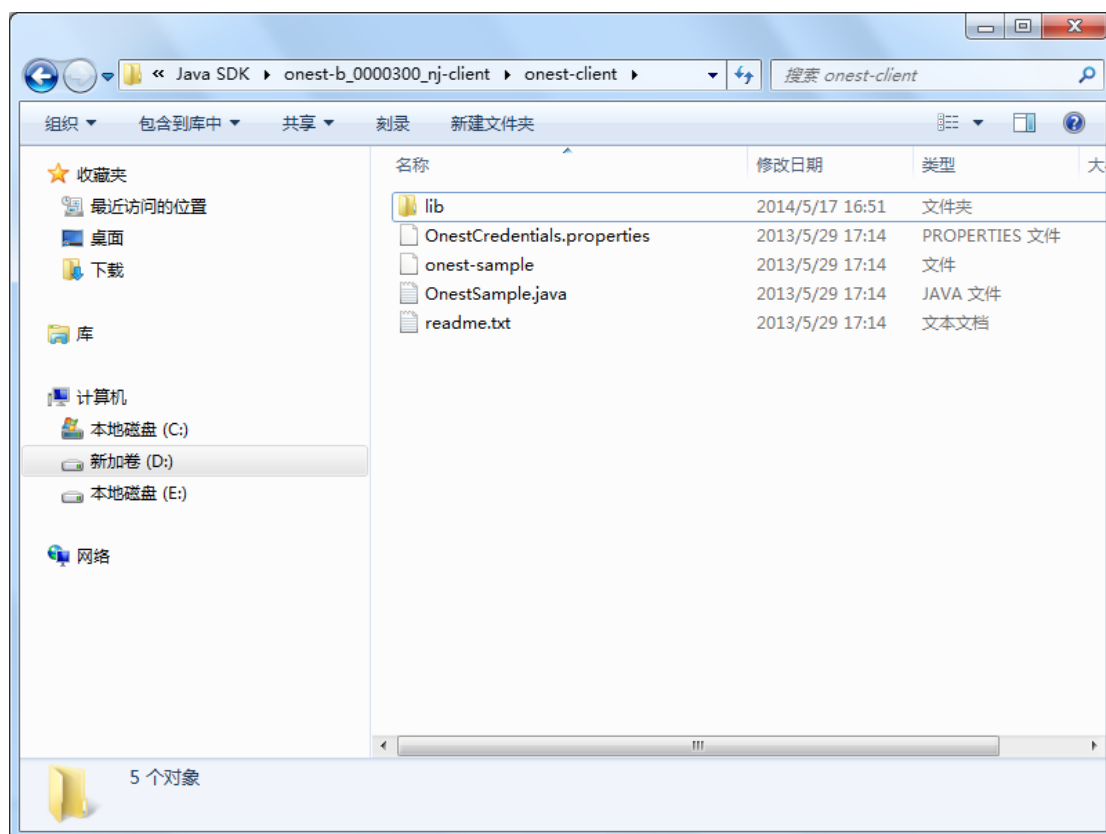
oNest Java SDK 快速开发手册

1、概述

本文主要介绍在 Eclipse 中使用 oNest Java SDK 的方法，同时给出一些常用接口的使用示例，力求为 Java 语言开发者提供快速入门指引。需要指出的是，本文中的示例并未涉及 SDK 中的全部接口，关于全部接口的详细说明和使用方法可参考《oNest Java ClientLib API Reference v1.1》。

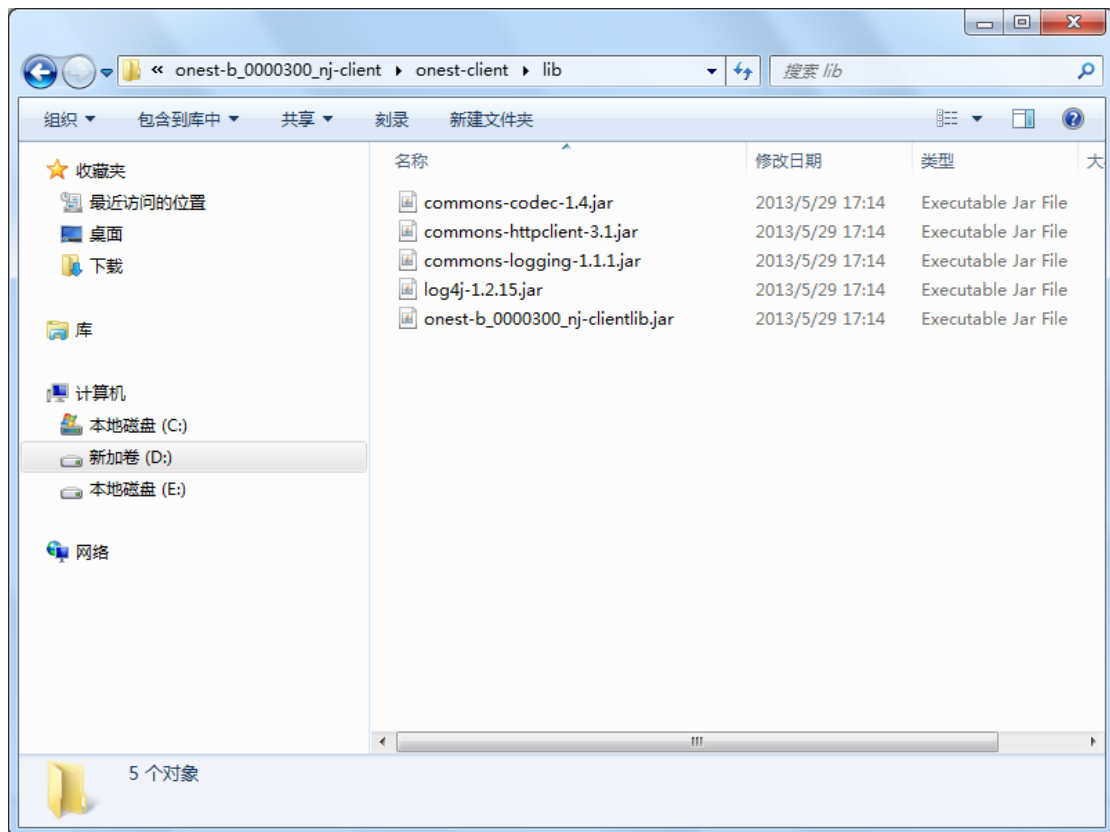
2、SDK 目录结构

在介绍 SDK 的使用方法之前，我们首先来看一下 SDK 中主要包括哪些内容。目前，SDK 的压缩包为 onest-b_0000300_nj-client.tar.gz，解压之后可以在文件夹 onest-client 看到如图一所示的内容。其中 OnestCredentials.properties 是 SDK 运行时需要使用的配置文件，OnestSample.java 是 SDK 自带的示例程序源文件，onest-sample 为示例程序的启动运行脚本，readme.txt 为 SDK 的简单说明文档。



图一 SDK 解压缩后的内容

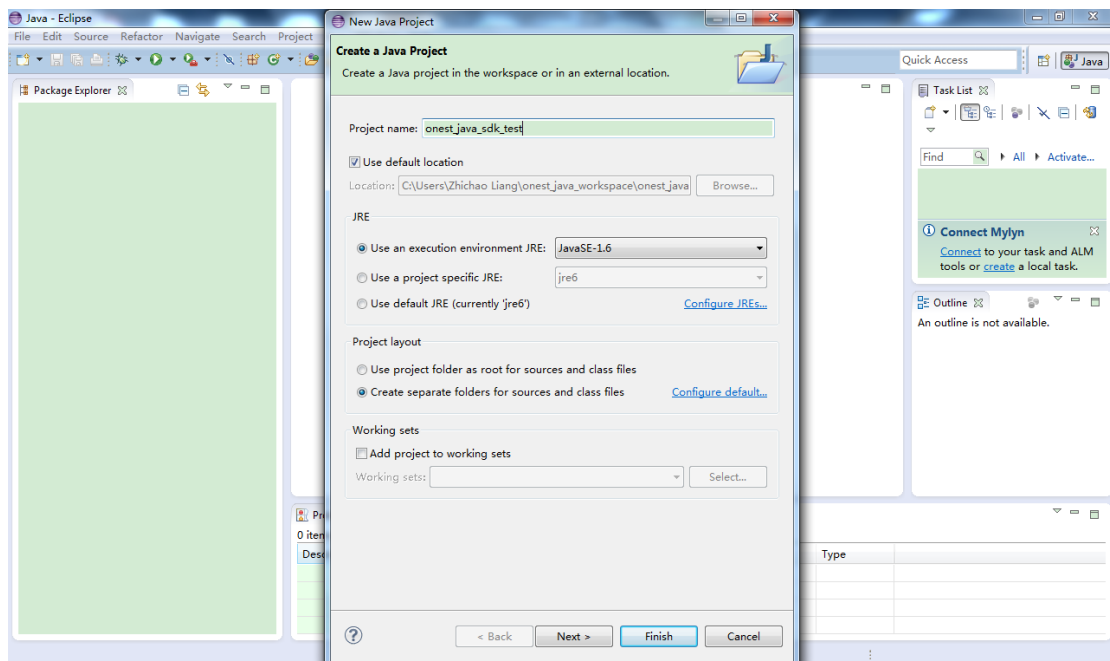
在 lib 目录中为 SDK 包含的所有库文件，如图二所示，onest-b_0000300_nj-clientlib.jar 即为 SDK 自身的 jar 包，其中提供了操作 oNest 的所有接口，其他文件则为 SDK 依赖的第三方 jar 包。



图二 目录 lib 中的内容

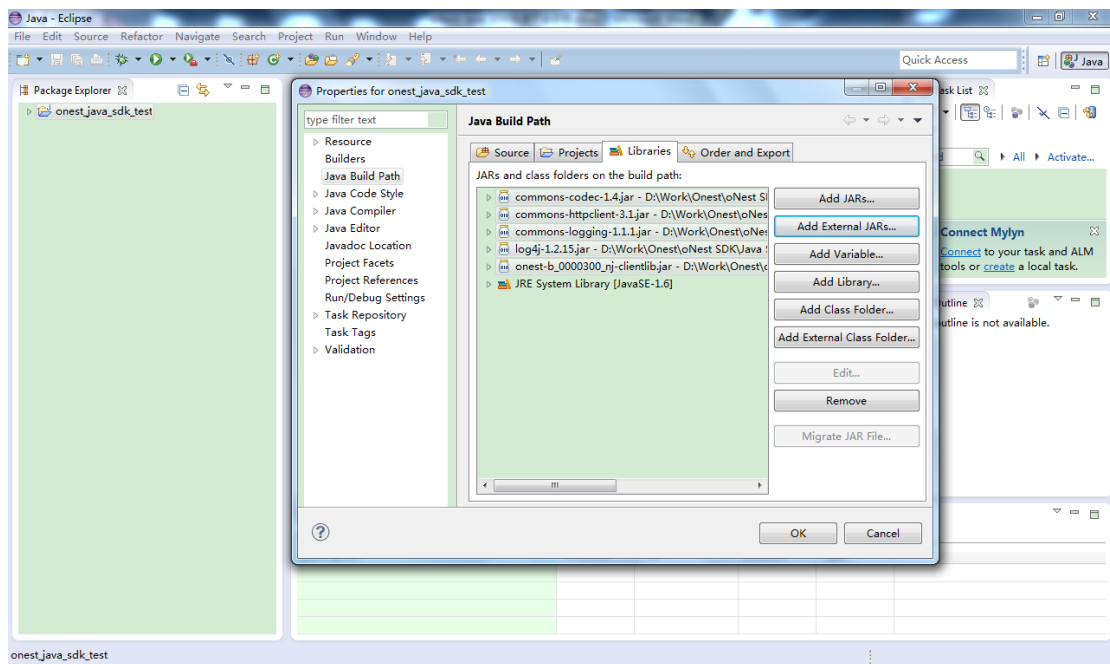
3、Eclipse 开发环境准备

在大致了解 SDK 包含的主要文件后，我们就来看一看如何在 Eclipse 中准备开发环境。首先，创建项目。打开 Eclipse 后通过如下方式 File->Java Project 创建一个新的 Java 项目，如图三所示，我们将该项目命名为 onest_java_sdk_test。



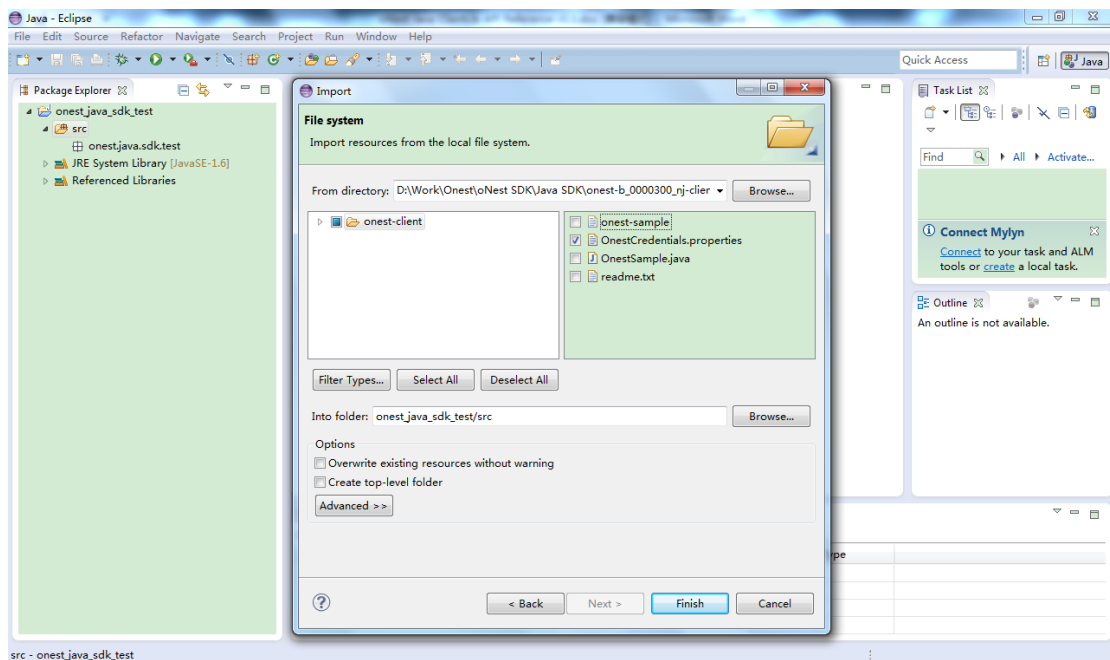
图三 创建新项目

其次，添加 jar 文件。通过右键项目 onest_java_sdk_test->Properties->Java Build Path->Libraries->Add External JARs，如图四所示，选择 SDK 中 lib 目录下的所有 jar 文件，将其添加到项目中。



图四 添加 jar 文件到项目

再次，导入配置文件 OnestCredentials.properties。通过右键 src->Import->General->File System，如图五所示，选择 SDK 中的 OnestCredentials.properties 文件，将其导入到项目中。



图五 导入配置文件

最后，配置文件 OnestCredentials.properties。打开 OnestCredentials.properties 文件，我们可以看到多个配置参数，仅有 AccessID、AccessSecretKey、hostAddr、userName 和 isIntraAddr 为必选的参数。前四个配置参数很容易理解，AccessID 和 AccessSecretKey 即为用户访问

oNest 系统对应的 AccessID 和 AccessSecretKey, hostAddr 为云存储管理组件地址, 而 userName 则为用户 ID, 如图六所示, 这些信息均可以通过中国移动公众服务云云存储控制台得到。



图六 示例用户控制台

最后一个配置参数 isIntraAddr 标明了 SDK 用户是否选择外网模式。这里的外网模式指的就是客户端和云存储系统不在同一个内网, 如果客户端和存储系统在同一个内网就是内网模式。若是外网模式 isIntraAddr 的值就为 true, 若是内网模式 isIntraAddr 的值就为 false。一般情况下, 公有云中使用时使用外网地址访问云存储的普通用户, isIntraAddr 均配置为 true 即可。以图六中给出的用户信息为例, 对应 OnestCredentials.properties 的文件内容如图七所示。



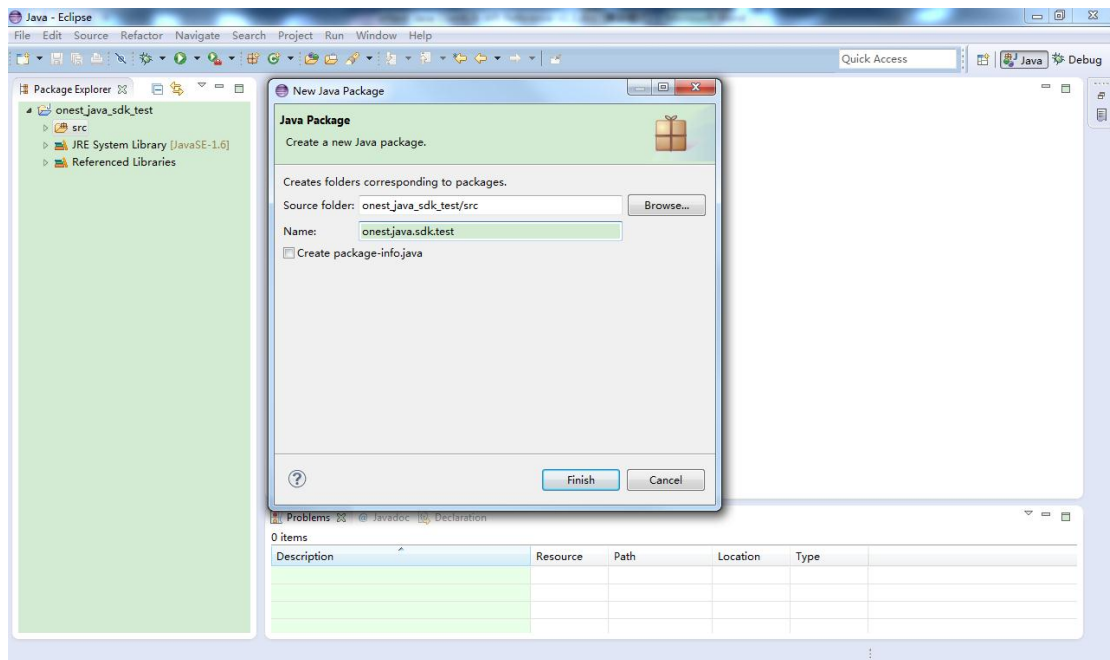
图七 OnestCredentials.properties 配置文件内容

4、oNest 客户端初始化

在完成 Eclipse 开发环境的配置之后, 我们就可以使用 SDK 进行云存储的应用开发了, 而开发的第一步就是初始化一个到 oNest 系统的客户端, 之后针对 oNest 的所有操作均通过该客户端来完成。oNest 客户端的初始化有两种方法, 一种是通过云存储管理组件地址初始化, 另一种是通过云存储服务地址初始化, 这两个地址分别对应的就是图六中的 SDK API 访问地址和 REST API 访问地址。这里推荐采用第一种方式来初始化 oNest 客户端, 本文档给出的所有代码示例也全部基于第一种初始化方法。第二种初始化方法可以参照文档《oNest Java ClientLib API Reference v1.1》。

在编写 oNest 客户端初始化代码前, 我们首先在 Eclipse 中创建一个 package, 将本文档中用到的所有代码都存放在这个 package 下。通过右键 src->New->Package 来创建 package,

如图八所示，我们将 package 命名为 onest.java.sdk.test。



图八 创建新 package

在创建完 package 后，我们在该 package 中创建一个 Java 类 InitializeOnestClient，其内容如示例一所示。在该类的 main 方法中，我们通过第一种方式初始化 oNest 客户端并通过该客户端列出 frankey 用户已创建的所有桶。

```
package onest.java.sdk.test;

import java.io.IOException;
import java.util.List;

import com.onest.auth.PropertiesCredentials;
import com.onest.client.Onest;
import com.onest.client.OnestClientFactory;
import com.onest.client.OnestClientException;
import com.onest.metainfo.Bucket;

public class InitializeOnestClient {
    public static void main(String[] args) throws IOException {
        Onest onest = null;
        /*获取oNest客户端工厂实例*/
        OnestClientFactory ocf = OnestClientFactory.getInstance();
        /*读取配置文件OnestCredentials.properties*/
        PropertiesCredentials onestProperites = new
        PropertiesCredentials(InitializeOnestClient.class.getResourceAsStream(".././.././../OnestCredentials
        .properties"));
        ocf.setFactoryInfo(onestProperites, onestProperites.getOnestHostAddr());
        /*读取配置文件中参数isIntraAddr和isIntraSrvLocation的参数值*/
    }
}
```

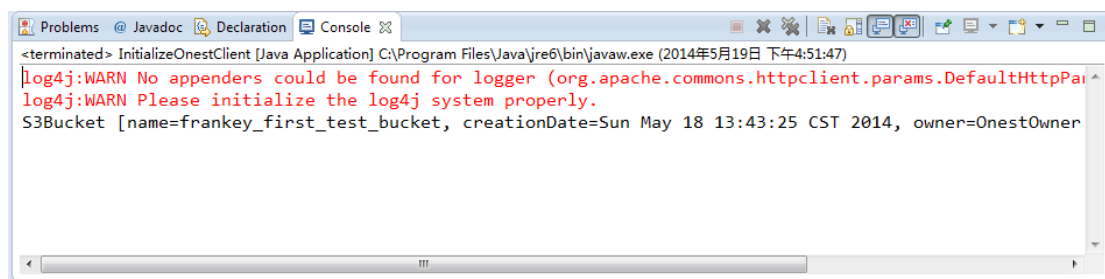
```

    boolean isIntraAddr = false;
    boolean isIntraSrvLocation = false;
    if(onestProperites.getIsIntraAddr() != null) {
        isIntraAddr = onestProperites.getIsIntraAddr().equals("true") ? true : false;
    }
    if(onestProperites.getIsIntraSvrAddr() != null) {
        isIntraSrvLocation = onestProperites.getIsIntraSvrAddr().equals("true") ? true :
false;
    }
    try {
        /*创建oNest客户端*/
        onest = ocf.getLocation(isIntraAddr, isIntraSrvLocation);
        /*通过oNest客户端获取用户frankey的所有容器并打印*/
        List<Bucket> buckets = onest.listBuckets();
        for(Bucket bucket : buckets) {
            System.out.println(bucket);
        }
    } catch (OnestClientException ce) {
        ce.printStackTrace();
    }
}
}

```

示例一 初始化 oNest 客户端并打印用户已有容器

在 Eclipse 中执行以上程序，图九给出了程序执行结果，从中可知我们成功初始化了 oNest 客户端并打印出了用户 frankey 已创建的所有容器。



图九 示例一的执行结果

5、主要接口使用示例

在掌握 oNest 客户端的初始化方法之后，我们就可以通过客户端提供的接口来实现对 oNest 系统的操作了。这里给出几种常用接口的使用代码示例，更多示例可参考文档《oNest Java ClientLib API Reference v1.1》。

5.1、 创建容器

用户在 oNest 系统存储数据的时候需要首先创建容器(bucket)，容器的名称需要全网唯一且容器的命名需要符合以下条件：

- 1、必须以字母和数字作为起始字符；
- 2、长度范围为[4-64]；
- 3、可以由大小写字母，数字，”_”构成。

需要注意的是容器一旦创建则无法重命名，所以要对容器重命名的话只能先删除该容器，然后再创建一个重命名后的新容器。示例二给出了创建容器的代码示例，其中为用户 **frankey** 新建了一个名为 **frankey_second_test_bucket** 的容器，之后获取用户 **frankey** 已创建的所有容器并打印。

```
package onest.java.sdk.test;

import java.io.IOException;
import java.util.List;

import com.onest.auth.PropertiesCredentials;
import com.onest.client.Onest;
import com.onest.client.OnestClientFactory;
import com.onest.client.OnestClientException;
import com.onest.metainfo.Bucket;

public class CreateBucket {
    public static void main(String[] args) throws IOException {
        Onest onest = null;
        /*获取oNest客户端工厂实例*/
        OnestClientFactory ocf = OnestClientFactory.getInstance();
        /*读取配置文件OnestCredentials.properties*/
        PropertiesCredentials onestProperites = new
PropertiesCredentials(CreateBucket.class.getResourceAsStream("../../../OnestCredentials.proper
ties"));
        ocf.setFactoryInfo(onestProperites, onestProperites.getOnestHostAddr());
        /*读取配置文件中参数isIntraAddr和isIntraSrvLocation的参数值*/
        boolean isIntraAddr = false;
        boolean isIntraSrvLocation = false;
        if(onestProperites.getIsIntraAddr() != null) {
            isIntraAddr = onestProperites.getIsIntraAddr().equals("true") ? true : false;
        }
        if(onestProperites.getIsIntraSvrAddr() != null) {
            isIntraSrvLocation = onestProperites.getIsIntraSvrAddr().equals("true") ? true :
false;
        }
        try {
            /*创建oNest客户端*/
            onest = ocf.getLocation(isIntraAddr, isIntraSrvLocation);
            /*通过oNest客户端为用户frankey创建一个新的容器，容器名为
frankey_second_test_bucket*/
        }
    }
}
```

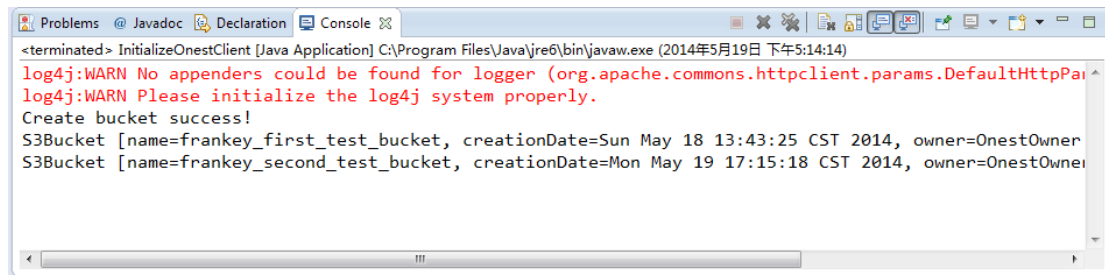
```

        if(null != onest.createBucket("frankey_second_test_bucket")) {
            System.out.println("Create bucket success!");
        }
        /*通过oNest客户端获取用户frankey的所有容器并打印*/
        List<Bucket> buckets = onest.listBuckets();
        for(Bucket bucket : buckets) {
            System.out.println(bucket);
        }
    } catch (OnestClientException ce) {
        ce.printStackTrace();
    }
}
}

```

示例二 创建新容器

在 Eclipse 中执行以上程序，图十给出了程序执行结果，从中可知我们成功创建了容器，新建的容器 frankey_second_test_bucket 已经可以打印出来。



图十 示例二的执行结果

5.2、 获取容器中对象列表

用户可以通过获取容器中对象列表接口来查询一个容器中包含的所有对象的列表，其中包含了每个对象的 Key、大小、创建时间等信息。示例三给出了获取容器中对象列表的代码示例，其中获取容器 frankey_first_test_bucket 中包含的所有对象的列表并打印所有对象名和对象大小。

```

package onest.java.sdk.test;

import java.io.IOException;
import java.util.List;

import com.onest.auth.PropertiesCredentials;
import com.onest.client.Onest;
import com.onest.client.OnestClientFactory;
import com.onest.client.OnestClientException;
import com.onest.metainfo.ObjectListing;
import com.onest.metainfo.OnestObjectSummary;

public class ListObjects {

```



```

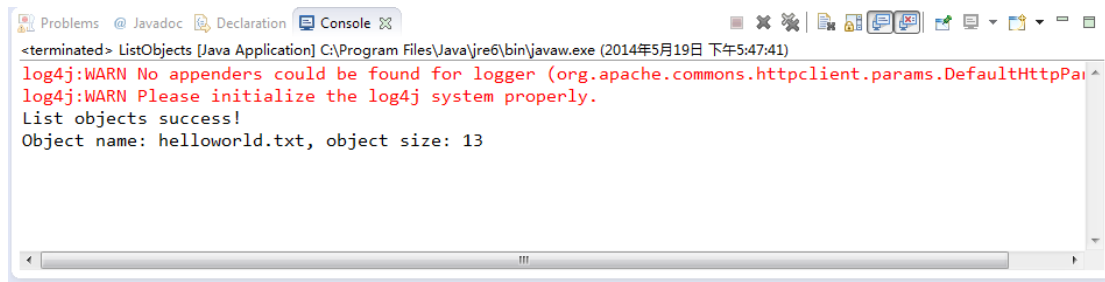
public static void main(String[] args) throws IOException {
    Onest onest = null;
    /*获取oNest客户端工厂实例*/
    OnestClientFactory ocf = OnestClientFactory.getInstance();
    /*读取配置文件OnestCredentials.properties*/
    PropertiesCredentials onestProperites = new
PropertiesCredentials(ListObjects.class.getResourceAsStream("../../../OnestCredentials.properties"));

    ocf.setFactoryInfo(onestProperites, onestProperites.getOnestHostAddr());
    /*读取配置文件中参数isIntraAddr和isIntraSrvLocation的参数值*/
    boolean isIntraAddr = false;
    boolean isIntraSrvLocation = false;
    if(onestProperites.getIsIntraAddr() != null) {
        isIntraAddr = onestProperites.getIsIntraAddr().equals("true") ? true : false;
    }
    if(onestProperites.getIsIntraSvrAddr() != null) {
        isIntraSrvLocation = onestProperites.getIsIntraSvrAddr().equals("true") ? true :
false;
    }
    try {
        /*创建oNest客户端*/
        onest = ocf.getLocation(isIntraAddr, isIntraSrvLocation);
        /*获取容器frankey_first_test_bucket中的所有对象的列表*/
        ObjectListing objectList = onest.listObjects("frankey_first_test_bucket");
        if(null != objectList) {
            System.out.println("List objects success!");
        }
        /*获取所有对象的名称和大小并打印*/
        List<OnestObjectSummary> objects = objectList.getObjectSummaries();
        for(OnestObjectSummary object : objects) {
            System.out.println("Object name: " + object.getKey() + ", object size: " +
object.getSize());
        }
    } catch (OnestClientException ce) {
        ce.printStackTrace();
    }
}

```

示例三 获取容器中对象列表

在 Eclipse 中执行以上程序，图十一给出了程序执行结果，从中可知我们成功获取了容器 frankey_first_test_bucket 中的对象列表，打印出了对象的 Key 和对象大小。



图十一 示例三的执行结果

5.3、 创建对象

对象是用户存储在 oNest 系统中数据的最小单位，通常是一个文件。对象存储在 bucket 中，对象有 Key，数据和元数据信息，对象的 Key 在容器中是唯一的。示例四给出了创建对象的代码示例，其中上传一个本地文件到容器 frankey_first_test_bucket 中，之后通过获取容器 frankey_first_test_bucket 中的对象列表来确认文件是否上传成功。

```

Package onest.java.sdk.test;

import java.io.File;
import java.io.IOException;
import java.util.List;

import com.onest.auth.PropertiesCredentials;
import com.onest.client.Onest;
import com.onest.client.OnestClientFactory;
import com.onest.client.OnestClientException;
import com.onest.metainfo.ObjectListing;
import com.onest.metainfo.OnestObjectSummary;
import com.onest.metainfo.PutObjectResult;

public class CreateObject {
    public static void main(String[] args) throws IOException {
        Onest onest = null;
        /*获取oNest客户端工厂实例*/
        OnestClientFactory ocf = OnestClientFactory.getInstance();
        /*读取配置文件OnestCredentials.properties*/
        PropertiesCredentials onestProperites = new
        PropertiesCredentials(CreateObject.class.getResourceAsStream("../OnestCredentials.properties"));

        ocf.setFactoryInfo(onestProperites, onestProperites.getOnestHostAddr());
        /*读取配置文件中参数isIntraAddr和isIntraSrvLocation的参数值*/
        boolean isIntraAddr = false;
        boolean isIntraSrvLocation = false;
        if(onestProperites.getIsIntraAddr() != null) {
            isIntraAddr = onestProperites.getIsIntraAddr().equals("true") ? true : false;

```

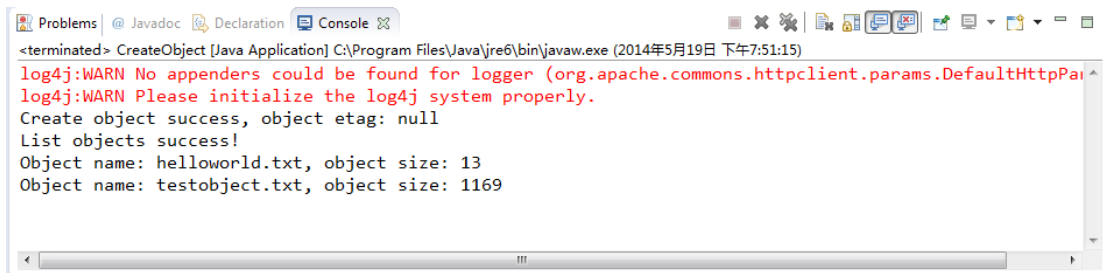
```

    }
    if(onestProperites.getIsIntraSvrAddr() != null) {
        isIntraSrvLocation = onestProperites.getIsIntraSvrAddr().equals("true") ? true :
false;
    }
    try {
        /*创建oNest客户端*/
        onest = ocf.getLocation(isIntraAddr, isIntraSrvLocation);
        /*打开一个本地文件testobject.txt*/
        File file = new File("C:/Users/Zhichao Liang/testobject.txt");
        /*将文件testobject.txt上传到容器frankey_first_test_bucket中*/
        PutObjectResult result = onest.putObject("frankey_first_test_bucket",
"testobject.txt", file);
        if(null != result) {
            System.out.println("Create object success, object etag: " + result.getETag());
        }
        /*获取容器frankey_first_test_bucket中的所有对象的列表*/
        ObjectListing objectList = onest.listObjects("frankey_first_test_bucket");
        if(null != objectList) {
            System.out.println("List objects success!");
        }
        /*获取所有对象的名称和大小并打印*/
        List<OnestObjectSummary> objects = objectList.getObjectSummaries();
        for(OnestObjectSummary object : objects) {
            System.out.println("Object name: " + object.getKey() + ", object size: " +
object.getSize());
        }
    } catch (OnestClientException ce) {
        ce.printStackTrace();
    }
}
}

```

示例四 创建对象

在 Eclipse 中执行以上程序，图十二给出了程序执行结果，从中可知我们成功上传了本地文件到容器 frankey_first_test_bucket 中，该对象大小为 1169 个字节。



```

<terminated> CreateObject [Java Application] C:\Program Files\Java\jre6\bin\javaw.exe (2014年5月19日 下午7:51:15)
log4j:WARN No appenders could be found for logger (org.apache.commons.httpclient.params.DefaultHttpPar
log4j:WARN Please initialize the log4j system properly.
Create object success, object etag: null
List objects success!
Object name: helloworld.txt, object size: 13
Object name: testobject.txt, object size: 1169

```

图十二 示例四的执行结果

5.4、 读取对象

用户可以通过指定容器名和对象名从 oNest 系统读取对象的内容，示例五给出了读取对象的代码示例，其中我们从容器 `frankey_first_test_bucket` 读取在示例四中上传的文件 `testobject.txt`。

```
package onest.java.sdk.test;

import java.io.IOException;
import java.io.InputStream;

import com.onest.auth.PropertiesCredentials;
import com.onest.client.Onest;
import com.onest.client.OnestClientFactory;
import com.onest.client.OnestClientException;
import com.onest.metainfo.OnestObject;

public class GetObject {
    public static void main(String[] args) throws IOException {
        Onest onest = null;
        /*获取oNest客户端工厂实例*/
        OnestClientFactory ocf = OnestClientFactory.getInstance();
        /*读取配置文件OnestCredentials.properties*/
        PropertiesCredentials onestProperites = new
PropertiesCredentials(GetObject.class.getResourceAsStream("../../../OnestCredentials.properties
"));
        ocf.setFactoryInfo(onestProperites, onestProperites.getOnestHostAddr());
        /*读取配置文件中参数isIntraAddr和isIntraSrvLocation的参数值*/
        boolean isIntraAddr = false;
        boolean isIntraSrvLocation = false;
        if(onestProperites.getIsIntraAddr() != null) {
            isIntraAddr = onestProperites.getIsIntraAddr().equals("true") ? true : false;
        }
        if(onestProperites.getIsIntraSvrAddr() != null) {
            isIntraSrvLocation = onestProperites.getIsIntraSvrAddr().equals("true") ? true :
false;
        }
        try {
            /*创建oNest客户端*/
            onest = ocf.getLocation(isIntraAddr, isIntraSrvLocation);
            /*读取容器frankey_first_test_bucket中的对象testobject.txt*/
            OnestObject object = onest.getObject("frankey_first_test_bucket", "testobject.txt");
            if(null != object) {
                System.out.println("Get object success!");
            }
        }
    }
}
```

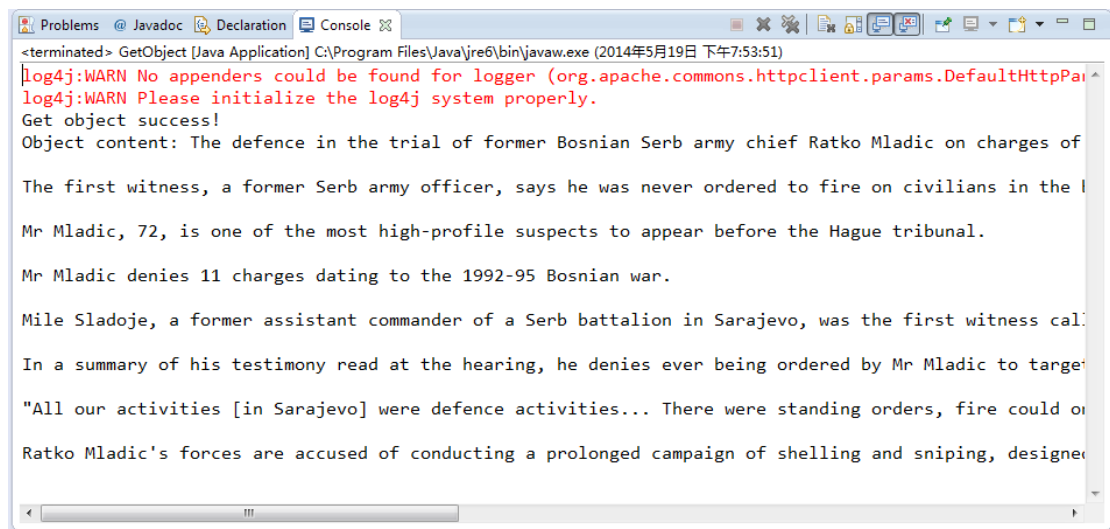
```

        /*获取对象数据输入流*/
        InputStream in = object.getObjectContent();
        /*分配字节数组用于读取对象数据*/
        byte objectdata[] = new
byte[(int)object.getObjectMetadata().getLength();
        /*读取对象数据*/
        in.read(objectdata);
        /*打印对象内容*/
        System.out.println("Object content: " + new String(objectdata));
    } catch (OnestClientException ce) {
        ce.printStackTrace();
    }
}
}

```

示例五 读取对象

在 Eclipse 中执行以上程序，图十三给出了程序执行结果，我们成功读取了对象 testobject.txt 中的全部内容。



图十三 示例五的执行结果

5.5、 删除对象

用户可以通过指定容器名和对象名从 oNest 系统删除一个对象，示例六给出了删除对象的代码示例，其中我们从容器 `frankey_first_test_bucket` 删除在示例四中上传的文件 `testobject.txt`，之后通过获取容器 `frankey_first_test_bucket` 中的对象列表来确认文件是否删除成功。

```

package onest.java.sdk.test;

import java.io.IOException;
import java.util.List;

```

```

import com.onest.auth.PropertiesCredentials;
import com.onest.client.Onest;
import com.onest.client.OnestClientFactory;
import com.onest.client.OnestClientException;
import com.onest.metainfo.ObjectListing;
import com.onest.metainfo.OnestObjectSummary;

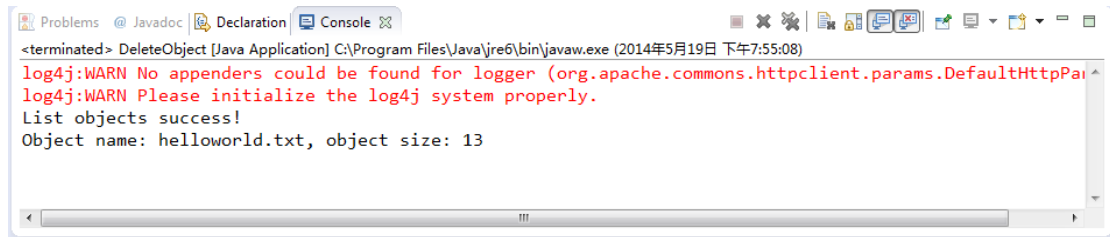
public class DeleteObject {
    public static void main(String[] args) throws IOException {
        Onest onest = null;
        /*获取oNest客户端工厂实例*/
        OnestClientFactory ocf = OnestClientFactory.getInstance();
        /*读取配置文件OnestCredentials.properties*/
        PropertiesCredentials onestProperites = new
PropertiesCredentials(DeleteObject.class.getResourceAsStream("../../OnestCredentials.properties"));
        ocf.setFactoryInfo(onestProperites, onestProperites.getOnestHostAddr());
        /*读取配置文件中参数isIntraAddr和isIntraSrvLocation的参数值*/
        boolean isIntraAddr = false;
        boolean isIntraSrvLocation = false;
        if(onestProperites.getIsIntraAddr() != null) {
            isIntraAddr = onestProperites.getIsIntraAddr().equals("true") ? true : false;
        }
        if(onestProperites.getIsIntraSvrAddr() != null) {
            isIntraSrvLocation = onestProperites.getIsIntraSvrAddr().equals("true") ? true :
false;
        }
        try {
            /*创建oNest客户端*/
            onest = ocf.getLocation(isIntraAddr, isIntraSrvLocation);
            /*从容器frankey_first_test_bucket中删除对象testobject.txt*/
            onest.deleteObject("frankey_first_test_bucket", "testobject.txt");
            /*获取容器frankey_first_test_bucket中的所有对象的列表*/
            ObjectListing objectList = onest.listObjects("frankey_first_test_bucket");
            if(null != objectList) {
                System.out.println("List objects success!");
            }
            /*获取所有对象的名称和大小并打印*/
            List<OnestObjectSummary> objects = objectList.getObjectSummaries();
            for(OnestObjectSummary object : objects) {
                System.out.println("Object name: " + object.getKey() + ", object size: " +
object.getSize());
            }
        } catch (OnestClientException ce) {

```

```
        ce.printStackTrace();
    }
}
}
```

示例六 删除对象

在 Eclipse 中执行以上程序，图十四给出了程序执行结果，我们成功删除了对象 testobject.txt。



图十四 示例六的执行结果