

许可证 License

1 引言

1.1 编写目的

该组件为软件复用课程的课程项目，撰写此文档的目的在于使得用户能够更好的复用该组件。

2 组件简介

2.1 功能

该构件可让用户设置系统证书资源的上限，重新设置证书资源上限之后，构件使用数将重新开始计数。注：设置资源上限时，上限值只能是正数（大于 0 的正整数）；若设置为 0 或负数，那么设置操作将失败。

2.2 输入

应用程序的性能指标

2.3 输出

根据 license 中设置的规则，返回相应的结果

2.4 使用方法

首先我们需要获得 LicenseManager 实例，代码如下

```
LicenseManager licenseManager = LicenseManager.getInstance();
```

客户端需要将自身的一些信息填入 CallerMessage 类中（当前类中仅包含客户端 ID 字段信息），使得系统了解证书的申请者信息，操作代码如下：

```
CallerMessage callerMessage = new CallerMessage("THFERDXEWS-98754456");
```

```
// THFERDXEWS-98754456 为申请证书资源客户端的 ID 号。
```

通过 `LicenseManager` 向系统发起证书申请，代码如下：

```
RequestResultMessage rrm = licenseManager.requestLicense(callerMessage);
```

另外我们还可以对系统证书资源的上限值做一些改变，调用方法如下：

```
LicenseManager licenseManager = LicenseManager.getInstance();  
//设置系统证书资源的上限值为 100（默认为 10）  
boolean bResult1 = licenseManager.setLicenseCapacity(100);  
if (bResult1 == true){  
    System.out.println("证书资源上限设置成功");  
} else {  
    System.out.println("证书资源上限设置失败");  
}  
//获取当前证书资源的上限值  
int licenseCount = licenseManager.getLicenseCapacity();  
System.out.println("当前系统证书资源的上限值为: " + liecnseCount);  
//获取当前剩余的系统资源数量  
int restLicenseNum = licenseManager.getRestLicenseCapacity();  
System.out.println("当前系统剩余的证书资源为: " + restLicenseNum);
```

2.5 说明

`LicenseManager` 的 `requestLicense` 方法要求客户端传入一个 `CallerMessage` 类的实例，该实例中包含了发出证书申请客户端的一些信息。随后向系统发出证书申请，该方法将会返回一个 `RequestResultMessage` 对象，该对象中包含了证书申请成功与否的信息。如上，可调用 `RequestResultMessage` 的 `isSuccess()` 方法来判断此次证书申请过程是否成功。若 `isSuccess()` 返回 `True`，那么表示此次证书申请成功；否则，证书申请失败。

当 `RequestResultMessage` 的 `isSuccess()` 返回 `False` 时，我们可以调用 `RequestResultMessage` 的 `getInfo()` 方法来获取证书申请失败的相关信息。如果该信息为 "Request License Failed"，那么说明此时系统的证书资源已经耗尽。若该信息为 "License Request Exception, Error Message:"，那么说明此时系统资源仍有剩余，问题出在客户端向系统申请资源的过程中，具体的出错原因由 `getInfo()` 获得的 `Exception` 信息具体判断。另外还可通过

RequestResultMessage 的 `getResponseTime()` 方法获得系统对证书申请做出回馈的时间。

2.6 示例

```
public class TestLicense {
    public static void main(String[] args){
        //1. 获得 LicenseManager 实例，用于接下来的证书申请操作。
        LicenseManager licenseManager = LicenseManager.getInstance();

        //默认系统证书资源上限为 10.
        System.out.println("默认系统证书资源上限为:" +
licenseManager.getLicenseCapacity());
        //设置系统证书资源的上限值为 100（默认为 10）
        boolean bResult1 = licenseManager.setLicenseCapacity(100);
        if (bResult1 == true) {
            System.out.println("证书资源上限设置成功，当前系统证书上限为：" +
licenseManager.getLicenseCapacity());
        } else {
            System.out.println("证书资源上限设置失败");
        }
        System.out.println("当前系统剩余的证书资源为(请求操作前)：" +
licenseManager.getRestLicenseCapacity());

        //2. 生成 CallerMessage 实例，填充做出证书申请客户端的一些基本信息。
        CallerMessage callerMessage = new CallerMessage("THFERDXEWWWS-
98754456");
        //3. 向系统发起证书申请操作。
        RequestResultMessage rrm = licenseManager.requestLicense(callerMessage);
        //4. 检查此次证书申请是否成功。
        if(rrm.isSuccess()){//证书申请成功
            System.out.println("证书申请成功，申请时间：" +rrm.getResponseTime());
        }else{//证书申请失败
            System.out.println("证书申请失败，出错信息：" +rrm.getInfo()+"，申请时
间：" +rrm.getResponseTime());
        }

        System.out.println("当前系统剩余的证书资源为(请求操作后)：" +
licenseManager.getRestLicenseCapacity());
    }
}
```

上诉代码的输出如下：

默认系统证书资源上限为:10

证书资源上限设置成功, 当前系统证书上限为: 100

当前系统剩余的证书资源为(请求操作前): 100

证书申请成功, 申请时间: Thu May 14 21:27:41 CST 2015

当前系统剩余的证书资源为(请求操作后): 99