Protocol Gateway SDK Manual

Maxvision Technology Corp.

New Technology Research Institute-Software Development Department

	Change the content		date
V1.0	found	Yu	2024.11.08
V1.1	Add new Ip settings, operation (seal/unseal) instructions, and adjust	Yu	2024.11.15
V1.2	Add device mode switching	Yu	2024.11.22

Catalogue

1. Quote SDK

Put the sdk in the specified directory and configure the corresponding path.

```
<dependency>
    <groupId>com.maxvision</groupId>
    <artifactId>maxvision-edge-protocol-lock-sdk</artifactId>
    <version>1.0.0-SNAPSHOT</version>
    <scope>system</scope>
    <systemPath>/lib/maxvision-edge-protocol-gateway-service.jar</systemPath>
</dependency>
```

maxvision-edge
-protocol-gatew

Add related dependencies.

```
<dependency>
    <groupId>org.projectlombok</groupId>
    <artifactId>lombok</artifactId>
    <version>1.18.24</version>
</dependency>

<dependency>
    <groupId>io.netty</groupId>
    <artifactId>netty-all</artifactId>
    <version>4.1.78.Final</version>
</dependency>

<dependency>
    <groupId>com.github.ben-manes.caffeine</groupId>
    <artifactId>caffeine</artifactId>
    <version>3.1.8</version>
</dependency>

<dependency>
    <groupId>cn.hutool</groupId>
    <artifactId>hutool-all</artifactId>
    <version>5.7.3</version>
</dependency>
```

```
<dependency>
    <groupId>org.apache.commons</groupId>
    <artifactId>commons-lang3</artifactId>
    <version>3.12.0</version>
</dependency>

<dependency>
    <groupId>com.google.guava</groupId>
    <artifactId>guava</artifactId>
    <version>28.1-jre</version>
</dependency>
```

2. Configuration file and log Settings

The SDK connects to the lock through netty. Custom netty port is supported. The configuration is as follows:

```
mvgateway:
  servers:
# Port
    - port: 8910
    # Whether the channel is encrypted
      encrypt: true
      # Application name
      name: maxvision
```

Note: Do not change the application name

Put the log configuration file in src/main/resource

logback-config.xml

3. Interface specification

Provides a lock setting service (LockSettingService) and a lock message reporting service (LockReportService).

3.1. Lock Setting Service

3.1.1. Description

The lock setting interface includes temporary sealing/unsealing card authorization, abnormal proximity distance setting, gps report interval setting, IpTcpUdp setting, OTA setting, RFID setting, sleep time setting, and SmsVip setting.

3.1.2. Interface agent

```
/**
* Authorization for temporary sealing/unsealing card
* @param encoderModel, temporary sealing/unsealing card authorization object
 */
void authSealOrUnsealCard(AuthSealOrUnsealCardEncoderModel encoderModel) throws
CustomException;

/**
* GPS reporting interval Settings
* @param encoderModel gps report interval setting object
 */
void gpsIntervalSetting(GpsIntervalSettingEncoderModel encoderModel) throws
CustomException;

/**
* SmsVip Settings
* @param encoderModel SmsVip sets the object
 */
void smsViperSetting(SmsVipSettingEncoderModel encoderModel) throws CustomException;

/**
* Multiple IP Settings
* @param encoderModel Multi-ip setting object
 */
void multiIpSetting(MultiIpEncoderModel encoderModel) throws CustomException;

/**
* operational order
* @param encoderModel Operation instruction object
 */
void operateCommand(OperateCommandEncoderModel encoderModel) throws
CustomException;
```

```
/**
* Switch work mode
* @param encoderModel Switching object for the working mode
 */
void changeDeviceMode(ChangeDeviceModeEncoderModel encoderModel) throws
CustomException;
```

3.1.3. Parameter declaration

3.1.3.1. Basic objects

```
public class BaseModel implements Serializable {
	/**
* Operation log primary key ID
	 */
	private String commandLogId;

	/**
:: Lock number
	 */
	private String lockCode;

	/**
* Type of operation instruction
	 */
	private String commandType;
}
```

3.1.3.2. Temporary seal/unseal card authorization object

```
public class AuthSealOrUnsealCardEncoderModel extends BaseModel implements Serializable {

	/**
* Type of authorization instruction for temporary sealing/unsealing card
	 */
	private String subCmdType;

	/**
* Bind or unbind
```

```
        */
       private Boolean binding;

       /**
:: Card type, universal (universal), temporary (temporary)
        */
       private String cardType;

       /**
IC card number
        */
       private String cardNo;

}
```

3.1.3.3. Set the GPS report interval object

```
public class GpsIntervalSettingEncoderModel extends BaseModel implements Serializable {

       /**
* GPS time interval
        */
       private String gpsInterval;

}
```

3.1.3.4. Set the object for SmsVip

```
public class SmsVipSettingEncoderModel extends BaseModel implements Serializable{

       /**
* VIP SMS list
        */
       private List<SmsVipSettingObj> smsVipList;

}
```

```
public class SmsVipSettingObj implements Serializable {

       /**
* Slot index
        */
```

```
    private String index;

    /**
* VIP number
     */
    private String phone;

    /**
* Whether alarm push is enabled
     */
    private Boolean pushAlarm = false;
}
```

3.1.3.5. Multiple IP Settings Object

```
public class MultiIpEncoderModel extends BaseModel implements Serializable {

    /**
:: Main IP
     */
    private MultiIpSettingObj primary;

    /**
:: Secondary IP
     */
    private MultiIpSettingObj secondary;

}


public class MultiIpSettingObj implements Serializable {

    private static final long serialVersionUID = -6896397748032325613L;

    /**
* The IP address bound to the device
     */
    private String ip;

    /**
* The TCP port number to which the device is bound
     */
    private String tcpPort;
```

```
    /**
     * APN
     */
    private String apn;

    /**
* APN account number
     */
    private String account;

    /**
* APN password
     */
    private String password;

    /**
:: Use of domain names
     */
    private boolean domain;
}
```

3.1.3.6. Operation (seal/unseal) instruction object

```
public class OperateCommandEncoderModel extends BaseModel implements Serializable {

    /**
* Command word type
     */
    private String cmdType;

    /**
* The consignment list number, which is used for sealing/unsealing/checking status/removing alarm. It is default 1234567890123
     */
    private String cargoCode = "1234567890123";

    /**
* Route number for sealing/unsealing/inspection status/alarm release, which is default 1234
     */
    private String routeCode = "1234";
```

```
    /**
* Card port, not required
     */
    private String gate;

    /**
* Operation object, not required
     */
    private Boolean forTrip;
}
```

3.1.3.7. Switch the device mode object

```
public class ChangeDeviceModeEncoderModel extends BaseModel implements Serializable {

    /**
* Equipment mode, 0-end of trip, 1-activation of trip
     */
    private int deviceMode;
}
```

3.2. LockReportService

3.2.1. Description

Lock the message reporting interface.

3.2.2. Interface agent

```
public interface LockReportService {

    /**
* Lock message reported
* @param jsonStr, message json string
     */
    void reportLockMsg(String jsonStr);
}
```

3.2.3. Method of calling

The LockReportService interface can be obtained to obtain the real-time report message.

```
/**
* Smart lock reporting is implemented
 */
@Service
public class LockReportServiceImpl implements LockReportService {
    @Override
    public void reportLockMsg(String model) {
        // todo Implement the business logic of response by yourself
                System.out.println("reportLockMsg:\n" + model);
    }
}
```

4. Error code

Error code	meaning	explain
1001	device not connected	This prompt is triggered by sending a command when the device is not connected
1002	commandLogId is error	The primary key ID of the operation log is empty
1003	lockCode is error	The lock number is incorrect; 4 letters + 10 digits
1004	commandType is error	The operation instruction type is incorrect. See Appendix I of Interface Input Parameter Description
2001	subCmdType is error	The subcommand type is incorrect and must be consistent with commandType
2002	cardType is error	Card type error; universal: universal card
2003	cardNo is error	Card number error; 8-digit hexadecimal number (capital letters)
3001	gpsIntervalSetting is error	GPS time interval error; any integer value in [10,65535]
4001	smsVipList cannot be empty	The VIP SMS list cannot be empty
4002	index is error	Slot index error; any integer value in [1,5]
4003	phone is error	VIP number is incorrect; length 4 to 20
5001	primary cannot be null	The main IP cannot be empty
5002	ip is error	IP error. Please check whether the IP format is correct
5003	tcpPort is error	TCP port error; 1-4 digit numbers

5004	apn is error	APN error; maximum length 50
5005	account is error	APN account error; maximum length 50
5006	password is error	APN password is incorrect; maximum length 50
6001	cmdType is error	The command type is incorrect and should be consistent with commandType; seal/unseal
7001	deviceMode is error	The device is in the wrong mode: 0 for end of trip, 1 for trip activation

5. Frequently asked questions

1. After introducing the sdk, you need to add annotations in the startup class

```
ComponentScan ("The path of the component to be scanned", "com.maxvision.edge.gateway"})
```

2、The generateBeanName method needs to be rewritten to generate the name of the Bean in the Spring container. The code is as follows:

```
@Slf4j
@ComponentScan({"com.maxvision.sdk.test", "com.maxvision.edge.gateway"})
Spring Boot // Spring Boot itself
public class SdkDemoApplication {

    public static class CustomGenerator implements BeanNameGenerator {
                @Override
                public String generateBeanName(BeanDefinition definition,
BeanDefinitionRegistry registry) {
                        return definition.getBeanClassName();
        }
    }

    public static void main(String[] args) {
                new SpringApplicationBuilder(SdkDemoApplication.class)
                                .beanNameGenerator(new CustomGenerator())
                                .run(args);
                log.info("--------------------sdk-demo--------------------");
    }

}
```

6. SDK business flow chart