设备type对应的设备类型

{

type: '12',

deviceTypeName: 'H10',

},

{

type: '13',

deviceTypeName: 'HOH77',

},

{

type: '16',

deviceTypeName: 'US3-6',

},

{

type: '18',

deviceTypeName: '1800',

},

{

type: '19',

deviceTypeName: 'H10-S',

},

{

type: '21',

deviceTypeName: 'T20',

},

{

type: '22',

deviceTypeName: 'H10-MSM',

},

{

type: '25',

deviceTypeName: 'T80',

},

{

type: '30',

deviceTypeName: 'L3',

},

{

type: '110',

deviceTypeName: 'H10-C',

},

@property (nonatomic, strong) NSString \*deviceName;//设备名称

@property (nonatomic, strong) NSString \*deviceCode;//设备code

@property (nonatomic, strong) NSString \*deviceId;//设备ID

@property (nonatomic, strong) NSString \*deviceType;//设备类型21是网关

@property (nonatomic, strong) NSString \*gatewayCode;//网关code

@property (nonatomic, assign) NSInteger battery;//电量

@property (nonatomic, strong) NSString \*isFirstConnection;//1在线 0离线

@property (nonatomic, strong) NSString \*jurisdiction;//"权限 1管理员，2家人"

@property (nonatomic, strong) NSString \*openStatus;//1常开，0||null非常开",

@property (nonatomic, strong) NSString \*deviceMac;//设备mac地址

@property (nonatomic, strong) NSString \*isBing;

@property (nonatomic, strong) NSString \*id;//设备ID

@property (nonatomic, strong) NSString \*name;//设备ID

@property (nonatomic, strong) NSString \*mac;//设备mac

@property (nonatomic, strong) NSString \*ip;//设备ip

@property (nonatomic, strong) NSString \*code;//设备序列号

@property (nonatomic, strong) NSString \*status;//1常开，0||null非常开"

@property (nonatomic, strong) NSString \*openGroup;//1组合 0任意

@property (nonatomic, strong) NSString \*checkingInId;

@property (nonatomic, strong) NSMutableArray<ADLLockModel \*> \*devices;//门锁数组

#pragma mark ------ 计算文字宽高 ------

+ (CGSize)calculateString:(NSString \*)string rectSize:(CGSize)rectSize fontSize:(CGFloat)fontSize {

    if (string.length > 0) {

        return [string boundingRectWithSize:rectSize options:NSStringDrawingUsesLineFragmentOrigin attributes:@{NSFontAttributeName:[UIFont systemFontOfSize:fontSize]} context:nil].size;

    } else {

        return CGSizeZero;

    }

}

#pragma mark ------ 计算富文本宽高 ------

+ (CGSize)calculateAttributeString:(NSAttributedString \*)attributeString rectSize:(CGSize)rectSize {

    if (attributeString.length > 0) {

        return [attributeString boundingRectWithSize:rectSize options:NSStringDrawingUsesLineFragmentOrigin context:nil].size;

    } else {

        return CGSizeZero;

    }

}

[ADLToast showMessage:@"设备离线,操作失败!" duration:3];

**酒店版 1800 名称为H10-C5 deviceType为111**

\_\_weak typeof(self)weakSelf = self;

\_\_strong typeof(self) strongSelf = weakSelf;

**加载中 取消**

[ADLToast showLoadingMessage:ADLString(@"loading")];

 [ADLToast hide];

[params setValue:[ADLUtils handleParamsSign:params] forKey:@"sign"];

 [ADLNetWorkManager postWithPath:ADEL\_verifyPasswrod parameters:params autoToast:YES success:^(NSDictionary \*responseDict) {

        if ([self.scrollView.mj\_header isRefreshing]) {

            [self.scrollView.mj\_header endRefreshing];

        }

        [ADLToast hide];

        if ([responseDict[@"code"] integerValue] == 10000) {

            NSDictionary \*dict = responseDict[@"data"];

            [ws dataOpenType:dict[@"validateCode"]];

        }

    } failure:^(NSError \*error) {

        if ([self.scrollView.mj\_header isRefreshing]) {

            [self.scrollView.mj\_header endRefreshing];

        }

    }];

校验是否存在密码

{

    code = 10000;

    currentPage = null;

    data =     {

        isExist = 1;

    };

    msg = OK;

    pageSize = null;

    sequenceNum = null;

    totalPage = null;

    totalSize = null;

}

**public static final int DEV\_L0\_CODE = 11;//L0 仅用于L2升级时用  
    public static final int DEV\_H10\_CODE = 12;//H10 WFL10  
    public static final int DEV\_HOH77\_CODE = 13;//HOH77 WFL7  
    public static final int DEV\_USB12\_CODE = 14;//USB12  
    public static final int DEV\_BOX\_CODE = 15;//BOX  
    public static final int DEV\_US36\_CODE = 16;//US3-6美国6号  
    public static final int DEV\_LS99\_CODE = 17;//LS99 = H10S  
    public static final int DEV\_H1800\_CODE = 18;//1800  
    public static final int DEV\_H10S\_CODE = 19;//H10S  
    public static final int DEV\_GATEWAY\_CODE = 22;//网关(H10MSM L2)  
    public static final int DEV\_GECKO\_CODE = 21;//壁虎  
    public static final int DEV\_GECKO2\_CODE = 233;//壁虎二代(T20-2)  
    public static final int DEV\_T80\_CODE = 25;//T80  
    public static final int DEV\_H10C\_CODE = 110;//h10c //相当于US13  
    public static final int DEV\_L3\_CODE = 30;//L3 外机（内机暂定用壁虎）1:室内，0：室外 L3Type = 3 \* 10 + ioType  
    public static final int DEV\_L2PLUS\_CODE = 24;//L2PLUS  
    public static final int DEV\_STORAGE\_BOX\_CODE = 51;//储物箱（未定参数）  
    public static final int DEV\_GAS\_VALVE\_CODE = 41;//燃气阀（未定参数）  
    public static final int DEV\_H1800C\_CODE = 111;//h1800c  
    public static final int DEV\_L3\_IN\_CODE = 31;//L3内机**

 NSString \*deviceId = [[NSUserDefaults standardUserDefaults] objectForKey:FAMILY\_DEVICE];

 [ADLUtils saveValue:self.model.deviceId forKey:FAMILY\_DEVICE];

ADD\_CHILD\_GATEWAY\_CODE

ADD\_CHILD\_GATEWAY\_ID

获取开门方式 == {

    code = 10000;

    currentPage = null;

    data =     {

        openApp = 1;

        openCard = 1;

        openFace = "-1";

        openFingerprint = "-1";

        openGroup = 0;

        openPassword = "-1";

        openStatus = 0;

        securityLevel = 4;

    };

    msg = OK;

    pageSize = null;

    sequenceNum = null;

    totalPage = null;

    totalSize = null;

}