

1. Basic Command Format

web port: 61080 call method: POST UTF-8 Code

json-rpc 2.0

1.1. Remote Open Door (Basic Command Format)

1.1.1. Send

```
{
  "jsonrpc": "2.0",
  "method": "RemoteOpenDoor",
  "params": [
    {
      "ControllerSN": 252123001,
      "DoorNO": 1
    }
  ],
  "id": 1001
}
```

parameter		type	Required	Remarks
jsonrpc	2.0	String	yes	fixed
method	RemoteOpenDoor	String	yes	(Function command)
params		Arrays	yes	parameters for the operation
ControllerSN	Actual 9-digit controller SN	number		must be 9 digits
DoorNO	1	number		Value range: 1,2,3,4
id	command sequence number	number	yes	

1.1.2. Return

1.1.2.1. (Success)

```
{
  "jsonrpc": "2.0",
  "result": {
    "success": true,
    "msg": "controller instructions execute successfully ok",
    "code": 1,
    "ControllerSN": 252123001
  },
  "id": 1001
}
```

parameter		type	Remarks
jsonrpc	2.0	String	fixed
result	{}	Object	(Result information)
success	true	boolean	true-Success; false-Failed
msg	controller instructions execute successfully ok	String	
code	1	number	Value range: 0, 1; 1- Success; 0-Failed
ControllerSN	Actual 9-digit Controller SN	number	must be 9 digits
id	command sequence number	number	

1.1.2.2. (Fail)

1.1.2.2.1. Reason 1: "Controller Communicate Failed"

```
{
  "jsonrpc": "2.0",
  "result": {
    "success": false,
    "msg": "Controller Communicate Failed",
    "code": 0,
    "ControllerSN": 252123001
  },
  "id": 1001
}
```

parameter		type	Remarks
jsonrpc	2.0	String	fixed
result	{}	Object	(Result information)
success	false	boolean	true-Success; false-Failed
msg	Controller Communicate Failed	String	
code	0	number	Value range: 0, 1; 1-Success; 0- Failed
ControllerSN	Actual 9-digit Controller SN	number	must be 9 digits
id	command sequence number	number	

1.1.2.2.2. Reason 2: "Invalid Command Format"

```
{
  "jsonrpc": "2.0",
  "error": {
    "code": -32700,
    "message": "Parse error: Invalid Command Format",
    "version": "8.241106"
  },
  "id": null
}
```

parameter		type	Remarks
jsonrpc	2.0	String	fixed
error	{}	Object	(Error information)
code	-32700	number	-32700: Invalid Command Format
message	Parse error: Invalid Command Format	String	
version	8.241106	String	Dcm Server Software Version
id	null		fixed

2. jsonrpc instructions

eg: Controller SN = 252123001

2.1. RemoteOpenDoor

2.1.1. Send

```
{
  "jsonrpc": "2.0",
  "method": "RemoteOpenDoor",
  "params": [
    {
      "ControllerSN": 252123001,
      "DoorNO": 1
    }
  ],
  "id": 1001
}
```

2.1.2. Return

```
{
  "jsonrpc": "2.0",
  "result": {
    "success": true,
    "msg": "controller instructions execute successfully ok",
    "code": 1,
    "ControllerSN": 252123001
  },
  "id": 1001
}
```

2.2. Force Normal Open

2.2.1. Send

```
{
  "jsonrpc": "2.0",
  "method": "SetDoorControlMode",
  "params": [
    {
      "ControllerSN": 252123001,
      "DoorNO": 1,
      "DoorControlMode": "NormalOpen",
      "OpenDelay(seconds)": 3
    }
  ],
  "id": 1004
}
```

2.2.2. Return

```
{
  "jsonrpc": "2.0",
  "result": {
    "success": true,
    "msg": "controller instructions execute successfully ok",
    "code": 1,
    "ControllerSN": 252123001
  },
  "id": 1004
}
```

2.3. Restore To Online

2.3.1. Send

```
{
  "jsonrpc": "2.0",
  "method": "SetDoorControlMode",
  "params": [
    {
      "ControllerSN": 252123001,
      "DoorNO": 1,
      "DoorControlMode": "OnLine",
      "OpenDelay(seconds)": 3
    }
  ],
  "id": 1004
}
```

2.3.2. Return

```
{
  "jsonrpc": "2.0",
  "result": {
    "success": true,
    "msg": "controller instructions execute successfully ok",
    "code": 1,
    "ControllerSN": 252123001
  },
  "id": 1004
}
```

2.4. QueryControllerStatus

2.4.1. Send

```
{
  "jsonrpc": "2.0",
  "method": "QueryControllerStatus",
  "params": [
    {
      "ControllerSN": 252123001
    }
  ],
  "id": 1002
}
```

2.4.2. Return

```
{
  "jsonrpc": "2.0",
  "result": {
    "success": true,
    "msg": "controller instructions execute successfully ok",
    "code": 1,
    "ControllersSN": 252123001,
    "ControllerInformation": {
      "Time": "2025-01-20 17:41:11",
      "Door1#": "open",
      "Door2#": "open",
      "LastSixRecords": [
        {
          "RecordIndex": 82,
          "RecordType": 2,
          "Description": "Remote Open Door",
          "CardNO": 3094927669,
          "DoorNO": 1,
          "InExit": "In",
          "Valid": "Pass",
          "Time": "2025-01-20 17:39:07"
        },
        {
          "RecordIndex": 81,
          "RecordType": 2,
          "Description": "Remote Open Door",
          "CardNO": 3094927669,
          "DoorNO": 1,
          "InExit": "In",
          "Valid": "Pass",
          "Time": "2025-01-20 17:39:03"
        },
        {
          "RecordIndex": 80,
          "RecordType": 2,
          "Description": "Remote Open Door",
          "CardNO": 3094927669,
          "DoorNO": 2,
          "InExit": "In",
          "Valid": "Pass",
          "Time": "2025-01-20 17:35:39"
        },
        {
          "RecordIndex": 79,
          "RecordType": 2,
          "Description": "Remote Open Door",
          "CardNO": 3094927669,
          "DoorNO": 1,
          "InExit": "In",
          "Valid": "Pass",
          "Time": "2025-01-20 17:35:39"
        }
      ]
    }
  }
}
```

```

        "RecordIndex": 78,
        "RecordType": 1,
        "Description": "Denied Access:No PRIVILEGE",
        "CardNO": 2152191830,
        "DoorNO": 1,
        "InExit": "In",
        "Valid": "Pass",
        "Time": "2025-01-20 17:35:34"
    },
    {
        "RecordIndex": 77,
        "RecordType": 1,
        "Description": "Denied Access:No PRIVILEGE",
        "CardNO": 2152191830,
        "DoorNO": 1,
        "InExit": "In",
        "Valid": "Pass",
        "Time": "2025-01-20 17:35:33"
    }
]
},
{id": 1002
}

```

2.5. AdjustTime

2.5.1. Send

```

{
    "jsonrpc": "2.0",
    "method": "AdjustTime",
    "params": [
        {
            "ControllersSN": 252123001
        }
    ],
    "id": 1003
}

```

2.5.2. Return

```

{
    "jsonrpc": "2.0",
    "result": {
        "success": true,
        "msg": "controller instructions execute successfully ok",
        "code": 1,
        "ControllersSN": 252123001
    },
    "id": 1003
}

```

2.6. SetDoorControlMode

2.6.1. Send--OnLine

```
{
  "jsonrpc": "2.0",
  "method": "SetDoorControlMode",
  "params": [
    {
      "ControllerSN": 252123001,
      "DoorNO": 1,
      "DoorControlMode": "OnLine",
      "OpenDelay(seconds)": 3
    }
  ],
  "id": 1004
}
```

2.6.2. Send--NormalOpen

```
{
  "jsonrpc": "2.0",
  "method": "SetDoorControlMode",
  "params": [
    {
      "ControllerSN": 252123001,
      "DoorNO": 1,
      "DoorControlMode": "NormalOpen",
      "OpenDelay(seconds)": 3
    }
  ],
  "id": 1004
}
```

2.6.3. Send--NormalClosed

```
{
  "jsonrpc": "2.0",
  "method": "SetDoorControlMode",
  "params": [
    {
      "ControllerSN": 252123001,
      "DoorNO": 1,
      "DoorControlMode": "NormalClosed",
      "OpenDelay(seconds)": 3
    }
  ],
  "id": 1004
}
```

2.6.4. Return

```
{
  "jsonrpc": "2.0",
  "result": {
    "success": true,
    "msg": "controller instructions execute successfully ok",
    "code": 1,
    "ControllersSN": 252123001
  },
  "id": 1004
}
```

2.7. GetPrivilegeTotal

2.7.1. Send

```
{
  "jsonrpc": "2.0",
  "method": "GetPrivilegeTotal",
  "params": [
    {
      "ControllersSN": 252123001
    }
  ],
  "id": 4003
}
```

2.7.2. Return

```
{
  "jsonrpc": "2.0",
  "result": {
    "success": true,
    "msg": "controller instructions execute successfully ok",
    "code": 1,
    "ControllersSN": 252123001,
    "PrivilegeTotal": 6
  },
  "id": 4003
}
```

2.8. AddEditPrivilege

2.8.1. Send

```
{
  "jsonrpc": "2.0",
  "method": "AddEditPrivilege",
  "params": [
    {
```

```

        "ControllerSN": 252123001,
        "Privilege": [
            {
                "CardNO": [ 12345
                ],
                "BeginDateTime": "2022-05-08 12:12:10",
                "EndDateTime": "2029-12-31 23:59:00",
                "Door1TimeProfileID": 1,
                "Door2TimeProfileID": 1,
                "Door3TimeProfileID": 1,
                "Door4TimeProfileID": 1,
                "UserPassword": 0
            }
        ]
    },
    ],
    "id": 4001
}

```

2.8.2. Return

```

{
    "jsonrpc": "2.0",
    "result": {
        "success": true,
        "msg": "controller instructions execute successfully ok",
        "code": 1,
        "ControllerSN": 252123001
    },
    "id": 4001
}

```

2.9. UploadAllPrivileges (5Units)

2.9.1. Send

```

{
    "jsonrpc": "2.0",
    "method": "UploadAllPrivileges",
    "params": [
        {
            "ControllerSN": 252123001,
            "Privilege": [
                {
                    "CardNO": [ 12345
                    ],
                    "BeginDateTime": "2022-05-08 12:12:10",
                    "EndDateTime": "2029-12-31 23:59:00",
                    "Door1TimeProfileID": 1,
                    "Door2TimeProfileID": 1,
                    "Door3TimeProfileID": 1,
                    "Door4TimeProfileID": 1,
                    "UserPassword": 0
                }
            ]
        }
    ]
}

```

```

    },
    {
      "CardNO": [ 12346
      ],
      "BeginDateTime": "2022-05-08 12:12:10",
      "EndDateTime": "2029-12-31 23:59:00",
      "Door1TimeProfileID": 1,
      "Door2TimeProfileID": 1,
      "Door3TimeProfileID": 1,
      "Door4TimeProfileID": 1,
      "UserPassword": 0
    },
    {
      "CardNO": [ 12347
      ],
      "BeginDateTime": "2022-05-08 12:12:10",
      "EndDateTime": "2029-12-31 23:59:00",
      "Door1TimeProfileID": 1,
      "Door2TimeProfileID": 1,
      "Door3TimeProfileID": 1,
      "Door4TimeProfileID": 1,
      "UserPassword": 0
    },
    {
      "CardNO": [ 12348
      ],
      "BeginDateTime": "2022-05-08 12:12:10",
      "EndDateTime": "2029-12-31 23:59:00",
      "Door1TimeProfileID": 1,
      "Door2TimeProfileID": 1,
      "Door3TimeProfileID": 1,
      "Door4TimeProfileID": 1,
      "UserPassword": 0
    },
    {
      "CardNO": [ 12349
      ],
      "BeginDateTime": "2022-05-08 12:12:10",
      "EndDateTime": "2029-12-31 23:59:00",
      "Door1TimeProfileID": 1,
      "Door2TimeProfileID": 1,
      "Door3TimeProfileID": 1,
      "Door4TimeProfileID": 1,
      "UserPassword": 0
    }
  ]
},
{
  "id": 4004
}

```

2.9.2. Return

```
{
  "jsonrpc": "2.0",
  "result": {
    "success": true,
    "msg": "controller instructions execute successfully ok",
    "code": 1,
    "uploadStep": "Upload Privileges Completely",
    "uploadedPrivilegeNumber": 5,
    "controllersSN": 252123001
  },
  "id": 4004
}
```

2.10. GetPrivilegeTotal

2.10.1. Send

```
{
  "jsonrpc": "2.0",
  "method": "GetPrivilegeTotal",
  "params": [
    {
      "controllersSN": 252123001
    }
  ],
  "id": 4003
}
```

2.10.2. Return

```
{
  "jsonrpc": "2.0",
  "result": {
    "success": true,
    "msg": "controller instructions execute successfully ok",
    "code": 1,
    "controllersSN": 252123001,
    "privilegeTotal": 5
  },
  "id": 4003
}
```

2.11. QueryPrivilege

2.11.1. Send

```
{
  "jsonrpc": "2.0",
  "method": "QueryPrivilege",
  "params": [
    {
      "ControllerSN": 252123001,
      "CardNO": [
        12345
      ]
    }
  ],
  "id": 4003
}
```

2.11.2. Return

```
{
  "jsonrpc": "2.0",
  "result": {
    "success": true,
    "msg": "controller instructions execute successfully ok",
    "code": 1,
    "ControllerSN": 252123001,
    "Privilege": [
      {
        "CardNO": 12345,
        "BeginDateTime": "2022-05-08",
        "EndDateTime": "2029-12-31",
        "Door1TimeProfileID": 1,
        "Door2TimeProfileID": 1,
        "Door3TimeProfileID": 0,
        "Door4TimeProfileID": 0,
        "UserPassword": 0
      }
    ]
  },
  "id": 4003
}
```

2.12. DeletePrivilege

2.12.1. Send

```
{
  "jsonrpc": "2.0",
  "method": "DeletePrivilege",
  "params": [
    {
```

```

        "ControllerSN": 252123001,
        "Privilege": [
            {
                "CardNO": [ 12345
            ]
        ]
    },
    ],
    "id": 4002
}

```

2.12.2. Return

```

{
    "jsonrpc": "2.0",
    "result": {
        "success": true,
        "msg": "controller instructions execute successfully ok",
        "code": 1,
        "ControllerSN": 252123001
    },
    "id": 4002
}

```

2.13. GetPrivilege

2.13.1. Send

```

{
    "jsonrpc": "2.0",
    "method": "GetPrivilege",
    "params": [
        {
            "ControllerSN": 252123001
        }
    ],
    "id": 4005
}

```

2.13.2. Return

```

{
    "jsonrpc": "2.0",
    "result": {
        "success": true,
        "msg": "controller instructions execute successfully ok",
        "code": 1,
        "ControllerSN": 252123001,
        "GotPrivilegeNumber": 4,
        "LastPrivilegeIndex": 5,
        "RecordInformation": [

```

```

    {
      "ID": 2,
      "CardNO": 12346,
      "BeginDateTime": "2022-05-08 12:12:10",
      "EndDateTime": "2029-12-31 23:59:00",
      "Door1TimeProfileID": 1,
      "Door2TimeProfileID": 1,
      "Door3TimeProfileID": 0,
      "Door4TimeProfileID": 0,
      "UserPassword": 0
    },
    {
      "ID": 3,
      "CardNO": 12347,
      "BeginDateTime": "2022-05-08 12:12:10",
      "EndDateTime": "2029-12-31 23:59:00",
      "Door1TimeProfileID": 1,
      "Door2TimeProfileID": 1,
      "Door3TimeProfileID": 0,
      "Door4TimeProfileID": 0,
      "UserPassword": 0
    },
    {
      "ID": 4,
      "CardNO": 12348,
      "BeginDateTime": "2022-05-08 12:12:10",
      "EndDateTime": "2029-12-31 23:59:00",
      "Door1TimeProfileID": 1,
      "Door2TimeProfileID": 1,
      "Door3TimeProfileID": 0,
      "Door4TimeProfileID": 0,
      "UserPassword": 0
    },
    {
      "ID": 5,
      "CardNO": 12349,
      "BeginDateTime": "2022-05-08 12:12:10",
      "EndDateTime": "2029-12-31 23:59:00",
      "Door1TimeProfileID": 1,
      "Door2TimeProfileID": 1,
      "Door3TimeProfileID": 0,
      "Door4TimeProfileID": 0,
      "UserPassword": 0
    }
  ]
},
"id": 4005
}

```

2.14. UploadAllTimeProfiles (10 Units)

2.14.1. Send

```
{
  "jsonrpc": "2.0",
  "method": "uploadAllTimeProfiles",
  "params": [
    {
      "ControllerSN": 252123001,
      "TimeProfile": [
        {
          "TimeProfileID": 2,
          "BeginDate": "2022-05-10",
          "EndDate": "2099-12-31",
          "Mon": 1,
          "Tue": 1,
          "Wed": 1,
          "Thu": 1,
          "Fri": 1,
          "Sat": 1,
          "Sun": 1,
          "TimeZone1BeginTime": "08:30",
          "TimeZone1EndTime": "12:00",
          "TimeZone2BeginTime": "14:00",
          "TimeZone2EndTime": "17:30",
          "TimeZone3BeginTime": "00:00",
          "TimeZone3EndTime": "00:00",
          "NextLinkedTimeProfileID": 0
        },
        {
          "TimeProfileID": 3,
          "BeginDate": "2022-05-10",
          "EndDate": "2099-12-31",
          "Mon": 1,
          "Tue": 1,
          "Wed": 1,
          "Thu": 1,
          "Fri": 1,
          "Sat": 1,
          "Sun": 1,
          "TimeZone1BeginTime": "08:30",
          "TimeZone1EndTime": "12:00",
          "TimeZone2BeginTime": "14:00",
          "TimeZone2EndTime": "17:30",
          "TimeZone3BeginTime": "00:00",
          "TimeZone3EndTime": "00:00",
          "NextLinkedTimeProfileID": 0
        },
        {
          "TimeProfileID": 4,
          "BeginDate": "2022-05-10",
          "EndDate": "2099-12-31",
          "Mon": 1,
          "Tue": 1,
```

```

        "Wed": 1,
        "Thu": 1,
        "Fri": 1,
        "Sat": 1,
        "Sun": 1,
        "TimeZone1BeginTime": "08:30",
        "TimeZone1EndTime": "12:00",
        "TimeZone2BeginTime": "14:00",
        "TimeZone2EndTime": "17:30",
        "TimeZone3BeginTime": "00:00",
        "TimeZone3EndTime": "00:00",
        "NextLinkedTimeProfileID": 0
    },
    {
        "TimeProfileID": 5,
        "BeginDate": "2022-05-10",
        "EndDate": "2099-12-31",
        "Mon": 1,
        "Tue": 1,
        "Wed": 1,
        "Thu": 1,
        "Fri": 1,
        "Sat": 1,
        "Sun": 1,
        "TimeZone1BeginTime": "08:30",
        "TimeZone1EndTime": "12:00",
        "TimeZone2BeginTime": "14:00",
        "TimeZone2EndTime": "17:30",
        "TimeZone3BeginTime": "00:00",
        "TimeZone3EndTime": "00:00",
        "NextLinkedTimeProfileID": 0
    },
    {
        "TimeProfileID": 6,
        "BeginDate": "2022-05-10",
        "EndDate": "2099-12-31",
        "Mon": 1,
        "Tue": 1,
        "Wed": 1,
        "Thu": 1,
        "Fri": 1,
        "Sat": 1,
        "Sun": 1,
        "TimeZone1BeginTime": "08:30",
        "TimeZone1EndTime": "12:00",
        "TimeZone2BeginTime": "14:00",
        "TimeZone2EndTime": "17:30",
        "TimeZone3BeginTime": "00:00",
        "TimeZone3EndTime": "00:00",
        "NextLinkedTimeProfileID": 0
    },
    {
        "TimeProfileID": 7,
        "BeginDate": "2022-05-10",
        "EndDate": "2099-12-31",
        "Mon": 1,

```

```

    "Tue": 1,
    "Wed": 1,
    "Thu": 1,
    "Fri": 1,
    "Sat": 1,
    "Sun": 1,
    "TimeZone1BeginTime": "08:30",
    "TimeZone1EndTime": "12:00",
    "TimeZone2BeginTime": "14:00",
    "TimeZone2EndTime": "17:30",
    "TimeZone3BeginTime": "00:00",
    "TimeZone3EndTime": "00:00",
    "NextLinkedTimeProfileID": 0
  },
  {
    "TimeProfileID": 8,
    "BeginDate": "2022-05-10",
    "EndDate": "2099-12-31",
    "Mon": 1,
    "Tue": 1,
    "Wed": 1,
    "Thu": 1,
    "Fri": 1,
    "Sat": 1,
    "Sun": 1,
    "TimeZone1BeginTime": "08:30",
    "TimeZone1EndTime": "12:00",
    "TimeZone2BeginTime": "14:00",
    "TimeZone2EndTime": "17:30",
    "TimeZone3BeginTime": "00:00",
    "TimeZone3EndTime": "00:00",
    "NextLinkedTimeProfileID": 0
  },
  {
    "TimeProfileID": 9,
    "BeginDate": "2022-05-10",
    "EndDate": "2099-12-31",
    "Mon": 1,
    "Tue": 1,
    "Wed": 1,
    "Thu": 1,
    "Fri": 1,
    "Sat": 1,
    "Sun": 1,
    "TimeZone1BeginTime": "08:30",
    "TimeZone1EndTime": "12:00",
    "TimeZone2BeginTime": "14:00",
    "TimeZone2EndTime": "17:30",
    "TimeZone3BeginTime": "00:00",
    "TimeZone3EndTime": "00:00",
    "NextLinkedTimeProfileID": 0
  },
  {
    "TimeProfileID": 10,
    "BeginDate": "2022-05-10",
    "EndDate": "2099-12-31",

```

```

        "Mon": 1,
        "Tue": 1,
        "Wed": 1,
        "Thu": 1,
        "Fri": 1,
        "Sat": 1,
        "Sun": 1,
        "TimeZone1BeginTime": "08:30",
        "TimeZone1EndTime": "12:00",
        "TimeZone2BeginTime": "14:00",
        "TimeZone2EndTime": "17:30",
        "TimeZone3BeginTime": "00:00",
        "TimeZone3EndTime": "00:00",
        "NextLinkedTimeProfileID": 0
    },
    {
        "TimeProfileID": 11,
        "BeginDate": "2022-05-10",
        "EndDate": "2099-12-31",
        "Mon": 1,
        "Tue": 1,
        "Wed": 1,
        "Thu": 1,
        "Fri": 1,
        "Sat": 1,
        "Sun": 1,
        "TimeZone1BeginTime": "08:30",
        "TimeZone1EndTime": "12:00",
        "TimeZone2BeginTime": "14:00",
        "TimeZone2EndTime": "17:30",
        "TimeZone3BeginTime": "00:00",
        "TimeZone3EndTime": "00:00",
        "NextLinkedTimeProfileID": 0
    }
]
},
{
    "id": 5001
}

```

2.14.2. Return

```

{
    "jsonrpc": "2.0",
    "result": {
        "success": true,
        "msg": "controller instructions execute successfully ok",
        "code": 1,
        "ControllerSN": 252123001
    },
    "id": 5001
}

```

2.15. GetTimeProfiles

2.15.1. Send

```
{
  "jsonrpc": "2.0",
  "method": "GetTimeProfiles",
  "params": [
    {
      "ControllerSN": 252123001
    }
  ],
  "id": 4005
}
```

2.15.2. Return

```
{
  "jsonrpc": "2.0",
  "result": {
    "success": true,
    "msg": "controller instructions execute successfully ok",
    "code": 1,
    "ControllerSN": 252123001,
    "GotTimeProfileNumber": 10,
    "RecordInformation": [
      {
        "TimeProfileID": 2,
        "BeginDate": "2022-05-10",
        "EndDate": "2099-12-31",
        "Mon": 1,
        "Tue": 1,
        "Wed": 1,
        "Thu": 1,
        "Fri": 1,
        "Sat": 1,
        "Sun": 1,
        "TimeZone1BeginTime": "08:30",
        "TimeZone1EndTime": "12:00",
        "TimeZone2BeginTime": "14:00",
        "TimeZone2EndTime": "17:30",
        "TimeZone3BeginTime": "00:00",
        "TimeZone3EndTime": "00:00",
        "NextLinkedTimeProfileID": 0,
        "HolidayControl": 1
      },
      {
        "TimeProfileID": 3,
        "BeginDate": "2022-05-10",
        "EndDate": "2099-12-31",
        "Mon": 1,
        "Tue": 1,
        "Wed": 1,
        "Thu": 1,

```

```

        "Fri": 1,
        "Sat": 1,
        "Sun": 1,
        "TimeZone1BeginTime": "08:30",
        "TimeZone1EndTime": "12:00",
        "TimeZone2BeginTime": "14:00",
        "TimeZone2EndTime": "17:30",
        "TimeZone3BeginTime": "00:00",
        "TimeZone3EndTime": "00:00",
        "NextLinkedTimeProfileID": 0,
        "HolidayControl": 1
    },
    {
        "TimeProfileID": 4,
        "BeginDate": "2022-05-10",
        "EndDate": "2099-12-31",
        "Mon": 1,
        "Tue": 1,
        "Wed": 1,
        "Thu": 1,
        "Fri": 1,
        "Sat": 1,
        "Sun": 1,
        "TimeZone1BeginTime": "08:30",
        "TimeZone1EndTime": "12:00",
        "TimeZone2BeginTime": "14:00",
        "TimeZone2EndTime": "17:30",
        "TimeZone3BeginTime": "00:00",
        "TimeZone3EndTime": "00:00",
        "NextLinkedTimeProfileID": 0,
        "HolidayControl": 1
    },
    {
        "TimeProfileID": 5,
        "BeginDate": "2022-05-10",
        "EndDate": "2099-12-31",
        "Mon": 1,
        "Tue": 1,
        "Wed": 1,
        "Thu": 1,
        "Fri": 1,
        "Sat": 1,
        "Sun": 1,
        "TimeZone1BeginTime": "08:30",
        "TimeZone1EndTime": "12:00",
        "TimeZone2BeginTime": "14:00",
        "TimeZone2EndTime": "17:30",
        "TimeZone3BeginTime": "00:00",
        "TimeZone3EndTime": "00:00",
        "NextLinkedTimeProfileID": 0,
        "HolidayControl": 1
    },
    {
        "TimeProfileID": 6,
        "BeginDate": "2022-05-10",
        "EndDate": "2099-12-31",

```

```

        "Mon": 1,
        "Tue": 1,
        "Wed": 1,
        "Thu": 1,
        "Fri": 1,
        "Sat": 1,
        "Sun": 1,
        "TimeZone1BeginTime": "08:30",
        "TimeZone1EndTime": "12:00",
        "TimeZone2BeginTime": "14:00",
        "TimeZone2EndTime": "17:30",
        "TimeZone3BeginTime": "00:00",
        "TimeZone3EndTime": "00:00",
        "NextLinkedTimeProfileID": 0,
        "HolidayControl": 1
    },
    {
        "TimeProfileID": 7,
        "BeginDate": "2022-05-10",
        "EndDate": "2099-12-31",
        "Mon": 1,
        "Tue": 1,
        "Wed": 1,
        "Thu": 1,
        "Fri": 1,
        "Sat": 1,
        "Sun": 1,
        "TimeZone1BeginTime": "08:30",
        "TimeZone1EndTime": "12:00",
        "TimeZone2BeginTime": "14:00",
        "TimeZone2EndTime": "17:30",
        "TimeZone3BeginTime": "00:00",
        "TimeZone3EndTime": "00:00",
        "NextLinkedTimeProfileID": 0,
        "HolidayControl": 1
    },
    {
        "TimeProfileID": 8,
        "BeginDate": "2022-05-10",
        "EndDate": "2099-12-31",
        "Mon": 1,
        "Tue": 1,
        "Wed": 1,
        "Thu": 1,
        "Fri": 1,
        "Sat": 1,
        "Sun": 1,
        "TimeZone1BeginTime": "08:30",
        "TimeZone1EndTime": "12:00",
        "TimeZone2BeginTime": "14:00",
        "TimeZone2EndTime": "17:30",
        "TimeZone3BeginTime": "00:00",
        "TimeZone3EndTime": "00:00",
        "NextLinkedTimeProfileID": 0,
        "HolidayControl": 1
    },

```

```
{
  "TimeProfileID": 9,
  "BeginDate": "2022-05-10",
  "EndDate": "2099-12-31",
  "Mon": 1,
  "Tue": 1,
  "Wed": 1,
  "Thu": 1,
  "Fri": 1,
  "Sat": 1,
  "Sun": 1,
  "TimeZone1BeginTime": "08:30",
  "TimeZone1EndTime": "12:00",
  "TimeZone2BeginTime": "14:00",
  "TimeZone2EndTime": "17:30",
  "TimeZone3BeginTime": "00:00",
  "TimeZone3EndTime": "00:00",
  "NextLinkedTimeProfileID": 0,
  "HolidayControl": 1
},
{
  "TimeProfileID": 10,
  "BeginDate": "2022-05-10",
  "EndDate": "2099-12-31",
  "Mon": 1,
  "Tue": 1,
  "Wed": 1,
  "Thu": 1,
  "Fri": 1,
  "Sat": 1,
  "Sun": 1,
  "TimeZone1BeginTime": "08:30",
  "TimeZone1EndTime": "12:00",
  "TimeZone2BeginTime": "14:00",
  "TimeZone2EndTime": "17:30",
  "TimeZone3BeginTime": "00:00",
  "TimeZone3EndTime": "00:00",
  "NextLinkedTimeProfileID": 0,
  "HolidayControl": 1
},
{
  "TimeProfileID": 11,
  "BeginDate": "2022-05-10",
  "EndDate": "2099-12-31",
  "Mon": 1,
  "Tue": 1,
  "Wed": 1,
  "Thu": 1,
  "Fri": 1,
  "Sat": 1,
  "Sun": 1,
  "TimeZone1BeginTime": "08:30",
  "TimeZone1EndTime": "12:00",
  "TimeZone2BeginTime": "14:00",
  "TimeZone2EndTime": "17:30",
  "TimeZone3BeginTime": "00:00",
```

```

        "TimeZone3EndTime": "00:00",
        "NextLinkedTimeProfileID": 0,
        "HolidayControl": 1
    }
]
},
"id": 4005
}

```

2.16. CreateQRCode

2.16.1. Send

```

{
  "jsonrpc": "2.0",
  "method": "CreateQRCode",
  "params": [
    {
      "CardNO": 75583150550,
      "EndDateTime": "2025-01-20 17:58:29"
    }
  ],
  "id": 4006
}

```

2.16.2. Return

```

{
  "jsonrpc": "2.0",
  "result": {
    "success": true,
    "msg": "run success OK",
    "code": 1,
    "CardNO": 75583150550,
    "EndDateTime": "2025-01-20 17:58:29",
    "QRCodeInformation": "389D54158B24078C75222E336B641050"
  },
  "id": 4006
}

```

2.17. Enabled QR Code Transparent Transmission

2.17.1. Send

```

{
  "jsonrpc": "2.0",
  "method": "FunctionConfig",
  "params": [
    {
      "ControllerSN": 252123001,
      "QRCodeTransparentTransmission": "Enabled"
    }
  ]
}

```

```
],  
  "id": 1004  
}
```

2.17.2. Return

```
{  
  "jsonrpc": "2.0",  
  "result": {  
    "success": true,  
    "msg": "controller instructions execute successfully ok",  
    "code": 1,  
    "ControllersSN": 252123001  
  },  
  "id": 1004  
}
```

2.18. Disabled QR Code Transparent Transmission

2.18.1. Send

```
{  
  "jsonrpc": "2.0",  
  "method": "FunctionConfig",  
  "params": [  
    {  
      "ControllersSN": 252123001,  
      "QRCodeTransparentTransmission": "Forbidden"  
    }  
  ],  
  "id": 1004  
}
```

2.18.2. Return

```
{  
  "jsonrpc": "2.0",  
  "result": {  
    "success": true,  
    "msg": "controller instructions execute successfully ok",  
    "code": 1,  
    "ControllersSN": 252123001  
  },  
  "id": 1004  
}
```

2.19. Enabled Input 6 Digits Password As CardNO

2.19.1. Send

```
{
  "jsonrpc": "2.0",
  "method": "FunctionConfig",
  "params": [
    {
      "ControllerSN": 252123001,
      "Input6DigitsPasswordAsCardNO": "Enabled"
    }
  ],
  "id": 1004
}
```

2.19.2. Return

```
{
  "jsonrpc": "2.0",
  "result": {
    "success": true,
    "msg": "controller instructions execute successfully ok",
    "code": 1,
    "ControllerSN": 252123001
  },
  "id": 1004
}
```

2.20. test input password

pwd: 123456 only Once in the date of 2024-06-09

2.20.1. Send

```
{
  "jsonrpc": "2.0",
  "method": "AddEditPrivilege",
  "params": [
    {
      "ControllerSN": 252123001,
      "Privilege": [
        {
          "CardNO": [ 123456
          ],
          "BeginDateTime": "2025-01-20 00:00:00",
          "EndDateTime": "2025-01-20 23:59:59",
          "Door1TimeProfileID": 1,
          "Door2TimeProfileID": 1,
          "Door3TimeProfileID": 1,
          "Door4TimeProfileID": 1,
          "Onlyonce": "Enabled"
        }
      ]
    }
  ]
}
```

```

    ]
  }
],
"id": 4001
}

```

2.20.2. Return

```

{
  "jsonrpc": "2.0",
  "result": {
    "success": true,
    "msg": "controller instructions execute successfully ok",
    "code": 1,
    "ControllersSN": 252123001
  },
  "id": 4001
}

```

2.21. Disabled Input 6 Digits Password As CardNO

2.21.1. Send

```

{
  "jsonrpc": "2.0",
  "method": "FunctionConfig",
  "params": [
    {
      "ControllersSN": 252123001,
      "Input6DigitsPasswordAsCardNO": "Forbidden"
    }
  ],
  "id": 1004
}

```

2.21.2. Return

```

{
  "jsonrpc": "2.0",
  "result": {
    "success": true,
    "msg": "controller instructions execute successfully ok",
    "code": 1,
    "ControllersSN": 252123001
  },
  "id": 1004
}

```

2.22. Get Swipe

Please reference C# Demo or Java Demo

3. Realtime-monitoring (websocket)

The demo as websocket client connects to the websocket server (port: 61081) on the DcmCloudServer.

```
private static final String SERVER = "ws://localhost:61081"; //Dcm websocket service ip/port
```

run
WEBAPI_JavaDemoJDK1.8_2024English\src\websocketTestOnLocalHost_RealtimeMonitoring.java
in the java ide (eg.: IntelliJ IDEA 2022.3.2(Community Edition))

3.1. WS Server push Information to the Client

3.1.1. Timed push information (default 30 second cycle) door status

		[{ "SN" : 252123001, "1#Door" : "Closed", "2#Door" : "Open", "RefreshTime" : "2023-05-22 11:57:56" }]	
	Arrays		
SN	Number	252123001	Controller SN(9 digits)
1#Door	String	"Closed"	range: "Open","Closed" No.1 door status: Closed
2#Door	String	"Open"	No.2 door status: Open
RefreshTime	String	"2025-01-20 18:12:34"	Information Refresh Time of the PC

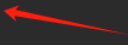
3.1.2. Actively pushed record information (RecordType==1)

After swiping ic cards, you can get record information.

```

Mon Jan 20 18:12:23 CST 2025 Timed push information (default 30 second cycle) door status
Mon Jan 20 18:12:23 CST 2025 [ {
  "SN" : 252123001,
  "Door1#" : "Open",
  "Door2#" : "Open",
  "RefreshTime" : "2025-01-20 18:11:35"
} ]
Mon Jan 20 18:12:34 CST 2025 Timed push information (default 30 second cycle) door status
Mon Jan 20 18:12:34 CST 2025 [ {
  "SN" : 252123001,
  "Door1#" : "Closed",
  "Door2#" : "Open",
  "RefreshTime" : "2025-01-20 18:12:34"
} ]
Mon Jan 20 18:12:34 CST 2025 Timed push information (default 30 second cycle) door status
Mon Jan 20 18:12:34 CST 2025 [ {
  "SN" : 252123001,
  "Door1#" : "Open",
  "Door2#" : "Open",
  "RefreshTime" : "2025-01-20 18:12:34"
} ]

```

 new record information

###

		[{ "SN" : 252123001, "RecordIndex" : 83, "RecordType" : 1, "Description" : "Denied Access:No PRIVILEGE", "CardNO" : 2152191830, "DoorNO" : 1, "InExit" : "In", "Valid" : "Forbidden", "Time" : "2025-01-20 18:15:26" }]	
	Arrays		
SN	Number	252123001	Controller SN(9 digits)
Index	Number	83	record index
RecordType	Number	1	range: 1,2,3 =1 swipe record =2 Events(remote open, push button, Door Status, Reboot) =3 Warns(Fire etc.)
Description	String	"Denied Access: No PRIVILEGE"	
CardNO	Number	2152191830	
DoorNO	Number	1	range: 1,2,3,4
InExit	String	"In"	range: "In","Exit"
Valid	String	"Forbidden"	range: "Pass","Forbidden"

Time	String	"2025-01-20 18:15:26"	time format: yyyy-MM-dd HH:mm:ss
------	--------	-----------------------	-------------------------------------

3.1.3 Dealing record information

3.1.3.1. source code for dealing

the function dealDataFromDcmCloudServer is followed.

You can analyse "long vcard = recinfo.get("CardNO").asLong();" and decide whether to open door.

```

static void dealDataFromDcmCloudServer(WebSocket ws) //deal data
information
{
    String msg;
    try {
        synchronized (queue) {
            msg = queue.poll();
        }
        ObjectMapper infojsn = new ObjectMapper();
        JsonNode node = infojsn.readTree(msg);
        if (!node.findPath("RefreshTime").isMissingNode()) { //----
            //Door synchronization information: Door status
            Acswebapi.logInfo("Timed push information (default 30 second
cycle) door status");
            Acswebapi.logInfoInJson(msg);
            for(int i=0; i<node.size(); i++)
            {
                //Return Information
                //    [{
                //        "SN": 225043096,
                //        "Door1#": "Open",
                //        "Door2#": "Open",
                //        "RefreshTime": "2021-12-03 10:10:38"
                //    }]
                JsonNode doorinfo = node.get(i);
                SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd
hh:mm:ss");
                java.util.Date refreshtime =
sdf.parse(doorinfo.get("RefreshTime").asText());
                java.util.Date curtime =new java.util.Date();
                int controllersSN = doorinfo.get("SN").asInt();
                if (Math.abs(refreshtime.getTime()-curtime.getTime())>
60*1000) //communication failed
                {
                    Acswebapi.logInfo(String.format("ControllersSN = %d
communication failed???", controllersSN));
                }
                //doorinfo.get("Door1#"); //Get Door Status
            }
        }
        else {
            Acswebapi.logInfoInJson(msg); //----
            //Record information
            //    [

```

```

//      {
//          "SN": 225043096,
//          "RecordIndex": 194787,
//          "RecordType": 1,
//          "Description": "Swipe",
//          "CardNO": 12800026,
//          "DoorNO": 1,
//          "InExit": "In",
//          "Valid": "Pass",
//          "Time": "2021-12-03 10:12:26"
//      }
//  ]
for (int i = 0; i < node.size(); i++) {
    JsonNode recinfo = node.get(i);
    if (!recinfo.findPath("RecordType").isMissingNode()) {
        if (recinfo.get("RecordType").asInt() == 1) { //Only
focus on swipe records

            //new record information
            if
(recinfo.get("Valid").asText().equals("Forbidden")) //no pass
            {
                //???????????????? Issue VirtualCardNO door
opening command according to customer rules ?????????????????????????????????
                int controllerSN = recinfo.get("SN").asInt();
                long vcard = recinfo.get("CardNO").asLong();
                int doorNO = recinfo.get("DoorNO").asInt();
                String inout = recinfo.get("InExit").asText();

                ObjectMapper mapper = new ObjectMapper();
                ObjectNode webcmd = mapper.createObjectNode();
                webcmd.put("jsonrpc", "2.0");
                webcmd.put("method", "RemoteOpenDoor");
                ArrayNode params = webcmd.putArray("params");
                ObjectNode param = mapper.createObjectNode();
                param.put("ControllerSN", controllerSN);
                param.put("DoorNO", doorNO);
                param.put("InExit", inout.equals("In") ? 1 : 2);
                param.put("VirtualCardNO", vcard);
                params.add(param);
                webcmd.put("id", 1001);
                ws.send(webcmd.toString()); //remote open door
with virtual cardNO

                Acswebapi.logInfo(String.format("Issue remote
door opening command for invalid card - SN=%d, swipe card number=%d",
controllerSN, vcard));
            }
        } else if (recinfo.get("RecordType").asInt() == 2) {
//remote open door
            if
(recinfo.get("Description").asText().indexOf("RemoteOpenDoor") > 0) {
                int controllerSN = recinfo.get("SN").asInt();
                long vcard = recinfo.get("CardNO").asLong();
                int doorNO = recinfo.get("DoorNO").asInt();

```

```

        Acswebapi.logInfo(String.format("Remote door
opening successful - card number=%d, SN=%d, door number=%d", vcard,controllerSN,
doorNO));
    }
}
}
else if (!recinfo.findPath("PrimitHexData").isMissingNode())
//Transparent data transmission
{
    //QRCode
    String qrinfoHex = recinfo.get("PrimitHexData").asText();
    String qrinfo =
recinfo.get("ConvertedToString").asText();

    //????????????????????Perform the door opening operation
after analyzing the QR code
    int controllersSN = recinfo.get("SN").asInt();
    int doorNO = recinfo.get("DoorNO").asInt();
    long vcard = 90010020011; //Virtual CardNO
recinfo.get("CardNO").asLong();
    String inout = recinfo.get("InExit").asText();

    ObjectMapper mapper = new ObjectMapper();
    ObjectNode webcmd = mapper.createObjectNode();
    webcmd.put("jsonrpc", "2.0");
    webcmd.put("method", "RemoteOpenDoor");
    ArrayNode params = webcmd.putArray("params");
    ObjectNode param = mapper.createObjectNode();
    param.put("ControllerSN", controllersSN);
    param.put("DoorNO", doorNO);
    param.put("InExit", inout.equals("In") ? 1 : 2);
    param.put("VirtualCardNO", vcard);
    params.add(param);
    webcmd.put("id", 1001);
    ws.send(webcmd.toString()); //remote open door with
virtual cardNO

    Acswebapi.logInfo(String.format("Issue remote door
opening command for QR code - QR code information=%s, VirtualCardNO=%d, SN=%d,
doorNO=%d",qrinfo, vcard, controllersSN, doorNO));
}
}
}
//logInfo((new ObjectMapper()).readTree(info).toPrettyString());
} catch (JsonProcessingException e) {
    //throw new RuntimeException(e);
} catch (ParseException e) {
    //throw new RuntimeException(e);
}
}
}

```

3.1.3.2. remote open door

In the demo, if the property of "Valid" is "Forbidden", the door will be remote opened.

"Issue remote door opening command for invalid card - SN=252123001, swipe card number=2152191830"

The RecordType of the new record is 2 (events). It indicates that the operation is completed successfully.

```
Mon Jan 20 18:15:26 CST 2025 [ {
  "SN" : 252123001,
  "RecordIndex" : 83,
  "RecordType" : 1,
  "Description" : "Denied Access:No PRIVILEGE",
  "CardNO" : 2152191830,
  "DoorNO" : 1,
  "InExit" : "In",
  "Valid" : "Forbidden",|
  "Time" : "2025-01-20 18:15:26"
} ]
Mon Jan 20 18:15:26 CST 2025 Issue remote door opening command for invalid card - SN=252123001, swipe card number=2152191830
Mon Jan 20 18:15:27 CST 2025 [ {
  "SN" : 252123001,
  "RecordIndex" : 84,
  "RecordType" : 2,
  "Description" : "Remote Open Door",
  "CardNO" : 2152191830,
  "DoorNO" : 1,
  "InExit" : "In",
  "Valid" : "Pass",
  "Time" : "2025-01-20 18:15:26"
} ]
```

3.1.3.2.1. post (no reply)

		{ "jsonrpc" : "2.0", "method" : "RemoteOpenDoor", "params" : [{ "ControllerSN" : 252123001, "DoorNO" : 1, "InExit" : 1, "VirtualCardNO" : 2152191830 }], "id" : 1001 }	
jsonrpc	String	"2.0"	const
method	String	RemoteOpenDoor	
params	Object		
ControllerSN	Number	252123001	Controller SN(9 digits)
DoorNO	Number	1	range: 1,2,3,4
InExit	Number	1	range:1,2 =1 "In", =2 "Exit"

VirtualCardNO	Number	2152191830	
id	Number	1001	Can be used as a serial number [synchronous on return]

3.1.3.2.2. new record(RecordType==2)

if pushed record information (RecordType==2), it indicated that the operation is completed successfully.

		[{ "SN" : 252123001, "Index" : 84, "RecordType" : 2, "Description" : "Remote Open Door", "CardNO" : 2152191830, "DoorNO" : 1, "InExit" : "In", "Valid" : "Pass", "Time" : "2025-01-20 18:15:26" }]	
	Arrays		
SN	Number	452123001	Controller SN(9 digits)
Index	Number	84	record index
RecordType	Number	2	range: 1,2,3 =1 swipe record =2 Events(remote open, push button, Door Status, Reboot) =3 Warns(Fire etc.)
Description	String	"Remote Open Door"	
CardNO	Number	2152191830	
DoorNO	Number	1	range: 1,2,3,4
InExit	String	"In"	range: "In","Exit"
Valid	String	"Pass"	range: "Pass","Forbidden"
Time	String	"2025-01-20 18:15:26"	time format: yyyy-MM-dd HH:mm:ss

4. History