

## HWUG Electronic Signing and Approval of UG05 Service Description

them e	HWUG Electronic Signing and Approval of UG05 Service Description
vers ion	V1.0.0.2
cont ent	HWUG Electronic Signing and Approval of UG05 Service Description
creation time	February 21, 2020
founde r	Leon
turnover time	September 17, 2021

### Document change record

Changed by	date	Change content

### Main review opinions of documents

Product group

reviewers	date	idea

QA group

reviewers	date	idea

1

--	--	--

## development platform

### Development environment and tools

tool	function
Java Script	Demo sample development language

### SDK

Document name	Function
HWUG05SignServer.exe	Sign and approve plug-in service

## HWUG05SignServer plug-in environment support

system	Windows XP 、 Windows 7 、 Windows 8 、 Windows 10
browser	IE8 and above、 chrome 、 Firefox etc.
Port number	10001
communication mode	WebSocket

## WebSocket communication command description

1

- Establish a WebSocket connection

```
/******  
*   establish a connection with HWUGSignServer  
*   Parameter: Plug-in service address  
*   wsUri:                                     "ws://127.0.0.1:10001/"  
*****/ new WebSocket(wsUri);
```

2

- Start signing

```
/******  
*   Send the start signing command to the plug-in service.  
*   Parameter: Start signing command  
*   messageStr: { "typeName": "UgeeStartSign", "message":  
{ "penwidth": "5" }}  
*****/ websocket.send(messageStr);
```

3

- Began to hold back fingerprints

```
/******  
*   Send the command to start restraining to the plug-in service.  
*   Parameter: Start to suppress fingerprints.  
*   messageStr:  
{ "typeName": "UgeeStartFinger", "message": { "quality": "50" }}  
*****/ websocket.send(messageStr);
```

4

- Start signing and stamping

```
/******  
*   Send the command to start signing and restrain to the plug-in  
service.  
*   Parameter: Start signing and suppress fingerprints.  
*   messageStr:
```

```
{“typeName”:”UgeeStartSignFinger”,”message”:{ ”penwidth”:”5”,”quality”:  
  “50”}}*****/  
websocket.send(messageStr);
```

---

5

● Close the signature

```
/******  
*   Send the command to close the signature to the plug-in service.  
*   Parameter: Close the signing command  
*   messageStr:      { “typeName ” : ” UgeeCloseSign ” }  
*****/ websocket.send(messageStr);
```

---

6

● Close link

```
/******  
*   send the close link command to the plug-in service.  
*   Parameter: Close Link Command  
*   messageStr:    { “typeName ” : ” UgeeCloseSocket ” }  
*   Return value:  
*****/ websocket.send(messageStr);
```

---

7

● Get Finger ANSITemplate

```
/******  
  
*   Send the command to Get Finger AnsITemplate  
*   messageStr:  { "typeName " : " UgeeGetANSITemplate " }  
*   Return value: ANSITemplate  
*   *****/ websocket.send(messageStr);
```

---

---

8

● Get Finger ISOTemplate

```
/******  
  
*   Send the command to Get Finger ISOTemplate  
*   messageStr:  { "typeName " : " UgeeGetISOTemplate " }  
*   Return value: ISOTemplate  
*****/ websocket.send(messageStr);
```

---

---

9

● Compare Finger Templates

```
/******  
  
*   Send the command to Compare Finger Templates  
*   messageStr:  { "typeName " : " UgeeCompareTemplates " " message " :  
{ " template1" : "template1Value", " template2" : "template2Value" }}  
*   Return value: compare result  
*****/ websocket.send(messageStr);
```

---

## Description of message returned by plug-in service

---

1

---

- Websocket listens for messages returned by plug-in services.

/\*\*\*\*\*

\* receive messages returned by plug-in services.

\* Parameter: Return message packet

\* event: message returned by plug-in service.

JSON string description returned by event.data

1. {"typename": "ugeesignBase64", "message": "signed picture base64 stream"}
2. {"typename": "ugeecancel"} The user canceled the signature.
3. {"typename": "ugeeclear"} User re-sign
4. {"typename": "ugeeok"} User confirms.
5. {"typename": "ugeeunsign", "message": "Please confirm whether to sign"}
6. {"typename": "error", "message": "error message"}

\* Return value:

\*\*\*\*\*/ websocket.onmessage(event);

---

