



Chat API Specification

PRIVATE AND CONFIDENTIAL

This document contains information, which is the property of Cellfind (Pty) Ltd. No part of this document may be reproduced, transmitted or divulged to third parties in any form or by any means, without written permission from the owner.

Prepaid API

SUMMARY

Document Status	FIRST DRAFT – FOR REVIEW
Version:	1.1
Filename:	Cellfind Chat Technical API.doc
Last Revision Date:	12/09/2019
Reviewer(s)	<ul style="list-style-type: none">• Mervyn Fourie

DOCUMENT VERSION CONTROL

Version(s):	Author(s)	Date Issued	Summary of Additions / Changes
1.0	Mervyn Fourie	12/09/2019	New document

DOCUMENT SIGNOFF

The contents of this Functional Specification are accepted and acknowledged by:

NAME	CAPACITY	COMMENTS	SIGNATURE	DATE
Mervyn Fourie	Developer	New document creation		11 Sep 2019
Clayton Barnard	Developer			
Kobus Kleynhans	Developer			
Lilian Kambatika	Development Manager			

Chat API Specification

Contents

1.	LIST OF ABBREVIATIONS	4
2.	DEFINITION OF TERMS.....	4
3.	API ACCESS	4
4.	PRODUCT OVERVIEW	4
4.1	<i>Purpose.....</i>	4
4.2	<i>Product Description</i>	4
4.3	<i>Background</i>	4
5.	API FUNCTION SUMMARY	5
6.	FUNCTIONS	6
6.1	<i>Security.....</i>	6
6.1.1	<i>Login</i>	6
6.2	<i>Register</i>	7
6.2.1	<i>SubscribeIndividual.....</i>	7
6.3	<i>Query.....</i>	9
6.3.1	<i>Departments.....</i>	9
6.3.2	<i>MessagesQueued</i>	10
6.3.3	<i>MemberCard.....</i>	11
6.3.4	<i>UpdateIMRead</i>	12
6.4	<i>Messaging</i>	14
6.4.1	<i>IM</i>	14
6.5	<i>Profile</i>	15
6.5.1	<i>ProfileIndividualUpdate.....</i>	15
6.6	<i>File.....</i>	16
6.6.1	<i>FileUpload.....</i>	16
6.6.2	<i>FileDownload.....</i>	18
7.	APPLICATION FLOW.....	20
7.1	<i>New Message/ File Download/ Device Status Flow</i>	20
7.1.1	<i>New Text Message.....</i>	20
7.1.2	<i>New File Download.....</i>	21
7.1.3	<i>Device Status Update.....</i>	23
8.	APPLICATION LAYOUT AND IMPLEMENTATION OF CONTRACT	24
8.1	<i>Initial Registration</i>	24
8.2	<i>Department List.....</i>	24
8.3	<i>Menu Options.....</i>	25
8.4	<i>Message View</i>	25
8.5	<i>File View</i>	26
8.6	<i>Member Card</i>	26
8.7	<i>Member Card Sharing</i>	27
8.8	<i>Edit Details</i>	27
9.	APPENDIX A.....	28
9.1	<i>Tag Details (Payload)</i>	28
9.2	<i>Tag Details (Response).....</i>	29

Chat API Specification

1. LIST OF ABBREVIATIONS

Abbreviation	Meaning
API	Application Programming Interface
NA	Not Applicable
Contract	Refers to the JSON String that forms the request and response.
ClientID	Refers to the unique Device ID assigned to the application by the Device.

2. DEFINITION OF TERMS

Term	Definition
Web Service	A software system designed to support interoperable machine-to-machine interaction over a network.

3. API ACCESS

Dev Server

<https://qa-mobile.cellfind.co.za/mag/>

Production Server

<https://mobile.cellfind.co.za/mag/service.asmx>

4. PRODUCT OVERVIEW

4.1 PURPOSE

The purpose of this document is to capture and detail functionality, architectural design, logical flow and the method of integration to services exposed by the Chat API.

4.2 PRODUCT DESCRIPTION

Cellfind Chat service.

4.3 BACKGROUND

Cellfind is exposing a Web Service that will expose functions that will allow for integration with its internal products.

Chat API Specification

5. API FUNCTION SUMMARY

The C# web service exposes a number of functions which allows for a third party client to interface into the internal Cellfind products/systems. The web service acts as a router that channels communication between the external party and the internal products/systems.

Method	Description & available payloads
Security	<ul style="list-style-type: none">• Login : Authenticate and initiate a session in order to utilise the methods of this service
Register	<ul style="list-style-type: none">• SubscribeIndividual: This method should be performed after a session token has been received.• This method should be called every time the application opens to ensure the latest device detail is obtained.
Query	<ul style="list-style-type: none">• Departments : This method needs to be called after Register in order to get the list of available department for the member to chat to. This does not have to be called with every request, merely at registration• MessagesQueued : This method can be called at any point after registration and will fetch any messages that may not have been pushed to the application.• MemberCard : This call can be made any time after registration in order to check and pull any member cards that may exist. (This is not required if you do not have a member card)• UpdateImRead : This call is made to mark the message as read once the message has been viewed in app.
Message	<ul style="list-style-type: none">• IM : This method is used to start a new message with the Agent Dashboard as most messages would be initiated by the customer.
File	<ul style="list-style-type: none">• FileUpload : This method is used to upload files to the Agent Dashboard. Supported files docx, xlsx, pdf, png, jpg.• FileDownload : This method is used to retrieve files from the Agent Dashboard. Supported files docx, xlsx, pdf, png, jpg.
Profile	<ul style="list-style-type: none">• Profile : This Method will be used in conjunction with the register method to set the toggle state for the user as they have the option to accept or decline incoming messages from the Dashboard

Chat API Specification

6. FUNCTIONS

6.1 SECURITY

6.1.1 LOGIN

Verify the username and password for a specified service and return a session token if authentication is successful. This session token is required to activate the rest of the functionality on the interface. NB: Session Tokens are Valid for 24 hours so please ensure that your logic caters for a login retry should the token not be valid on the Methods that follow.

Payload Structure

```
{
  "ServiceRequest": {
    "Application": "demo", ("Will be specified on the creation of the account as this is a unique Identifier for the application.")
    "Category": "Security",
    "DeviceID": "76C20E26-90CB-404A-A922-33112C8C7DA6",
    "EntryCount": "1",
    "Requests": {
      "Request": {
        "Login": {
          "Password": "Username", ("Will be supplied when setting up the account")
          "Username": "Password" ("Will be supplied when setting up the account")
        },
        "Action": "Execute",
        "DateTime": "2019-09-12T12:29:33.836+02:00",
        "Type": "Login"
      }
    },
    "Version": "2"
  }
}
```

The response will have two success results, the result inside would be for the specific method called which would return true or false this will also contain a message with a result for the action performed. The result on the outside would be a result pertaining to the entire contract, in some cases this result will be set to true meaning the JSON contract parsed but the inner request may have failed for a specific reason.

Response Structure

```
{
  "ServiceResponse": {
    "Version": "2",
    "Category": "Security",
    "Application": "demo",
    "DeviceID": "76C20E26-90CB-404A-A922-33112C8C7DA6",
    "EntryCount": "1",
    "Responses": {
      "Response": {
        "DateTime": "2019-09-12T12:29:32.4806022+02:00",
        "Reference": "RPu1xY3f1BQVVRXJ5Z694twpXicMj9ZOCm4omKiM",
        "Type": "Login",
        "Result": {
          "Success": "true",
          "Code": "1",
          "Message": "Login Successful"
        }
      }
    },
    "Success": "true"
  }
}
```

Chat API Specification

6.2 REGISTER

6.2.1 SUBSCRIBEINDIVIDUAL

This contract and Method used for both device registration and some details pertaining to the Individual that will be subscribed. Additional validation will be required in your application to validate Telephone numbers and to Validate email addresses so that these conform to their respective standards. More importantly we will require the DeviceID which is stipulated in two sections of the below JSON Contract and the Token which will be used for pushing to the respective pushing services. All Values in this contract are required.

Payload Structure

```
{
  "ServiceRequest": {
    "Application": "demo",
    "Category": "Register",
    "DeviceID": "76C20E26-90CB-404A-A922-33112C8C7DA6",
    "EntryCount": "1",
    "Requests": {
      "Request": {
        "SubscribeIndividual": {
          "Individual": {
            "Device": {
              "ID": "76C20E26-90CB-404A-A922-33112C8C7DA6",
              "Locale": "en-ZA",
              "Manufacturer": "Mervyn's iPhone",
              "Model": "iPhone Xr",
              "OS": "Apple",
              "Platform": "12.4.1",
              "TimeZone": "Africa/Johannesburg",
              "Token": "ExponentPushToken[-PlpkZM8ZwpU3ZhM-fmax4]"
            }
            If Exponent is used the Token Value will sent as seen below
            If Firebase is used the Token Value will sent as seen below
            "Token": "FireBasePushToken[cbDqpbDCiQw:APA91bGr-B6N18QkVWVjKy0ZmwpKTngqrd3U4-6D0NPcQAwWnRn9yKrDBf49sPfeYtoyI6MryVpw4fNj5JdewJU1qADVMRBV9L0K0C0RMOZeJMAj6iIS0nymyCcv9ShYJP57OzCWQIzLlF]"
          },
          "Email": "Mervyn@cellfind.co.za",
          "LastName": "Fourie",
          "MSISDN": "0821234867",
          "MetaData": {
            "Entries": {
              "Entry": [
                {
                  "Name": "MemberNo",
                  "Value": "Qamervyn"
                },
                {
                  "Name": "BeneficiaryNo",
                  "Value": "00"
                }
              ]
            }
          },
          "Name": "Mervyn"
        }
      }
    },
    "Action": "Execute",
    "Type": "SubscribeIndividual"
  }
},
"SessionReference": "RPu1xY3f1BQVVRXJ5Z694twpXicMj9ZOCm4omKiM",
"Version": "3"
```

Chat API Specification

```
}  
}
```

The response will have two success results, the result inside would be for the specific method called which would return true or false this will also contain a message with a result for the action performed. The result on the outside would be a result pertaining to the entire contract, in some cases this result will be set to true meaning the JSON contract parsed but the inner request may have failed for a specific reason.

Response Structure

```
{  
  "ServiceResponse": {  
    "Version": "3",  
    "Category": "Register",  
    "Application": "demo",  
    "DeviceID": "76C20E26-90CB-404A-A922-33112C8C7DA6",  
    "SessionReference": "RPu1xY3f1BQVVRXJ5Z694twpXicMj9ZOCm4omKiM",  
    "EntryCount": "1",  
    "Responses": {  
      "Response": {  
        "Type": "SubscribeIndividual",  
        "SubscribeIndividual": {  
          "Individual": {  
            "MSISDN": "0821234867",  
            "Name": "Mervyn",  
            "LastName": "Fourie",  
            "Email": "Mervyn@cellfind.co.za",  
            "MetaData": {  
              "Entries": {  
                "Entry": [  
                  {  
                    "Name": "MemberNo",  
                    "Value": "Qamervyn"  
                  },  
                  {  
                    "Name": "BeneficiaryNo",  
                    "Value": "00"  
                  }  
                ]  
              }  
            }  
          }  
        }  
      },  
      "Device": {  
        "ID": "76C20E26-90CB-404A-A922-33112C8C7DA6",  
        "Type": "SmartPhone",  
        "OS": "Apple",  
        "Token": "ExponentPushToken[-PlpkZM8ZwpU3ZhM-fmax4]",  
        "Manufacturer": "Mervyn's iPhone",  
        "Model": "iPhone Xr",  
        "Locale": "en-ZA",  
        "TimeZone": "Africa/Johannesburg",  
        "Result": { "Success": "true" }  
      },  
      "Result": {  
        "Success": "true",  
        "Message": "Registration Successful"  
      }  
    }  
  }  
},  
  "Success": "true"  
}
```

Chat API Specification

6.3 QUERY

6.3.1 DEPARTMENTS

This method will be used to return the Departments that are available for the specified company.

Payload Structure

```
{
  "ServiceRequest": {
    "Application": "demo",
    "Category": "Query",
    "DeviceID": "76C20E26-90CB-404A-A922-33112C8C7DA6",
    "EntryCount": "1",
    "Requests": {
      "Request": {
        "Info": "Departments",
        "MetaData": {

        },
        "Action": "Execute",
        "Type": "GetInfo"
      }
    },
    "SessionReference": "RPu1xY3f1BQVVRXJ5Z694twpXicMj9ZOCm4omKiM",
    "Version": "1"
  }
}
```

The response will have two success results, the result inside would be for the specific method called which would return true or false this will also contain a message with a result for the action performed. The result on the outside would be a result pertaining to the entire contract, in some cases this result will be set to true meaning the JSON contract parsed but the inner request may have failed for a specific reason.

Response Structure

```
{
  "ServiceResponse": {
    "Version": "1",
    "Category": "Query",
    "Application": "demo",
    "DeviceID": "76C20E26-90CB-404A-A922-33112C8C7DA6",
    "SessionReference": "RPu1xY3f1BQVVRXJ5Z694twpXicMj9ZOCm4omKiM",
    "EntryCount": "1",
    "Responses": {
      "Response": {
        "Type": "GetInfo",
        "Info": "Departments",
        "InfoList": {
          "Info": {
            "Entries": {
              "Entry": [
                {
                  "Name": "im_inbox",
                  "Value": "Inbox"
                },
                {
                  "Name": "dem",
                  "Value": "Demo"
                }
              ]
            }
          }
        }
      }
    }
  }
}
```

Chat API Specification

```
    }
  }
},
"Result": {
  "Success": "true"
}
},
"Success": "true"
}
}
```

6.3.2 MESSAGESQUEUED

This method will be used to retrieve messages that would not have been fetched automatically. This would normally be triggered by a Silent push. An example of payload can be seen below. An example of this flow is described in 7.1.1 of this document.

Payload Structure

```
{
  "ServiceRequest": {
    "Application": "demo",
    "Category": "Query",
    "DeviceID": "76C20E26-90CB-404A-A922-33112C8C7DA6",
    "EntryCount": "1",
    "Requests": {
      "Request": {
        "Info": "MessagesQueued",
        "MetaData": {

        }
      },
      "Action": "Execute",
      "Type": "GetInfo"
    }
  },
  "SessionReference": "RPu1xY3f1BQVVRXJ5Z694twpXicMj9ZOCm4omKiM",
  "Version": "1"
}
```

The response will have two success results, the result inside would be for the specific method called which would return true or false this will also contain a message with a result for the action performed. The result on the outside would be a result pertaining to the entire contract, in some cases this result will be set to true meaning the JSON contract parsed but the inner request may have failed for a specific reason.

Response Structure

```
{
  "ServiceResponse": {
    "Version": "1",
    "Category": "Query",
    "Application": "Demo",
    "DeviceID": "76C20E26-90CB-404A-A922-33112C8C7DA6",
    "SessionReference": "RPu1xY3f1BQVVRXJ5Z694twpXicMj9ZOCm4omKiM",
    "EntryCount": "1",
    "Responses": {
      "Response": {
        "Type": "GetInfo",
        "Info": "MessagesQueued",
        "InfoList": {
          "Info": {
            "Entries": {
              "Entry": {
```

Chat API Specification

```
{
  {
    "Name": "Chat_ID",
    "Value": "0"
  },
  {
    "Name": "gen",
    "Value": "Thank you for using the instant messaging facility. Please take note that this facility closed at 16:00 and will re-
open at 8:30 the next working day. "
  }
]
}
},
"Result": { "Success": "true" }
}
},
"Success": "true"
}
}
```

6.3.3 MEMBERCARD

This method will be used to retrieve the member card for an individual if the specified company has member cards loaded on our content gateway. If you do not have member cards for your users this won't be required.

Payload Structure

```
{
  "ServiceRequest": {
    "Application": "demo",
    "Category": "Query",
    "DeviceID": "76C20E26-90CB-404A-A922-33112C8C7DA6",
    "EntryCount": "1",
    "Requests": {
      "Request": {
        "Info": "MemberCard",
        "MetaData": {
        },
      },
      "Action": "Execute",
      "Type": "GetInfo"
    }
  },
  "SessionReference": "RPu1xY3f1BQVVRXJ5Z694twpXicMj9ZOCm4omKiM",
  "Version": "1"
}
```

The response will have two success results, the result inside would be for the specific method called which would return true or false this will also contain a message with a result for the action performed. The result on the outside would be a result pertaining to the entire contract, in some cases this result will be set to true meaning the JSON contract parsed but the inner request may have failed for a specific reason.

Response Structure

```
{
  "ServiceResponse": {
    "Version": "1",
    "Category": "Query",
    "Application": "demo",
    "DeviceID": "76C20E26-90CB-404A-A922-33112C8C7DA6",
    "SessionReference": "RPu1xY3f1BQVVRXJ5Z694twpXicMj9ZOCm4omKiM",
    "EntryCount": "1",
  }
}
```

Chat API Specification

```
        "Name": "chat_id",
        "Value": "rkZ5iSxjAIJ3rbAI5HzaRRT97R7m2yT"
      }
    },
    "Action": "Execute",
    "Type": "GetInfo"
  }
},
{
  "Request": {
    "Info": "UpdateIMRead",
    "MetaData": {
      "Entries": {
        "Entry": {
          "Name": "chat_id",
          "Value": "3z17tfo3bksukzbd7gmdsxtyl1jonh"
        }
      }
    },
    "Action": "Execute",
    "Type": "GetInfo"
  }
},
"SessionReference": "RPu1xY3f1BQVVRXJ5Z694twpXicMj9ZOCm4omKiM",
"Version": "2"
}
}
```

The response will have two success results, the result inside would be for the specific method called which would return true or false this will also contain a message with a result for the action performed. The result on the outside would be a result pertaining to the entire contract, in some cases this result will be set to true meaning the JSON contract parsed but the inner request may have failed for a specific reason.

Response Structure

```
{
  "ServiceResponse": {
    "Version": "2",
    "Category": "Query",
    "Application": "demo",
    "DeviceID": "76C20E26-90CB-404A-A922-33112C8C7DA6",
    "SessionReference": "RPu1xY3f1BQVVRXJ5Z694twpXicMj9ZOCm4omKiM",
    "EntryCount": "1",
    "Responses": {
      "Response": {
        "Type": "GetInfo",
        "Info": "UpdateIMRead",
        "Result": {
          "Success": "true",
          "Message": "Message Status updated to Read"
        }
      }
    },
    "Success": "true"
  }
}
```

Chat API Specification

6.4 MESSAGING

6.4.1 IM

This method is used to initiate a chat message to the dashboard. The request will need to contain the department that the user wishes to chat to. The response will contain the unique chat identifier used on the agent dashboard. This identifier will be used to update the message state on the dashboard when calling the Query/UpdateIMRead method in 6.3.4 of this document.

Payload Structure

```
{
  "ServiceRequest": {
    "Application": "demo",
    "Category": "Messaging",
    "DeviceID": "76C20E26-90CB-404A-A922-33112C8C7DA6",
    "EntryCount": "1",
    "Requests": {
      "Request": {
        "IM": {
          "Message": "Test",
          "MetaData": {
            "Entries": {
              "Entry": {
                "Name": "Department",
                "Value": "dem"
              }
            }
          }
        },
        "Reference": "bf015725-6353-42c9-89da-80c5032f0acd",
        "Source": "PHONE"
      },
      "Action": "Execute",
      "Type": "IM"
    }
  },
  "SessionReference": "RPu1xY3f1BQVVRXJ5Z694twpXicMj9ZOCm4omKiM",
  "Version": "1"
}
```

The response will have two success results, the result inside would be for the specific method called which would return true or false this will also contain a message with a result for the action performed. The result on the outside would be a result pertaining to the entire contract, in some cases this result will be set to true meaning the JSON contract parsed but the inner request may have failed for a specific reason.

Response Structure

```
{
  "ServiceResponse": {
    "Version": "1",
    "Category": "Messaging",
    "Application": "demo",
    "DeviceID": "76C20E26-90CB-404A-A922-33112C8C7DA6",
    "SessionReference": "RPu1xY3f1BQVVRXJ5Z694twpXicMj9ZOCm4omKiM",
    "EntryCount": "1",
    "Responses": {
      "Response": {
        "Type": "IM",
        "Result": {
          "Success": "true",
          "Message": "Instant Message Sent to Scheme"
        }
      }
    }
  }
}
```

Chat API Specification

```
"IM": {
  "Source": "PHONE",
  "Reference": "bf015725-6353-42c9-89da-80c5032f0acd",
  "Message": "Test",
  "MetaData": {
    "Entries": {
      "Entry": [
        {
          "Name": "Department",
          "Value": "dem"
        },
        {
          "Name": "ChatID",
          "Value": "rkZ5iSxjAlJ3rbAl5HzaRRT97R7m2yT"
        }
      ]
    }
  }
},
"Success": "true"
}
```

6.5 PROFILE

6.5.1 PROFILEINDIVIDUALUPDATE

This contract is used to accept or decline chat messages that would be initiated by the agent from the dashboard. This is mandatory for this application and would be initiated at any point after the Register method has been called.

Payload Structure

```
{
  "ServiceRequest": {
    "Application": "demo",
    "Category": "Profile",
    "DeviceID": "76C20E26-90CB-404A-A922-33112C8C7DA6",
    "EntryCount": "1",
    "Requests": {
      "Request": {
        "ProfileIndividualUpdate": {
          "Email": "Mervyn@cellfind.co.za",
          "LastName": "Fourie",
          "MSISDN": "0821234867",
          "MetaData": {
            "Entries": {
              "Entry": [
                {
                  "Name": "MemberNumber",
                  "Value": "Qamervyn"
                },
                {
                  "Name": "ToggleState",
                  "Value": "true"
                }
              ]
            }
          },
          "Name": "Mervyn"
        },
        "Action": "Execute",
        "Type": "ProfileIndividualUpdate"
      }
    }
  },
}
```

Chat API Specification

```
"SessionReference": "RPu1xY3f1BQVVRXJ5Z694twpXicMj9ZOCm4omKiM",
"Version": "2"
}
}
```

The response will have two success results, the result inside would be for the specific method called which would return true or false this will also contain a message with a result for the action performed. The result on the outside would be a result pertaining to the entire contract, in some cases this result will be set to true meaning the JSON contract parsed but the inner request may have failed for a specific reason.

Response Structure

```
{
  "ServiceResponse": {
    "Version": "2",
    "Category": "Profile",
    "Application": "demo",
    "DeviceID": "76C20E26-90CB-404A-A922-33112C8C7DA6",
    "SessionReference": "RPu1xY3f1BQVVRXJ5Z694twpXicMj9ZOCm4omKiM",
    "EntryCount": "1",
    "Responses": {
      "Response": {
        "Type": "ProfileIndividualUpdate",
        "ProfileIndividualUpdate": {
          "MSISDN": "0821234867",
          "Email": "Mervyn@cellfind.co.za",
          "LastName": "Fourie",
          "Result": { "Success": "true" }
        },
        "Result": {
          "Success": "true",
          "Message": "ProfileUpdated Successfully for Device Identifier 76C20E26-90CB-404A-A922-33112C8C7DA6"
        }
      }
    },
    "Success": "true"
  }
}
```

6.6 FILE

6.6.1 FILEUPLOAD

This contract will be used to upload files from the application to the agent dashboard. The file needs to be passed as a base64 string from the application. This will be passed as Data please see part marked in yellow. Supported file types on the dashboard are as follows (.doc, .docx, .xls, .xlsx, pdf, .png, .jpg).

Payload Structure

```
{
  "ServiceRequest": {
    "Application": "FED",
    "Category": "File",
    "DeviceID": "da752725-38f0-4323-ab63-3ec584bad468",
    "EntryCount": "1",
    "Requests": {
      "Request": {
        "FileUpload": {
          "FileList": {
            "Files": {
              "File": {
                "Compressed": "false",
                "Data": "Base64 String",
              }
            }
          }
        }
      }
    }
  }
}
```

Chat API Specification

```
"Extension": "jpg",
"Hash": "b9b13348e0c8e4da5bed3b401605cfe7",
"MIMEType": "image/jpeg",
"Parts": {
  "Hash": "b9b13348e0c8e4da5bed3b401605cfe7",
  "Part": "1",
  "Total": "1"
},
"Size": "75709"
}
},
"MetaData": {
  "Entries": {
    "Entry": {
      "Name": "Department",
      "Value": "gen"
    }
  }
},
"Reference": "d33a148b-a65d-4704-b9cb-4900279bd7dc",
"Source": "PHONE"
},
"Action": "Execute",
"Type": "FileUpload"
}
},
"SessionReference": "34ZObl8NGHki2d2P2HpSkF7SKowhbEhyMYNKtrZw",
"Version": "1"
}
}
```

The response will have two success results, the result inside would be for the specific method called which would return true or false this will also contain a message with a result for the action performed. The result on the outside would be a result pertaining to the entire contract, in some cases this result will be set to true meaning the JSON contract parsed but the inner request may have failed for a specific reason.

Response Structure

```
{
  "ServiceResponse": {
    "Version": "1",
    "Category": "File",
    "Application": "FED",
    "DeviceID": "da752725-38f0-4323-ab63-3ec584bad468",
    "SessionReference": "34ZObl8NGHki2d2P2HpSkF7SKowhbEhyMYNKtrZw",
    "EntryCount": "1",
    "Responses": {
      "Response": {
        "Type": "FileUpload",
        "FileUpload": {
          "Source": "PHONE",
          "Reference": "d33a148b-a65d-4704-b9cb-4900279bd7dc",
          "FileList": {
            "Files": {
              "File": {
                "ID": "9ccd5330-348a-42c5-a3e3-d0ca3b17f408",
                "Extension": "jpg",
                "MIMEType": "image/jpeg",
                "Size": "75709",
                "Hash": "b9b13348e0c8e4da5bed3b401605cfe7",
                "Compressed": "false",
                "Parts": {
                  "Part": "1",
                  "Total": "1",
```

Chat API Specification

```
    "Hash": "b9b13348e0c8e4da5bed3b401605cfe7"
  },
  "Result": {
    "Success": "true",
    "Message": "File Successfully Sent"
  }
},
"MetaData": {
  "Entries": {
    "Entry": [
      {
        "Name": "Department",
        "Value": "gen"
      },
      {
        "Name": "ChatID",
        "Value": "muWEWtnNY3BTZ10rRGj91tcquAcy077"
      }
    ]
  }
},
"Result": {
  "Success": "true",
  "Message": "File Sent"
}
},
"Success": "true"
}
```

6.6.2 FILEDOWNLOAD

This contract will be used to retrieve images sent from the dashboard. The files will be received as a base64 string and will need to be rendered by your application. This will be initiated when new messages are downloaded and the message from the dashboard contains certain parameters. An example of this flow is described in 7.1.2 of this document.

Payload Structure

```
{
  "ServiceRequest": {
    "Application": "demo",
    "Category": "File",
    "DeviceID": "76C20E26-90CB-404A-A922-33112C8C7DA6",
    "EntryCount": "1",
    "Requests": {
      "Request": {
        "FileDownload": {
          "FileList": {
            "Files": {
              "File": {
                "Compressed": "false",
                "FileName": "Snip20190909_11.png.png",
                "ID": "dc91f394-135a-4851-af59-79c5a5e6596e",
                "MIMEType": "image/png",
                "Parts": {
                  "Part": "1",
                  "Total": "1"
                },
                "Size": "65053"
              }
            }
          }
        }
      }
    }
  },
}
```

Chat API Specification

```
"MetaData": {
  "Entries": {
    "Entry": {
      "Name": "ChatID",
      "Value": "rkz5isxjalj3rbal5hzarrt97r7m2yt"
    }
  }
},
"Reference": "29345086-7c6e-bd29-01ac-dd664395421e",
"Source": "PHONE"
},
"Action": "Execute",
"Type": "FileDownload"
}
},
"SessionReference": "RPu1xY3f1BQVVRXJ5Z694twpXicMj9ZOCm4omKiM",
"Version": "1"
}
}
```

The response will have two success results, the result inside would be for the specific method called which would return true or false this will also contain a message with a result for the action performed. The result on the outside would be a result pertaining to the entire contract, in some cases this result will be set to true meaning the JSON contract parsed but the inner request may have failed for a specific reason.

Response Structure

```
{
  "ServiceResponse": {
    "Version": "1",
    "Category": "File",
    "Application": "demo",
    "DeviceID": "76C20E26-90CB-404A-A922-33112C8C7DA6",
    "SessionReference": " RPu1xY3f1BQVVRXJ5Z694twpXicMj9ZOCm4omKiM",
    "EntryCount": "1",
    "Responses": {
      "Response": {
        "Type": "FileDownload",
        "FileDownload": {
          "Source": "PHONE",
          "Reference": "29345086-7c6e-bd29-01ac-dd664395421e ",
          "FileList": {
            "Files": {
              "File": {
                "ID": " dc91f394-135a-4851-af59-79c5a5e6596e",
                "FileName": "sample",
                "Extension": "png",
                "MimeType": "image/png",
                "Size": "48213",
                "Hash": "78865c7cbef4785711353515b8b266b6",
                "Data": "iVBORw0KGgoAAAANSUHEUgAAAJQAAACWCAYAAAA",
                "Compressed": "false",
                "Parts": {
                  "Part": "1",
                  "Total": "1"
                }
              },
              "Result": { "Success": "true" }
            }
          }
        }
      },
      "MetaData": {
        "Entries": {
          "Entry": {
            "Name": "ChatID",
            "Value": " rkz5isxjalj3rbal5hzarrt97r7m2yt"
          }
        }
      }
    }
  }
}
```

Chat API Specification

```
    }
  }
},
"Result": {
  "Success": "true",
  "Message": "File Downloaded"
}
},
"Success": "true"
}
```

7. APPLICATION FLOW

7.1 NEW MESSAGE/ FILE DOWNLOAD/ DEVICE STATUS FLOW

7.1.1 NEW TEXT MESSAGE

On receiving a new message from the agent Dahnboard a push message is sent to the device with the following payload. Take note of the cmd "FetchMessage" as this is what you will use to trigger a new request to query for messages.

PushMessage Received

```
DataObject {
  "cmd": "FetchMessage",
  "ref": "2281351"
}
```

Initiate Query to get queued messages

```
{
  "ServiceRequest": {
    "Application": "demo",
    "Category": "Query",
    "DeviceID": "76C20E26-90CB-404A-A922-33112C8C7DA6",
    "EntryCount": "1",
    "Requests": {
      "Request": {
        "Info": "MessagesQueued",
        "MetaData": {

        },
      },
      "Action": "Execute",
      "Type": "GetInfo"
    }
  },
  "SessionReference": "RPu1xY3f1BQVVRXJ5Z694twpXicMj9ZOCm4omKiM",
  "Version": "1"
}
```

Response

```
{
  "ServiceResponse": {
    "Version": "1",
    "Category": "Query",
    "Application": "Demo",
    "DeviceID": "76C20E26-90CB-404A-A922-33112C8C7DA6",
    "SessionReference": "RPu1xY3f1BQVVRXJ5Z694twpXicMj9ZOCm4omKiM",
    "EntryCount": "1",
    "Responses": {
      "Response": {
```

Chat API Specification

```
"Type": "GetInfo",
"Info": "MessagesQueued",
"InfoList": {
  "Info": {
    "Entries": {
      "Entry": [
        {
          "Name": "Chat_ID",
          "Value": "0"
        },
        {
          "Name": "gen",
          "Value": "Thank you for using the instant messaging facility. Please take note that this facility closed at 16:00 and will re-
open at 8:30 the next working day. "
        }
      ]
    }
  }
},
"Result": { "Success": "true" }
},
"Success": "true"
}
```

7.1.2 NEW FILE DOWNLOAD

On receiving a file from the agent Dabsboard a push message is sent with the following payload. Take note of the cmd "FetchMessage" as this is what you will use to trigger a new request to query for messages.

PushMessage Received

```
DataObject {
  "cmd": "FetchMessage",
  "ref": "2281351"
}
```

Initiate Query to get queued messages

```
{
  "ServiceRequest": {
    "Application": "demo",
    "Category": "Query",
    "DeviceID": "76C20E26-90CB-404A-A922-33112C8C7DA6",
    "EntryCount": "1",
    "Requests": {
      "Request": {
        "Info": "MessagesQueued",
        "MetaData": {

        },
        "Action": "Execute",
        "Type": "GetInfo"
      }
    },
    "SessionReference": "RPu1xY3f1BQVVRXJ5Z694twpXicMj9ZOCm4omKiM",
    "Version": "1"
  }
}
```

The response in this case would be used as a trigger a file download and would also be used to build your FileDownload Request.

```
{
```

Chat API Specification

```
"ServiceResponse": {
  "Version": "1",
  "Category": "Query",
  "Application": "demo",
  "DeviceID": "76C20E26-90CB-404A-A922-33112C8C7DA6",
  "SessionReference": "RPu1xY3f1BQVVRXJ5Z694twpXicMj9ZOCm4omKiM",
  "EntryCount": "1",
  "Responses": {
    "Response": {
      "Type": "GetInfo",
      "Info": "MessagesQueued",
      "InfoList": {
        "Info": {
          "Entries": {
            "Entry": [
              {
                "Name": "Chat_ID",
                "Value": "rkz5isxjalj3rbal5hzarrt97r7m2yt"
              },
              {
                "Name": "[IM_FILEUPLOAD]",
                "Value": "[IM_FILEUPLOAD]{\"DEPARTMENT\": \"dem\", \"IM_FILEUPLOAD\": {\"ID\": \"dc91f394-135a-4851-af59-79c5a5e6596e\", \"FileName\": \"Snip20190909_11.png.png\", \"MIMEType\": \"image/png\", \"Size\": \"65053\"}}}"
              }
            ]
          }
        }
      },
      "Result": { "Success": "true" }
    }
  },
  "Success": "true"
}
```

FileDownload Request built using paramters passed in the queued message response as seen above.

```
{
  "ServiceRequest": {
    "Application": "demo",
    "Category": "File",
    "DeviceID": "76C20E26-90CB-404A-A922-33112C8C7DA6",
    "EntryCount": "1",
    "Requests": {
      "Request": {
        "FileDownload": {
          "FileList": {
            "Files": {
              "File": {
                "Compressed": "false",
                "FileName": "Snip20190909_11.png.png",
                "ID": "dc91f394-135a-4851-af59-79c5a5e6596e",
                "MIMEType": "image/png",
                "Parts": {
                  "Part": "1",
                  "Total": "1"
                },
                "Size": "65053"
              }
            }
          },
          "MetaData": {
            "Entries": {
              "Entry": {
                "Name": "ChatID",
                "Value": "rkz5isxjalj3rbal5hzarrt97r7m2yt"
              }
            }
          }
        }
      }
    }
  }
}
```

Chat API Specification

```
    }
  }
},
"Reference": "29345086-7c6e-bd29-01ac-dd664395421e",
"Source": "PHONE"
},
"Action": "Execute",
"Type": "FileDownload"
}
},
"SessionReference": "RPu1xY3f1BQVVRXJ5Z694twpXicMj9ZOCm4omKiM",
"Version": "1"
}
}
```

7.1.3 DEVICE STATUS UPDATE

On receiving a new message from the agent Dashboard a push message is sent to the device with the following payload. Take note of the cmd "FetchMessage" as this is what you will use to trigger a new request to query for messages.

PushMessage Received

```
DataObject {
  "cmd": "FetchMessage",
  "ref": "2281351"
}
```

Initiate Query to get queued messages

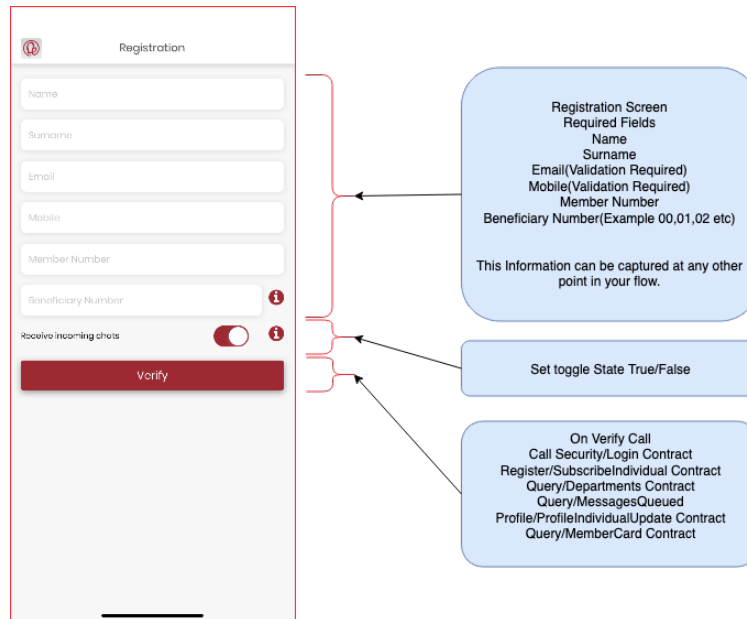
```
{
  "ServiceRequest": {
    "Application": "demo",
    "Category": "Query",
    "DeviceID": "76C20E26-90CB-404A-A922-33112C8C7DA6",
    "EntryCount": "1",
    "Requests": {
      "Request": {
        "Info": "MessagesQueued",
        "MetaData": {

        },
      },
      "Action": "Execute",
      "Type": "GetInfo"
    }
  },
  "SessionReference": "RPu1xY3f1BQVVRXJ5Z694twpXicMj9ZOCm4omKiM",
  "Version": "1"
}
```

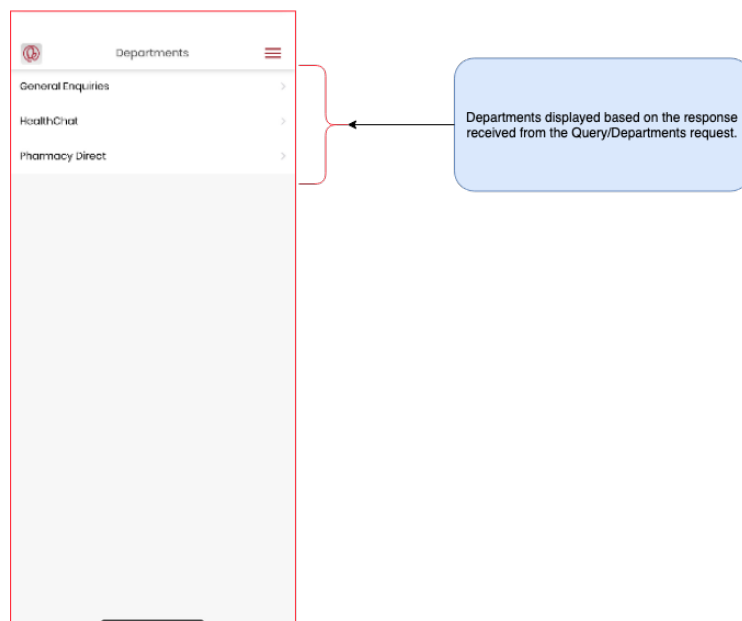
Chat API Specification

8. APPLICATION LAYOUT AND IMPLEMENTATION OF CONTRACT

8.1 INITIAL REGISTRATION

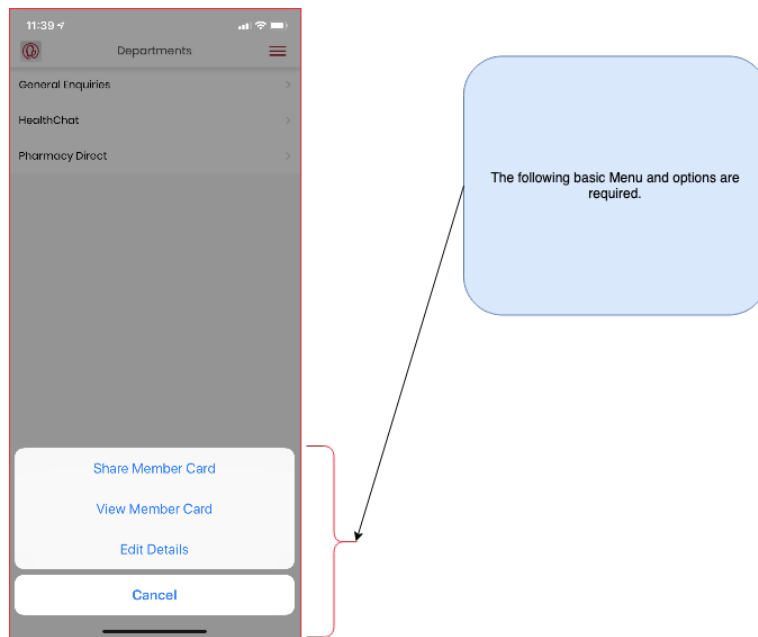


8.2 DEPARTMENT LIST

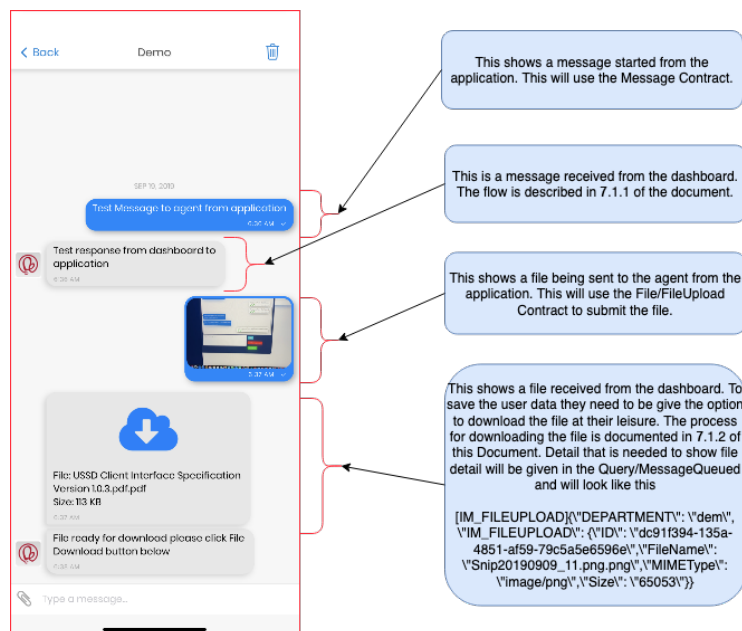


Chat API Specification

8.3 MENU OPTIONS

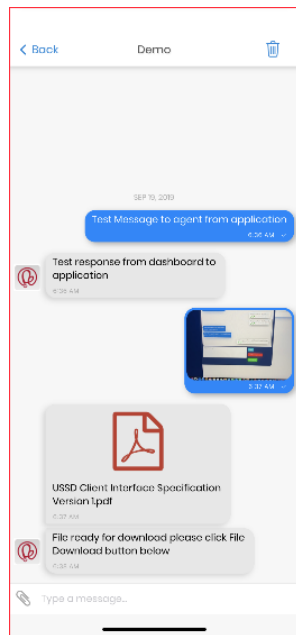


8.4 MESSAGE VIEW



Chat API Specification

8.5 FILE VIEW



This shows an example of a downloaded Document. This is represented with an icon of the document type. It would be up to the user to have a viewer on their device to view different file types.

Current supported types
.doc, .docx, .xls, .xlsx, .pdf, .png, .jpg

We have attempted to cater for other file types but this would be a best effort.

8.6 MEMBER CARD

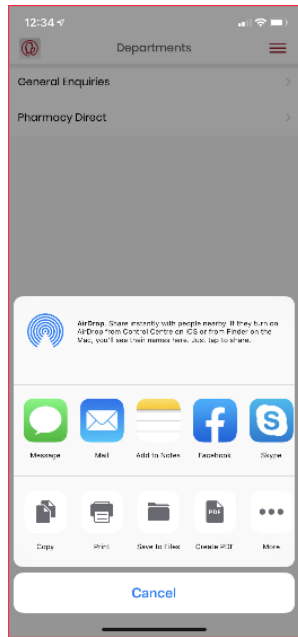


Member cards will be received on the app in html format this will need to be rendered in the application.

The Member card will be received in the Query/MemberCard response.

Chat API Specification

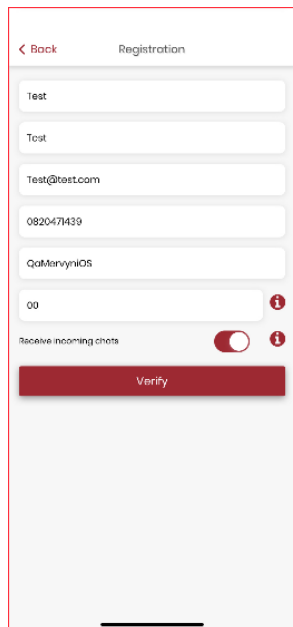
8.7 MEMBER CARD SHARING



Member cards will be received on the app in html format this will need to be rendered in the application.

The Member card will be received in the Query/MemCard response.

8.8 EDIT DETAILS



This screen is presented to the user after registration as a menu option allowing them to change their details at any given time this follows the same flow as the initial registration

Required Fields
 Name
 Surname
 Email(Validation Required)
 Mobile(Validation Required)
 Member Number
 Beneficiary Number(Example 00,01,02 etc)

This Information can be captured at any other point in your flow.

Set toggle State True/False

On Verify Call
 Call Security/Login Contract
 Register/SubscribeIndividual Contract
 Query/Departments Contract
 Query/MessagesQueued
 Profile/ProfileIndividualUpdate Contract
 Query/MemCard Contract

Chat API Specification

9. APPENDIX A

9.1 TAG DETAILS (PAYLOAD)

Type	Name	Description
Payload	Version	Contract/Payload version. <i>(Provided by Cellfind : default = 1)</i>
Payload	Category	Cellfind backend system identifier. <i>(Provided by Cellfind : Security, Register, Query, Profile, Messaging, File)</i>
Payload	Application	Client application identifier. <i>(Provided by Cellfind : default =)</i>
Payload	DeviceID	Device Identifier obtained from the OS.
Payload	SessionReference	The SessionToken supplied on Login. <i>(Generated by Cellfind[Login])</i>
Payload	EntryCount	Number of transaction entries (Request tags) for the payload.
Payload	Requests	Transaction request set identifier
Payload	Request	Transaction request/record identifier. Each <Request> tag equates to 1 transaction.
Payload	Type	Defines the method to be executed within the specified system. <i>(Provided by Cellfind = Login, Logout, SubscribeIndividual, GetInfo, ProfileIndividualUpdate, IM, FileUpload, FileDownload).</i> <i>GetInfo will have sub methods also provided in each contract that specifies what is being queried (Departments, MessagesQueued, MemberCard, UpdateIMRead)</i>
Payload	Action	Defines the type of action to be performed by the system for the request. <i>(Provided by Cellfind : default = Execute)</i>
Payload	DateTime	Timestamp for the transaction. This can also be used to schedule a transaction.

Chat API Specification

9.2 TAG DETAILS (RESPONSE)

Action	Type	Parameters
Response	Version	Contract/Payload version. <i>(Provided by Cellfind : default = 1)</i>
Response	Category	Cellfind backend system identifier. <i>(Provided by Cellfind : Security, Register, Query, Profile, Messaging, File)</i>
Response	Application	Client application identifier. <i>(Provided by Cellfind : default =)</i>
Response	DeviceID	Device Identifier obtained from the OS.
Response	SessionReference	The SessionToken supplied on Login. <i>(Generated by Cellfind[Login])</i>
Response	Success	Global Tag: True/False identifier indicating the status of the whole Payload (True/False).
Response	EntryCount	Number of transaction entries for the payload.
Response	Responses	Transaction response set identifier.
Response	Response	Transaction response/record identifier. Each <Response> tag equates to 1 transaction.
Response	Type	Defines the method to be executed within the specified system. <i>(Provided by Cellfind = Login, Logout, SubscribeIndividual, GetInfo, ProfileIndividualUpdate, IM, FileUpload, FileDownload).</i> <i>GetInfo will have sub methods also provided in each contract that specifies what is being queried (Departments, MessagesQueued, MemberCard, UpdateIMRead).</i>
Response	Result	Per transaction result set for a specific transaction.
Response	Success	Result Tag: True/False identifier indicating the status of the specific transaction in the set.
Response	Code	Optional error code, if any.
Response	Message	Optional error message, if any.