

Interfaces

Server Messaging Service

The interface between the server and client is via HTTP requests as part of the REST interface.

The JSON format for data transfer is detailed below.

```
[] : list  
{ } : dictionary
```

```
Message : {  
    timestamp: int,  
    from_user: int,  
    to_user: int,  
    message: string,  
    lat: float,  
    lon: float  
}
```

POST upload	Send a message to another user.
Return	-
Body	Message

Example Request:

POST <http://api.emergencyalarm.com/1/messages/send>

```
{  
    "timestamp": 1376298387,  
    "from_user": 12,  
    "to_user": 14,  
    "message": "I'm OK."  
    "lat": 37.7833,  
    "lon": 144.9667  
}
```

Example Reply:

ACCEPTED 202

Client Messaging Component

The interface to send messages from the client is via a Java API.

The Java API is detailed below.

```
/**
 * API to access message services provided by a remote server.
 */
public class MessageService

/**
 * Send a message to specific user, who is using another device.
 *
 * @param String message The text message to be sent.
 * @param UserIdentifier user The identifier for the recipient
 * of the message.
 */
public static void sendMessageToUser(String message, UserIdentifier
user)
```

Example Java API usage:

```
String messageText = textField.contents();
UserIdentifier carerId = currentUser().primaryCarer().id();
MessageServer.sendMessageToUser(messageText, carerId);
```