GeoReady Web Service Document

Document Version: 1.0.1

Prepared by: YANG Wei

Company: CoordSafe Pte Ltd

Created Date: 25/10/2013

[1. INTRODUCTION 5](#_Toc372212067)

[2. FUNCTIONS 5](#_Toc372212068)

[2.1 User Registration & Login 5](#_Toc372212069)

[2.1.1 Login 5](#_Toc372212070)

[2.1.2 Logout 5](#_Toc372212071)

[2.1.3 Registration 5](#_Toc372212072)

[2.3 Circle Function 6](#_Toc372212073)

[2.3.1 Get circle by login user 6](#_Toc372212074)

[2.3.2 Get simple circle 6](#_Toc372212075)

[2.3.3 Create circle 6](#_Toc372212076)

[2.3.4 Update circle 6](#_Toc372212077)

[2.3.5 Delete circle 6](#_Toc372212078)

[2.3.6 Get circle members & other detail 6](#_Toc372212079)

[2.3.7 Send invite 7](#_Toc372212080)

[2.3.8 Add Ward 7](#_Toc372212081)

[2.3.9 Approve/reject invite 7](#_Toc372212082)

[2.4 Guardian Function 7](#_Toc372212083)

[2.4.1 Update profile 7](#_Toc372212084)

[2.4.2 Update profile picture 7](#_Toc372212085)

[2.4.3 Change password 7](#_Toc372212086)

[2.4.4 Forget password 7](#_Toc372212087)

[2.4.5 ~~Reset password~~ 8](#_Toc372212088)

[2.4.6 Get User Setting – ~~(backup, otherwise stored on device)~~ 8](#_Toc372212089)

[2.4.7 Update setting 8](#_Toc372212090)

[2.5 Ward Function 8](#_Toc372212091)

[2.5.1 Quit circle 8](#_Toc372212092)

[2.5.2 Send panic alarm 8](#_Toc372212093)

[2.5.3 Send check-in 8](#_Toc372212094)

[2.6 Zone & Geofence 9](#_Toc372212095)

[2.6.1 Retrieve zones created by user 9](#_Toc372212096)

[2.6.2 Delete zones created by user 9](#_Toc372212097)

[2.6.3 Retrieve users this zone applied 9](#_Toc372212098)

[2.6.4 Retrieve zones applied to user 9](#_Toc372212099)

[2.6.5 Apply/remove zone to/from ward 9](#_Toc372212100)

[2.6.6 Create circular zone from phone 9](#_Toc372212101)

[2.6.7 Update circular zone 10](#_Toc372212102)

[2.6.8 Enter zone 10](#_Toc372212103)

[2.6.9 Exit zone 10](#_Toc372212104)

[2.7 Social Event 10](#_Toc372212105)

[2.7.1 Create event 10](#_Toc372212106)

[2.7.2 Cancel/delete event 10](#_Toc372212107)

[2.7.3 Invite member to event -> *equivalent to add to circle: 2.3.8* 10](#_Toc372212108)

[2.7.4 Send Apology/Delay arrival 10](#_Toc372212109)

[2.8 Location Update (HTTPGateway) 11](#_Toc372212110)

[2.8.1 Update from Phone 11](#_Toc372212111)

[2.8.2 Update from Device 11](#_Toc372212112)

[2.8.3 Status Update (battery, phone states, etc) 11](#_Toc372212113)

[2.9 Panic Alarm (refer to 2.5.2) 11](#_Toc372212114)

[2.10 Check-in 11](#_Toc372212115)

[2.10.1 Send (stealth) check-in request 11](#_Toc372212116)

[2.11 Low Battery 11](#_Toc372212117)

[2.11.1 Send Low Battery 11](#_Toc372212118)

[2.11.2 Dismiss Low Battery 11](#_Toc372212119)

[2.12 Message (mqtt) 12](#_Toc372212120)

[2.12.1 Broadcast to circle 12](#_Toc372212121)

[2.12.2 Send to single member 12](#_Toc372212122)

[2.12.3 Send to multiple member 12](#_Toc372212123)

[2.12.4 Retrieve message history by circle 12](#_Toc372212124)

[2.12.5 Retrieve message history with certain guardian 12](#_Toc372212125)

[2.12.6 ~~Delete message history~~ 12](#_Toc372212126)

[2.13 Safe Distance 12](#_Toc372212127)

[2.13.1 Set Safe distance to ward 12](#_Toc372212128)

[2.13.2 Update Safe distance 13](#_Toc372212129)

[2.13.3 Cancel Safe distance 13](#_Toc372212130)

[2.14 Locator function 13](#_Toc372212131)

[2.14.1 Get locator info & status 13](#_Toc372212132)

[2.14.2 Get locator’s history between certain time period 13](#_Toc372212133)

[2.14.3 Get locator’s location (lat, lng, speed, alt) at a past time 13](#_Toc372212134)

[2.14.4 Get locator A’s trips (basic: start, end, duration) within certain period 13](#_Toc372212135)

[2.14.5 Get locator’s current trip (start time, destination, etc) 13](#_Toc372212136)

[2.14.6 Get arbitrary trip by Id 13](#_Toc372212137)

[2.14.7 Get trip detail for locator between certain time period 14](#_Toc372212138)

[2.15 Statistic & Report function 14](#_Toc372212139)

[2.15.1 Get locator’s mileage within certain time 14](#_Toc372212140)

[2.15.2 Get locator A’s running time within certain period 14](#_Toc372212141)

[2.16 Event (Activity) 14](#_Toc372212142)

[2.16.1 Retrieve notifications for user C between certain period 14](#_Toc372212143)

# INTRODUCTION

GeoReady service platform provides RESTFul style web service to enable 3rd parties to read/write data to our server, either from web or mobile client.

* Base URL (for dev): <http://137:132.145.228/geoready/api>
* Topic: start with /CsMt
* API Key:
  + Each request (except for login) has to append an API key at the end in order to gain access, each user holds a different API key and application can obtain that key after successful log in
  + Key will be revoked at the server in case of being abused
* Transmission data is in JSON format
* Date and Time: UTC time in “dd-MM-yyyy” and “hh:mi:ss” format

# FUNCTIONS

## User Registration & Login

* + 1. Login
    - URL: /user/login&from
    - Method: POST (HeaderParam: “name”, “password”)
    - Example: /user/login?from=iphone|android
    - Return: UserCredential object -

*{"nick\_name":null,"id":"1","company\_id":"1","company":"CS","login\_name":"admin","addr":null,"contact":"90297718","key":"master-1234qwer"}*

* + 1. Logout
    - URL: /user/logout
    - Method: GET
    - Example:
    - Return: Boolean, true=success, false=fail
    1. Registration
    - URL: /user/create
    - Method: POST (User object)
    - Example:
    - Return: User object
  1. ~~Create/activate device~~
     + ~~URL: /device/add/{deviceId}~~
     + ~~Method: POST (device password)~~
     + ~~Example:~~
     + ~~Return: Device object~~

## Circle Function

* + 1. Get circle by login user
    - URL: /circle/guardian/{guardianId}
    - Method: GET
    - Example: ../circle/guardian/001
    - Return: List<Circle>
    1. Get simple circle
    - URL: /circle/guardian/{guardianId}?type=simple
    - Method: GET
    - Example: ../circle/guardian/001?type=simple
    - Return: List<SimpleCircle>
    1. Create circle
    - URL: /circle
    - Method: POST
    - Example:
    - Return: Circle object
    1. Update circle
    - URL: /circle/{circleId}
    - Method: PUT
    - Example:
    - Return: HTTP 200
    1. Delete circle
    - URL: /circle/{circleId}
    - Method: DELETE
    - Example:
    - Return: HTTP 200
    1. Get circle members & other detail
    - URL: /circle/{circleId}
    - Method: GET
    - Example: ../circle/001
    - Return: Circle object
    1. Send invite
    - URL: /circle/{circleId}/invite/{EmailOrPhone}
    - Method: GET
    - Example: ../circle/001/invite/xxx@yyy.com
    - Return: Invite object
    1. Add Ward
    - URL: /circle/{circleId}/add/{wardId}
    - Method: GET
    - Example: ../circle/001/add/001
    - Return: boolean
    1. Approve/reject invite
    - URL: /invite/{inviteId}/approve|reject
    - Method: GET
    - Example:
    - Return: boolean

## Guardian Function

* + 1. Update profile
    - URL: /user/{userId}
    - Method: PUT (UserInformation object)
    - Example:
    - Return: UserInformation object
    1. Update profile picture
    - URL: /user/{userId}/display
    - Method: POST (picture binary data)
    - Example:
    - Return: User object
    1. Change password
    - URL: /user/{userId}?action=changepass
    - Method: POST (oldPass, newPass)
    - Example:
    - Return: boolean
    1. Forget password
    - URL: /user/{userId}?action=forgetpass
    - Method: GET
    - Example:
    - Return: HTTP 200
    1. ~~Reset password~~
    - ~~URL: /user/{userId}?action=resetpass~~
    - ~~Method: POST (newPass)~~
    - ~~Example:~~
    - ~~Return: boolean~~
    1. Get User Setting – ~~(backup, otherwise stored on device)~~
    - URL: /user/setting/{userId}
    - Method: GET
    - Example:
    - Return: UserSetting object
    1. Update setting
    - URL: /user/setting/{userId}
    - Method: PUT (UserSetting object)
    - Example:
    - Return: User object

## Ward Function

* + 1. Quit circle
    - URL: /circle/{circleId}/quit
    - Method: POST (HeaderParam: “wardId”)
    - Example:
    - Return: HTTP 200
    1. Send panic alarm
    - Topic: /panic/{wardId}
    - Message: PanicAlarm object in JSON
    - Example:
    - Return:
    1. Send check-in
    - Topic: /check-in/{wardId}
    - Message: Location object in JSON
    - Example:
    - Return:

## Zone & Geofence

* + 1. Retrieve zones created by user
    - URL: /zone/guardian/{guardianId}
    - Method: GET
    - Example:
    - Return: List<Geofence>
    1. Delete zones created by user
    - URL: /zone/{zoneId}
    - Method: DELETE
    - Example:
    - Return: HTTP 200
    1. Retrieve users this zone applied
    - URL: /zone/{zoneId}/list
    - Method: GET
    - Example:
    - Return: List<Guardian>
    1. Retrieve zones applied to user
    - URL: /zone/ward/{wardId}
    - Method: GET
    - Example:
    - Return: List<Geofence>
    1. Apply/remove zone to/from ward
    - URL: /zone/{zoneId}/add|remove
    - Method: POST (HeaderParam: “wardId”)
    - Example:
    - Return: HTTP 200
    1. Create circular zone from phone
    - URL: /zone
    - Method: POST (HeaderParam: “centerPoint(double, double)”, “radius”)
    - Example:
    - Return: Geofence object
    1. Update circular zone
    - URL: /zone/{zoneId}
    - Method: PUT
    - Example:
    - Return: HTTP 200
    1. Enter zone
    - Topic: /place/in/{wardId}
    - Message: ?
    - Example:
    - Return:
    1. Exit zone
    - Topic: /place/out/{wardId}
    - Method: ?
    - Example:
    - Return:

## Social Event

* + 1. Create event
    - URL: /social-event/create
    - Method: POST
    - Example:
    - Return: SocialEvent object
    1. Cancel/delete event
    - URL: /social-event/{eventId}
    - Method: DELETE
    - Example:
    - Return: HTTP 200
    1. Invite member to event -> *equivalent to add to circle: 2.3.8*
    2. Send Apology/Delay arrival
    - Topic: /social-event/{eventId}/apology|delay
    - Message: ?
    - Example:
    - Return:

## Location Update (HTTPGateway)

* + 1. Update from Phone
    - URL: /ward/location
    - Method: POST
    - Example:
    - Return: boolean
    1. Update from Device
    2. Status Update (battery, phone states, etc)
    - URL: /ward/location
    - Method: POST
    - Example:
    - Return: boolean

## Panic Alarm (refer to 2.5.2)

## Check-in

* + 1. Send (stealth) check-in request
    - Topic /check-in/request/{wardId}/{stealth}
    - Message: ?
    - Example:
    - Return:

## Low Battery

* + 1. Send Low Battery
    - Topic /battery/{wardId}
    - Message: ? (battery level)
    - Example:
    - Return:
    1. Dismiss Low Battery
    - URL: /battery?action=dismiss
    - Method: POST (headparam: deviceId)
    - Example:
    - Return: boolean

## Message (mqtt)

* + 1. Broadcast to circle
    - Topic: ../chat/{circleId}
    - Message: ChatMessage object in JSON
    - Example:
    - Return:
    1. Send to single member
    - Topic: ../chat/{memberId}
    - Message:
    - Example:
    - Return:
    1. Send to multiple member

*(Iterate each topic for all recipients)*

* + 1. Retrieve message history by circle
    - URL: /message-history/circle/{circleId}/{startDt, endDt}
    - Method: GET
    - Example:
    - Return: List<InstantMessage>
    1. Retrieve message history with certain guardian
    - URL: /message-history/private/{guardianId}/{startDt, endDt}
    - Method: GET
    - Example:
    - Return: List<ChatMessage>
    1. ~~Delete message history~~

## Safe Distance

* + 1. Set Safe distance to ward
    - URL: /safe-distance/ward/{wardId}/{distance(xx m)}?action=add|cancel
    - Method: GET
    - Example:
    - Return: HTTP 200
    1. Update Safe distance
    2. Cancel Safe distance

## Locator function

* + 1. Get locator info & status
    - URL: /locator/{locatorId}
    - Method: GET
    - Example: ../locator/21
    - Return: Locator object
    1. Get locator’s history between certain time period
    - URL: /history/{locatorId}/{start time, end time}
    - Method: GET
    - Example: ../history/18/01-27-2013 09:00,01-27-2013 10:00
    - Return: List<LocatorLocationHistory>
    1. Get locator’s location (lat, lng, speed, alt) at a past time
    - URL: /location/{locatorId}/{time}
    - Method: GET
    - Example: ../Location/18/01-27-2013 09:30
    - Return: LocatorLocationHistory
    1. Get locator A’s trips (basic: start, end, duration) within certain period
    - URL: /trip/{locatorId}/{timeRange}
    - Method: GET
    - Example: ../trip/26/01-27-2013 02:00,01-27-2013 10:00
    - Return: List<Trip>
    1. Get locator’s current trip (start time, destination, etc)
    - URL: /trip-detail/{locatorId}/curOrLast
    - Method: GET
    - Example: ../trip-detail/26/curOrLast
    - Return: TripDetail object
    1. Get arbitrary trip by Id
    - URL: /trip/{tripId}
    - Method: GET
    - Example: ../trip/374683
    - Return: TripDetail
    1. Get trip detail for locator between certain time period
    - URL: /trip-detail/{locatorId}/{timeRange}
    - Method: GET
    - Example: ../trip-detail/26/01-27-2013 02:00,01-27-2013 10:00
    - Return: List<TripDetail>

## Statistic & Report function

* + 1. Get locator’s mileage within certain time
    - URL: /mileage/{locatorId}/{timeRange}
    - Method: GET
    - Example: ../mileage/26/01-27-2013 02:00,01-27-2013 10:00
    - Return: xx km
    1. Get locator A’s running time within certain period
    - URL: /move/{locatorId}/{date}
    - Method: GET
    - Example: ../move/26/01-27-2013
    - Return: xx mins

## Event (Activity)

* + 1. Retrieve notifications for user C between certain period
    - URL: /event/{locatorId}/{date}
    - Method: GET
    - Example: ../event/26/01-27-2013
    - Return: List<Event>