# iTime Developer Documentation

## *Introduction*

This documentation is written by Yuming Shen (Ethan Shin). The objective of writing this documentation is to provide a basic architecture, API and models that are used in the iOS application "**iTime**".

The whole documentation consists of three parts: Architecture, Views and API(models).

## *Tools used in the development*

- **Swift**
  - Swift is the programming language used in this project.
- **Realm**
  - The Object-Oriented Database (OODB) used for our local database.
- **Lean cloud**
  - The mobile backend as a service (BaaS) provider, which manages all functionalities of the notification.
- **Google/Firebase**
  - Several Google APIs have been used in the project
- **Apple developer account/Testflight**
  - Responsible for testing the application
  - other configurations

# *Architecture*

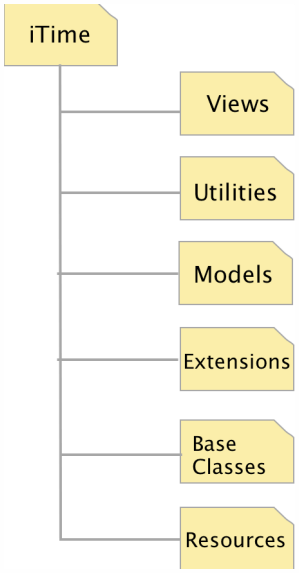



Figure 1: Project Architecture in Xcode

The project architecture in Xcode has been shown in Figure 1. The whole project comprised of six main parts. The Views part includes all view controller files. All utility functions are in Utilities folder and models used in the application are in Models folder. Files in Extensions folder extends some classes with some convenient functions. BaseClasses and Resources contains other important resources for instance, icon images.

---

## Views

### Tabbar ViewController


Figure 2 Tabbar

- There are five main components in the tabbar
  - **Meetings, Activities, Functional Button, Calendar and Contacts.**

- In this project, the tabbar is simply created by using storyboard (**Main.storyboard**)

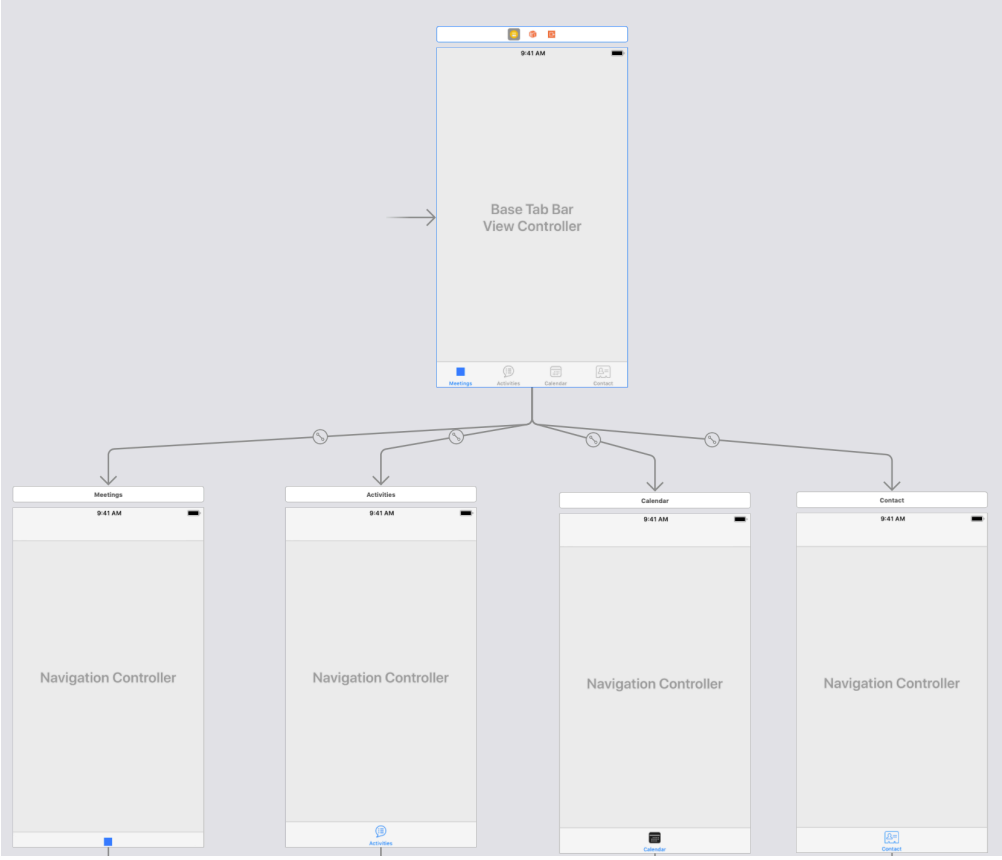



Figure 2: Main Storyboard

- Tabbar-related codes are written in **BaseTabBarViewController.swift**

```
class BaseTabBarViewController: UITabBarController {
  override func viewDidLoad(){
    //Notification setting
    if UserUtility.isLoggedin(){
      // Utility initiation
      // setup socket
    }
    // UI part
  }
  func syncFunctions{
    // sync functions
  }
  @objc func longPressed(){
    // long-press
  }
```

## Meetings ViewController

- The Meeting module is one of the most important parts of the iTime. The Meeting module comprises of three independent components: **Invitations module, Hosting module and Coming module.**
- The Invitation part displays all events that the current user has been invited. And all meetings that are created by the user will be shown in the Hosting module. In addition, if the user has voted for a certain meeting, and the event has not outdated, those events will be moved into the Coming module.
- In order to provide the convenience for hosting events, the information about how many people have **voted/refused/no reply** will be shown in the event cell.
- In addition, outdated events will be presented after the "out of date" line so that users can manage them quickly.
- If the user confirmed to attend an event, the left part of the event cell will change from **`Vote`** to **`Going`**, as you can see in the left screen of the figure 3.1.

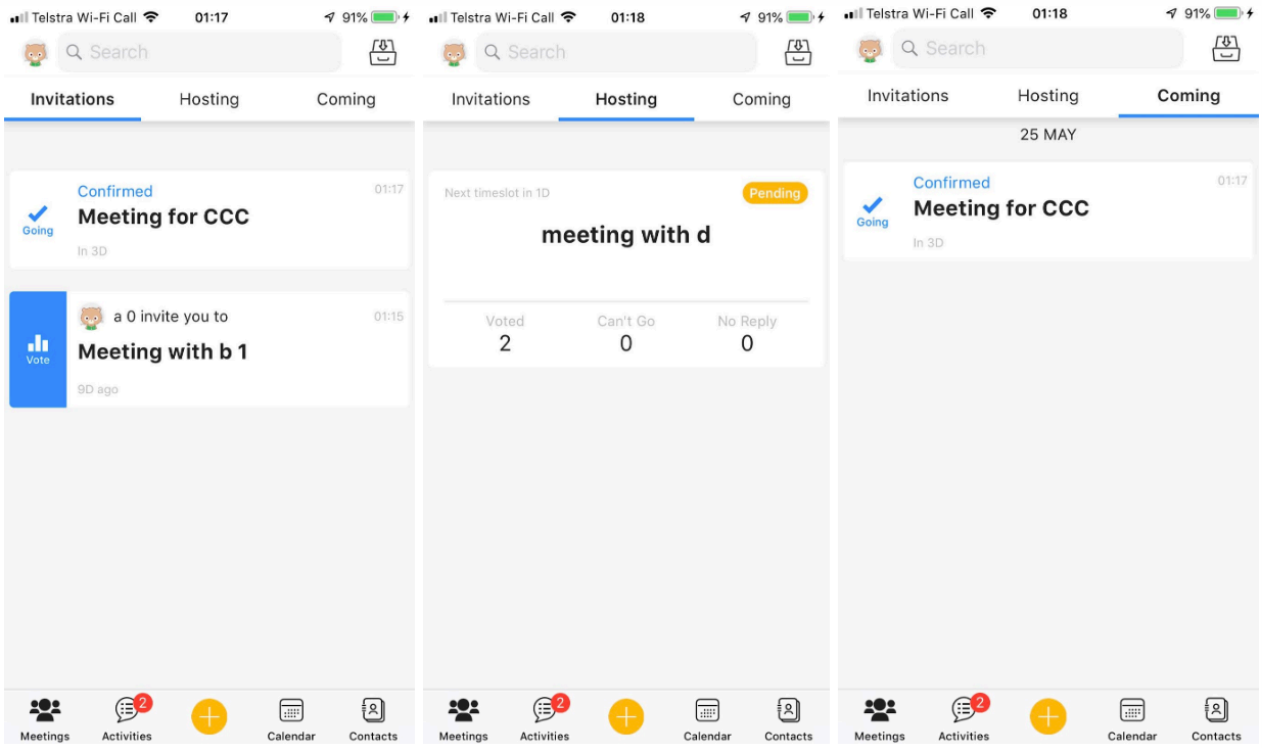



Figure 3.1 . Meeting Module

- **Code structure**
  - XLPagerTabStrip is a Container View Controller that allows us to switch easily among a collection of view controllers.
  - In this case, the BaseMeetingViewController is the container view and Invitation VC, Hosting VC and ComingVC are subVCs of the MeetingVC.

```
let subViewController1 = MeetingInvitationViewController()

let subViewController2 = MeetingHostingViewController()

let subViewController3 = MeetingComingViewController()
```

- BaseMeetingViewController is the VC holds the whole structure of the meeting views.

```
class BaseMeetingViewController:
ButtonBarPagerTabStripViewController {
      override func viewDidLoad() {
        // setupButtonBar should be called before
super.viewDidLoad()
        setupButtonBar()
        super.viewDidLoad()
        setupBasicUI()
        setupNavigationBar()
        setupNotification()
    }

    // UI functions
  func setButtonBar {

    }
    func setNavigationBar{

    }
    //@objc func functions
    @objc func searchBarTapped() {

    }
}
```

## Activities ViewController

- All event-related notification messages will be presented in the ActivityModule.
- It can be seen from Figure 3.2 that there are two main parts: the last cell is for official messages (from the iTime) and others are for recording event-related messages. When the user taps the event, three cells containing information of changes will appear and tap any of them will jump to the event detail page. The iTime messages view contains all other changes such as share calendar creation and permission setting changes.

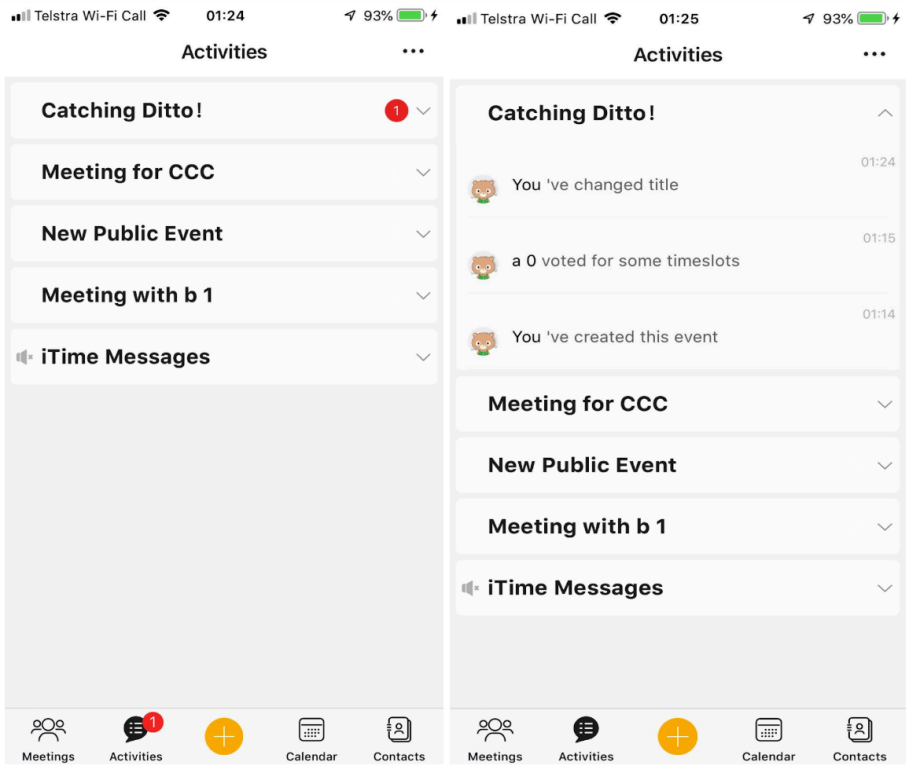


Figure 3.2 Activity Module

- Code Structure

```
class ActivityViewController: UIViewController,
UITableViewDelegate, UITableViewDataSource  {
  override func viewDidLoad() {
    // add notification observer
    // reloadActivities
    // set UI
  }
}
```

```
// Second part is about right corner ... button
var rightConnerBtnVC : ActivityShowMoreViewController?
// The Third part is about cell
}

extension ActivityViewController:
UIPopoverPresentationControllerDelegate{
  // pop over effect
}
```

## Main Functional Button ViewController

- The orange button in the middle of the tabbar is the main function button (addBtn).
- When the user taps the button just once, the event creation buttons appear, and the user can not only create public events, but also group meetings and personal event. which shows in Figure 3.3.

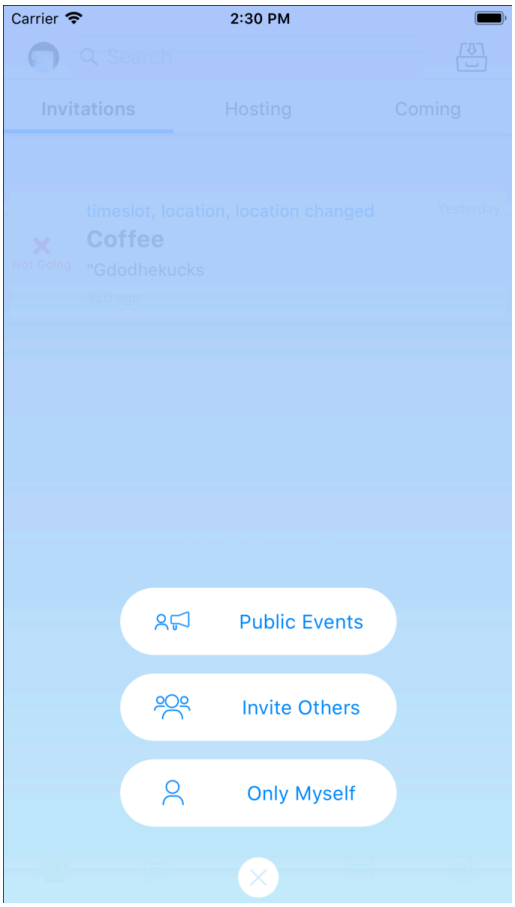



Figure 3.3 CreateEventViewController

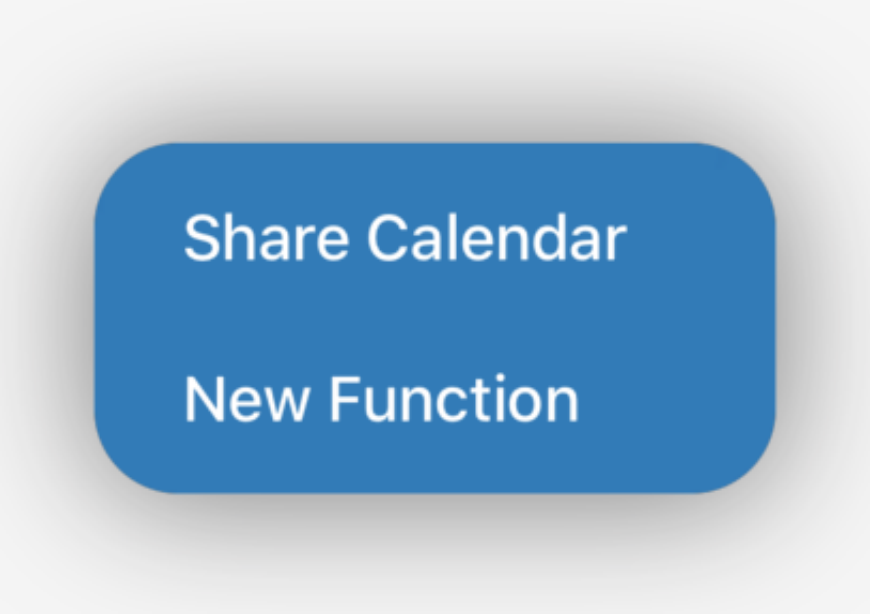
- code for Main Functional Button ViewController

```
// Add Button
let selfBar = self.tabBar as! CustomTabBar
        addButton = selfBar.addButton
        addButton.addTarget(self, action:
#selector(self.addBtnTapped(_:)), for: .touchUpInside)
        let longGesture = UILongPressGestureRecognizer(target:
self, action: #selector(longPressed))
        addButton.addGestureRecognizer(longGesture)
```

```
// Create Event VC
class CreateEventViewController: UIViewController {
    // did load
    private func setupBasic(){
        self.view.backgroundColor = UIColor.clear

        // solo event button
        let soloView = UIView(frame: CGRect())
        let soloImageView = UIImageView(frame: CGRect())
        soloImageView.image = #imageLiteral(resourceName:
"icon_event_onlyme")
        let soloLabel = UILabel(frame: CGRect())
        // UI setup part
        createSoloBtn.addSubview(soloView)

        // group event button
        let groupView = UIView(frame: CGRect())
        let groupImageView = UIImageView(frame: CGRect())
        groupImageView.image = #imageLiteral(resourceName:
"icon_event_inviteothers")
        let groupLabel = UILabel(frame: CGRect())
        // UI setup part
        createGroupBtn.addSubview(groupView)
```

```
        //The public event button
        let publicView = UIView(frame: CGRect())
        let publicImageView = UIImageView(frame: CGRect())

        publicImageView.image = #imageLiteral(resourceName:
"icon_popover_publicevent")
        let publicLabel = UILabel(frame: CGRect())

        // UI setup part
    }
}
```

- When the user taps the button longer than 0.5 seconds, a new pop menu appears with some other hotkeys (Figure 3.4).

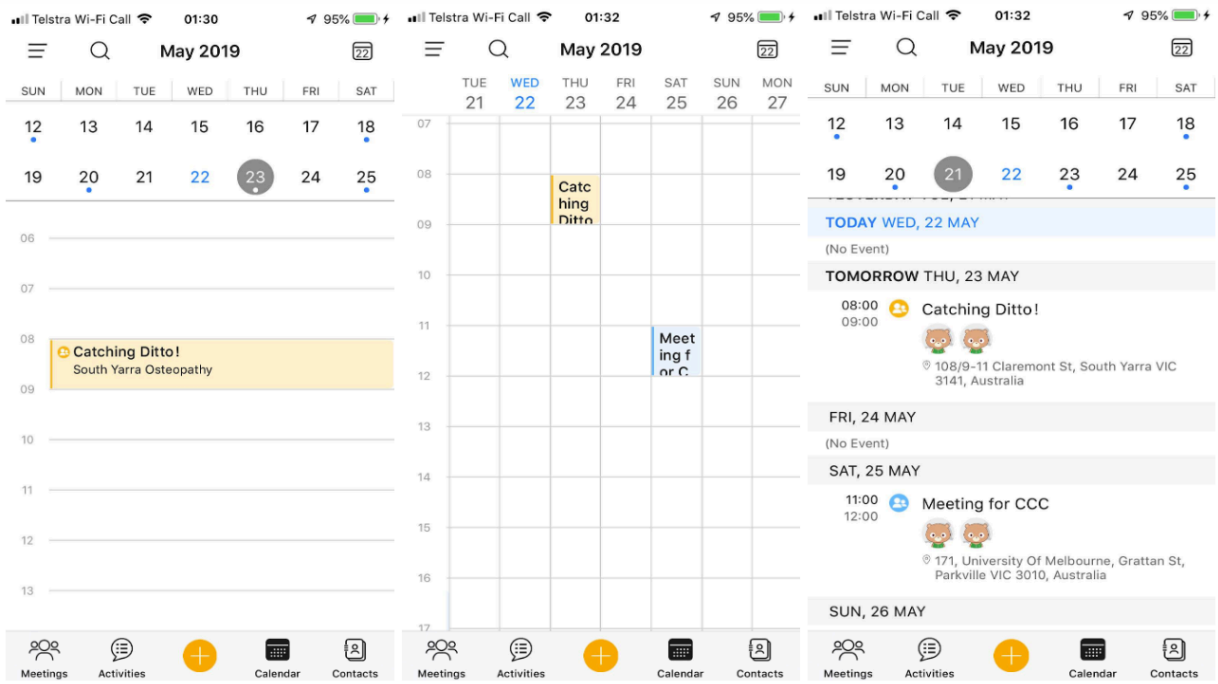



- Figure 3.4: LongPress Effect

```
@objc func longPressed(action:UILongPressGestureRecognizer){
        let actions: [PopMenuDefaultAction] = [
            // PopMenuDefaultActions
        ]
        if action.state == .began {
            // when the action begins
        } else if action.state == .ended{
            // when the action ends
        }
    }
```

## Calendar ViewController

- In the calendar view, all events will be presented in the corresponding cells in the view. User is able to see the calendar in day, week or agenda type. It can be seen from the figure 3.5 that the left screenshot is the calendar in day type, the middle one shows the calendar in week type and the right one is in agenda type.



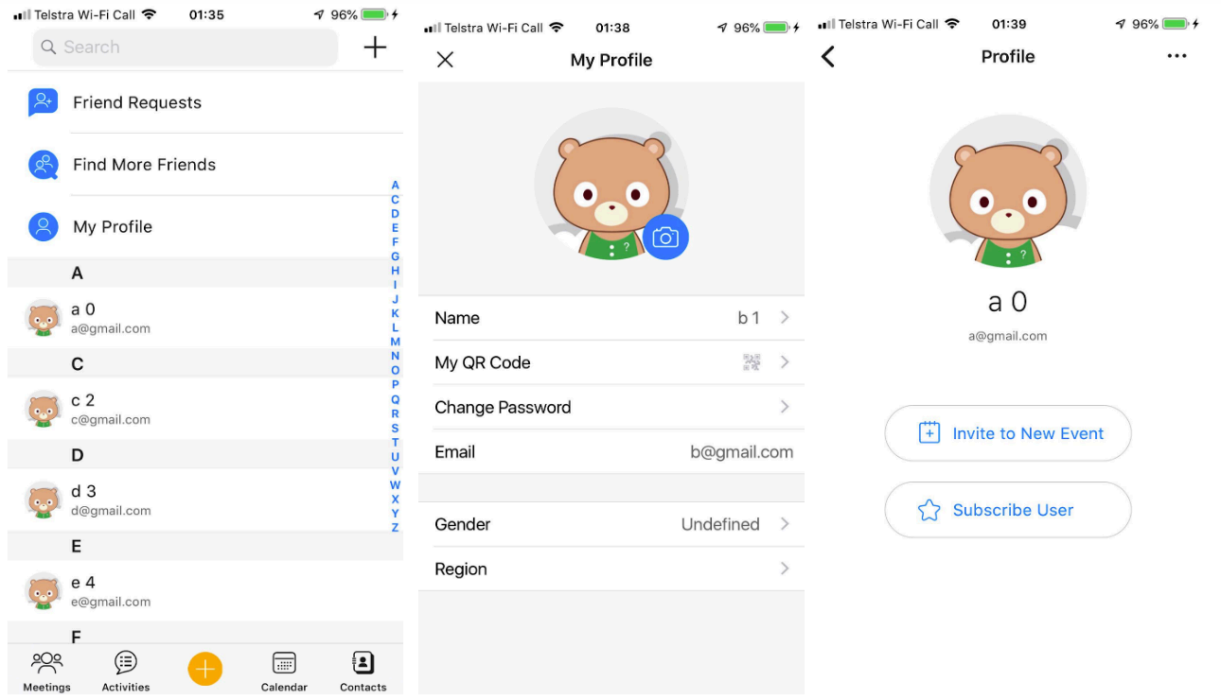

- Figure 3.5: Calendar Module

```
enum CalendarViewType{
    case day
    case week
    case agenda
}
```

```
// monthView appears when the calendar type is day or agenda
var upperMonthView: BasicCalendarView!
var agendaVC: AgendaTableViewController!
var agendaView: UIView!
// addChildViewController part -- in viewdidload part
addChildViewController(agendaVC)
agendaVC.agendaViewDelegate = self
agendaVC.didMove(toParentViewController: self)
agendaView = agendaVC.view
view.addSubview(agendaView)
```

## Contact ViewController

- The Contacts Module displays all contact information, users can add or find users that they want to connect with. If the user click any one of the user cells, the app will jump to this user's profile detail page. And the user can directly invite that user to the new Event or subscribe this user.




- Figure 3.6: Contact Module

```
class ContactsViewController: UIViewController{

    var tableView = UITableView()
    var viewModel = ContactsViewModel()
    var popoverController:UIPopoverPresentationController!

    override func viewDidLoad() {
        super.viewDidLoad()

        setupBasic()
        setupNaviBar()
        setupTableView()
        reloadContactsViewController()
    }
```

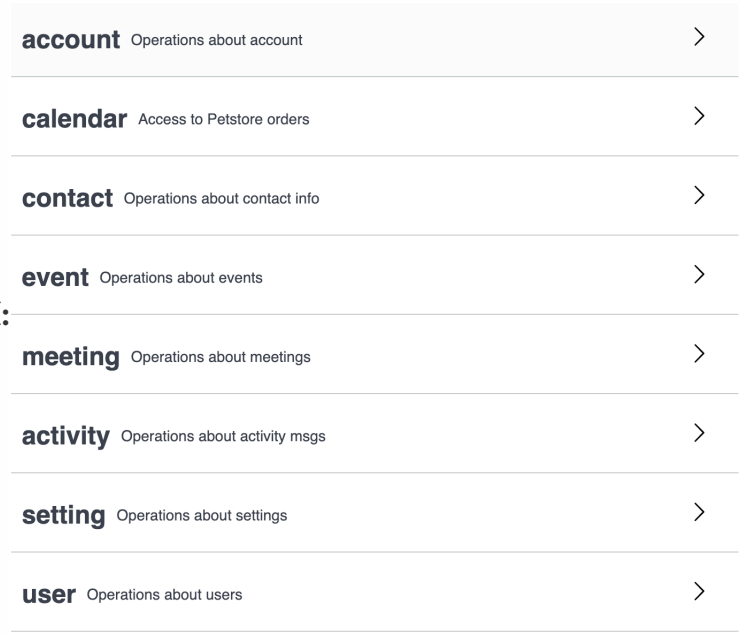
```
    func setupBasic(){
      // UI & notification
    }

    func setupNaviBar(){
      // search bar and addbtn
    }

    func setupTableView(){
      // UI
    }

    // @objc functions
    @objc func searchBarTapped() {
    }

    @objc func addFriendsBtnTapped(_ sender: UIBarButtonItem){
    }

    @objc func reRenderContactsViewNotification(_ notification:
Notification){
    }
}

// Tableview delegate
extension ContactsViewController: UITableViewDelegate,
UITableViewDataSource{
}

// popover effect
extension ContactsViewController:
UIPopoverPresentationControllerDelegate,
ContactsMoreOptionsDelegate{
}
```

## Utilities

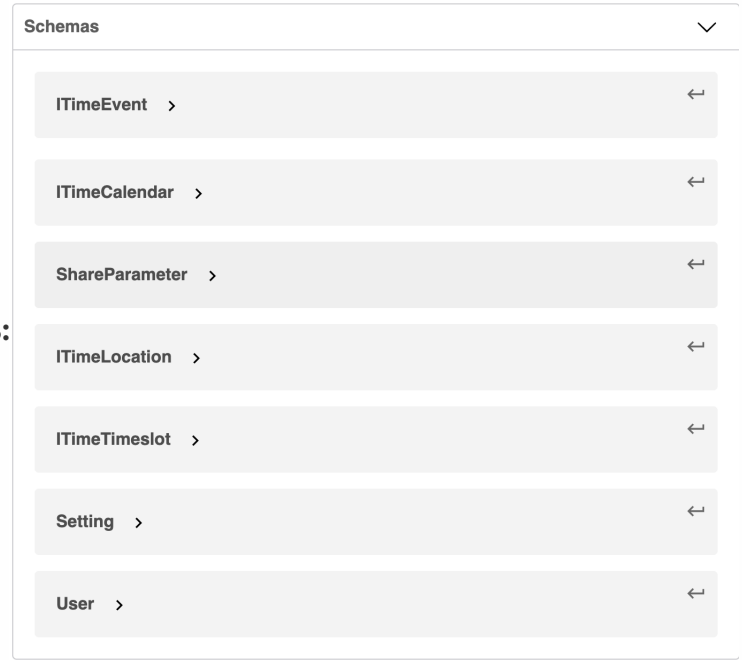
- The Utilities functions contains most of utility functions that can be called in other places
- Take the **`EventUtility`** as an example, following are some utility functions in the **`EventUtility`** class:
  - `class func initContacts()`
    - Get all contact information from the local database (realm).
  - `class func initBlockedUser()`
    - Get the list of blocked users from the local database.
  - `class func initFriendRequest()`
    - Get all friend request information from the local database.
  - `class func postRerenderContactsViewNotification()`
    - post notification to tell corresponding views to re-render.
  - `static func getContactsWithAlphabetOrder()`
    - used when reload the view model, put the new contact information in alphabetic order.
  - `static func checkIfSubscribed(_ contactUid: String)->Bool`
    - returns a boolean value to tell whether the user has subscribed a specific user or not.
  - `class func listBlockedUsersRemote(parameters)`
    - This is a HTTP GET request to the server to get all blocked users.
  - `class func listContactsRemote`
    - Same as above, a HTTP GET request to get all contact information.
  - `class func sendFriendRequest()`
    - A POST request to the server to send a friend request to other users.
  - `class func deleteContact()`
    - Delete a specific contact with a user.
  - `class func unSubscribeUser()`
    - A corresponding function to the subscribe function.
  - `class func updateContacts`
    - When sync an updated contact from the server, update it to the view.

# API & Model

- `baseUrl = ConfigurationManager(environment:.prod).baseUrl()`
- `baseApiUrl = "\(baseUrl)/api"`
- `baseSocketUrl = "http://socket.timagic.net"`
- **API:**




- **Models:**




- NOTE: The API documentation showing below is not completely same as what was written in the project, responses used in the project are comprised of only "success" and "fail". The response part in the API doc showing below is just a template.

Swagger Supported by SMARTBEAR

http://127.0.0.1:8080/ Explore

# iTime 1.0.0 OAS3

http://127.0.0.1:8080/dist/iTimeAPI.json

This is an API documentation for the mobile application named "iTime"

Terms of service
Contact the developer
Apache 2.0

## account Operations about account

**GET** `/account/list` Get binding accounts

Parameters | Try it out

No parameters

Responses

| Code | Description | Links |
|---|---|---|
| 200 | *successful operation*<br>Controls Accept header. | *No links* |
| 405 | *Validation exception* | *No links* |

**POST** `/account/google/unbind/{account_uid}` Delete binding acc of google

Parameters | Try it out

| Name | Description |
|---|---|
| account_uid * required<br>string<br>(path) | Uid of the account to be deleted |

Responses

| Code | Description | Links |
|---|---|---|
| 200 | *successfully deleted*<br><br>Controls Accept header. | *No links* |
| 405 | *Validation exception* | *No links* |

POST `/account/university/unbind/{account_uid}` Delete binding acc of unimelb

**Parameters** Try it out

| Name | Description |
|---|---|
| **account_uid** * required<br>`string`<br>*(path)* | Uid of the account to be deleted |

**Responses**

| Code | Description | Links |
|---|---|---|
| 200 | *successfully deleted*<br><br>Controls Accept header. | *No links* |
| 405 | *Validation exception* | *No links* |

~~POST~~ ~~`/account/google/bind/{authcode}`~~ import the calendar from google calendar

Warning: Deprecated

**Parameters** Try it out

| Name | Description |
|---|---|
| **authcode** * required<br>`string`<br>*(path)* | authentication code from google |

**Responses**

| Code | Description | Links |
|---|---|---|
| 200 | ***successfully binded*** | *No links* |
| | Controls `Accept` header. | |
| 405 | ***Validation exception*** | *No links* |

~~POST~~ ~~`/account/university/bind/unimelb`~~ import the calendar from unimelb calendar

**Warning: Deprecated**

**Parameters** Try it out

No parameters

Request body application/json

**Example Value** | Schema

```
{
  "username": "string",
  "password": "string"
}
```

**Responses**

| Code | Description | Links |
|---|---|---|
| 200 | ***successfully binded*** | *No links* |
| | Controls `Accept` header. | |
| 405 | ***Validation exception*** | *No links* |

## calendar Access to Petstore orders

GET `/calendar/list` Get the whole calendar

**Parameters** Try it out

No parameters

### Responses

| Code | Description | Links |
|---|---|---|
| 200 | ***successful operation*** <br> Controls Accept header. | *No links* |
| 405 | ***Validation exception*** | *No links* |

**POST** `/calendar/insert` insert a new calendar

**Parameters** Try it out

No parameters

Request body */*

iTimeCalendar

**Example Value** | Schema

```
{
  "uid": "string",
  "visibility": 0,
  "color": "string",
  "access": "string",
  "status": "string",
  "extra": "string",
  "summary": "string",
  "source": "string",
  "calendarUid": "string",
  "userUid": 0,
  "createdAt": "string",
  "updatedAt": "string",
  "accountUid": "string",
  "deleteLevel": 0,
  "email": "string"
}
```

### Responses

| Code | Description | Links |
|---|---|---|
| 200 | ***successful operation*** <br> Controls Accept header. | *No links* |
| 405 | ***Validation exception*** | *No links* |

**POST** `/calendar/share/{calendarUid}/change` change permission level of a user

Parameters | Try it out

| Name | Description |
| --- | --- |
| **calendarUid** * required<br>`string`<br>*(path)* | Uid of the calendar |

Request body | */*

info to share

**Example Value** | Schema

```
{
  "sharedUser": [
    "string"
  ],
  "access": "string"
}
```

**Responses**

| Code | Description | Links |
| --- | --- | --- |
| 200 | ***successfully operation***<br>Controls `Accept` header. | *No links* |
| 405 | ***Validation exception*** | *No links* |

**POST** `/calendar/share/{calendarUid}` share calendar with added people

Parameters | Try it out

| Name | Description |
| --- | --- |
| **calendarUid** * required<br>`string`<br>*(path)* | Uid of the calendar |

Request body | */*

info to share

**Example Value** | Schema

```
{
  "sharedUser": [
    "string"
  ],
  "access": "string"
}
```


### Responses

| Code | Description | Links |
|---|---|---|
| 200 | ***successfully operation*** <br> Controls `Accept` header. | *No links* |
| 405 | ***Validation exception*** | *No links* |

**POST** `/calendar/unshare/{calendarUid}/{userUid}` unshare calendar with user

### Parameters

Try it out

| Name | Description |
|---|---|
| **calendarUid** * required <br> `string` <br> *(path)* | Uid of the calendar |
| **userUid** * required <br> `integer` <br> *(path)* | Uid of the user |

### Responses

| Code | Description | Links |
|---|---|---|
| 200 | ***successfully operation*** <br> Controls `Accept` header. | *No links* |
| 405 | ***Validation exception*** | *No links* |

**POST** `/calendar/share/{calendarUid}/revoke` revoke the shared calendar

### Parameters

Try it out

| Name | Description |
| --- | --- |
| **calendarUid** * required<br>`string`<br>*(path)* | Uid of the calendar |

Request body — application/json

**Example Value** | Schema

```
{
  "parameters": {
    "additionalProp1": "string",
    "additionalProp2": "string",
    "additionalProp3": "string"
  }
}
```

**Responses**

| Code | Description | Links |
| --- | --- | --- |
| 200 | *successfully operation*<br>Controls Accept header. | *No links* |
| 405 | *Validation exception* | *No links* |

POST `/calendar/share/{calendarUid}/status` get the calendar's status

**Parameters** — Try it out

| Name | Description |
| --- | --- |
| **calendarUid** * required<br>`string`<br>*(path)* | Uid of the calendar |

**Responses**

| Code | Description | Links |
| --- | --- | --- |
| 200 | *successfully operation*<br>Controls Accept header. | *No links* |

| Code | Description | Links |
|---|---|---|
| 405 | *Validation exception* | *No links* |

**POST** `/calendar/update/{calendarUid}` update the calendar

**Parameters** Try it out

| Name | Description |
|---|---|
| **calendarUid** * required<br>`string`<br>*(path)* | Uid of the calendar |

Request body */*

**Example Value** | Schema


```
{
  "uid": "string",
  "visibility": 0,
  "color": "string",
  "access": "string",
  "status": "string",
  "extra": "string",
  "summary": "string",
  "source": "string",
  "calendarUid": "string",
  "userUid": 0,
  "createdAt": "string",
  "updatedAt": "string",
  "accountUid": "string",
  "deleteLevel": 0,
  "email": "string"
}
```


**Responses**

| Code | Description | Links |
|---|---|---|
| 200 | *successfully operation*<br>Controls `Accept` header. | *No links* |
| 405 | *Validation exception* | *No links* |

**POST** `/calendar/delete/{calendarUid}` delete the calendar

**Parameters** Try it out

| Name | Description |
| --- | --- |
| **calendarUid** * required<br>`string`<br>*(path)* | Uid of the calendar |

Request body */*

**Example Value** | Schema


```
{
  "uid": "string",
  "visibility": 0,
  "color": "string",
  "access": "string",
  "status": "string",
  "extra": "string",
  "summary": "string",
  "source": "string",
  "calendarUid": "string",
  "userUid": 0,
  "createdAt": "string",
  "updatedAt": "string",
  "accountUid": "string",
  "deleteLevel": 0,
  "email": "string"
}
```


**Responses**

| Code | Description | Links |
| --- | --- | --- |
| 200 | *successfully operation*<br>Controls `Accept` header. | *No links* |
| 405 | *Validation exception* | *No links* |

POST `/calendar/unsubscribe/{calendarUid}` unsubscribe the calendar

**Parameters** Try it out

| Name | Description |
| --- | --- |
| **calendarUid** * required<br>`string`<br>*(path)* | Uid of the calendar |

**Responses**

| Code | Description | Links |
|---|---|---|
| 200 | ***successfully operation***<br><br>Controls `Accept` header. | *No links* |
| 405 | ***Validation exception*** | *No links* |

## contact Operations about contact info

**GET** `/contact/user/list` get the user list -- (NOT SURE)

Parameters | Try it out

| Name | Description |
|---|---|
| syncToken<br>*(query)* | ["syncToken": syncToken as AnyObject] |

Responses

| Code | Description | Links |
|---|---|---|
| 200 | ***successfully operation***<br><br>Controls `Accept` header. | *No links* |
| 405 | ***Validation exception*** | *No links* |

**GET** `/contact/friend_request/list` list all friend requests

Parameters | Try it out

| Name | Description |
|---|---|
| syncToken<br>**string**<br>*(query)* | ["syncToken": syncToken as AnyObject] |

Responses

| Code | Description | Links |
|---|---|---|
| 200 | ***successfully operation***<br><br>Controls `Accept` header. | *No links* |
| 405 | ***Validation exception*** | *No links* |

**POST** `/contact/friend_request/send/{friendUserUid}/{invitationSource}` send the request to add a friend

**Parameters** Try it out

| Name | Description |
|---|---|
| **friendUserUid** * required<br>`string`<br>*(path)* | UserUid |
| **invitationSource** * required<br>`string`<br>*(path)* | source of invitation, e.g "itime" |

**Responses**

| Code | Description | Links |
|---|---|---|
| 200 | ***successfully operation***<br><br>Controls `Accept` header. | *No links* |
| 405 | ***Validation exception*** | *No links* |

**POST** `/contact/friend_request/confirm/{freqUid}` send the confirm ack

**Parameters** Try it out

| Name | Description |
|---|---|
| **freqUid** * required<br>`string`<br>*(path)* | Uid of FriendRequest |

**Responses**

| Code | Description | Links |
|---|---|---|
| 200 | ***successfully operation*** | *No links* |
| | Controls `Accept` header. | |
| 405 | ***Validation exception*** | *No links* |

**POST** `/contact/friend_request/reject/{freqUid}` reject the friend request

**Parameters** Try it out

| Name | Description |
|---|---|
| **freqUid** * required<br>`string`<br>*(path)* | Uid of FriendRequest |

**Responses**

| Code | Description | Links |
|---|---|---|
| 200 | ***successfully operation*** | *No links* |
| | Controls `Accept` header. | |
| 405 | ***Validation exception*** | *No links* |

**POST** `/contact/delete/{contactUid}` delete the contact

**Parameters** Try it out

| Name | Description |
|---|---|
| **contactUid** * required<br>`string`<br>*(path)* | Uid of the user in contact |

**Responses**

| Code | Description | Links |
|---|---|---|

| Code | Description | Links |
|---|---|---|
| 200 | ***successfully operation*** | *No links* |
| | Controls `Accept` header. | |
| 405 | ***Validation exception*** | *No links* |

## event Operations about events

**GET** `/event/getLink/{eventUid}` get the public event link

**Parameters** Try it out

| Name | Description |
|---|---|
| **eventUid** * required<br>`string`<br>*(path)* | Uid of the event |
| **syncToken** * required<br>`string`<br>*(query)* | synchronized token |

**Responses**

| Code | Description | Links |
|---|---|---|
| 200 | ***successfully operation*** | *No links* |
| | Controls `Accept` header. | |
| 405 | ***Validation exception*** | *No links* |

**GET** `/event/getPublicEvent/{eventUid}` get the public event

**Parameters** Try it out

| Name | Description |
|---|---|
| **eventUid** * required<br>`string`<br>*(path)* | Uid of the event |

| Name | Description |
|---|---|
| **syncToken** * required<br>`string`<br>*(query)* | synchronized token |

**Responses**

| Code | Description | Links |
|---|---|---|
| 200 | ***successfully operation***<br>Controls `Accept` header. | *No links* |
| 405 | ***Validation exception*** | *No links* |

**GET** `/event/list/{calendarUid}` get the newest event list from the server

**Parameters** Try it out

| Name | Description |
|---|---|
| **calendarUid** * required<br>`string`<br>*(path)* | Uid of the calendar |
| **syncToken** * required<br>`string`<br>*(query)* | synchronized token |

**Responses**

| Code | Description | Links |
|---|---|---|
| 200 | ***successfully operation***<br>Controls `Accept` header. | *No links* |
| 405 | ***Validation exception*** | *No links* |

**POST** `/event/insert` create a new event

**Parameters** Try it out

No parameters

Request body application/json

iTimeEvent

**Example Value** | Schema


```
{
  "id": "string",
  "host": "string",
  "recurrence": [
    "string"
  ],
  "status": "string",
  "summary": "string",
  "reason": "string",
  "start": 0,
  "end": 0,
  "duration": 0,
  "greeting": "string",
  "eventDescription": "string",
  "note": "string",
  "url": "string",
  "extra": "string",
  "location": {
    "lcoationNote": "string",
    "locationString1": "string",
    "locationString2": "string",
    "locationLatitude": "string",
    "locationLongitude": "string"
  },
  "reminder": 0,
  "source": "string",
  "eventUid": "string",
  "hostUserUid": 0,
```


**Responses**

| Code | Description | Links |
|---|---|---|
| 200 | *successfully operation*<br>Controls `Accept` header. | *No links* |
| 405 | *Validation exception* | *No links* |

POST `/event/update/{previousCalendarUId}/{eventUid}` create a new event

**Parameters** Try it out

| Name | Description |
|---|---|
| **previousCalendarUId** * required<br>`string`<br>*(path)* | Uid of the previous calendar |

| Name | Description |
| --- | --- |
| **eventUid** * required<br>`string`<br>*(path)* | Uid of the event |

Request body — application/json

iTimeEvent

**Example Value** | Schema


```
{
  "id": "string",
  "host": "string",
  "recurrence": [
    "string"
  ],
  "status": "string",
  "summary": "string",
  "reason": "string",
  "start": 0,
  "end": 0,
  "duration": 0,
  "greeting": "string",
  "eventDescription": "string",
  "note": "string",
  "url": "string",
  "extra": "string",
  "location": {
    "lcoationNote": "string",
    "locationString1": "string",
    "locationString2": "string",
    "locationLatitude": "string",
    "locationLongitude": "string"
  },
  "reminder": 0,
  "source": "string",
  "eventUid": "string",
  "hostUserUid": 0,
```


**Responses**

| Code | Description | Links |
| --- | --- | --- |
| 200 | *successfully operation*<br>Controls `Accept` header. | *No links* |
| 405 | *Validation exception* | *No links* |

**POST** `/event/invite/{eventUid}` invite users to a public event

**Parameters** — Try it out

| Name | Description |
| --- | --- |

| Name | Description |
|---|---|
| **eventUid** * required<br>`string`<br>*(path)* | Uid of the event |
| **invitees** * required<br>`array[object]`<br>*(query)* | users to be inviteds |

**Responses**

| Code | Description | Links |
|---|---|---|
| 200 | ***successfully operation***<br>Controls `Accept` header. | *No links* |
| 405 | ***Validation exception*** | *No links* |

**POST** `/event/publish_event` create a new public event

**Parameters** Try it out

No parameters

Request body application/json

iTimeEvent

**Example Value** | Schema


```
{
  "id": "string",
  "host": "string",
  "recurrence": [
    "string"
  ],
  "status": "string",
  "summary": "string",
  "reason": "string",
  "start": 0,
  "end": 0,
  "duration": 0,
  "greeting": "string",
  "eventDescription": "string",
  "note": "string",
  "url": "string",
  "extra": "string",
  "location": {
    "lcoationNote": "string",
    "locationString1": "string",
    "locationString2": "string",
    "locationLatitude": "string",
    "locationLongitude": "string"
  },
  "reminder": 0,
  "source": "string",
  "eventUid": "string",
  "hostUserUid": 0,
```

### Responses

| Code | Description | Links |
|---|---|---|
| 200 | ***successfully operation***<br><br>Controls `Accept` header. | *No links* |
| 405 | ***Validation exception*** | *No links* |

**POST** `/event/updatePublicEvent/{calendarUid}/{eventUid}` update a public event

### Parameters

Try it out

| Name | Description |
|---|---|
| **calendarUid** * required<br>`string`<br>*(path)* | Uid of the calendar |
| **eventUid** * required<br>`string`<br>*(path)* | Uid of the event |

Request body — application/json

iTimeEvent

**Example Value** | Schema


```
{
  "id": "string",
  "host": "string",
  "recurrence": [
    "string"
  ],
  "status": "string",
  "summary": "string",
  "reason": "string",
  "start": 0,
  "end": 0,
  "duration": 0,
  "greeting": "string",
  "eventDescription": "string",
  "note": "string",
  "url": "string",
  "extra": "string",
  "location": {
    "lcoationNote": "string",
    "locationString1": "string",
    "locationString2": "string",
    "locationLatitude": "string",
    "locationLongitude": "string"
  },
  "reminder": 0,
  "source": "string",
  "eventUid": "string",
  "hostUserUid": 0,
```

### Responses

| Code | Description | Links |
|---|---|---|
| 200 | ***successfully operation*** <br> Controls `Accept` header. | *No links* |
| 405 | ***Validation exception*** | *No links* |

**POST** `/update_cover_photo/{calendarUid}/{eventUid}` update a cover photo

### Parameters

Try it out

| Name | Description |
|---|---|
| **calendarUid** * required <br> `string` <br> *(path)* | Uid of the calendar |
| **eventUid** * required <br> `string` <br> *(path)* | Uid of the event |
| **coverPhotoUrl** * required <br> `string` <br> *(query)* | url of the new cover photo |

### Responses

| Code | Description | Links |
|---|---|---|
| 200 | ***successfully operation*** <br> Controls `Accept` header. | *No links* |
| 405 | ***Validation exception*** | *No links* |

**POST** `/event/delete/{calendarUid}/{eventUid}` delete the event by host users

### Parameters

Try it out

| Name | Description |
|---|---|
| **calendarUid** * required<br>`string`<br>*(path)* | Uid of the calendar |
| **eventUid** * required<br>`string`<br>*(path)* | Uid of the event |
| **syncToken** * required<br>`string`<br>*(query)* | synchronized token |

**Responses**

| Code | Description | Links |
|---|---|---|
| 200 | *successfully operation*<br>Controls `Accept` header. | *No links* |
| 405 | *Validation exception* | *No links* |

**POST** `/event/invitee/delete/{calendarUid}/{eventUid}` delete the event by non-host users

**Parameters** Try it out

| Name | Description |
|---|---|
| **calendarUid** * required<br>`string`<br>*(path)* | Uid of the calendar |
| **eventUid** * required<br>`string`<br>*(path)* | Uid of the event |
| **syncToken** * required<br>`string`<br>*(query)* | synchronized token |

**Responses**

| Code | Description | Links |
|---|---|---|

| Code | Description | Links |
|---|---|---|
| 200 | ***successfully operation*** | *No links* |
| | Controls `Accept` header. | |
| 405 | ***Validation exception*** | *No links* |

**POST** `/event/deletePublicEvent/{calendarUid}/{eventUid}` delete the public event

**Parameters** Try it out

| Name | Description |
|---|---|
| **calendarUid** * required<br>`string`<br>*(path)* | Uid of the calendar |
| **eventUid** * required<br>`string`<br>*(path)* | Uid of the event |
| **syncToken** * required<br>`string`<br>*(query)* | synchronized token |

**Responses**

| Code | Description | Links |
|---|---|---|
| 200 | ***successfully operation*** | *No links* |
| | Controls `Accept` header. | |
| 405 | ***Validation exception*** | *No links* |

**POST** `/event/photo/update/{calendarUid}/{eventUid}/{photoUid}` update a photo

**Parameters** Try it out

| Name | Description |
|---|---|
| **calendarUid** * required<br>`string`<br>*(path)* | Uid of the calendar |

| Name | Description |
|---|---|
| **eventUid** * required<br>`string`<br>*(path)* | Uid of the event |
| **photoUid** * required<br>`string`<br>*(path)* | Uid of the photo |

**Responses**

| Code | Description | Links |
|---|---|---|
| 200 | ***successfully operation***<br>Controls `Accept` header. | *No links* |
| 405 | ***Validation exception*** | *No links* |

**POST** `/event/confirm/{calendarUid}/{eventUid}/{timeslotUid}` confirme the invitation of the group event

**Parameters** Try it out

| Name | Description |
|---|---|
| **calendarUid** * required<br>`string`<br>*(path)* | Uid of the calendar |
| **eventUid** * required<br>`string`<br>*(path)* | Uid of the event |
| **timeslotUid** * required<br>`string`<br>*(path)* | timestamp |
| **syncToken** * required<br>`string`<br>*(query)* | synchronized token |

**Responses**

| Code | Description | Links |
|---|---|---|

| Code | Description | Links |
|---|---|---|
| 200 | ***successfully operation*** | *No links* |
| | Controls `Accept` header. | |
| 405 | ***Validation exception*** | *No links* |

**POST** `/event/timeslot/accept/{calendarUid}/{eventUid}` request if invitee accepts the time slot

**Parameters** Try it out

| Name | Description |
|---|---|
| **calendarUid** * required<br>`string`<br>*(path)* | Uid of the calendar |
| **eventUid** * required<br>`string`<br>*(path)* | Uid of the event |
| **timeslot** * required<br>`object`<br>*(path)* | time slot of the event |

**Responses**

| Code | Description | Links |
|---|---|---|
| 200 | ***successfully operation*** | *No links* |
| | Controls `Accept` header. | |
| 405 | ***Validation exception*** | *No links* |

**POST** `/event/timeslot/reject/{calendarUid}/{eventUid}` reject time slot

**Parameters** Try it out

| Name | Description |
|---|---|
| **calendarUid** * required<br>`string`<br>*(path)* | Uid of the calendar |

| Name | Description |
|---|---|
| **eventUid** * required<br>`string`<br>*(path)* | Uid of the event |
| reason<br>`object`<br>*(query)* | reason of rejecting the timeslot |

**Responses**

| Code | Description | Links |
|---|---|---|
| 200 | ***successfully operation***<br>Controls `Accept` header. | *No links* |
| 405 | ***Validation exception*** | *No links* |

**POST** `/event/invitee/accept/{calendarUid}{eventUid}` confirm to attend the event

**Parameters** Try it out

| Name | Description |
|---|---|
| **calendarUid** * required<br>`string`<br>*(path)* | Uid of the calendar |
| **eventUid** * required<br>`string`<br>*(path)* | Uid of the event |
| **syncToken** * required<br>`string`<br>*(query)* | synchronized token |

**Responses**

| Code | Description | Links |
|---|---|---|
| 200 | ***successfully operation***<br>Controls `Accept` header. | *No links* |
| 405 | ***Validation exception*** | *No links* |

POST `/event/addPublicEvent/{calendarUid}/{eventUid}` confirm to attend the public event

**Parameters** Try it out

| Name | Description |
|---|---|
| **calendarUid** * required<br>`string`<br>*(path)* | Uid of the calendar |
| **eventUid** * required<br>`string`<br>*(path)* | Uid of the event |
| **syncToken** * required<br>`string`<br>*(query)* | synchronized token |

**Responses**

| Code | Description | Links |
|---|---|---|
| 200 | ***successfully operation***<br>Controls `Accept` header. | *No links* |
| 405 | ***Validation exception*** | *No links* |

POST `/event/accept/{calendarUid}/{eventUid}` the invitee dicide to attend the public event

**Parameters** Try it out

| Name | Description |
|---|---|
| **calendarUid** * required<br>`string`<br>*(path)* | Uid of the calendar |
| **eventUid** * required<br>`string`<br>*(path)* | Uid of the event |
| **syncToken** * required<br>`string`<br>*(query)* | synchronized token |

### Responses

| Code | Description | Links |
|---|---|---|
| 200 | ***successfully operation*** <br> Controls Accept header. | *No links* |
| 405 | ***Validation exception*** | *No links* |

**POST** `/event/invitee/quit/{calendarUid}/{eventUid}` not going the confirmed event

### Parameters

Try it out

| Name | Description |
|---|---|
| **calendarUid** * required <br> `string` <br> *(path)* | Uid of the calendar |
| **eventUid** * required <br> `string` <br> *(path)* | Uid of the event |
| **syncToken** * required <br> `string` <br> *(query)* | synchronized token |

### Responses

| Code | Description | Links |
|---|---|---|
| 200 | ***successfully operation*** <br> Controls Accept header. | *No links* |
| 405 | ***Validation exception*** | *No links* |

**POST** `/event/removePublicEvent/{calendarUid}/{eventUid}` user rejects to go to the public event

### Parameters

Try it out

| Name | Description |
|---|---|
| **calendarUid** * required <br> `string` <br> *(path)* | Uid of the calendar |

| Name | Description |
|---|---|
| **eventUid** * required<br>`string`<br>*(path)* | Uid of the event |
| **syncToken** * required<br>`string`<br>*(query)* | synchronized token |

**Responses**

| Code | Description | Links |
|---|---|---|
| 200 | ***successfully operation***<br>Controls `Accept` header. | *No links* |
| 405 | ***Validation exception*** | *No links* |

**POST** `/event/reject/{calendarUid}/{eventUid}` the invitee rejects to go to the public event

**Parameters** Try it out

| Name | Description |
|---|---|
| **calendarUid** * required<br>`string`<br>*(path)* | Uid of the calendar |
| **eventUid** * required<br>`string`<br>*(path)* | Uid of the event |
| **syncToken** * required<br>`string`<br>*(query)* | synchronized token |

**Responses**

| Code | Description | Links |
|---|---|---|
| 200 | ***successfully operation***<br>Controls `Accept` header. | *No links* |

| Code | Description | Links |
|---|---|---|
| 405 | *Validation exception* | *No links* |

**POST** `/event/pin/{calendarUid}/{eventUid}/{pinInt}` the user pin an event

**Parameters** Try it out

| Name | Description |
|---|---|
| **calendarUid** * required<br>`string`<br>*(path)* | Uid of the calendar |
| **eventUid** * required<br>`string`<br>*(path)* | Uid of the event |
| **pinInt** * required<br>`string`<br>*(path)* | 1/0 for isPin or not |
| **syncToken** * required<br>`string`<br>*(query)* | synchronized token |

**Responses**

| Code | Description | Links |
|---|---|---|
| 200 | *successfully operation*<br>Controls `Accept` header. | *No links* |
| 405 | *Validation exception* | *No links* |

**POST** `/event/remind_all/{calendarUid}/{eventUid}` send a notification to everyone in the event group

**Parameters** Try it out

| Name | Description |
|---|---|
| **calendarUid** * required<br>`string`<br>*(path)* | Uid of the calendar |

| Name | Description |
|---|---|
| **eventUid** * required<br>`string`<br>*(path)* | Uid of the event |
| **syncToken** * required<br>`string`<br>*(query)* | synchronized token |

**Responses**

| Code | Description | Links |
|---|---|---|
| 200 | ***successfully operation***<br><br>Controls `Accept` header. | *No links* |
| 405 | ***Validation exception*** | *No links* |

**POST** `/event/pin/{calendarUid}/{eventUid}/{archiveInt}` the user archieve an event

**Parameters** Try it out

| Name | Description |
|---|---|
| **calendarUid** * required<br>`string`<br>*(path)* | Uid of the calendar |
| **eventUid** * required<br>`string`<br>*(path)* | Uid of the event |
| **archiveInt** * required<br>`string`<br>*(path)* | 1/0 for isArchieve or not |
| **syncToken** * required<br>`string`<br>*(query)* | synchronized token |

**Responses**

| Code | Description | Links |
|---|---|---|

| Code | Description | Links |
|---|---|---|
| 200 | ***successfully operation*** | *No links* |
| | Controls `Accept` header. | |
| 405 | ***Validation exception*** | *No links* |

**POST** `/event/pin/{calendarUid}/{eventUid}/{muteInt}` the user mute an event

**Parameters** Try it out

| Name | Description |
|---|---|
| **calendarUid** * required<br>`string`<br>*(path)* | Uid of the calendar |
| **eventUid** * required<br>`string`<br>*(path)* | Uid of the event |
| **muteInt** * required<br>`string`<br>*(path)* | 1/0 for isMute or not |
| **syncToken** * required<br>`string`<br>*(query)* | synchronized token |

**Responses**

| Code | Description | Links |
|---|---|---|
| 200 | ***successfully operation*** | *No links* |
| | Controls `Accept` header. | |
| 405 | ***Validation exception*** | *No links* |

**POST** `/event/timeslot/recommend` fetch rec timeslotes from server

**Parameters** Try it out

| Name | Description |
|---|---|

| Name | Description |
|---|---|
| parameters<br>**object**<br>*(header)* | The user to create. |

**Responses**

| Code | Description | Links |
|---|---|---|
| 200 | ***successfully operation***<br>Controls Accept header. | *No links* |
| 405 | ***Validation exception*** | *No links* |

## meeting Operations about meetings

**GET** `/meeting/list/` get the newest meeting list from the server

**Parameters** Try it out

| Name | Description |
|---|---|
| **syncToken** * required<br>**string**<br>*(query)* | synchronized token |

**Responses**

| Code | Description | Links |
|---|---|---|
| 200 | ***successfully operation***<br>Controls Accept header. | *No links* |
| 405 | ***Validation exception*** | *No links* |

## activity Operations about activity msgs

**GET** `/message_group/list` sync activities from the server

**Parameters** Try it out

| Name | Description |
|---|---|
| **syncToken** * required<br>`string`<br>*(query)* | synchronized token |

**Responses**

| Code | Description | Links |
|---|---|---|
| 200 | ***successfully operation***<br>Controls `Accept` header. | *No links* |
| 405 | ***Validation exception*** | *No links* |

**POST** `/message_group/read` read msg group

**Parameters** Try it out

| Name | Description |
|---|---|
| parameters<br>`object`<br>*(header)* | paramters |
| **syncToken** * required<br>`string`<br>*(query)* | synchronized token |

**Responses**

| Code | Description | Links |
|---|---|---|
| 200 | ***successfully operation***<br>Controls `Accept` header. | *No links* |
| 405 | ***Validation exception*** | *No links* |

**POST** `/message_group/read_all` read all msg group

Parameters Try it out

| Name | Description |
|---|---|
| parameters<br>**object**<br>*(header)* | paramters |
| **syncToken** * required<br>**string**<br>*(query)* | synchronized token |

**Responses**

| Code | Description | Links |
|---|---|---|
| 200 | ***successfully operation***<br>Controls Accept header. | *No links* |
| 405 | ***Validation exception*** | *No links* |

## setting Operations about settings

**GET** `/setting/get` fetch setting info from server

Parameters Try it out

No parameters

**Responses**

| Code | Description | Links |
|---|---|---|
| 200 | ***successfully operation***<br>Controls Accept header. | *No links* |
| 405 | ***Validation exception*** | *No links* |

**POST** `/setting/update` update setting configuration to server

Parameters Try it out

No parameters

Request body application/json

setting

**Example Value** | Schema


```
{
  "enablePreviewText": true,
  "enableEventAlert": true,
  "enableNotification": true,
  "showPreviewText": true,
  "appAlertSound": true,
  "systemSound": true,
  "systemVibrate": true,
  "defaultSoloAlertTime": 0,
  "defaultGroupAlertTime": 0,
  "defaultAllDayAlertTime": 0,
  "enableFriendRequestEmail": true,
  "enableEventInvitationEmail": true,
  "enableEventConfirmEmail": true,
  "showCalendarOnStartup": true,
  "showUncomfirmedEvents": true
}
```


**Responses**

| Code | Description | Links |
|---|---|---|
| 200 | *successfully operation*<br>Controls `Accept` header. | *No links* |
| 405 | *Validation exception* | *No links* |

## user Operations about users

GET `/user/get/{userUid}` get the user info from remote

**Parameters** Try it out

| Name | Description |
|---|---|
| **userUid** * required<br>`string`<br>*(path)* | Uid of the event |

**Responses**

| Code | Description | Links |
|---|---|---|
| 200 | ***successfully operation*** <br> Controls Accept header. | *No links* |
| 405 | ***Validation exception*** | *No links* |

**GET** `/user/domain/list` get the domain list

**Parameters** Try it out

No parameters

**Responses**

| Code | Description | Links |
|---|---|---|
| 200 | ***successfully operation*** <br> Controls Accept header. | *No links* |
| 405 | ***Validation exception*** | *No links* |

**POST** `/user/password/update` update password

**Parameters** Try it out

| Name | Description |
|---|---|
| parameters<br>**object**<br>*(header)* | paramters |

**Responses**

| Code | Description | Links |
|---|---|---|
| 200 | ***successfully operation*** <br> Controls Accept header. | *No links* |

| Code | Description | Links |
|---|---|---|
| 405 | Validation exception | No links |

**POST** `/user/signin` Logs user into the system

**Parameters** Try it out

| Name | Description |
|---|---|
| **username** * required<br>string<br>*(query)* | The user name for login |
| **password** * required<br>string<br>*(query)* | The password for login in clear text |

**Responses**

| Code | Description | Links |
|---|---|---|
| 200 | successfully operation<br><br>Controls Accept header. | No links |
| 405 | Validation exception | No links |

**POST** `/user/signup` logs out from the system

**Parameters** Try it out

| Name | Description |
|---|---|
| userInfoForSignUp<br>object<br>*(header)* | paramters |

**Responses**

| Code | Description | Links |
|---|---|---|

| Code | Description | Links |
|---|---|---|
| 200 | ***successfully operation*** | *No links* |
| | Controls `Accept` header. | |
| 405 | ***Validation exception*** | *No links* |

**POST** `/user/validate` validate the email for signUpp

**Parameters** Try it out

| Name | Description |
|---|---|
| **userId** * required<br>`string`<br>*(query)* | email |

**Responses**

| Code | Description | Links |
|---|---|---|
| 200 | ***successfully operation*** | *No links* |
| | Controls `Accept` header. | |
| 405 | ***Validation exception*** | *No links* |

**POST** `/user/profile/update` update user's info to server

**Parameters** Try it out

| Name | Description |
|---|---|
| **user** * required<br>`object`<br>*(header)* | user info |

**Responses**

| Code | Description | Links |
|---|---|---|

| Code | Description | Links |
|---|---|---|
| 200 | ***successfully operation*** | *No links* |
| | Controls `Accept` header. | |
| 405 | ***Validation exception*** | *No links* |

**POST** `/user/search/{query}` search user

**Parameters** Try it out

| Name | Description |
|---|---|
| **query** * required<br>`string`<br>*(path)* | user's name for search |

**Responses**

| Code | Description | Links |
|---|---|---|
| 200 | ***successfully operation*** | *No links* |
| | Controls `Accept` header. | |
| 405 | ***Validation exception*** | *No links* |

**Schemas**

**ITimeEvent** ⌄ {

| | |
|---|---|
| id | string |
| host | string |
| recurrence | ⌄ [string] |
| status | string |
| summary | string |
| reason | string |
| start | number($date-time) |
| end | number($date-time) |
| duration | integer |
| greeting | string |
| eventDescription | string |
| note | string |
| url | string |
| extra | string |
| location | **ITimeLocation** ⌄ { |
| | lcoationNote string |
| | locationString1 string |
| | locationString2 string |
| | locationLatitude string |
| | locationLongitude string |
| | } |
| reminder | integer |
| source | string |
| eventUid | string |
| hostUserUid | integer |
| userUid | integer |
| calendarUid | string |
| recurringEventUid | string |
| isAllday | boolean |
| eventType | string |
| inviteeVisibility | integer |
| freebusyAccess | integer |
| showLevel | integer |
| deleteLevel | integer |
| isPinned | boolean |
| isMute | boolean |
| isArchived | boolean |
| archivedAt | string |
| coverPhoto | string |
| createdAt | string |
| updatedAt | string |
| is_subscribed | boolean |
| is_added | boolean |
| adminList | ⌄ [integer] |

}

**ITimeCalendar** ⌄ {

| | |
|---|---|
| uid | string |
| visibility | integer($binary) |
| color | string |
| access | string |
| | access type |
| status | string |
| extra | string |
| summary | string |
| | title |
| source | string |
| calendarUid | string |
| userUid | integer($int16) |
| createdAt | string |
| updatedAt | string |
| accountUid | string |
| deleteLevel | integer($int16) |
| email | string |

}

### ShareParameter { 

```
    sharedUser                  [string]
    access                      string
}
```

### ITimeLocation { 

```
    lcoationNote                string
    locationString1             string
    locationString2             string
    locationLatitude            string
    locationLongitude           string
}
```

### ITimeTimeslot { 

```
    start                       number
    end                         number
    status                      string
    rate                        integer
    timeslotUid                 string
    eventUid                    string
    acceptedNum                 integer
    rejectedNum                 integer
    pendingNum                  integer
    totalNum                    integer
    isConfirmed                 boolean
    isAllDay                    boolean
    isSystemSuggested           boolean
    inviteeUid                  string
    userUid                     integer
}
```

### Setting { 

```
    enablePreviewText           boolean
    enableEventAlert            boolean
    enableNotification          boolean
    showPreviewText             boolean
    appAlertSound               boolean
    systemSound                 boolean
    systemVibrate               boolean
    defaultSoloAlertTime        integer
    defaultGroupAlertTime       integer
    defaultAllDayAlertTime      integer
    enableFriendRequestEmail    boolean
    enableEventInvitationEmail  boolean
    enableEventConfirmEmail     boolean
    showCalendarOnStartup       boolean
    showUncomfirmedEvents       boolean
}
```

**User** ⌄ {

| | |
|---|---|
| userUid | integer |
| userId | string |
| password | string |
| personalAlias | string |
| email | string |
| phone | string |
| photo | string |
| source | string |
| deviceToken | string |
| deviceId | string |
| averageRatingValue | string |
| timezone | string |
| lastSigninTime | integer |
| signinCount | integer |
| gender | integer |
| region | string |
| status | string |
| createdAt | string |
| updatedAt | string |
| language | string |

}