# 在线安装：

1. 安装插件background-mode： cordova plugin add <https://github.com/katzer/cordova-plugin-background-mode.git>
2. 安装插件localnotification：cordova plugin add <https://github.com/katzer/cordova-plugin-local-notifications.git>。安装后如果编译不通过，那么先删除android/ios平台再重新添加android/ios平台。
3. 安装插件badge：cordova plugin add https://github.com/katzer/cordova-plugin-badge.git
4. 安装angular-socket-io：bower install –save angular-socket-io
5. 将socket.io.js和push\_service.js文件拷贝放入www/lib文件夹下

# 离线安装：

1. 在下载好的background-mode插件（http://172.28.3.10/svn/Component/message\_push/resources/cordova-plugin-background-mode）、app-event插件（http://172.28.3.10/svn/Component/message\_push/resources/cordova-plugin-app-event）、localnotification插件（http://172.28.3.10/svn/Component/message\_push/resources/de.appplant.cordova.plugin.local-notification）和badge插件（http://172.28.3.10/svn/Component/message\_push/resources/cordova-plugin-badge）解压缩后放入项目的同级目录
2. 安装background-mode插件：cordova plugin add ../ cordova-plugin-background-mode
3. 安装app-event插件：cordova plugin add ../ cordova-plugin-app-event
4. 安装localnotification插件：cordova plugin add ../ de.appplant.cordova.plugin.local-notification。安装后如果编译不通过，那么先删除android/ios平台再重新添加android/ios平台。
5. 安装badge插件：cordova plugin add ../cordova-plugin-badge
6. 将下载好的angular-socket-io（http://172.28.3.10/svn/Component/message\_push/resources/angular-socket-io）解压缩后拷贝到www/lib文件夹下，并将socket.io.js（http://172.28.3.10/svn/Component/message\_push/resources/socket.io.js）和push\_service.js（http://172.28.3.10/svn/Component/message\_push/push\_service.js）文件拷贝放入www/lib文件夹下

# 接口说明：

nari\_push：websocket连接管理、消息推送服务

|  |  |  |
| --- | --- | --- |
| 方法名 | 参数定义 | 功能说明 |
| init |  | 建立、维持与消息推送服务端的连接，在连接断开后自动尝试重连并重新登录 |
| login | @in userid 字符型变量，用于消息推送服务端对websocket连接进行唯一标识 | 在消息推送服务端注册唯一标识 |
| logout |  | 在消息推送服务端注销唯一标识 |
| waitMessages | @in showDefaultNotification，布尔型变量，默认为true，用于表示是否使用默认的推送设置 | 接收并缓存推送过来的消息；通知用户程序获取新消息 |
| setBackgroundReady | @in options，程序后台运行时通知栏上显示的内容{titie:’’,text:’’} | 设置允许后台模式（在$ionicPlatform.ready()里使用） |
| getUnreadCount |  | 获取后台运行模式下未处理的消息数目 |

new\_messages：提供缓存的推送过来的消息

|  |  |  |
| --- | --- | --- |
| 方法名 | 参数定义 | 功能说明 |
| getMessages | @return 数组，每一个数组项都是一条消息。示例：  [{msgID:’’,data{}}] | 获取推送过来的未被处理的消息 |

触发的事件

|  |  |  |
| --- | --- | --- |
| 事件名 | 参数定义 | 功能说明 |
| newMessages | @in e，事件对象。  @in d，数据对象，此处为null。 | 此事件触发时，说明有新的消息被推送过来，用户应该调用new\_messages.getMessages方法获取被推送过来的数据 |
| broadcastMessages | @in e，事件对象  @in d，数据对象，此处为被推送过来的广播消息。示例：{msgID:’’,data{}} | 此事件触发，说明有广播消息被推送过来。 |
| $cordovaLocalNotification:click |  | 此事件触发，说明点击了通知栏，用户侦听此事件以实现通知栏点击后触发的动作 |

# 使用示例：

1. 引入相应js文件

|  |
| --- |
| //index.html  …  <script src="lib/socket.io.js"></script>  <script src="lib/angular-socket-io/socket.js"></script>  <script src="js/push\_service.js"></script>  … |

1. 消息推送客户端初始化、后台模式设立以及接收推送过来的消息

|  |
| --- |
| //app.js  angular.module('starter', ['ionic', 'ngCordova', 'btford.socket-io', 'pushService', …])  .run([…,’nari\_push',function(…,nari\_push){  …  //消息推送客户端初始化  nari\_push.init();  //开始接收推送过来的消息（使用默认的消息栏提示）  nari\_push.waitMessages();  //登录  nari\_push.login(‘00060924’);  …  $ionicPlatform.ready(function() {  …  var defaults = {  title: '智慧园区',  text: '智慧园区正在后台运行'  };  //后台模式设立  nari\_push.setBackgroundReady(defaults);  //设置点击通知栏后所需触发的动作  var toPage=function(pageConditions){  if(pageConditions.condition=='bus'){  console.log('Enter the bus page');  $state.go('shuttleTab.list');  }else if(pageConditions.condition=='meeting'){  console.log('Enter the meeting page');  $state.go('meettab.mine');  }else{  console.log('Enter the main page');  $state.go('app.main');  }  }  var pageConditions={condition:'meeting'};  $scope.$on('$cordovaLocalNotification:click ',function(e,d){  toPage(pageConditions);  });  …  })  } |

1. 用户程序侦听newMessages或者broadcastMessages事件，处理推送过来的消息

|  |
| --- |
| //controllers.js  …  .controller('MainCtrl', ['$scope', 'new\_messages', …,  function($scope, new\_messages, …) {  …  $scope.$on('newMessages',function(e,d){  console.log("New messages!");  var messages=new\_messages.getMessages();  console.log(messages.length);  })  $scope.$on('broadcastMessages',function(e,d){  console.log('Receive broadcast data!');  console.log(d);  })  …  }])  … |

完整版示例参见：

http://172.28.3.10/svn/Component/message\_push/demo