| | | | | --- | --- | --- | | **Property** | **Type** | **Description** | | devices | Array | A list of all the devices that are newly discovered. |

Device Object

| | | | | --- | --- | --- | | **Property** | **Type** | **Description** | | name | String | Name of the bluetooth device.(For some devices, there's no name.) | | deviceName (Compatible with initial version) | String | Name of the bluetooth device. | | localName | String | Name of the local device. | | deviceId | String | Device ID. | | RSSI | Number | Received Signal Strength Indicator. | | advertisData | Hex String | Advertisement data of the device. |

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev/api_device_bluetooth_bluetooth_onbluetoothdevicefound

---

my.onCompassChange {#myoncompasschange}

Last updated: 2022-07-03

Path: miniprogram_gcash

my.onCompassChange

2022-07-03 18:44

Use this API to listen to the compass data change event. After the interface call, listening is automatically started. The callback interval is 500ms. You can use my.offCompassChange to stop listening.

Sample Code

copy

```
my.onCompassChange(function (res) {
  console.log(res.direction);
})
```

Parameters

The property is a callback function which uses object properties with the following property:

| | | | | --- | --- | --- | | **Property** | **Type** | **Description** | | direction | Number | The degree of the direction that you are facing. |

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev/api_device_compass_oncompasschange

---

my.onCompassChange {#myoncompasschange}

Last updated: 2021-05-09

Path: miniprogram_gcash

my.onCompassChange

2021-05-09 18:43

Use this API to listen to the compass data change event. After the interface call, listening is automatically started. The callback interval is 500ms. You can use my.offCompassChange to stop listening.

Sample Code

copy

```
my.onCompassChange(function (res) {
  console.log(res.direction);
})
```

Parameters

The property is a callback function which uses object properties with the following property:

Property	**Type**	**Description**
direction	Number	The degree of the direction that you are facing.

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev-old/api_device_compass_oncompasschange

---

my.onError {#myonerror}

Last updated: 2022-07-03

Path: miniprogram_gcash

my.onError

2022-07-03 18:44

Listen for the event that errors occur in the mini program. Currently only JS errors are supported. The triggered time and required parameters are the same with those of the onError() method.

Sample code

Sample for general errors:

copy

```
// .js
App({
  onLaunch() {
    // Type of error is String.
    my.onError(function(error) {
      // Errors occur when running the mini program.
      console.error(error);
    });
  }
})
```

Sample for JS errors:

copy

```
onShow: function() {

    // Errors occur when running the mini program.
    my.onError(function(error) {
      // Error message displayed on the page
      console.warn(error, '1212');
    });
  }
```

Parameters

Property	**Type**	**Description**
callback	Function	The callback function for the event that JS errors occur in the mini program.

Returned value

Property	**Type**	**Description**
error	String	Error message.

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev/api_event_onerror

---

my.onMemoryWarning {#myonmemorywarning}

Last updated: 2022-07-03

Path: miniprogram_gcash

my.onMemoryWarning

2022-07-03 18:44

Use this API to listen to the insufficient memory alarm event.

For Android, two alarm levels exist:

- `TRIM_MEMORY_RUNNING_LOW`
- `TRIM_MEMORY_RUNNING_CRITICAL`

No alarm level is available for iOS.

Sample Code

copy

```
// API-DEMO page/API/memory-warning/memory-warning.json
{
   "defaultTitle": "OnMemoryWarning"
}
```

copy

```
<!-- API-DEMO page/API/memory-warning/memory-warning.axml-->
<view class="page">

  <button type="primary" onTap="onMemoryWarning">
    Listen to Insufficient Memory Alarm Event
  </button>

</view>
```

copy

```
// API-DEMO page/API/memory-warning/memory-warning.js
Page({
  onLoad() {
    this.callback = (res) => {
        var levelString = 'iOS device, No alarm level exists.';
        switch (res.level) {
          case 10:
            levelString = 'Android device, level = 
TRIM_MEMORY_RUNNING_LOW';
            break;
          case 15:
            levelString = 'Android device, level = 
TRIM_MEMORY_RUNNING_CRITICAL';
            break;
        }
        my.alert({
          title: 'Received insufficient memory alarm',
          content: levelString
        });
    };
    this.isApiAvailable = my.canIUse('onMemoryWarning');
  },
  onMemoryWarning() {
    if (this.isApiAvailable) {
      my.onMemoryWarning(this.callback);
    } else {
      my.alert({
        title: 'Client version is too low',
        content: 'my.onMemoryWarning() and my.offMemoryWarning() need 
10.1.35 or higher versions'
      });
    }
  },
  onUnload() {
    if (this.isApiAvailable) {
      my.offMemoryWarning(this.callback);
    }
  }
});
```

Parameters

The property is a callback function which uses object properties with the following property:

| | | | | --- | --- | --- | | **Property** | **Type** | **Description** | | level | Number | Memory alarm level, only available in Android. |

Android alarm levels that correspond to the system macro definitions:

copy

```
int TRIM_MEMORY_RUNNING_LOW = 10
int TRIM_MEMORY_RUNNING_CRITICAL = 15
```

九色鹿

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev/api_device_memory-warning_onmemorywarning

---

my.onMemoryWarning {#myonmemorywarning}

Last updated: 2021-05-09

Path: miniprogram_gcash

my.onMemoryWarning

2021-05-09 18:43

Use this API to listen to the insufficient memory alarm event.

For Android, two alarm levels exist:

- `TRIM_MEMORY_RUNNING_LOW`
- `TRIM_MEMORY_RUNNING_CRITICAL`

No alarm level is available for iOS.

Sample Code

copy

```
// API-DEMO page/API/memory-warning/memory-warning.json
{
   "defaultTitle": "OnMemoryWarning"
}
```

copy

```
<!-- API-DEMO page/API/memory-warning/memory-warning.axml-->
<view class="page">

  <button type="primary" onTap="onMemoryWarning">
    Listen to Insufficient Memory Alarm Event
  </button>

</view>
```

copy

```
// API-DEMO page/API/memory-warning/memory-warning.js
Page({
  onLoad() {
    this.callback = (res) => {
        var levelString = 'iOS device, No alarm level exists.';
        switch (res.level) {
          case 10:
            levelString = 'Android device, level =
TRIM_MEMORY_RUNNING_LOW';
            break;
          case 15:
            levelString = 'Android device, level =
TRIM_MEMORY_RUNNING_CRITICAL';
            break;
        }
        my.alert({
          title: 'Received insufficient memory alarm',
          content: levelString
        });
    };
    this.isApiAvailable = my.canIUse('onMemoryWarning');
  },
  onMemoryWarning() {
    if (this.isApiAvailable) {
      my.onMemoryWarning(this.callback);
    } else {
      my.alert({
        title: 'Client version is too low',
        content: 'my.onMemoryWarning() and my.offMemoryWarning() need
10.1.35 or higher versions'
      });
    }
  },
  onUnload() {
    if (this.isApiAvailable) {
      my.offMemoryWarning(this.callback);
    }
  }
});
```

Parameters

The property is a callback function which uses object properties with the following property:

| | | | | --- | --- | --- | | **Property** | **Type** | **Description** | | level | Number | Memory alarm level, only available in Android. |

Android alarm levels that correspond to the system macro definitions:

copy

```
int TRIM_MEMORY_RUNNING_LOW = 10
int TRIM_MEMORY_RUNNING_CRITICAL = 15
```

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev-old/api_device_memory-warning_onmemorywarning

---

my.onSocketClose {#myonsocketclose}

Last updated: 2022-07-03

Path: miniprogram_gcash

my.onSocketClose

2022-07-03 18:44

Use this API to listen to the event of disabling the WebSocket connection.

Sample Code

copy

```
onLoad() {
    // Note: The registration of callback only needs to be done once
during the start-up phase of the Mini Program, multiple calls have
multiple callbacks.
    my.onSocketClose((res) => {
      my.alert({content: 'The connection is disabled!'});
      this.setData({
        sendMessageAbility: false,
        closeLinkAbility: false,
      });
    });
    // Note: The registration of callback only needs to be done once
during the start-up phase of the Mini Program, multiple calls have
multiple callbacks.
    my.onSocketOpen((res) => {
      my.alert({content: 'The connection is enabled!'});
      this.setData({
        sendMessageAbility: true,
        closeLinkAbility: true,
      });
    });

    my.onSocketError(function(res){
      my.alert('The webSocket connection fails. Please check!' + res);
    });

    // Note: The registration of callback only needs to be done once
during the start-up phase of the Mini Program, multiple calls have
multiple callbacks.
    my.onSocketMessage((res) => {
      my.alert({content: 'Data received!' + JSON.stringify(res)});
    });
  }

connect_start() {
    my.connectSocket({
      url: 'Server URL', // The developer server interface URL must
use WSS protocol and the domain name must be a legitimate one
configured in the backend.
      success: (res) => {
        my.showToast({
          content: 'success', // Text content
        });
      },
      fail:()=>{
        my.showToast({
          content: 'fail', // Text content
        });
      }
    });
  },
```

Note: The case is only for reference. Please use your own URL to test.

Parameters

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Required** | **Description** | | callback | Function | Yes | The callback function for the event of disabling the WebSocket connection. |

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev/API_Network_onSocketClose

---

my.onSocketClose {#myonsocketclose}

Last updated: 2021-05-09

Path: miniprogram_gcash

my.onSocketClose

2021-05-09 18:43

Use this API to listen to the event of disabling the WebSocket connection.

Sample Code

copy

```
onLoad() {
    // Note: The registration of callback only needs to be done once
during the start-up phase of the Mini Program, multiple calls have
multiple callbacks.
    my.onSocketClose((res) => {
      my.alert({content: 'The connection is disabled!'});
      this.setData({
        sendMessageAbility: false,
        closeLinkAbility: false,
      });
    });
    // Note: The registration of callback only needs to be done once
during the start-up phase of the Mini Program, multiple calls have
multiple callbacks.
    my.onSocketOpen((res) => {
      my.alert({content: 'The connection is enabled!'});
      this.setData({
        sendMessageAbility: true,
        closeLinkAbility: true,
      });
    });

    my.onSocketError(function(res){
      my.alert('The webSocket connection fails. Please check!' + res);
    });

    // Note: The registration of callback only needs to be done once
during the start-up phase of the Mini Program, multiple calls have
multiple callbacks.
    my.onSocketMessage((res) => {
      my.alert({content: 'Data received!' + JSON.stringify(res)});
    });
  }

connect_start() {
    my.connectSocket({
      url: 'Server URL', // The developer server interface URL must
use WSS protocol and the domain name must be a legitimate one
configured in the backend.
      success: (res) => {
        my.showToast({
          content: 'success', // Text content
        });
      },
      fail:()=>{
        my.showToast({
          content: 'fail', // Text content
        });
      }
    });
  },
```

Note: The case is only for reference. Please use your own URL to test.

Parameters

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Required** | **Description** | | callback | Function | Yes | The callback function for the event of disabling the WebSocket connection. |

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev-old/api_network_onsocketclose

---

my.onSocketError {#myonsocketerror}

Last updated: 2022-07-03

Path: miniprogram_gcash

my.onSocketError

2022-07-03 18:44

Use this API to listen to WebSocket error events.

Sample Code

copy

```
my.connectSocket({
  url: 'Developer's server URL'
});

my.onSocketOpen(function(res){
  console.log('The WebSocket connection is enabled!');
});

my.onSocketError(function(res){
  console.log('The WebSocket connection failed. Please check!');
});
```

Note: The case is only for reference. Please use your own URL to test.

Parameters

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Required** | **Description** | | callback | Function | Yes | The callback function for the WebSocket error event. |

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev/api_network_onsocketerror

---

my.onSocketError {#myonsocketerror}

Last updated: 2021-05-09

Path: miniprogram_gcash

my.onSocketError

2021-05-09 18:43

Use this API to listen to WebSocket error events.

Sample Code

copy

```
my.connectSocket({
  url: 'Developer's server URL'
});

my.onSocketOpen(function(res){
  console.log('The WebSocket connection is enabled!');
});

my.onSocketError(function(res){
  console.log('The WebSocket connection failed. Please check!');
});
```

Note: The case is only for reference. Please use your own URL to test.

Parameters

Property	**Type**	**Required**	**Description**
callback	Function	Yes	The callback function for the WebSocket error event.

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev-old/api_network_onsocketerror

---

my.onSocketMessage {#myonsocketmessage}

Last updated: 2022-07-03

Path: miniprogram_gcash

my.onSocketMessage

2022-07-03 18:44

Use this API to listen to the event of receiving server messages by WebSocket.

Callback Return Value

Property	**Type**	**Description**
data	String / ArrayBuffer	Messages returned by the server, in text string or base64 encoded string.
isBuffer	Boolean	If the value of this field is `true`, the `data` field represents the received base64 encoded binary strings. If the value of this field is empty, the `data` field represents normal strings.

Sample Code

copy

```
my.connectSocket({
  url: 'Server URL'
})

my.onSocketMessage(function(res) {
  console.log('Server content received  ' + res.data)
})
```

Note: The case is only for reference. Please use your own URL to test.

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev/api_network_onsocketmessage

my.onSocketMessage {#myonsocketmessage}

Last updated: 2021-05-09

Path: miniprogram_gcash

my.onSocketMessage

2021-05-09 18:43

Use this API to listen to the event of receiving server messages by WebSocket.

Callback Return Value

| | | | | --- | --- | --- | | **Property** | **Type** | **Description** | | data | String / ArrayBuffer | Messages returned by the server, in text string or base64 encoded string. | | isBuffer | Boolean | If the value of this field is `true`, the `data` field represents the received base64 encoded binary strings. If the value of this field is empty, the `data` field represents normal strings. |

Sample Code

copy

```
my.connectSocket({
  url: 'Server URL'
})

my.onSocketMessage(function(res) {
  console.log('Server content received  ' + res.data)
})
```

Note: The case is only for reference. Please use your own URL to test.

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev-old/api_network_onsocketmessage

---

my.onSocketOpen {#myonsocketopen}

Last updated: 2022-07-03

Path: miniprogram_gcash

my.onSocketOpen

2022-07-03 18:44

Use this API to listen to the event of enabling the WebSocket connection.

Sample Code

copy

```
my.connectSocket({
  url: 'test.php',
});

my.onSocketOpen(function(res) {
  console.log('The WebSocket connection is enabled!');
});
```

Note: The case is only for reference. Please use your own URL to test.

Parameters

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Required** | **Description** | | callback | Function | Yes | The callback function for the event of enabling the WebSocket connection. |

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev/api_network_onsocketopen

---

my.onSocketOpen {#myonsocketopen}

Path: miniprogram_gcash

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev-old/api_network_onsocketopen

---

my.onUnhandledRejection {#myonunhandledrejection}

Last updated: 2022-07-03

Path: miniprogram_gcash

my.onUnhandledRejection

2022-07-03 18:44

Listen for the *unhandledrejection* event. The *unhandledrejection* event is triggered when a JavaScript `Promise` that has no rejection handler is rejected. The triggered time of the callback for the *unhandledrejection* event and required parameters are the same with those of the onUnhandledRejection() method.

Sample code

copy

```
//.js
App({
  onShow(options) {
    my.onUnhandledRejection((res) => {
      console.log(res.reason);
      console.log(res.promise);
    });
  }
})
```

Parameters

| | | | | --- | --- | --- | | **Property** | **Type** | **Description** | | callback | Function |
The *unhandledrejection* event is triggered when a JavaScript `Promise` that has no rejection handler is rejected. |

Callback function

| | | | | --- | --- | --- | | **Property** | **Type** | **Description** | | reason | String | Rejection reason, which usually is the error object. | | promise | Promise | The rejected Promise object. |

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev/api_event_onunhandledrejection

---

my.onUserCaptureScreen {#myonusercapturescreen}

Last updated: 2022-07-03

Path: miniprogram_gcash

my.onUserCaptureScreen

2022-07-03 18:44

my.onUserCaptureScreen

Listen to the user-initiated active screen capture event. This will receive all the screen capture event notification of the system or a third-party tool.

Sample Code

copy

```
<!-- API-DEMO page/API/user-capture-screen/user-capture-screen.axml-->
<view class="page">
  <view class="page-description">User screen capture event API</view>
  <view class="page-section">
    <view class="page-section-title">my.onUserCaptureScreen</view>
    <view class="page-section-demo">
      <view>Current status: {{ condition ? "listening on" : 'Listening
off' }}</view>
      <view a:if="{{condition}}">
        <button type="primary" onTap="offUserCaptureScreen">Cancel
screen capture listening event</button>
      </view>
      <view a:else>
        <button type="primary" onTap="onUserCaptureScreen">Turn on
screen capture listening event</button>
      </view>
    </view>
  </view>
</view>
```

copy

```
// API-DEMO page/API/user-capture-screen/user-capture-screen.js
Page({
  data: {
    condition: false,
  },
  onReady() {
    my.onUserCaptureScreen(() => {
      my.alert({
        content: 'Received user screen capture',
      });
    });
  },
  offUserCaptureScreen() {
    my.offUserCaptureScreen();
    this.setData({
      condition: false,
    });
  },
  onUserCaptureScreen() {
    my.onUserCaptureScreen(() => {
      my.alert({
        content: 'Received user screen capture'
      });
    });
    this.setData({
      condition: true,
    });
  },
});
```

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev/API_Device_Capture_onUserCaptureScreen

---

my.onUserCaptureScreen {#myonusercapturescreen}

Last updated: 2021-05-09

Path: miniprogram_gcash

my.onUserCaptureScreen

2021-05-09 18:43

my.onUserCaptureScreen

Listen to the user-initiated active screen capture event. This will receive all the screen capture event notification of the system or a third-party tool.

Sample Code

copy

```
<!-- API-DEMO page/API/user-capture-screen/user-capture-screen.axml-->
<view class="page">
  <view class="page-description">User screen capture event API</view>
  <view class="page-section">
    <view class="page-section-title">my.onUserCaptureScreen</view>
    <view class="page-section-demo">
      <view>Current status: {{ condition ? "listening on" : 'Listening off' }}</view>
      <view a:if="{{condition}}">
        <button type="primary" onTap="offUserCaptureScreen">Cancel screen capture listening event</button>
      </view>
      <view a:else>
        <button type="primary" onTap="onUserCaptureScreen">Turn on screen capture listening event</button>
      </view>
    </view>
  </view>
</view>
```

copy

```
// API-DEMO page/API/user-capture-screen/user-capture-screen.js
Page({
  data: {
    condition: false,
  },
  onReady() {
    my.onUserCaptureScreen(() => {
      my.alert({
        content: 'Received user screen capture',
      });
    });
  },
  offUserCaptureScreen() {
    my.offUserCaptureScreen();
    this.setData({
      condition: false,
    });
  },
  onUserCaptureScreen() {
    my.onUserCaptureScreen(() => {
      my.alert({
        content: 'Received user screen capture'
      });
    });
    this.setData({
      condition: true,
    });
  },
});
```

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev-old/api_device_capture_onusercapturescreen

---

my.openBluetoothAdapter {#myopenbluetoothadapter}

Last updated: 2022-07-03

Path: miniprogram_gcash

my.openBluetoothAdapter

2022-07-03 18:44

Use this API to initialize the Bluetooth module in the mini program. You can call the following mini program API and receive event callbacks that are related to the Bluetooth module in the effective period when you the API `my.openBluetoothAdapter` is called. The effective period is ended when the API `my.closeBluetoothAdapter` is called.

Instructions :

- If you call other APIs in the bluetooth module before you call the API `my.openBluetoothAdapter` you get the following error:
- Error Code: 10000
- Error Description: The Bluetooth adapter is not initialized.
- Solution: call the API `my.openBluetoothAdapter`
- When users do not switch on the Bluetooth or the Bluetooth function is not supported on the user's mobile phone, an error is returned after you call the API `my.openBluetoothAdapter`. For more information about error codes, see the error code table. After the Bluetooth module is initialized, you can use the API my.onBluetoothAdapterStateChange to monitor changes of the Bluetooth status.

Note:

Currently simulation in IDE is not supported. Please debug in the production environment.

Code Sample

copy

```
/* .acss */
.help-info {
  padding:10px;
  color:#000000;
}
.help-title {
  padding:10px;
  color:#FC0D1B;
}
```

copy

```
// .json
{
    "defaultTitle": "Bluetooth"
}
```

copy

```
<!-- .axml-->
<view class="page">
  <view class="page-description">Bluetooth API</view>
  <view class="page-section">
    <view class="page-section-title">The Bluetooth state</view>
    <view class="page-section-demo">
       <button type="primary" onTap="openBluetoothAdapter">Initialize
Bluetooth</button>
       <button type="primary" onTap="closeBluetoothAdapter">Close
Bluetooth</button>
       <button type="primary" onTap="getBluetoothAdapterState">Obtain
Bluetooth state</button>
    </view>
    <view class="page-section-title">Scan the Bluetooth device</view>
    <view class="page-section-demo">
       <button type="primary"
onTap="startBluetoothDevicesDiscovery">Start searching</button>
       <button type="primary" onTap="getBluetoothDevices">All devices
found</button>
       <button type="primary" onTap="getConnectedBluetoothDevices">All
devices connected</button>
       <button type="primary"
onTap="stopBluetoothDevicesDiscovery">Stop searching</button>
    </view>
    <view class="page-section-title">Connect the device</view>
    <view class="page-section-demo">
       <input class="input" onInput="bindKeyInput" type="{{text}}"
placeholder="Enter the device ID of the device to connect"></input>
       <button type="primary" onTap="connectBLEDevice">Connect the
device</button>
       <button type="primary" onTap="getBLEDeviceServices">Obtain
device services</button>
       <button type="primary"
onTap="getBLEDeviceCharacteristics">Obtain read and write
characteristics</button>
       <button type="primary" onTap="disconnectBLEDevice">Disconnect
the device</button>
    </view>
     <view class="page-section-title">Read and write data</view>
     <view class="page-section-demo">
       <button type="primary"
onTap="notifyBLECharacteristicValueChange">Listens to the
characteristic data change</button>
```

```
       <button type="primary" onTap="readBLECharacteristicValue">Read
data</button>
       <button type="primary"
onTap="writeBLECharacteristicValue">Write data</button>
       <button type="primary"
onTap="offBLECharacteristicValueChange">Un-listens to characteristic
value</button>
    </view>
     <view class="page-section-title">Other events</view>
     <view class="page-section-demo">
       <button type="primary"
onTap="bluetoothAdapterStateChange">Changes of the Bluetooth
state</button>
       <button type="primary"
onTap="offBluetoothAdapterStateChange">Un-listens to Bluetooth
state</button>
       <button type="primary"
onTap="BLEConnectionStateChanged">Changes of Bluetooth connection
state</button>
       <button type="primary" onTap="offBLEConnectionStateChanged">Un-
listens to Bluetooth connection state</button>

    </view>
  </view>
</view>
```

copy

```
// .js
Page({
  data: {
    devid: '0D9C82AD-1CC0-414D-9526-119E08D28124',
    serid: 'FEE7',
    notifyId: '36F6',
    writeId: '36F5',
    charid: '',
    alldev: [{ deviceId: '' }],
  },

  //Obtain the Bluetooth state
  openBluetoothAdapter() {
    my.openBluetoothAdapter({
      success: res => {
        if (!res.isSupportBLE) {
          my.alert({ content: 'Sorry, your mobile Bluetooth is
unavailable temporarily' });
          return;
        }
        my.alert({ content: 'Succeeded to initialize!' });
      },
      fail: error => {
        my.alert({ content: JSON.stringify(error) });
      },
    });
  },
  closeBluetoothAdapter() {
    my.closeBluetoothAdapter({
      success: () => {
        my.alert({ content: 'Bluetooth closed!' });
      },
      fail: error => {
        my.alert({ content: JSON.stringify(error) });
      },
    });
  },
  getBluetoothAdapterState() {
    my.getBluetoothAdapterState({
      success: res => {
        if (!res.available) {
          my.alert({ content: 'Sorry, your mobile Bluetooth is
unavailable temporarily' });
          return;
        }
        my.alert({ content: JSON.stringify(res) });
      },
      fail: error => {
        my.alert({ content: JSON.stringify(error) });
      },
    });
  },

  //Scan the Bluetooth device
  startBluetoothDevicesDiscovery() {
    my.startBluetoothDevicesDiscovery({
      allowDuplicatesKey: false,
      success: () => {
        my.onBluetoothDeviceFound({
          success: res => {
            // my.alert({content:'Listens to new
device'+JSON.stringify(res)});
            var deviceArray = res.devices;
            for (var i = deviceArray.length - 1; i >= 0; i--) {
              var deviceObj = deviceArray[i];
              //Pair the target device with the device name or
broadcast data, and then record the device ID for later use.
              if (deviceObj.name == this.data.name) {
                my.alert({ content: 'Target device is found' });
                my.offBluetoothDeviceFound();
                this.setData({
                  deviceId: deviceObj.deviceId,
                });
                break;
              }
            }
          },
          fail: error => {
            my.alert({ content: 'Failed to listen to new device' +
JSON.stringify(error) });
          },
        });
      },
      fail: error => {
        my.alert({ content: 'Failed to start scanning' +
JSON.stringify(error) });
      },
    });
  },

  //Stop scanning
  stopBluetoothDevicesDiscovery() {
    my.stopBluetoothDevicesDiscovery({
      success: res => {
        my.offBluetoothDeviceFound();
        my.alert({ content: 'Succeeded!' });
      },
      fail: error => {
        my.alert({ content: JSON.stringify(error) });
      },
    });
  },

  //Obtain the connected device
  getConnectedBluetoothDevices() {
    my.getConnectedBluetoothDevices({
      success: res => {
        if (res.devices.length === 0) {
          my.alert({ content: 'No connecting devices!' });
          return;
        }
        my.alert({ content: JSON.stringify(res) });
        devid = res.devices[0].deviceId;
      },
      fail: error => {
        my.alert({ content: JSON.stringify(error) });
      },
    });
  },

  //Obtain all searched devices
  getBluetoothDevices() {
    my.getBluetoothDevices({
      success: res => {
        my.alert({ content: JSON.stringify(res) });
      },
      fail: error => {
        my.alert({ content: JSON.stringify(error) });
      },
    });
  },
  bindKeyInput(e) {
    this.setData({
      devid: e.detail.value,
    });
  },

  //Connect the device
  connectBLEDevice() {
    my.connectBLEDevice({
      deviceId: this.data.devid,
      success: res => {
        my.alert({ content: 'Succeeded to connect!' });
      },
      fail: error => {
        my.alert({ content: JSON.stringify(error) });
      },
    });
  },

  //Disconnect the device
  disconnectBLEDevice() {
    my.disconnectBLEDevice({
      deviceId: this.data.devid,
      success: () => {
        my.alert({ content: 'Succeeded to disconnect!' });
      },
      fail: error => {
        my.alert({ content: JSON.stringify(error) });
      },
    });
  },

  //Obtain the services of the connected device
  getBLEDeviceServices() {
    my.getConnectedBluetoothDevices({
      success: res => {
        if (res.devices.length === 0) {
          my.alert({ content: 'No connected devices' });
          return;
        }
        my.getBLEDeviceServices({
          deviceId: this.data.devid,
          success: res => {
            my.alert({ content: JSON.stringify(res) });
            this.setData({
              serid: res.services[0].serviceId,
            });
          },
          fail: error => {
            my.alert({ content: JSON.stringify(error) });
          },
        });
      },
    });
  },

  //Obtain the char ID of the connected device, read and write
characteristics are respectively screened out.
  getBLEDeviceCharacteristics() {
    my.getConnectedBluetoothDevices({
      success: res => {
        if (res.devices.length === 0) {
          my.alert({ content: 'No connected devices' });
          return;
        }
        this.setData({
          devid: res.devices[0].deviceId,
        });
        my.getBLEDeviceCharacteristics({
          deviceId: this.data.devid,
          serviceId: this.data.serid,
          success: res => {
            my.alert({ content: JSON.stringify(res) });
            //See the related document for more information of the
properties of the characteristics. Pair the characteristics according
to the properties and record the value for later use.
            this.setData({
              charid: res.characteristics[0].characteristicId,
            });
          },
          fail: error => {
            my.alert({ content: JSON.stringify(error) });
          },
        });
      },
    });
  },

  //Read and write data
  readBLECharacteristicValue() {
    my.getConnectedBluetoothDevices({
      success: res => {
        if (res.devices.length === 0) {
          my.alert({ content: 'No connected devices' });
          return;
        }
        this.setData({
          devid: res.devices[0].deviceId,
        });
        my.readBLECharacteristicValue({
          deviceId: this.data.devid,
          serviceId: this.data.serid,
          characteristicId: this.data.notifyId,
          //1  Android reading service
          // serviceId:'0000180d-0000-1000-8000-00805f9b34fb',
          // characteristicId:'00002a38-0000-1000-8000-00805f9b34fb',
          success: res => {
            my.alert({ content: JSON.stringify(res) });
          },
          fail: error => {
            my.alert({ content: 'Failed to read' +
JSON.stringify(error) });
          },
        });
      },
    });
  },
  writeBLECharacteristicValue() {
    my.getConnectedBluetoothDevices({
      success: res => {
        if (res.devices.length === 0) {
          my.alert({ content: 'No connected devices' });
          return;
        }
        this.setData({
          devid: res.devices[0].deviceId,
        });
        my.writeBLECharacteristicValue({
          deviceId: this.data.devid,
          serviceId: this.data.serid,
          characteristicId: this.data.charid,
          //Android writing service
          //serviceId:'0000180d-0000-1000-8000-00805f9b34fb',
          //characteristicId:'00002a39-0000-1000-8000-00805f9b34fb',
          value: 'ABCD',
          success: res => {
            my.alert({ content: 'Succeeded to write data!' });
          },
          fail: error => {
            my.alert({ content: JSON.stringify(error) });
          },
        });
      },
    });
  },
  notifyBLECharacteristicValueChange() {
    my.getConnectedBluetoothDevices({
      success: res => {
```

```
    if (res.devices.length === 0) {
     my.alert({ content: 'No connected devices' });
     return;
    }
    this.setData({
     devid: res.devices[0].deviceId,
    });
    my.notifyBLECharacteristicValueChange({
     state: true,
     deviceId: this.data.devid,
     serviceId: this.data.serid,
     characteristicId: this.data.notifyId,
     success: () => {
      //Listens to characteristic change events
      my.onBLECharacteristicValueChange({
       success: res => {
        //  my.alert({content: 'Changes of
characteristics  '+JSON.stringify(res)});
        my.alert({ content: 'Obtain the response data = ' +
res.value });
       },
      });
      my.alert({ content: 'Succeeded to listen' });
     },
     fail: error => {
      my.alert({ content: 'Failed to listen' +
JSON.stringify(error) });
     },
    });
   },
  });
 },
 offBLECharacteristicValueChange() {
  my.offBLECharacteristicValueChange();
 },

 //Other events
 bluetoothAdapterStateChange() {

my.onBluetoothAdapterStateChange(this.getBind('onBluetoothAdapterState
 },
 onBluetoothAdapterStateChange() {
  if (res.error) {
   my.alert({ content: JSON.stringify(error) });
  } else {
   my.alert({ content: 'Changes of the Bluetooth state  ' +
JSON.stringify(res) });
  }
 },
 offBluetoothAdapterStateChange() {
```

```
my.offBluetoothAdapterStateChange(this.getBind('onBluetoothAdapterState
  },
  getBind(name) {
    if (!this[`bind${name}`]) {
      this[`bind${name}`] = this[name].bind(this);
    }
    return this[`bind${name}`];
  },
  BLEConnectionStateChanged() {

my.onBLEConnectionStateChanged(this.getBind('onBLEConnectionStateChange
  },
  onBLEConnectionStateChanged(res) {
    if (res.error) {
      my.alert({ content: JSON.stringify(error) });
    } else {
      my.alert({ content: 'Changes of connection state  ' +
JSON.stringify(res) });
    }
  },
  offBLEConnectionStateChanged() {

my.offBLEConnectionStateChanged(this.getBind('onBLEConnectionStateChan
  },
  onUnload() {
    this.offBLEConnectionStateChanged();
    this.offBLECharacteristicValueChange();
    this.offBluetoothAdapterStateChange();
    this.closeBluetoothAdapter();
  },
});
```

Parameters

The input parameters are displayed in the following table:

| | | | | --- | --- | --- | --- | | **Property** | **Type** | **Required** | **Description** | | autoClose | Boolean | No | Indicates whether to automatically disconnect Bluetooth when you leave the current page. The default value is true. Note : Only Android is supported. | | success | Function | No | The callback function for a successful API call. | | fail | Function | No | The callback function for a failed API call. | | complete | Function | No | The callback function for a completed API call (Regardless of whether the call is successful or not). |

Success Callback Function

The input parameters are displayed in the following table:

| | | | --- | --- | --- | | **Property** | **Type** | **Description** | | isSupportBLE | Boolean | Indicates whether BLE is supported. |

Error Code

The input parameters are displayed in the following table:

| | | | | --- | --- | --- | | **Error Code** | **Description** | **Solution** | | 12 | Bluetooth is not turned on. | Try again to turn on Bluetooth. | | 13 | Connection to the system service is temporarily lost. | Try again to reconnect. | | 14 | Not authorized to use Bluetooth. | Authorize app to use Bluetooth. | | 15 | Unknown error. | |

九色鹿

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev/API_Device_Bluetooth_Bluetooth_openBluetoothAdapter

---

my.openBluetoothAdapter {#myopenbluetoothadapter}

Last updated: 2021-05-09

Path: miniprogram_gcash

my.openBluetoothAdapter

2021-05-09 18:43

Use this API to initialize the Bluetooth module in the mini program. You can call the following mini program API and receive event callbacks that are related to the Bluetooth module in the effective period when you the API `my.openBluetoothAdapter` is called. The effective period is ended when the API `my.closeBluetoothAdapter` is called.

Instructions :

- If you call other APIs in the bluetooth module before you call the API `my.openBluetoothAdapter` you get the following error:
- Error Code: 10000
- Error Description: The Bluetooth adapter is not initialized.
- Solution: call the API `my.openBluetoothAdapter`
- When users do not switch on the Bluetooth or the Bluetooth function is not supported on the user's mobile phone, an error is returned after you call the API `my.openBluetoothAdapter`. For more information about error codes, see the error code table. After the Bluetooth module is initialized, you can use the API my.onBluetoothAdapterStateChange to monitor changes of the Bluetooth status.

Note:

Currently simulation in IDE is not supported. Please debug in the production environment.

Code Sample

copy

```
/* .acss */
.help-info {
  padding:10px;
  color:#000000;
}
.help-title {
  padding:10px;
  color:#FC0D1B;
}
```

copy

```
// .json
{
    "defaultTitle": "Bluetooth"
}
```

copy

```
<!-- .axml-->
<view class="page">
  <view class="page-description">Bluetooth API</view>
  <view class="page-section">
    <view class="page-section-title">The Bluetooth state</view>
    <view class="page-section-demo">
       <button type="primary" onTap="openBluetoothAdapter">Initialize
Bluetooth</button>
       <button type="primary" onTap="closeBluetoothAdapter">Close
Bluetooth</button>
       <button type="primary" onTap="getBluetoothAdapterState">Obtain
Bluetooth state</button>
    </view>
    <view class="page-section-title">Scan the Bluetooth device</view>
    <view class="page-section-demo">
       <button type="primary"
onTap="startBluetoothDevicesDiscovery">Start searching</button>
       <button type="primary" onTap="getBluetoothDevices">All devices
found</button>
       <button type="primary" onTap="getConnectedBluetoothDevices">All
devices connected</button>
       <button type="primary"
onTap="stopBluetoothDevicesDiscovery">Stop searching</button>
    </view>
    <view class="page-section-title">Connect the device</view>
    <view class="page-section-demo">
       <input class="input" onInput="bindKeyInput" type="{{text}}"
placeholder="Enter the device ID of the device to connect"></input>
       <button type="primary" onTap="connectBLEDevice">Connect the
device</button>
       <button type="primary" onTap="getBLEDeviceServices">Obtain
device services</button>
       <button type="primary"
onTap="getBLEDeviceCharacteristics">Obtain read and write
characteristics</button>
       <button type="primary" onTap="disconnectBLEDevice">Disconnect
the device</button>
    </view>
     <view class="page-section-title">Read and write data</view>
     <view class="page-section-demo">
       <button type="primary"
onTap="notifyBLECharacteristicValueChange">Listens to the
characteristic data change</button>
       <button type="primary" onTap="readBLECharacteristicValue">Read
data</button>
       <button type="primary"
onTap="writeBLECharacteristicValue">Write data</button>
       <button type="primary"
onTap="offBLECharacteristicValueChange">Un-listens to characteristic
value</button>
    </view>
     <view class="page-section-title">Other events</view>
     <view class="page-section-demo">
       <button type="primary"
onTap="bluetoothAdapterStateChange">Changes of the Bluetooth
state</button>
       <button type="primary"
onTap="offBluetoothAdapterStateChange">Un-listens to Bluetooth
state</button>
       <button type="primary"
onTap="BLEConnectionStateChanged">Changes of Bluetooth connection
state</button>
       <button type="primary" onTap="offBLEConnectionStateChanged">Un-
listens to Bluetooth connection state</button>

    </view>
  </view>
</view>
```

copy

```
// .js
Page({
  data: {
    devid: '0D9C82AD-1CC0-414D-9526-119E08D28124',
    serid: 'FEE7',
    notifyId: '36F6',
    writeId: '36F5',
    charid: '',
    alldev: [{ deviceId: '' }],
  },

  //Obtain the Bluetooth state
  openBluetoothAdapter() {
    my.openBluetoothAdapter({
      success: res => {
        if (!res.isSupportBLE) {
          my.alert({ content: 'Sorry, your mobile Bluetooth is
unavailable temporarily' });
          return;
        }
        my.alert({ content: 'Succeeded to initialize!' });
      },
      fail: error => {
        my.alert({ content: JSON.stringify(error) });
      },
    });
  },
  closeBluetoothAdapter() {
    my.closeBluetoothAdapter({
      success: () => {
        my.alert({ content: 'Bluetooth closed!' });
      },
      fail: error => {
        my.alert({ content: JSON.stringify(error) });
      },
    });
  },
  getBluetoothAdapterState() {
    my.getBluetoothAdapterState({
      success: res => {
        if (!res.available) {
          my.alert({ content: 'Sorry, your mobile Bluetooth is
unavailable temporarily' });
          return;
        }
        my.alert({ content: JSON.stringify(res) });
      },
      fail: error => {
        my.alert({ content: JSON.stringify(error) });
      },
    });
  },

  //Scan the Bluetooth device
  startBluetoothDevicesDiscovery() {
    my.startBluetoothDevicesDiscovery({
      allowDuplicatesKey: false,
      success: () => {
        my.onBluetoothDeviceFound({
          success: res => {
            // my.alert({content:'Listens to new
device'+JSON.stringify(res)});
            var deviceArray = res.devices;
            for (var i = deviceArray.length - 1; i >= 0; i--) {
              var deviceObj = deviceArray[i];
              //Pair the target device with the device name or
broadcast data, and then record the device ID for later use.
              if (deviceObj.name == this.data.name) {
                my.alert({ content: 'Target device is found' });
                my.offBluetoothDeviceFound();
                this.setData({
                  deviceId: deviceObj.deviceId,
                });
                break;
              }
            }
          },
          fail: error => {
            my.alert({ content: 'Failed to listen to new device' +
JSON.stringify(error) });
          },
        });
      },
      fail: error => {
        my.alert({ content: 'Failed to start scanning' +
JSON.stringify(error) });
      },
    });
  },

  //Stop scanning
  stopBluetoothDevicesDiscovery() {
    my.stopBluetoothDevicesDiscovery({
      success: res => {
        my.offBluetoothDeviceFound();
        my.alert({ content: 'Succeeded!' });
      },
      fail: error => {
        my.alert({ content: JSON.stringify(error) });
      },
    });
  },

  //Obtain the connected device
  getConnectedBluetoothDevices() {
    my.getConnectedBluetoothDevices({
      success: res => {
        if (res.devices.length === 0) {
```

```
        my.alert({ content: 'No connecting devices!' });
        return;
      }
      my.alert({ content: JSON.stringify(res) });
      devid = res.devices[0].deviceId;
    },
    fail: error => {
      my.alert({ content: JSON.stringify(error) });
    },
  });
},

//Obtain all searched devices
getBluetoothDevices() {
  my.getBluetoothDevices({
    success: res => {
      my.alert({ content: JSON.stringify(res) });
    },
    fail: error => {
      my.alert({ content: JSON.stringify(error) });
    },
  });
},
bindKeyInput(e) {
  this.setData({
    devid: e.detail.value,
  });
},

//Connect the device
connectBLEDevice() {
  my.connectBLEDevice({
    deviceId: this.data.devid,
    success: res => {
      my.alert({ content: 'Succeeded to connect!' });
    },
    fail: error => {
      my.alert({ content: JSON.stringify(error) });
    },
  });
},

//Disconnect the device
disconnectBLEDevice() {
  my.disconnectBLEDevice({
    deviceId: this.data.devid,
    success: () => {
      my.alert({ content: 'Succeeded to disconnect!' });
    },
    fail: error => {
      my.alert({ content: JSON.stringify(error) });
      },
    });
  },

  //Obtain the services of the connected device
  getBLEDeviceServices() {
    my.getConnectedBluetoothDevices({
      success: res => {
        if (res.devices.length === 0) {
          my.alert({ content: 'No connected devices' });
          return;
        }
        my.getBLEDeviceServices({
          deviceId: this.data.devid,
          success: res => {
            my.alert({ content: JSON.stringify(res) });
            this.setData({
              serid: res.services[0].serviceId,
            });
          },
          fail: error => {
            my.alert({ content: JSON.stringify(error) });
          },
        });
      },
    });
  },

  //Obtain the char ID of the connected device, read and write
characteristics are respectively screened out.
  getBLEDeviceCharacteristics() {
    my.getConnectedBluetoothDevices({
      success: res => {
        if (res.devices.length === 0) {
          my.alert({ content: 'No connected devices' });
          return;
        }
        this.setData({
          devid: res.devices[0].deviceId,
        });
        my.getBLEDeviceCharacteristics({
          deviceId: this.data.devid,
          serviceId: this.data.serid,
          success: res => {
            my.alert({ content: JSON.stringify(res) });
            //See the related document for more information of the
properties of the characteristics. Pair the characteristics according
to the properties and record the value for later use.
            this.setData({
              charid: res.characteristics[0].characteristicId,
            });
          },
          fail: error => {
            my.alert({ content: JSON.stringify(error) });
          },
        });
      },
    });
  },

  //Read and write data
  readBLECharacteristicValue() {
    my.getConnectedBluetoothDevices({
      success: res => {
        if (res.devices.length === 0) {
          my.alert({ content: 'No connected devices' });
          return;
        }
        this.setData({
          devid: res.devices[0].deviceId,
        });
        my.readBLECharacteristicValue({
          deviceId: this.data.devid,
          serviceId: this.data.serid,
          characteristicId: this.data.notifyId,
          //1  Android reading service
          // serviceId:'0000180d-0000-1000-8000-00805f9b34fb',
          // characteristicId:'00002a38-0000-1000-8000-00805f9b34fb',
          success: res => {
            my.alert({ content: JSON.stringify(res) });
          },
          fail: error => {
            my.alert({ content: 'Failed to read' +
JSON.stringify(error) });
          },
        });
      },
    });
  },
  writeBLECharacteristicValue() {
    my.getConnectedBluetoothDevices({
      success: res => {
        if (res.devices.length === 0) {
          my.alert({ content: 'No connected devices' });
          return;
        }
        this.setData({
          devid: res.devices[0].deviceId,
        });
        my.writeBLECharacteristicValue({
          deviceId: this.data.devid,
          serviceId: this.data.serid,
          characteristicId: this.data.charid,
          //Android writing service
          //serviceId:'0000180d-0000-1000-8000-00805f9b34fb',
          //characteristicId:'00002a39-0000-1000-8000-00805f9b34fb',
          value: 'ABCD',
          success: res => {
            my.alert({ content: 'Succeeded to write data!' });
          },
          fail: error => {
            my.alert({ content: JSON.stringify(error) });
          },
        });
      },
    });
  },
  notifyBLECharacteristicValueChange() {
    my.getConnectedBluetoothDevices({
      success: res => {
        if (res.devices.length === 0) {
          my.alert({ content: 'No connected devices' });
          return;
        }
        this.setData({
          devid: res.devices[0].deviceId,
        });
        my.notifyBLECharacteristicValueChange({
          state: true,
          deviceId: this.data.devid,
          serviceId: this.data.serid,
          characteristicId: this.data.notifyId,
          success: () => {
            //Listens to characteristic change events
            my.onBLECharacteristicValueChange({
              success: res => {
                //  my.alert({content: 'Changes of 
characteristics  '+JSON.stringify(res)});
                my.alert({ content: 'Obtain the response data = ' + 
res.value });
              },
            });
            my.alert({ content: 'Succeeded to listen' });
          },
          fail: error => {
            my.alert({ content: 'Failed to listen' + 
JSON.stringify(error) });
          },
        });
      },
    });
  },
  offBLECharacteristicValueChange() {
    my.offBLECharacteristicValueChange();
  },

  //Other events
  bluetoothAdapterStateChange() {

my.onBluetoothAdapterStateChange(this.getBind('onBluetoothAdapterStateC
  },
  onBluetoothAdapterStateChange() {
    if (res.error) {
      my.alert({ content: JSON.stringify(error) });
    } else {
      my.alert({ content: 'Changes of the Bluetooth state  ' +
JSON.stringify(res) });
    }
  },
  offBluetoothAdapterStateChange() {

my.offBluetoothAdapterStateChange(this.getBind('onBluetoothAdapterState
  },
  getBind(name) {
    if (!this[`bind${name}`]) {
      this[`bind${name}`] = this[name].bind(this);
    }
    return this[`bind${name}`];
  },
  BLEConnectionStateChanged() {

my.onBLEConnectionStateChanged(this.getBind('onBLEConnectionStateChange
  },
  onBLEConnectionStateChanged(res) {
    if (res.error) {
      my.alert({ content: JSON.stringify(error) });
    } else {
      my.alert({ content: 'Changes of connection state  ' +
JSON.stringify(res) });
    }
  },
  offBLEConnectionStateChanged() {

my.offBLEConnectionStateChanged(this.getBind('onBLEConnectionStateChang
  },
  onUnload() {
    this.offBLEConnectionStateChanged();
    this.offBLECharacteristicValueChange();
    this.offBluetoothAdapterStateChange();
    this.closeBluetoothAdapter();
  },
});
```

Parameters

The input parameters are displayed in the following table:

Property	**Type**	**Required**	**Description**
autoClose	Boolean	No	Indicates whether to automatically disconnect Bluetooth when you leave the current page. The default value is true. Note : Only Android is supported.
success	Function	No	The callback function for a successful API call.
fail	Function	No	The callback function for a failed API call.
complete	Function	No	The callback function for a completed API call (Regardless of whether the call is successful or not).

Success Callback Function

The input parameters are displayed in the following table:

Property	**Type**	**Description**
isSupportBLE	Boolean	Indicates whether BLE is supported.

Error Code

The input parameters are displayed in the following table:

Error Code	**Description**	**Solution**
12	Bluetooth is not turned on.	Try again to turn on Bluetooth.
13	Connection to the system service is temporarily lost.	Try again to reconnect.
14	Not authorized to use Bluetooth.	Authorize app to use Bluetooth.
15	Unknown error.	

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev-old/api_device_bluetooth_bluetooth_openbluetoothadapter

---

my.openDocument {#myopendocument}

Last updated: 2021-05-09

Path: miniprogram_gcash

my.openDocument

2021-05-09 18:43

The my.openDocument API enables users to preview a PDF file within the mini program.

After clicking on the PDF file, the user can preview the file without leaving the mini program. After closing the preview, the user will still be at the current mini program page.

Sample code

.js

copy

```
// .js
my.downloadFile({
  // Sample url that is not real
      url: 'http://documentExample.com/alipay.pdf',
      success({ apFilePath }) {
        my.hideLoading();
        my.openDocument({
          filePath: apFilePath,
          fileType: 'pdf',
          success: (res) => {
            console.log('open document success')
            }
          })
        }
      })
```

Parameters

Object type

Property	**Type**	**Required**	**Description**
filePath	String	Yes	Obtain file path via my.downloadFile.
fileType	String	Yes	File type.
success	Function	No	Callback function upon call success.
fail	Function	No	Callback function upon call failure.
complete	Function	No	Callback function upon call completion (to be executed upon either call success or failure).

Legal values for fileType

Value	**Description**
pdf	PDF format.

Error code

Error code	**Description**	**Solution**
4011	File path is invalid or no permission to access incoming path.	Check the incoming file path.
4012	Preview files do not exist.	Make sure files that correspond to file path exist.
4013	File format is not supported.	Currently preview of PDF files is supported.

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev-old/api_file_myopendocument

---

my.openDocument {#myopendocument}

Last updated: 2022-07-03

Path: miniprogram_gcash

my.openDocument

2022-07-03 18:44

The my.openDocument API enables users to preview a PDF file within the mini program.

After clicking on the PDF file, the user can preview the file without leaving the mini program. After closing the preview, the user will still be at the current mini program page.

Sample code

.js

copy

```
// .js
my.downloadFile({
  // Sample url that is not real
      url: 'http://documentExample.com/alipay.pdf',
      success({ apFilePath }) {
        my.hideLoading();
        my.openDocument({
          filePath: apFilePath,
          fileType: 'pdf',
          success: (res) => {
            console.log('open document success')
            }
          })
        }
      })
```

Parameters

Object type

Property	**Type**	**Required**	**Description**
filePath	String	Yes	Obtain file path via my.downloadFile.
fileType	String	Yes	File type.
success	Function	No	Callback function upon call success.
fail	Function	No	Callback function upon call failure.
complete	Function	No	Callback function upon call completion (to be executed upon either call success or failure).

Legal values for fileType

Value	**Description**
pdf	PDF format.

Error code

Error code	**Description**	**Solution**
4011	File path is invalid or no permission to access incoming path.	Check the incoming file path.
4012	Preview files do not exist.	Make sure files that correspond to file path exist.
4013	File format is not supported.	Currently preview of PDF files is supported.

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev/api_file_myopendocument

---

my.openLocation {#myopenlocation}

Last updated: 2022-07-03

Path: miniprogram_gcash

my.openLocation

2022-07-03 18:44

View the location on the built-in map.

Sample codes

copy

```
// API-DEMO page/API/open-location/open-location.json
{
    "defaultTitle": "Open Location"
}
```

copy

```
<!-- API-DEMO page/API/open-location/open-location.axml-->
<view class="page">
  <view class="page-section">
    <view class="page-section-demo">
      <text>Longitude</text>
      <input type="text" disabled="{{true}}" value="{{longitude}}"
name="longitude"></input>
```

```
    </view>
    <view class="page-section-demo">
      <text>Latitude</text>
      <input type="text" disabled="{{true}}"  value="{{latitude}}"
name="latitude"></input>
    </view>
    <view class="page-section-demo">
      <text>Name</text>
      <input type="text" disabled="{{true}}"  value="{{name}}"
name="name"></input>
    </view>
    <view class="page-section-demo">
      <text>Address</text>
      <input type="text" disabled="{{true}}"  value="{{address}}"
name="address"></input>
    </view>
    <view class="page-section-btns">
      <view type="primary" formType="submit" onTap="openLocation">open
location</view>
    </view>
  </view>
</view>
```

copy

```
// API-DEMO page/API/open-location/open-location.js
Page({
  data: {
    longitude: '103.873834',
    latitude: '1.355572',
    name: 'Serangoon Stadium',
    address: '33 Yio Chu Kang Rd, Singapore',
  },

  openLocation() {
    my.openLocation({
      longitude: this.data.longitude,
      latitude: this.data.latitude,
      name: this.data.name,
      address: this.data.address,
    })
  }
})
```

Parameters

Property	**Type**	**Required**	**Description**
longitude	String	Yes	The longitude.
latitude	String	Yes	The latitude.
name	String	Yes	The location.
address	String	Yes	The detailed address of the location.
scale	Number		

No | The zoom level. The value ranges from 3 to 19 and is 15 by default. | | success | Function | No | The callback method indicates a successful call. | | fail | Function | No | The callback method indicates a failed call. | | complete | Function | No | The callback method that indicates the call is completed (this will be executed regardless of whether the call succeeds or fails). |

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev/api_location_openlocation

---

my.openLocation {#myopenlocation}

Last updated: 2021-05-10

Path: miniprogram_gcash

my.openLocation

2021-05-10 03:43

View the location on the built-in map.

Sample codes

copy

```
// API-DEMO page/API/open-location/open-location.json
{
    "defaultTitle": "Open Location"
}
```

copy

```
<!-- API-DEMO page/API/open-location/open-location.axml-->
<view class="page">
  <view class="page-section">
    <view class="page-section-demo">
      <text>Longitude</text>
      <input type="text" disabled="{{true}}" value="{{longitude}}"
name="longitude"></input>
    </view>
    <view class="page-section-demo">
      <text>Latitude</text>
      <input type="text" disabled="{{true}}"  value="{{latitude}}"
name="latitude"></input>
    </view>
    <view class="page-section-demo">
      <text>Name</text>
      <input type="text" disabled="{{true}}"  value="{{name}}"
name="name"></input>
    </view>
    <view class="page-section-demo">
      <text>Address</text>
      <input type="text" disabled="{{true}}"  value="{{address}}"
name="address"></input>
    </view>
    <view class="page-section-btns">
      <view type="primary" formType="submit" onTap="openLocation">open
location</view>
    </view>
  </view>
</view>
```

copy

```
// API-DEMO page/API/open-location/open-location.js
Page({
  data: {
    longitude: '103.873834',
    latitude: '1.355572',
    name: 'Serangoon Stadium',
    address: '33 Yio Chu Kang Rd, Singapore',
  },

  openLocation() {
    my.openLocation({
      longitude: this.data.longitude,
      latitude: this.data.latitude,
      name: this.data.name,
      address: this.data.address,
    })
  }
})
```

Parameters

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Required** | **Description** | | longitude | String | Yes | The longitude. | | latitude | String | Yes | The latitude. | | name | String | Yes | The location. | | address | String | Yes | The detailed address of the location. | | scale | Number | No | The zoom level. The value ranges from 3 to 19 and is 15 by default. | | success | Function | No | The callback method indicates a successful call. | | fail | Function | No | The callback method indicates a failed call. | | complete | Function | No | The callback method that indicates the call is completed (this will be executed regardless of whether the call succeeds or fails). |

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev-old/api_location_openlocation

---

my.openSetting {#myopensetting}

Last updated: 2021-05-09

Path: miniprogram_gcash

my.openSetting

2021-05-09 18:43

Use this API to open the Mini Program settings page and returns permission setting results. Only the permissions that have been requested by the Mini Program from the user are displayed on the settings page.

Sample Code

copy

```
my.openSetting({
  success: (res) => {
    /*
     * res.authSetting = {
     *   "userInfo": true,
     *   "location": true,
     *   ...
     * }
     */
  }
})
```

Parameters

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Required** | **Description** | | success | Function | No | The callback function for a successful API call. | | fail | Function | No | The callback function for a failed API call. | | complete | Function | No | The callback function used when the API call is completed. This function is always executed no matter the call succeeds or fails. |

Success callback function

| | | | | --- | --- | --- | | **Property** | **Type** | **Description** | | authSetting | Object | Results of user authorization. Keys are the values of `scopes` and values are boolean types, which shows whether the user gives the permission or not. See Scopes for details. |

Scopes

| | | | | --- | --- | --- | | **Scope** | **API** | **Description** | | location | my.getLocation | This field specifies whether to authorize access to geographic location. | | album | my.chooseImage、 my.saveImage | This field specifies whether to authorize to save images to the albums. | | camera | my.scan | This field specifies whether to authorize access to camera. | | userInfo | my.getOpenUserInfo | This field specifies whether to authorize access to user information. |

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev-old/api_device_setting_opensetting

---

my.openSetting {#myopensetting}

Last updated: 2022-07-03

Path: miniprogram_gcash

my.openSetting

2022-07-03 18:44

Use this API to open the Mini Program settings page and returns permission setting results. Only the permissions that have been requested by the Mini Program from the user are displayed on the settings page.

Sample Code

copy

```
my.openSetting({
  success: (res) => {
    /*
     * res.authSetting = {
     *   "userInfo": true,
     *   "location": true,
     *   ...
     * }
     */
  }
})
```

Parameters

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Required** | **Description** | | success | Function | No | The callback function for a successful API call. | | fail | Function | No | The callback function for a failed API call. | | complete | Function | No | The callback function used when the API call is completed. This function is always executed no matter the call succeeds or fails. |

Success callback function

| | | | | --- | --- | --- | | **Property** | **Type** | **Description** | | authSetting | Object | Results of user authorization. Keys are the values of `scopes` and values are boolean types, which shows whether the user gives the permission or not. See Scopes for details. |

Scopes

| | | | | --- | --- | --- | | **Scope** | **API** | **Description** | | location | my.getLocation | This field specifies whether to authorize access to geographic location. | | album | my.chooseImage、 my.saveImage | This field specifies whether to authorize to save images to the albums. | | camera | my.scan | This field specifies whether to authorize access to camera. | | userInfo | my.getOpenUserInfo | This field specifies whether to authorize access to user information. |

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev/api_device_setting_opensetting

---

my.pageScrollTo {#mypagescrollto}

Last updated: 2021-05-09

Path: miniprogram_gcash

my.pageScrollTo

2021-05-09 18:43

Scroll to the target position on the page

Note:

- The scrollTop has a higher priority than selector.

- When the my.pageScrollTo is used to jump to the top of the Mini Program, the scrollTop value must be set as a number greater than 0 to make jump possible.

Sample Code

copy

```
<!-- .axml -->
<view class="page">
  <view class="page-description">Page scroll API</view>

  <view class="page-section">
    <view class="page-section-title">
      my.pageScrollTo
    </view>
    <view class="page-section-demo">
      <input type="text" placeholder="key" name="key" value="
{{scrollTop}}" onInput="scrollTopChange"></input>
    </view>
    <view class="page-section-btns">
      <view onTap="scrollTo">Page scroll </view>
    </view>
  </view>

  <view style="height:1000px"/>
</view>
```

copy

```
//.js
Page({
  data: {
    scrollTop: 0,
  },
  scrollTopChange(e) {
    this.setData({
      scrollTop: e.detail.value,
    });
  },
  onPageScroll({ scrollTop }) {
    console.log('onPageScroll', scrollTop);
  },
  scrollTo() {
    my.pageScrollTo({
      scrollTop: parseInt(this.data.scrollTop),
      duration: 300,
    });
  },
});
```

Parameters

Object type with the following attributes:

| | | | | | | --- | --- | --- | --- | --- | | **Property** | **Type** | **Default** | **Required** | **Description** | | scrollTop | Number | - | No | Scroll to the target position on the page, in px When the my.pageScrollTo is used to jump to the top of the Mini Program, the scrollTop value must be set as a number greater than 0 to make jump possible. | | duration | Number | 0 | No | Duration of scroll animation, in ms. | | selector | string | - | No | Selector. | | success | Function | - | No | Callback function upon call success. | | fail | Function | - | No | Callback function upon call failure. | | complete | Function | - | No | Callback function upon call completion (to be executed upon either call success or failure). |

Selector Syntax

When the selector parameter is transferred, the framework executes document.querySelector(selector) to select the target node.

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev-old/api_ui_scroll_pagescrollto

---

my.pageScrollTo {#mypagescrollto}

Last updated: 2022-07-03

Path: miniprogram_gcash

my.pageScrollTo

2022-07-03 18:44

Scroll to the target position on the page

Note:

- The scrollTop has a higher priority than selector.

- When the my.pageScrollTo is used to jump to the top of the Mini Program, the scrollTop value must be set as a number greater than 0 to make jump possible.

Sample Code

copy

```
<!-- .axml -->
<view class="page">
  <view class="page-description">Page scroll API</view>
  <view class="page-section">
    <view class="page-section-title">
      my.pageScrollTo
    </view>
    <view class="page-section-demo">
      <input type="text" placeholder="key" name="key" value="
{{scrollTop}}" onInput="scrollTopChange"></input>
    </view>
    <view class="page-section-btns">
      <view onTap="scrollTo">Page scroll </view>
    </view>
  </view>

  <view style="height:1000px"/>
</view>
```

copy

```
//.js
Page({
  data: {
    scrollTop: 0,
  },
  scrollTopChange(e) {
    this.setData({
      scrollTop: e.detail.value,
    });
  },
  onPageScroll({ scrollTop }) {
    console.log('onPageScroll', scrollTop);
  },
  scrollTo() {
    my.pageScrollTo({
      scrollTop: parseInt(this.data.scrollTop),
      duration: 300,
    });
  },
});
```

Parameters

Object type with the following attributes:

| | | | | | | --- | --- | --- | --- | --- | | **Property** | **Type** | **Default** | **Required** | **Description** | | scrollTop | Number | - | No | Scroll to the target position on the page, in px When the my.pageScrollTo is used to jump to the top of the Mini Program, the scrollTop value must be set as a number greater than 0 to make jump possible. | | duration | Number | 0 | No | Duration of scroll animation, in ms. | | selector | string | - | No | Selector. | | success |

Function | - | No | Callback function upon call success. | | fail | Function | - | No | Callback function upon call failure. | | complete | Function | - | No | Callback function upon call completion (to be executed upon either call success or failure). |

Selector Syntax

When the selector parameter is transferred, the framework executes document.querySelector(selector) to select the target node.

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev/api_ui_scroll_pagescrollto

---

my.previewImage {#mypreviewimage}

Last updated: 2021-05-09

Path: miniprogram_gcash

my.previewImage

2021-05-09 18:43

The Preview image's "local image path" is not supported.

Sample Code

copy

```
//.json
{
    "defaultTitle": "Picture"
}
```

copy

```
<!-- .axml -->
<view class="page">
  <view class="page-section">
    <view class="page-section-btns">
      <view onTap="chooseImage">Select image</view>
      <view onTap="previewImage">Preview image</view>
      <view onTap="saveImage">Save image</view>
    </view>
  </view>
</view>
```

copy

```
//.js
Page({
  chooseImage() {
    my.chooseImage({
      sourceType: ['camera','album'],
      count: 2,
      success: (res) => {
        my.alert({
          content: JSON.stringify(res),

        });
      },
      fail:()=>{
        my.showToast({
          content: 'fail', // text contents
        });
      }
    })
  },
  previewImage() {
    my.previewImage({
      current: 2,
      urls: [\
        'https://img.example.com/example1.jpg',\
        'https://img.example.com/example2.jpg',\
        'https://img.example.com/example3.jpg'\
      ],
    });
  },
  saveImage() {
    my.saveImage({
      url: 'https://img.example.com/example1.jpg',
      showActionSheet: true,
      success: () => {
        my.alert({
          title: 'Save success',
        });
      },
    });
  }
});
```

Parameters

The incoming parameter is of the Object type with the following attributes:

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Required** | **Description** | | urls | Array | Yes | Http url list of the images to be previewed. | | current | Number | No | Index of the current displayed image, 0 by default, indicating the first image in the URLs. | | success | Function | No | Callback function upon call success. | | fail | Function | No | Callback function upon call failure. | | complete | Function | No | Callback function upon call completion (to be executed upon either call success or failure). |

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev-old/api_media_image_previewimage

---

my.previewImage {#mypreviewimage}

Last updated: 2022-07-03

Path: miniprogram_gcash

my.previewImage

2022-07-03 18:44

The Preview image's "local image path" is not supported.

Sample Code

copy

```
//.json
{
    "defaultTitle": "Picture"
}
```

copy

```
<!-- .axml -->
<view class="page">
  <view class="page-section">
    <view class="page-section-btns">
      <view onTap="chooseImage">Select image</view>
      <view onTap="previewImage">Preview image</view>
      <view onTap="saveImage">Save image</view>
    </view>
  </view>
</view>
```

copy

```
//.js
Page({
  chooseImage() {
    my.chooseImage({
      sourceType: ['camera','album'],
      count: 2,
      success: (res) => {
        my.alert({
          content: JSON.stringify(res),

        });
      },
      fail:()=>{
        my.showToast({
          content: 'fail', // text contents
        });
      }
    })
  },
  previewImage() {
    my.previewImage({
      current: 2,
      urls: [\
        'https://img.example.com/example1.jpg',\
        'https://img.example.com/example2.jpg',\
        'https://img.example.com/example3.jpg'\
      ],
    });
  },
  saveImage() {
    my.saveImage({
      url: 'https://img.example.com/example1.jpg',
      showActionSheet: true,
      success: () => {
        my.alert({
          title: 'Save success',
        });
      },
    });
  }
});
```

Parameters

The incoming parameter is of the Object type with the following attributes:

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Required** | **Description** | | urls | Array | Yes | Http url list of the images to be previewed. | | current | Number | No | Index of the current displayed image, 0 by default, indicating the first image in the URLs. | | success |

Function | No | Callback function upon call success. | | fail | Function | No | Callback function upon call failure. | | complete | Function | No | Callback function upon call completion (to be executed upon either call success or failure). |

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev/api_media_image_previewimage

---

my.prompt {#myprompt}

Last updated: 2021-05-09

Path: miniprogram_gcash

my.prompt

2021-05-09 18:43

Pop up a dialog to show the prompt message.

Sample Code

copy

```
my.prompt({
  title: 'Title',
  message: 'Explain the current status and prompt the user solution.
It is best not to exceed two lines.',
  placeholder: 'Leave a message to a friend',
  okButtonText: 'Confirm',
  cancelButtonText: 'Cancel',
  success: (result) => {
    my.alert({
      title: JSON.stringify(result),
    });
  },
});
```

Parameters

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Required** | **Description** | | title | String | No | Title of prompt box. | | message | String | Yes | Text of prompt box, which is “Enter contents here” by default. | | placeholder | String | No | Prompt text for the entry box. | | align | String | No | Message alignment. Valid values are:
- left

- center
- right

The default value is `center` for both iOS and Android. | | okButtonText | String | No | OK button text, which is **OK** by default. | | cancelButtonText | String | No | Cancel button text, which is **Cancel** by default. | | success | Function | No | Callback function upon call success. | | fail | Function | No | Callback function upon call failure. | | complete | Function | No | Callback function upon call completion (to be executed upon either call success or failure). |

Success Callback Function

The incoming parameter is of the Object type with the following attributes:

| | | | | --- | --- | --- | | **Property** | **Type** | **Description** | | ok | Boolean | Click OK to return true; click Cancel to return false. | | inputValue | String | When OK is true, return the user's entry. |

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev-old/api_ui_feedback_prompt

---

my.prompt {#myprompt}

Last updated: 2022-07-03

Path: miniprogram_gcash

my.prompt

2022-07-03 18:44

Pop up a dialog to show the prompt message.

Sample Code

copy

```
my.prompt({
  title: 'Title',
  message: 'Explain the current status and prompt the user solution.
It is best not to exceed two lines.',
  placeholder: 'Leave a message to a friend',
  okButtonText: 'Confirm',
  cancelButtonText: 'Cancel',
  success: (result) => {
    my.alert({
      title: JSON.stringify(result),
    });
  },
});
```

Parameters

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Required** | **Description** | | title | String | No | Title of prompt box. | | message | String | Yes | Text of prompt box, which is “Enter contents here” by default. | | placeholder | String | No | Prompt text for the entry box. | | align | String | No | Message alignment. Valid values are:
- left
- center
- right
The default value is `center` for both iOS and Android. | | okButtonText | String | No | OK button text, which is **OK** by default. | | cancelButtonText | String | No | Cancel button text, which is **Cancel** by default. | | success | Function | No | Callback function upon call success. | | fail | Function | No | Callback function upon call failure. | | complete | Function | No | Callback function upon call completion (to be executed upon either call success or failure). |

Success Callback Function

The incoming parameter is of the Object type with the following attributes:

| | | | | --- | --- | --- | | **Property** | **Type** | **Description** | | ok | Boolean | Click OK to return true; click Cancel to return false. | | inputValue | String | When OK is true, return the user's entry. |

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev/api_ui_feedback_prompt

---

my.reLaunch {#myrelaunch}

Last updated: 2022-07-03

Path: miniprogram_gcash

my.reLaunch

2022-07-03 18:44

Close page stacks and jump to the specified page within the application.

Sample Code

copy

```
my.reLaunch({
  url: '/page/index'
})
```

Parameters

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Required** | **Description** | | url | String | Yes | Page path If the page is not a tabbar page, the path can be followed by parameters. Rules for the parameters: The path and parameter are separated with
`?`
, the parameter key and the parameter value are connected with
`=`
, and different parameters must be separated with
`&`
, such as
`path?key1=value1&key2=value2`
. | | success | Function | No | Callback function upon call success. | | fail | Function | No | Callback function upon call failure. | | complete | Function | No | Callback function upon call completion (to be executed upon either call success or failure). |

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev/api_ui_route_relaunch

---

my.reLaunch {#myrelaunch}

Last updated: 2021-05-09

Path: miniprogram_gcash

my.reLaunch

2021-05-09 18:43

Close page stacks and jump to the specified page within the application.

Sample Code

copy

```
my.reLaunch({
  url: '/page/index'
})
```

Parameters

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Required** | **Description** | | url | String | Yes | Page path If the page is not a tabbar page, the path can be followed by parameters. Rules for the parameters: The path and parameter are separated with

`?`

, the parameter key and the parameter value are connected with

`=`

, and different parameters must be separated with

`&`

, such as

`path?key1=value1&key2=value2`

. | | success | Function | No | Callback function upon call success. | | fail | Function | No | Callback function upon call failure. | | complete | Function | No | Callback function upon call completion (to be executed upon either call success or failure). |

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev-old/api_ui_route_relaunch

---

my.readBLECharacteristicValue {#myreadblecharacteristicvalue}

Last updated: 2022-07-03

Path: miniprogram_gcash

my.readBLECharacteristicValue

2022-07-03 18:44

Use this API to read the data of Bluetooth Low Energy (BLE) device characteristics. The returned information must be obtained by using the callback in the my.onBLECharacteristicValueChange.

Instructions:

- The device characteristics must support read to use this API. See properties in my.getBLEDeviceCharacteristics for details.
- Multiple parallel reading and writing calls might cause failures.
- If the reading times out, the error code is 10015. However, my.onBLECharacteristicValueChange interface might return data later. Please consider how to handle this situation.

Note:

Currently simulation in IDE is not supported. Please debug in production environment.

Sample Code

copy

```
my.readBLECharacteristicValue({
  deviceId: deviceId,
  serviceId: serviceId,
  characteristicId: characteristicId,
  success: (res) => {
    console.log(res)
  },
  fail:(res) => {
  },
  complete: (res)=>{
  }
})
```

Parameters

| | | | | --- | --- | --- | --- | | **Property** | **Type** | **Required** | **Description** | | deviceId | String | Yes | The Bluetooth device ID. | | serviceId | String | Yes | The UUID of the service corresponding to a Bluetooth characteristic. | | characteristicId | String | Yes | The Bluetooth characteristic UUID. | | success | Function | No | The callback function for a successful API call. | | fail | Function | No | The callback function for a failed API call. | | complete | Function | No | The callback function used when the API call is completed. This function is always executed no matter the call succeeds or fails. |

Success Callback Function

| | | | --- | --- | --- | | **Property** | **Type** | **Description** | | characteristic | Object | Information of the device characteristic. |

characteristic

Information of the Bluetooth device characteristic

| | | | --- | --- | --- | | **Property** | **Type** | **Description** | | characteristicId | String | The Bluetooth characteristic UUID. | | serviceId | String | The UUID of the service corresponding to a Bluetooth characteristic. | | value | Hex String | The Bluetooth characteristic value. |

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev/API_Device_Bluetooth_BLE_readBLECharacteristicValue

---

my.readBLECharacteristicValue {#myreadblecharacteristicvalue}

Last updated: 2021-05-09

Path: miniprogram_gcash

my.readBLECharacteristicValue

2021-05-09 18:43

Use this API to read the data of Bluetooth Low Energy (BLE) device characteristics. The returned information must be obtained by using the callback in the my.onBLECharacteristicValueChange.

Instructions:

- The device characteristics must support read to use this API. See properties in my.getBLEDeviceCharacteristics for details.
- Multiple parallel reading and writing calls might cause failures.
- If the reading times out, the error code is 10015. However, my.onBLECharacteristicValueChange interface might return data later. Please consider how to handle this situation.

Note:

Currently simulation in IDE is not supported. Please debug in production environment.

Sample Code

copy

```
my.readBLECharacteristicValue({
  deviceId: deviceId,
  serviceId: serviceId,
  characteristicId: characteristicId,
  success: (res) => {
    console.log(res)
  },
  fail:(res) => {
  },
  complete: (res)=>{
  }
})
```

Parameters

| | | | | --- | --- | --- | --- | | **Property** | **Type** | **Required** | **Description** | | deviceId | String | Yes | The Bluetooth device ID. | | serviceId | String | Yes | The UUID of the service corresponding to a Bluetooth characteristic. | | characteristicId | String | Yes | The Bluetooth characteristic UUID. | | success | Function | No | The callback function for a successful API call. | | fail | Function | No | The callback function for a failed API call. | | complete | Function | No | The callback function used when the API call is completed. This function is always executed no matter the call succeeds or fails. |

Success Callback Function

| | | | --- | --- | --- | | **Property** | **Type** | **Description** | | characteristic | Object | Information of the device characteristic. |

characteristic

Information of the Bluetooth device characteristic

| | | | --- | --- | --- | | **Property** | **Type** | **Description** | | characteristicId | String | The Bluetooth characteristic UUID. | | serviceId | String | The UUID of the service corresponding to a Bluetooth characteristic. | | value | Hex String | The Bluetooth characteristic value. |

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev-old/api_device_bluetooth_ble_readblecharacteristicvalue

---

my.redirectTo {#myredirectto}

Last updated: 2021-05-09

Path: miniprogram_gcash

my.redirectTo

2021-05-09 18:43

Close the current page and jump to the specified page within the application.

Sample Code

copy

```
my.redirectTo({
  url: 'new_page?count=100'
})
```

Parameters

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Required** | **Description** | | url | String | Yes | The application for the jumping does not include the destination page path of the tabBar. The path can be followed by parameters. Rules for the parameters: The path and parameter are separated with `?`, the parameter key and the parameter value are connected with `=`, and different parameters must be separated with `&`, such as `path?key1=value1&key2=value2`. | | success | Function | No | Callback function upon call success. | | fail | Function | No | Callback function upon call failure. | | complete | Function | No | Callback function upon call completion (to be executed upon either call success or failure). |

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev-old/api_ui_route_redirectto

---

my.redirectTo {#myredirectto}

Last updated: 2022-07-03

Path: miniprogram_gcash

my.redirectTo

2022-07-03 18:44

Close the current page and jump to the specified page within the application.

Sample Code

copy

```
my.redirectTo({
  url: 'new_page?count=100'
})
```

Parameters

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Required** | **Description** | | url | String | Yes | The application for the jumping does not include the destination page path of the tabBar. The path can be followed by parameters. Rules for the parameters: The path and parameter are separated with `?`, the parameter key and the parameter value are connected with `=`, and different parameters must be separated with `&`, such as `path?key1=value1&key2=value2`. | | success | Function | No | Callback function upon call

success. | | fail | Function | No | Callback function upon call failure. | | complete | Function | No | Callback function upon call completion (to be executed upon either call success or failure). |

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev/api_ui_route_redirectto

---

my.removeSavedFile {#myremovesavedfile}

Last updated: 2021-05-09

Path: miniprogram_gcash

my.removeSavedFile

2021-05-09 18:43

Delete a saved file.

Sample Code

copy

```
my.getSavedFileList({
  success:(res)=>{
    my.removeSavedFile({
      apFilePath:res.fileList[0].apFilePath,
      success:(res)=>{
        console.log('remove success')
      }
    })
  }
});
```

Parameters

Object type with the following attributes:

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Required** | **Description** | | apFilePath | String | Yes | File path. | | success | Function | No | Callback function upon call success. | | fail | Function | No | Callback function upon call failure. | | complete | Function | No | Callback function upon call completion (to be executed upon either call success or failure). |

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev-old/api_file_removesavedfile

---

my.removeSavedFile {#myremovesavedfile}

Last updated: 2022-07-03

Path: miniprogram_gcash

my.removeSavedFile

2022-07-03 18:44

Delete a saved file.

Sample Code

copy

```
my.getSavedFileList({
  success:(res)=>{
    my.removeSavedFile({
      apFilePath:res.fileList[0].apFilePath,
      success:(res)=>{
        console.log('remove success')
      }
    })
  }
});
```

Parameters

Object type with the following attributes:

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Required** | **Description** | | apFilePath | String | Yes | File path. | | success | Function | No | Callback function upon call success. | | fail | Function | No | Callback function upon call failure. | | complete | Function | No | Callback function upon call completion (to be executed upon either call success or failure). |

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev/API_File_removeSavedFile

---

my.removeStorage {#myremovestorage}

Last updated: 2021-05-09

Path: miniprogram_gcash

my.removeStorage

2021-05-09 18:43

Remove cached data.

> This is an asynchronous interface.
>
> Removing the webview embedded storage data will not remove the storage data of the Mini Program.

Sample Code

copy

```
my.removeStorage({
  key: 'currentCity',
  success: function(){
    my.alert({content: 'remove success'});
  }
});
```

Parameters

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Required** | **Description** | | key | String | Yes | Cache data key. | | success | Function | No | Callback function upon call success. | | fail | Function | No | Callback function upon call failure. | | complete | Function | No | Callback function upon call completion (to be executed upon either call success or failure). |

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev-old/api_storage_removestorage

---

my.removeStorage {#myremovestorage}

Last updated: 2022-07-03

Path: miniprogram_gcash

my.removeStorage

2022-07-03 18:44

Remove cached data.

> This is an asynchronous interface.

> Removing the webview embedded storage data will not remove the storage data of the Mini Program.

Sample Code

copy

```
my.removeStorage({
  key: 'currentCity',
  success: function(){
    my.alert({content: 'remove success'});
  }
});
```

Parameters

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Required** | **Description** | | key | String | Yes | Cache data key. | | success | Function | No | Callback function upon call success. | | fail | Function | No | Callback function upon call failure. | | complete | Function | No | Callback function upon call completion (to be executed upon either call success or failure). |

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev/API_Storage_removeStorage

---

my.removeStorageSync {#myremovestoragesync}

Last updated: 2021-05-09

Path: miniprogram_gcash

my.removeStorageSync

2021-05-09 18:43

Remove cached data synchronously.

> This is a synchronous interface.

Sample Code

copy

```
my.removeStorageSync({
  key: 'currentCity',
});
```

Parameters

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Required** | **Description** | | key | String | Yes | Cache data key. |

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev-old/api_storage_removestoragesync

---

my.removeStorageSync {#myremovestoragesync}

Last updated: 2022-07-03

Path: miniprogram_gcash

my.removeStorageSync

2022-07-03 18:44

Remove cached data synchronously.

> This is a synchronous interface.

Sample Code

copy

```
my.removeStorageSync({
  key: 'currentCity',
});
```

Parameters

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Required** | **Description** | | key | String | Yes | Cache data key. |

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev/API_Storage_removeStorageSync

---

my.request {#myrequest}

Last updated: 2022-07-03

Path: miniprogram_gcash

my.request

2022-07-03 18:44

Network request of a Mini Program.

Notes

1. Configure the domain name whitelist at first under `Mini Program` > `Configuration` > `Server Domain Whitelist`. The Mini Program can only communicate with the domain names in the whitelist during the API calls: `my.request` (send HTTP request), `my.uploadFile` (upload file), and `my.downloadFile` (download file).
2. During the Mini Program development, in the developer tool, select whether to ignore the httpRequest domain name validity check under `Details` > `Domain name information`. If yes, the domain name validity will not be checked in the simulator, preview, and real machine debugging scenarios. However, before the Mini Program goes online, you must maintain the domain names in the whitelist, otherwise, the domain names cannot be effective in the official release.

Important: `my.request` request header is **{'content-type': 'application/json'}** by default,instead of **{'content-type': 'application/x-www-form-urlencoded'}**.

Sample Code

copy

```
my.request({
  url: 'https://httpbin.org/post',
  method: 'POST',
  data: {
    from: 'Mini Program',
    production: 'JSAPI',
  },
  dataType: 'json',
  success: function(res) {
    my.alert({content: 'success'});
  },
  fail: function(res) {
    my.alert({content: 'fail'});
  },
  complete: function(res) {
    my.hideLoading();
    my.alert({content: 'complete'});
  }
});

const task = my.request({url: 'https://httpbin.org/post'})
task.abort()
```

Parameters

| | | | | --- | --- | --- | --- | | **Property** | **Type** | **Required** | **Description** | | url | String | Yes | Target server url. | | headers | Object | No | Set the request HTTP header. The default value is
`{'content-type': 'application/json'}`
. | | method | String | No | The default value is GET. Both GET and POST are supported. | | data | Object | No | Request parameter. | | timeout | Number | No | Timeout period in ms. The default value is 30000. | | dataType | String | No | Expected format of the returned data. The following formats are supported:
- json
- text
- base64
The default format is json. | | success | Function | No | The callback function for a successful API call. | | fail | Function | No | The callback function for a failed API call. | | complete | Function | No | The callback function used when the API call is completed. This function is always executed no matter the call succeeds or fails. |

Data Parameter Description

Data transferred to the server is eventually expressed in String. If the type is not String, the data will be converted into String. Conversion rules are:

- If the method is `GET`, the data will be converted into query string: encodeURIComponent(k)=encodeURIComponent(v)&encodeURIComponent(k)=encodeURIComponent(v)...
- If the method is `POST` and the `headers['content-type']` is `application/json`, the data will be JSON serialized.
- If the method is `POST` and the `headers['content-type']` is `application/x-www-form-urlencoded`, the data will be converted into query string: encodeURIComponent(k)=encodeURIComponent(v)&encodeURIComponent(k)=encodeURIComponent(v)...

Success Callback Function

The type of the incoming parameter is Object with the following attributes:

| | | | --- | --- | --- | | **Property** | **Type** | **Description** | | data | String | Response data. The format depends on the value of `dataType` in the request. | | status | Number | Response code. | | headers | Object | Response header. |

Error Code

| | | | --- | --- | | **Error** | **Description** | | 2 | Incorrect parameter. Check whether the URL is in HTTPS format and all parameters are correct. | | 4 | Not authorized to call the interface. The error might result from the following:
- The server domain whitelist is not configured. In this case, configure the server domain whitelist under `Mini Program` > `Configuration` > `Server Domain Whitelist`. The mini program can only communicate with the domains in the whitelist when calling the `my.request` and `my.uploadFile` APIs. If the server domain whitelist is updated, the whitelist comes effective only when a new version of the mini program is published.
- Errors occur with your account and you cannot login to the mini program platform. In this case, when you debug in the Mini Program studio, select *Ignore httpRequest domain name validity check* or *Ignore Webview domain name validity check* under `Details` > `Domain name information`, and then preview the debugging request.
- Check whether the HTTP request domain name is incorrect. | | 12 | Network error. | | 13 | Timeout. | | 14 | Decoding failure. | | 19 | HTTP error. | | 20 | Request stopped/service end traffic limit. |

Note:

The error code of 14 might be returned in the following cases:

- When the value of `dataType` is `json`, the Mini Program framework firstly perform the `JSON.prase` operation on the returned results. If the parsing fails, the error code of 14 is returned.
- When the value of `dataType` is `text` and the returned content has a bad format, the error code of 14 is returned.

To solve the error, check whether the `dataType` setting is incorrect firstly.

Return Value

RequestTask

Network request task object.

Method

RequestTask.abort()

Note:

If **Not authorized to call the interface** is returned, configure the domain whitelist under `Mini Program` > `Configuration` > `Server Domain Whitelist` in the mini program platform.

Source:
https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev/api_network_request

---

my.request {#myrequest}

Last updated: 2021-05-09

Path: miniprogram_gcash

my.request

2021-05-09 18:43

Network request of a Mini Program.

Notes

1. Configure the domain name whitelist at first under `Mini Program` > `Configuration` > `Server Domain Whitelist`. The Mini Program can only communicate with the domain names in the whitelist during the API calls: `my.request` (send HTTP request), `my.uploadFile` (upload file), and `my.downloadFile` (download file).
2. During the Mini Program development, in the developer tool, select whether to ignore the httpRequest domain name validity check under `Details` > `Domain name information`. If yes, the domain name validity will not be checked in the simulator, preview, and real machine debugging scenarios. However, before the Mini Program goes online, you must maintain the domain names in the whitelist, otherwise, the domain names cannot be effective in the official release.

Important: `my.request` request header is **{'content-type': 'application/json'}** by default,instead of **{'content-type': 'application/x-www-form-urlencoded'}**.

Sample Code

copy

```
my.request({
  url: 'https://httpbin.org/post',
  method: 'POST',
  data: {
    from: 'Mini Program',
    production: 'JSAPI',
  },
  dataType: 'json',
  success: function(res) {
    my.alert({content: 'success'});
  },
  fail: function(res) {
    my.alert({content: 'fail'});
  },
  complete: function(res) {
    my.hideLoading();
    my.alert({content: 'complete'});
  }
});

const task = my.request({url: 'https://httpbin.org/post'})
task.abort()
```

Parameters

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Required** | **Description** | | url | String | Yes | Target server url. | | headers | Object | No | Set the request HTTP header. The default value is
`{'content-type': 'application/json'}`
. | | method | String | No | The default value is GET. Both GET and POST are supported. | | data | Object | No | Request parameter. | | timeout | Number | No | Timeout period in ms. The default value is 30000. | | dataType | String | No | Expected format of the returned data. The following formats are supported:
- json
- text
- base64

The default format is json. | | success | Function | No | The callback function for a successful API call. | | fail | Function | No | The callback function for a failed API call. | | complete | Function | No | The callback function used when the API call is completed. This function is always executed no matter the call succeeds or fails. |

Data Parameter Description

Data transferred to the server is eventually expressed in String. If the type is not String, the data will be converted into String. Conversion rules are:

- If the method is `GET`, the data will be converted into query string: encodeURIComponent(k)=encodeURIComponent(v)&encodeURIComponent(k)=encodeURIComponent(v)...
- If the method is `POST` and the `headers['content-type']` is `application/json`, the data will be JSON serialized.
- If the method is `POST` and the `headers['content-type']` is `application/x-www-form-urlencoded`, the data will be converted into query string: encodeURIComponent(k)=encodeURIComponent(v)&encodeURIComponent(k)=encodeURIComponent(v)...

Success Callback Function

The type of the incoming parameter is Object with the following attributes:

| | | | | --- | --- | --- | | **Property** | **Type** | **Description** | | data | String | Response data. The format depends on the value of `dataType` in the request. | | status | Number | Response code. | | headers | Object | Response header. |

Error Code

| | | | --- | --- | | **Error** | **Description** | | 2 | Incorrect parameter. Check whether the URL is in HTTPS format and all parameters are correct. | | 4 | Not authorized to call the interface. The error might result from the following:
- The server domain whitelist is not configured. In this case, configure the server domain whitelist under `Mini Program` > `Configuration` > `Server Domain Whitelist`. The mini program can only communicate with the domains in the whitelist when calling the `my.request` and `my.uploadFile` APIs. If the server domain whitelist is updated, the whitelist comes effective only when a new version of the mini program is published.
- Errors occur with your account and you cannot login to the mini program platform. In this case, when you debug in the Mini Program studio, select *Ignore httpRequest domain name validity check* or *Ignore Webview domain name validity check* under `Details` > `Domain name information`, and then preview the debugging request.
- Check whether the HTTP request domain name is incorrect. | | 12 | Network error. | | 13 | Timeout. | | 14 | Decoding failure. | | 19 | HTTP error. | | 20 | Request stopped/service end traffic limit. |

Note:

The error code of 14 might be returned in the following cases:

- When the value of `dataType` is `json`, the Mini Program framework firstly perform the `JSON.prase` operation on the returned results. If the parsing fails, the error code of 14 is returned.
- When the value of `dataType` is `text` and the returned content has a bad format, the error code of 14 is returned.

To solve the error, check whether the `dataType` setting is incorrect firstly.

Return Value

RequestTask

Network request task object.

Method

RequestTask.abort()

Note:

If **Not authorized to call the interface** is returned, configure the domain whitelist under `Mini Program` > `Configuration` > `Server Domain Whitelist` in the mini program platform.

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev-old/api_network_request

---

my.saveFile {#mysavefile}

Last updated: 2021-05-09

Path: miniprogram_gcash

my.saveFile

2021-05-09 18:43

Save file in a local position (total capacity of local file size is limited to 10 MB).

Sample Code

copy

```
my.chooseImage({
  success: (res) => {
    my.saveFile({
      apFilePath: res.apFilePaths[0],
      success: (res) => {
        console.log(JSON.stringify(res))
      },
    });
  },
});
```

Parameters

Object type with the following attributes:

Property	**Type**	**Required**	**Description**
apFilePath	String	Yes	File path.
success	Function	No	Callback function upon call success.
fail	Function	No	Callback function upon call failure.
complete	Function	No	Callback function upon call completion (to be executed upon either call success or failure).

Success Callback Function

The incoming parameter is of the Object type with the following attributes:

Property	**Type**	**Description**
apFilePath	String	File save path.

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev-old/api_file_savefile

my.saveFile {#mysavefile}

Last updated: 2022-07-03

Path: miniprogram_gcash

my.saveFile

2022-07-03 18:44

Save file in a local position (total capacity of local file size is limited to 10 MB).

Sample Code

copy

```
my.chooseImage({
  success: (res) => {
    my.saveFile({
      apFilePath: res.apFilePaths[0],
      success: (res) => {
        console.log(JSON.stringify(res))
      },
    });
  },
});
```

Parameters

Object type with the following attributes:

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Required** | **Description** | | apFilePath | String | Yes | File path. | | success | Function | No | Callback function upon call success. | | fail | Function | No | Callback function upon call failure. | | complete | Function | No | Callback function upon call completion (to be executed upon either call success or failure). |

Success Callback Function

The incoming parameter is of the Object type with the following attributes:

| | | | | --- | --- | --- | | **Property** | **Type** | **Description** | | apFilePath | String | File save path. |

Source:
https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev/api_file_savefile

---

my.saveImage {#mysaveimage}

Last updated: 2021-05-09

Path: miniprogram_gcash

my.saveImage

2021-05-09 18:43

Save the online images to a device camera gallery.

Sample Code

copy

```
my.saveImage({url:'https://img.example.com/example.jpg'});
```

Parameters

Property	**Type**	**Required**	**Description**
url	String	Yes	URL of the image to be saved.
showActionSheet	Boolean	No	Whether to show the menu when saving image, by default it is `true`.
success	Function	No	Callback function upon call success.
fail	Function	No	Callback function upon call failure.
complete	Function	No	Callback function upon call completion (to be executed upon either call success or failure).

Error Code

Error	**Description**
2	Invalid parameter, no url parameter transferred.
15	Album right not enabled (iOS only).
16	Insufficient album storage on cellphone (iOS only).
17	Other errors during picture saving.

九色鹿

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev-old/api_media_image_saveimage

---

my.saveImage {#mysaveimage}

Last updated: 2022-07-03

Path: miniprogram_gcash

my.saveImage

2022-07-03 18:44

Save the online images to a device camera gallery.

Sample Code

copy

```
my.saveImage({url:'https://img.example.com/example.jpg'});
```

Parameters

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Required** | **Description** | | url | String | Yes | URL of the image to be saved. | | showActionSheet | Boolean | No | Whether to show the menu when saving image; By default, it is `false`. | | success | Function | No | Callback function upon call success. | | fail | Function | No | Callback function upon call failure. | | complete | Function | No | Callback function upon call completion (to be executed upon either call success or failure). |

Error Code

| | | | --- | --- | | **Error** | **Description** | | 2 | Invalid parameter, no url parameter   transferred. | | 15 | Album right not enabled (iOS only). | | 16 | Insufficient album storage on cellphone (iOS only). | | 17 | Other errors during picture saving. |

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev/API_Media_Image_saveImage

---

my.scan {#myscan}

Last updated: 2022-07-03

Path: miniprogram_gcash

my.scan

2022-07-03 18:44

Call the scan QR code or bar code function.

Hint: after calling `my.scan` API, the `onHide()` and `onShow()` lifecycle callback function of `App` and `Page` will be executed. The following is the executing sequence:

`App.onHide()` -> `Page.onHide()` -> `App.onShow()` -> `Page.onShow()`.

Sample Code

copy

```
Page({
  scan() {
    my.scan({
      type: 'qr',
      success: (res) => {
        my.alert({ title: res.code });
      },
    });
  }
})
```

Parameters

| | | | | --- | --- | --- | --- | | **Property** | **Type** | **Required** | **Description** | | type | String | No | Type for scanning (`qr` by default):
1. `qr`: two-dimensional QR scanning frame.
2. `bar`: Linear barcode scanning frame. | | hideAlbum | Boolean | No | Hide album entry or not. If it is `false`, there will be an entry for user to select a picture from the album as the scanned picture. Otherwise, user will use camera to scan the content directly. By default, its value is `false`. | | success | Function | No | Callback function upon call success. | | fail | Function | No | Callback function upon call failure. | | complete | Function | No | Callback function upon call completion (to be executed upon either call success or failure). |

Success Callback Function

The incoming parameter is of the Object type with the following attributes:

| | | | --- | --- | --- | | **Property** | **Type** | **Description** | | code | String | Data returned from scanning. | | qrCode | String | QR code data returned from scanning. | | barCode | String | Bar code data returned from scanning. |

Error Code

| | | --- | --- | | **Error** | **Description** | | 10 | User cancellation. | | 11 | Operation failure. |

Source:
https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev/API_Device_Scan_scan

---

my.scan {#myscan}

Last updated: 2021-05-09

Path: miniprogram_gcash

my.scan

2021-05-09 18:43

Call the scan QR code or bar code function.

Hint: after calling `my.scan` API, the `onHide()` and `onShow()` lifecycle callback function of `App` and `Page` will be executed. The following is the executing sequence:

`App.onHide()` -> `Page.onHide()` -> `App.onShow()` -> `Page.onShow()`.

Sample Code

copy

```
Page({
  scan() {
    my.scan({
      type: 'qr',
      success: (res) => {
        my.alert({ title: res.code });
      },
    });
  }
})
```

Parameters

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Required** | **Description** | | type | String | No |
Type for scanning (`qr` by default):
1. `qr`: two-dimensional QR scanning frame.
2. `bar`: Linear barcode scanning frame. | | hideAlbum | Boolean | No | Hide album entry or not. If it is `false`, there will be an entry for user to select a picture from the album as the scanned picture. Otherwise, user will use camera to scan the content directly. By default, its value is `false`. | | success | Function | No | Callback function upon call success. | | fail | Function | No | Callback function upon call failure. | | complete | Function | No | Callback function upon call completion (to be executed upon either call success or failure). |

Success Callback Function

The incoming parameter is of the Object type with the following attributes:

| | | | | --- | --- | --- | | **Property** | **Type** | **Description** | | code | String | Data returned from scanning. | | qrCode | String | QR code data returned from scanning. | | barCode | String | Bar code data returned from scanning. |

Error Code

| | | | --- | --- | | **Error** | **Description** | | 10 | User cancellation. | | 11 | Operation failure. |

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev-old/api_device_scan_scan

---

my.sendSocketMessage {#mysendsocketmessage}

Last updated: 2021-05-09

Path: miniprogram_gcash

my.sendSocketMessage

2021-05-09 18:43

Use this API to send data over WebSocket connection. Call my.connectSocket first and send data after the my.onSocketOpen callback.

Sample Code

copy

```
my.sendSocketMessage({
    data: this.data.toSendMessage, // The data to be sent
    success: (res) => {
        my.alert({content: 'The data sent!' +
this.data.toSendMessage});
    },
});
```

Note: The case is only for reference. Please use your own URL to test.

Parameters

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Required** | **Description** | | data | String | Yes | The data to be sent, in text string or base64 encoded string. | | isBuffer | Boolean | No | If the value of this field is `true`, the `data` field represents the received base64 encoded binary strings. If the value of this field is empty, the `data` field represents normal strings. | | success | Function | No | The callback function for a successful API call. | | fail | Function | No | The callback function for a failed API call. | | complete | Function | No | The callback function used when the API call is completed. This function is always executed no matter the call succeeds or fails. |

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev-old/api_network_sendsocketmessage

---

my.sendSocketMessage {#mysendsocketmessage}

Last updated: 2022-07-03

Path: miniprogram_gcash

my.sendSocketMessage

2022-07-03 18:44

Use this API to send data over WebSocket connection. Call my.connectSocket first and send data after the my.onSocketOpen callback.

Sample Code

copy

```
my.sendSocketMessage({
    data: this.data.toSendMessage, // The data to be sent
    success: (res) => {
        my.alert({content: 'The data sent!' +
this.data.toSendMessage});
    },
});
```

Note: The case is only for reference. Please use your own URL to test.

Parameters

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Required** | **Description** | | data | String | Yes | The data to be sent, in text string or base64 encoded string. | | isBuffer | Boolean | No | If the value of this field is `true`, the `data` field represents the received base64 encoded binary strings. If the value of this field is empty, the `data` field represents normal strings. | | success | Function | No | The callback function for a successful API call. | | fail | Function | No | The callback function for a failed API call. | | complete | Function | No | The callback function used when the API call is completed. This function is always executed no matter the call succeeds or fails. |

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev/api_network_sendsocketmessage

---

my.setBackgroundColor {#mysetbackgroundcolor}

Last updated: 2021-05-09

Path: miniprogram_gcash

my.setBackgroundColor

2021-05-09 18:43

Dynamically set window background color.

Sample Code

copy

```
my.setBackgroundColor({
    backgroundColor: '#ff0000',
    backgroundColorTop: '#00ff00',
    backgroundColorBottom: '#ff00ff'
})
```

Parameters

The incoming parameter is of the Object type with the following attributes:

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Required** | **Description** | | backgroundColor | HexColor | No | Window background color. For Android, this property is mandatory. | | backgroundColorTop | HexColor | No | Top window background color, supported in iOS only. | | backgroundColorBottom | HexColor | No | Bottom window background color,

supported in iOS only. | | success | Function | No | Callback function upon call success. | | fail | Function | No | Callback function upon call failure. | | complete | Function | No | Callback function upon call completion (to be executed upon either call success or failure). |

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev-old/api_ui_set-background_setbackgroundcolor

---

my.setBackgroundColor {#mysetbackgroundcolor}

Last updated: 2022-07-03

Path: miniprogram_gcash

my.setBackgroundColor

2022-07-03 18:44

Dynamically set window background color.

Sample Code

copy

```
my.setBackgroundColor({
    backgroundColor: '#ff0000',
    backgroundColorTop: '#00ff00',
    backgroundColorBottom: '#ff00ff'
})
```

Parameters

The incoming parameter is of the Object type with the following attributes:

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Required** | **Description** | | backgroundColor | HexColor | No | Window background color. For Android, this property is mandatory. | | backgroundColorTop | HexColor | No | Top window background color, supported in iOS only. | | backgroundColorBottom | HexColor | No | Bottom window background color, supported in iOS only. | | success | Function | No | Callback function upon call success. | | fail | Function | No | Callback function upon call failure. | | complete | Function | No | Callback function upon call completion (to be executed upon either call success or failure). |

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev/api_ui_set-background_setbackgroundcolor

---

my.setCanPullDown {#mysetcanpulldown}

Last updated: 2021-05-09

Path: miniprogram_gcash

my.setCanPullDown

2021-05-09 18:43

Set whether to support pull-down on the page (supported by default on Mini Program pages).

Sample Code

copy

```
my.setCanPullDown({
  canPullDown:true
})
```

Parameters

The incoming parameter is of the Object type with the following attributes:

Property	**Type**	**Required**	**Description**
canPullDown	Boolean	Yes	Support pull-down or not.

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev-old/api_ui_set-pulldown_setcanpulldown

---

my.setCanPullDown {#mysetcanpulldown}

Last updated: 2022-07-03

Path: miniprogram_gcash

my.setCanPullDown

2022-07-03 18:44

Set whether to support pull-down on the page (supported by default on Mini Program pages).

Sample Code

copy

```
my.setCanPullDown({
  canPullDown:true
})
```

Parameters

The incoming parameter is of the Object type with the following attributes:

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Required** | **Description** | | canPullDown | Boolean | Yes | Support pull-down or not. |

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev/api_ui_set-pulldown_setcanpulldown

---

my.setClipboard {#mysetclipboard}

Last updated: 2022-07-03

Path: miniprogram_gcash

my.setClipboard

2022-07-03 18:44

Set the clipboard data.

Sample Code

copy

```
Page({
  data: {
    text: '3.1415926',
    copy: '',
  },

  handleCopy() {
    my.setClipboard({
      text: this.data.text,
    });
  },
});
```

Parameters

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Required** | **Description** | | text | String | Yes | Clipboard data. | | success | Function | No | Callback function upon call success. | | fail | Function | No | Callback function upon call failure. | | complete | Function | No | Callback function upon call completion (to be executed upon either call success or failure). |

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev/api_device_clipboard_setclipboard

---

my.setClipboard {#mysetclipboard}

Last updated: 2021-05-09

Path: miniprogram_gcash

my.setClipboard

2021-05-09 18:43

Set the clipboard data.

Sample Code

copy

```
Page({
  data: {
    text: '3.1415926',
    copy: '',
  },

  handleCopy() {
    my.setClipboard({
      text: this.data.text,
    });
  },
});
```

Parameters

| | | | | --- | --- | --- | --- | | **Property** | **Type** | **Required** | **Description** | | text | String | Yes | Clipboard data. | | success | Function | No | Callback function upon call success. | | fail | Function | No | Callback function upon call failure. | | complete | Function | No | Callback function upon call completion (to be executed upon either call success or failure). |

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev-old/api_device_clipboard_setclipboard

---

my.setKeepScreenOn {#mysetkeepscreenon}

Last updated: 2021-05-10

Path: miniprogram_gcash

my.setKeepScreenOn

2021-05-10 03:43

Set whether to keep screen on Takes effect in the current Mini Program only.

Sample Code

copy

```
<!-- API-DEMO page/API/screen/screen.axml-->
<view class="page">
  <view class="page-description">Screen brightness API</view>
  <view class="page-section">
    <view class="page-section-title">Set whether to keep screen
on</view>
    <view class="page-section-demo">
      <switch checked="{{status}}" onChange="switchKeepScreenOn"/>
    </view>
  </view>
  <view class="page-section">
    <view class="page-section-title">Set screen brightness</view>
    <view class="page-section-demo">
      <slider value="{{brightness}}" max="1" min="0"
onChange="sliderChange" step="0.02"/>
    </view>
  </view>
  <view class="page-section">
    <view class="page-section-title">Get screen brightness</view>
    <view class="page-section-demo">
```

```
      <button type="primary" onTap="getBrightness">Get screen
brightness</button>
    </view>
  </view>
</view>
```

copy

```
// API-DEMO page/API/screen/screen.js
Page({
  data: {
    status: false,
    brightness: 1,
  },
  onLoad() {
    my.getScreenBrightness({
      success: res => {
        this.setData({
          brightness: res.brightness
        })
      },
    })
  },
  sliderChange(e) {
    my.setScreenBrightness({
      brightness: e.detail.value,
      success: (res) => {
        this.setData({
          brightness: e.detail.value,
        })
      }
    })
  },
  switchKeepScreenOn(e) {
    my.setKeepScreenOn({
      keepScreenOn: e.detail.value,
      success: (res) => {
        this.setData({
          status: e.detail.value,
        })
      }
    })
  },
  getBrightness() {
    my.getScreenBrightness({
      success: res => {
        my.alert({
          content: `Current screen brightness: ${res.brightness}`
        });
      }
    })
  }
});
```

Parameters

Object type with the following attributes:

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Required** | **Description** | | keepScreenOn | Boolean | Yes | Whether to keep screen on. | | success | Function | No | Callback function upon call success. | | fail | Function | No | Callback function upon call failure. | | complete | Function | No | Callback function upon call completion (to be executed upon either call success or failure). |

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev-old/api_device_screen_setkeepscreenon

---

my.setKeepScreenOn {#mysetkeepscreenon}

Last updated: 2022-07-03

Path: miniprogram_gcash

my.setKeepScreenOn

2022-07-03 18:44

Set whether to keep screen on Takes effect in the current Mini Program only.

Sample Code

copy

```
<!-- API-DEMO page/API/screen/screen.axml-->
<view class="page">
  <view class="page-description">Screen brightness API</view>
  <view class="page-section">
    <view class="page-section-title">Set whether to keep screen
on</view>
    <view class="page-section-demo">
      <switch checked="{{status}}" onChange="switchKeepScreenOn"/>
    </view>
  </view>
  <view class="page-section">
    <view class="page-section-title">Set screen brightness</view>
    <view class="page-section-demo">
```

```
      <slider value="{{brightness}}" max="1" min="0"
onChange="sliderChange" step="0.02"/>
    </view>
  </view>
  <view class="page-section">
    <view class="page-section-title">Get screen brightness</view>
    <view class="page-section-demo">
      <button type="primary" onTap="getBrightness">Get screen
brightness</button>
    </view>
  </view>
</view>
```

copy

```
// API-DEMO page/API/screen/screen.js
Page({
  data: {
    status: false,
    brightness: 1,
  },
  onLoad() {
    my.getScreenBrightness({
      success: res => {
        this.setData({
          brightness: res.brightness
        })
      },
    })
  },
  sliderChange(e) {
    my.setScreenBrightness({
      brightness: e.detail.value,
      success: (res) => {
        this.setData({
          brightness: e.detail.value,
        })
      }
    })
  },
  switchKeepScreenOn(e) {
    my.setKeepScreenOn({
      keepScreenOn: e.detail.value,
      success: (res) => {
        this.setData({
          status: e.detail.value,
        })
      }
    })
  },
  getBrightness() {
    my.getScreenBrightness({
      success: res => {
        my.alert({
          content: `Current screen brightness: ${res.brightness}`
        });
      }
    })
  }
});
```

Parameters

Object type with the following attributes:

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Required** | **Description** | | keepScreenOn | Boolean | Yes | Whether to keep screen on. | | success | Function | No | Callback function upon call success. | | fail | Function | No | Callback function upon call failure. | | complete | Function | No | Callback function upon call completion (to be executed upon either call success or failure). |

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev/api_device_screen_setkeepscreenon

---

my.setNavigationBar {#mysetnavigationbar}

Last updated: 2021-05-09

Path: miniprogram_gcash

my.setNavigationBar

2021-05-09 18:43

Set the navigation bar text and style.

Sample Code

copy

```
my.setNavigationBar({
  title: 'hello',
  backgroundColor: '#108ee9',
  success() {
    my.alert({
      content: 'Success',
    });
  },
  fail() {
    my.alert({
      content: 'Failed',
    });
  },
});
```

Parameters

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Required** | **Description** | | title | String | No | Navigation bar title. | | image | String | No | Picture link address, must be https. Use 3x high-definition pictures. If the image is set, the title parameter is inactive. | | backgroundColor | String | No | Navigation bar background color, supporting hex color value. | | borderBottomColor | String | No | Navigation bar bottom border color, supporting hex color value If the backgroundColor is set, the borderBottomColor does not take effect. The backgroundColor is used by default. | | reset | Boolean | No | Whether the navigation bar is reset to the default color, false by default. | | success | Function | No | Callback function upon call success. | | fail | Function | No | Callback function upon call failure. | | complete | Function | No | Callback function upon call completion (to be executed upon either call success or failure). |

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev-old/api_ui_navigationbar_setnavigationbar

---

my.setNavigationBar {#mysetnavigationbar}

Last updated: 2022-07-03

Path: miniprogram_gcash

my.setNavigationBar

2022-07-03 18:44

Set the navigation bar text and style.

Sample Code

copy

```
my.setNavigationBar({
  title: 'hello',
  backgroundColor: '#108ee9',
  success() {
    my.alert({
      content: 'Success',
    });
  },
  fail() {
    my.alert({
      content: 'Failed',
    });
  },
});
```

Parameters

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Required** | **Description** | | title | String | No | Navigation bar title. | | image | String | No | Picture link address, must be https. Use 3x high-definition pictures. If the image is set, the title parameter is inactive. Currently, SVG images are not supported. | | backgroundColor | String | No | Navigation bar background color, supporting hex color value. | | borderBottomColor | String | No | Navigation bar bottom border color, supporting hex color value If the backgroundColor is set, the borderBottomColor does not take effect. The backgroundColor is used by default. | | reset | Boolean | No | Whether the navigation bar is reset to the default color, false by default. | | success | Function | No | Callback function upon call success. | | fail | Function | No | Callback function upon call failure. | | complete | Function | No | Callback function upon call completion (to be executed upon either call success or failure). |

九色鹿

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev/API_UI_NavigationBar_setNavigationBar

---

my.setScreenBrightness {#mysetscreenbrightness}

Last updated: 2022-07-03

Path: miniprogram_gcash

my.setScreenBrightness

2022-07-03 18:44

Set screen brightness

Sample Code

copy

```
<!-- API-DEMO page/API/screen/screen.axml-->
<view class="page">
  <view class="page-description">Screen brightness API</view>
  <view class="page-section">
    <view class="page-section-title">Set whether to keep screen
on</view>
    <view class="page-section-demo">
      <switch checked="{{status}}" onChange="switchKeepScreenOn"/>
    </view>
  </view>
  <view class="page-section">
    <view class="page-section-title">Set screen brightness</view>
    <view class="page-section-demo">
      <slider value="{{brightness}}" max="1" min="0"
onChange="sliderChange" step="0.02"/>
    </view>
  </view>
  <view class="page-section">
    <view class="page-section-title">Get screen brightness</view>
    <view class="page-section-demo">
      <button type="primary" onTap="getBrightness">Get screen
brightness</button>
    </view>
  </view>
</view>
```

copy

```
// API-DEMO page/API/screen/screen.js
Page({
  data: {
    status: false,
    brightness: 1,
  },
  onLoad() {
    my.getScreenBrightness({
      success: res => {
        this.setData({
          brightness: res.brightness
        })
      },
    })
  },
  sliderChange(e) {
    my.setScreenBrightness({
      brightness: e.detail.value,
      success: (res) => {
        this.setData({
          brightness: e.detail.value,
        })
      }
    })
  },
  switchKeepScreenOn(e) {
    my.setKeepScreenOn({
      keepScreenOn: e.detail.value,
      success: (res) => {
        this.setData({
          status: e.detail.value,
        })
      }
    })
  },
  getBrightness() {
    my.getScreenBrightness({
      success: res => {
        my.alert({
          content: `Current screen brightness: ${res.brightness}`
        });
      }
    })
  }
});
```

Parameters

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Required** | **Description** | | brightness | Number | Yes | Screen brightness for the setting, range 0-1. | | success | Function | No | Callback function upon call success. | | fail | Function | No | Callback function upon call failure. | | complete | Function | No | Callback function upon call completion (to be executed upon either call success or failure). |

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev/api_device_screen_setscreenbrightness

---

my.setScreenBrightness {#mysetscreenbrightness}

Last updated: 2021-05-09

Path: miniprogram_gcash

my.setScreenBrightness

2021-05-09 18:43

Set screen brightness

Sample Code

copy

```
<!-- API-DEMO page/API/screen/screen.axml-->
<view class="page">
  <view class="page-description">Screen brightness API</view>
  <view class="page-section">
    <view class="page-section-title">Set whether to keep screen
on</view>
    <view class="page-section-demo">
      <switch checked="{{status}}" onChange="switchKeepScreenOn"/>
    </view>
  </view>
  <view class="page-section">
    <view class="page-section-title">Set screen brightness</view>
    <view class="page-section-demo">
      <slider value="{{brightness}}" max="1" min="0"
onChange="sliderChange" step="0.02"/>
    </view>
  </view>
  <view class="page-section">
    <view class="page-section-title">Get screen brightness</view>
    <view class="page-section-demo">
      <button type="primary" onTap="getBrightness">Get screen
brightness</button>
    </view>
  </view>
</view>
```

copy

```
// API-DEMO page/API/screen/screen.js
Page({
  data: {
    status: false,
    brightness: 1,
  },
  onLoad() {
    my.getScreenBrightness({
      success: res => {
        this.setData({
          brightness: res.brightness
        })
      },
    })
  },
  sliderChange(e) {
    my.setScreenBrightness({
      brightness: e.detail.value,
      success: (res) => {
        this.setData({
          brightness: e.detail.value,
        })
      }
    })
  },
  switchKeepScreenOn(e) {
    my.setKeepScreenOn({
      keepScreenOn: e.detail.value,
      success: (res) => {
        this.setData({
          status: e.detail.value,
        })
      }
    })
  },
  getBrightness() {
    my.getScreenBrightness({
      success: res => {
        my.alert({
          content: `Current screen brightness: ${res.brightness}`
        });
      }
    })
  }
});
```

Parameters

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Required** | **Description** | | brightness | Number | Yes | Screen brightness for the setting, range 0-1. | | success | Function | No | Callback function upon call success. | | fail | Function | No | Callback function upon call failure. | | complete | Function | No | Callback function upon call completion (to be executed upon either call success or failure). |

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev-old/api_device_screen_setscreenbrightness

---

my.setScreenOrientation {#mysetscreenorientation}

Last updated: 2022-07-04

Path: miniprogram_gcash

my.setScreenOrientation

2022-07-04 03:44

Call this API to set screen orientation.

> **Note:** For Android, calling this API to set screen orientation will be effective for all pages on the current mini program by default. When this API is called on a page to set the screen orientation, for example, to `portrait`, the screen orientation of all subpages entering from this page will also be `portrait`, the same as that of this page. Through certain processing, when returning from this page to previous pages, the orientation of these pages will not be affected.

Sample code

copy

```
my.setScreenOrientation({
  orientation: 'portrait',
  success: (res) => {
    my.alert({
      title: 'success',
      content: JSON.stringify(res)
    })
  },
  fail: (res) => {
    my.alert({
      title: 'fail',
      content: JSON.stringify(res)
    })
  }
})
```

Parameters

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Required** | **Description** | | orientation | String | Yes | Indicates the orientation of the screen, `portrait` or `landscape`. | | success | Function | No | Callback function upon call success | | fail | Function | No | Callback function upon call failure | | complete | Function | No | Callback function upon call completion (to be executed upon either call success or failure) |

Success callback function

| | | | | --- | --- | --- | | **Property** | **Type** | **Description** | | success | Boolean | Specifies whether the call is successful. When the value is `true`, the call is successful. |

Fail callback function

| | | | | --- | --- | --- | | **Property** | **Type** | **Description** | | error | Number | The error code for the failure | | errorMessage | String | The error message that describes the result code in detail |

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev/api_device_screen_setscreenorientation

---

my.setStorage {#mysetstorage}

Last updated: 2022-07-03

Path: miniprogram_gcash

my.setStorage

2022-07-03 18:44

Store the data with a specified key in the local cache. This will overlaps\ the original data using the same key.

> This is an asynchronous interface.
>
> Support the isolation of webview-embedded storage and the Mini Program storage. Specifying key storage data in embedded webview does not overlap the data corresponding to the same key of the Mini Program itself.

Sample Code

copy

```
my.setStorage({
  key: 'currentCity',
  data: {
    cityName: 'London',
    adCode: '330100',
    spell: ' London',
  },
  success: function() {
    my.alert({content: 'Set Success'});
  }
});
```

Parameters

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Required** | **Description** | | key | String | Yes | Cache data key. | | data | Object/String | Yes | Data to be cached. | | success | Function | No | Callback function upon call success. | | fail | Function | No | Callback function upon call failure. | | complete | Function | No | Callback function upon call completion (to be executed upon either call success or failure). |

> Note: When a single data is converted into a string, the string length is maximum 200*1024. For the same user, the cache of one Mini Program has an upper limit 10MB.

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev/api_storage_setstorage

---

my.setStorage {#mysetstorage}

Last updated: 2021-05-09

Path: miniprogram_gcash

my.setStorage

2021-05-09 18:43

Store the data with a specified key in the local cache. This will overlaps\ the original data using the same key.

> This is an asynchronous interface.
>
> Support the isolation of webview-embedded storage and the Mini Program storage. Specifying key storage data in embedded webview does not overlap the data corresponding to the same key of the Mini Program itself.

Sample Code

copy

```
my.setStorage({
  key: 'currentCity',
  data: {
    cityName: 'London',
    adCode: '330100',
    spell: ' London',
  },
  success: function() {
    my.alert({content: 'Set Success'});
  }
});
```

Parameters

| | | | | --- | --- | --- | --- | | **Property** | **Type** | **Required** | **Description** | | key | String | Yes | Cache data key. | | data | Object/String | Yes | Data to be cached. | | success | Function | No | Callback function upon call success. | | fail | Function | No | Callback function upon call failure. | | complete | Function | No | Callback function upon call completion (to be executed upon either call success or failure). |

> Note: When a single data is converted into a string, the string length is maximum 200*1024. For the same user, the cache of one Mini Program has an upper limit 10MB.

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev-old/api_storage_setstorage

---

my.setStorageSync {#mysetstoragesync}

Last updated: 2022-07-03

Path: miniprogram_gcash

my.setStorageSync

2022-07-03 18:44

Store synchronously the data in the specified key in the local cache.

> This is a synchronous interface.

Sample Code

copy

```
my.setStorageSync({
  key: 'currentCity',
  data: {
    cityName: 'London',
    adCode: '330100',
    spell: ' London',
  }
});
```

Parameters

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Required** | **Description** | | key | String | Yes | Cache data key. | | data | Object/String | Yes | Data to be cached. |

九色鹿

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev/api_storage_setstoragesync

---

my.setStorageSync {#mysetstoragesync}

Last updated: 2021-05-09

Path: miniprogram_gcash

my.setStorageSync

2021-05-09 18:43

Store synchronously the data in the specified key in the local cache.

> This is a synchronous interface.

Sample Code

copy

```
my.setStorageSync({
  key: 'currentCity',
  data: {
    cityName: 'London',
    adCode: '330100',
    spell: ' London',
  }
});
```

Parameters

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Required** | **Description** | | key | String | Yes | Cache data key. | | data | Object/String | Yes | Data to be cached. |

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev-old/api_storage_setstoragesync

my.setTabBarStyle {#mysettabbarstyle}

Last updated: 2022-07-03

Path: miniprogram_gcash

my.setTabBarStyle

2022-07-03 18:44

Use this API to set the overall style of the tab bar, such as the text color, background color and border style.

For more information, see Tab bar FAQ.

Note:

Currently simulation in IDE is not supported. Please debug in production environment.

Sample Code

copy

```
my.setTabBarStyle({
    color: '#FF0000',
    selectedColor: '#00FF00',
    backgroundColor: '#0000FF',
    borderStyle: 'white'
})
```

Parameters

Property	**Type**	**Required**	**Description**
color	HexColor	Yes	The default color of text on tab.
selectedColor	HexColor	Yes	The color of the selected text on tab.
backgroundColor	HexColor	Yes	The background color of tab.
borderStyle	String	Yes	The border color of tab bar, which only supports `black` / `white`.
success	Function	No	The callback function for a successful API call.
fail	Function	No	The callback function for a failed API call.
complete	Function	No	The callback function used when the API call is completed. This function is always executed no matter the call succeeds or fails.

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev/api_ui_tabbar_settabbarstyle

my.setTabBarStyle {#mysettabbarstyle}

Last updated: 2021-05-10

Path: miniprogram_gcash

my.setTabBarStyle

2021-05-10 03:43

Use this API to set the overall style of the tab bar, such as the text color, background color and border style.

For more information, see Tab bar FAQ.

Note:

Currently simulation in IDE is not supported. Please debug in production environment.

Sample Code

copy

```
my.setTabBarStyle({
    color: '#FF0000',
    selectedColor: '#00FF00',
    backgroundColor: '#0000FF',
    borderStyle: 'white'
})
```

Parameters

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Required** | **Description** | | color | HexColor | Yes | The default color of text on tab. | | selectedColor | HexColor | Yes | The color of the selected text on tab. | | backgroundColor | HexColor | Yes | The background color of tab. | | borderStyle | String | Yes | The border color of tab bar, which only supports `black` / `white`. | | success | Function | No | The callback function for a successful API call. | | fail | Function | No | The callback function for a failed API call. | | complete | Function | No | The callback function used when the API call is completed. This function is always executed no matter the call succeeds or fails. |

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev-old/api_ui_tabbar_settabbarstyle

---

my.showActionSheet {#myshowactionsheet}

Last updated: 2022-07-04

Path: miniprogram_gcash

my.showActionSheet

2022-07-04 03:44

Use this API to display the operation menu.

Sample Code

index.json

copy

```
// API-DEMO page/API/action-sheet/action-sheet.json
{
    "defaultTitle": "Operation Menu"
}
```

index.axml

copy

```
<!-- API-DEMO page/API/action-sheet/action-sheet.axml-->
<view class="page">
  <view class="page-description">Action sheet API</view>
  <view class="page-section">
    <view class="page-section-title">my.showActionSheet</view>
    <view class="page-section-demo">
      <button type="primary" onTap="showActionSheet">Show Action
Sheet</button>
    </view>
  </view>
</view>
```

index.js

copy

```
// API-DEMO page/API/action-sheet/action-sheet.js
Page({
  showActionSheet() {
    my.showActionSheet({
      title: 'ActionSheet',
      items: ['Menu 1', 'Menu 2', 'Menu3'],
      cancelButtonText: 'Cancel',
      success: (res) => {
        const btn = res.index === -1 ? 'Cancel' : 'No' + res.index +
'';
        my.alert({
          title: `You clicked ${btn} button`
        });
      },
    });
  },
});
```

Parameters

| | | | | --- | --- | --- | --- | | **Property** | **Type** | **Required** | **Description** | | title | String | No | The title of the menu. | | items | String Array | Yes | The text array of the menu button. | | cancelButtonText | String | No | Text on the cancel button, which is "Cancel" by default.
Note:
This field is invalid for Android and the cancel button will not be displayed. | | destructiveBtnIndex | Number | No | This field specifies the index number of a specific button. The value starts from 0.
Used for buttons to delete or clear data or the ones in similar scenarios. The default color is red. | | badges | Object Array | No | For the array with red option. See badges parameters below for the fields in the array. | | success | Function | No | The callback function for a successful API call. | | fail | Function | No | The callback function for a failed API call. | | complete | Function | No | The callback function used when the API call is completed. This function is always executed no matter the call succeeds or fails. |

`Badges` parameters

| | | | --- | --- | --- | | **Property** | **Type** | **Description** | | index | Number | The index of the option that needs use red marks, starting from 0. | | type | String | Red mark types. Valid values are:
- `none`: no red mark
- `point`: red mark
- `num`: numerical red mark
- `text`: texts in red | | text | String | Customized red mark:
- This field is optional when the value of type is `none`/ `point`/more;
- When the value of type is `num`, but the value of this field has decimals, not greater than 0, or not less than 100, this field is not displayed. |

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev/api_ui_feedback_showactionsheet

---

my.showActionSheet {#myshowactionsheet}

Last updated: 2021-05-09

Path: miniprogram_gcash

my.showActionSheet

2021-05-09 18:43

Use this API to display the operation menu.

Sample Code

index.json

copy

```
// API-DEMO page/API/action-sheet/action-sheet.json
{
    "defaultTitle": "Operation Menu"
}
```

index.axml

copy

```
<!-- API-DEMO page/API/action-sheet/action-sheet.axml-->
<view class="page">
  <view class="page-description">Action sheet API</view>
  <view class="page-section">
    <view class="page-section-title">my.showActionSheet</view>
    <view class="page-section-demo">
      <button type="primary" onTap="showActionSheet">Show Action
Sheet</button>
    </view>
  </view>
</view>
```

index.js

copy

```
// API-DEMO page/API/action-sheet/action-sheet.js
Page({
  showActionSheet() {
    my.showActionSheet({
      title: 'ActionSheet',
      items: ['Menu 1', 'Menu 2', 'Menu3'],
      cancelButtonText: 'Cancel',
      success: (res) => {
        const btn = res.index === -1 ? 'Cancel' : 'No' + res.index +
'';
```

```
        my.alert({
          title: `You clicked ${btn} button`
        });
      },
    });
  },
});
```

Parameters

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Required** | **Description** | | title | String | No | The title of the menu. | | items | String Array | Yes | The text array of the menu button. | | cancelButtonText | String | No | Text on the cancel button, which is "Cancel" by default.
Note:
This field is invalid for Android and the cancel button will not be displayed. | | destructiveBtnIndex | Number | No | This field specifies the index number of a specific button. The value starts from 0.
Used for buttons to delete or clear data or the ones in similar scenarios. The default color is red. | | badges | Object Array | No | For the array with red option. See badges parameters below for the fields in the array. | | success | Function | No | The callback function for a successful API call. | | fail | Function | No | The callback function for a failed API call. | | complete | Function | No | The callback function used when the API call is completed. This function is always executed no matter the call succeeds or fails. |

`Badges` parameters

| | | | | --- | --- | --- | | **Property** | **Type** | **Description** | | index | Number | The index of the option that needs use red marks, starting from 0. | | type | String | Red mark types. Valid values are:
- `none`: no red mark
- `point`: red mark
- `num`: numerical red mark
- `text`: texts in red | | text | String | Customized red mark:
- This field is optional when the value of type is `none`/ `point`/more;
- When the value of type is `num`, but the value of this field has decimals, not greater than 0, or not less than 100, this field is not displayed. |

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev-old/api_ui_feedback_showactionsheet

---

my.showAuthGuide {#myshowauthguide}

Last updated: 2021-05-09

Path: miniprogram_gcash

my.showAuthGuide

2021-05-09 18:43

Pop up dialog for user in form of (image, text, etc.) via the permission guide module. To advice the user to turn on the related permission. The permission guide is used to advice rather than to validate the permission. The call timing is when the service party confirms the required permission is limited. In addition, the permission guide pop-up is subject to fatigue and other factors.

Sample Code

copy

```
// API-DEMO page/API/show-auth-guide/show-auth-guide.json
{
    "defaultTitle": "Permission guide"
}
```

copy

```
<!-- API-DEMO page/API/show-auth-guide/show-auth-guide.axml-->
<view class="page">
  <view class="page-description">Permission guide API</view>
  <view class="page-section">
    <view class="page-section-title">my.showAuthGuide</view>
    <view class="page-section-demo">
      <button type="primary" onTap="showAuthGuide">Permission
guide</button>
    </view>
  </view>
</view>
```

copy

```
// API-DEMO page/API/show-auth-guide/show-auth-guide.js
Page({
  showAuthGuide() {
    my.showAuthGuide({
        authType:'LBS',
        success:(res)=>{
            //When shown is true, it indicates the permission guide
pop-up will be shown; when it is false, it indicates the user has
allowed the permission.
            my.alert({content: 'Call success: '+JSON.stringify(res),
});
        },
        fail:(error)=>{
            my.alert({content: 'Call failure:'+JSON.stringify(error),
});
        },
    });
  },
});
```

Parameters

String type with the following attributes:

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Required** | **Description** | | authType | String | Yes | Identifier of the permission under guide, used to identify the type of the permission (such as LBS). | | success | Function | No | Callback function upon call success. | | fail | Function | No | Callback function upon call failure. | | complete | Function | No | Callback function upon call completion (to be executed upon either call success or failure). |

Supported authType

| | | | --- | --- | | **Permission name** | **Permission code** | | Microphone permission | MICROPHONE | | Addressbook permission | ADDRESSBOOK | | Camera permission | CAMERA | | Photo permission | PHOTO | | Lbs switch | LBS |

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev-old/api_device_permission_showauthguide

---

my.showAuthGuide {#myshowauthguide}

Last updated: 2022-07-03

Path: miniprogram_gcash

my.showAuthGuide

2022-07-03 18:44

Pop up dialog for user in form of (image, text, etc.) via the permission guide module. To advice the user to turn on the related permission. The permission guide is used to advice rather than to validate the permission. The call timing is when the service party confirms the required permission is limited. In addition, the permission guide pop-up is subject to fatigue and other factors.

Sample Code

copy

```
// API-DEMO page/API/show-auth-guide/show-auth-guide.json
{
    "defaultTitle": "Permission guide"
}
```

copy

```
<!-- API-DEMO page/API/show-auth-guide/show-auth-guide.axml-->
<view class="page">
  <view class="page-description">Permission guide API</view>
  <view class="page-section">
    <view class="page-section-title">my.showAuthGuide</view>
    <view class="page-section-demo">
      <button type="primary" onTap="showAuthGuide">Permission
guide</button>
    </view>
  </view>
</view>
```

copy

```
// API-DEMO page/API/show-auth-guide/show-auth-guide.js
Page({
  showAuthGuide() {
    my.showAuthGuide({
        authType:'LBS',
        success:(res)=>{
            //When shown is true, it indicates the permission guide
pop-up will be shown; when it is false, it indicates the user has
allowed the permission.
            my.alert({content: 'Call success: '+JSON.stringify(res),
});
        },
        fail:(error)=>{
            my.alert({content: 'Call failure:'+JSON.stringify(error),
});
        },
    });
  },
});
```

Parameters

String type with the following attributes:

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Required** | **Description** | | authType | String | Yes | Identifier of the permission under guide, used to identify the type of the permission (such as LBS). | | success | Function | No | Callback function upon call success. | | fail | Function | No | Callback function upon call failure. | | complete | Function | No | Callback function upon call completion (to be executed upon either call success or failure). |

Supported authType

Permission name	**Permission code**
Microphone permission	MICROPHONE
Addressbook permission	ADDRESSBOOK
Camera permission	CAMERA
Photo permission	PHOTO
Lbs switch	LBS

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev/API_Device_Permission_showAuthGuide

---

my.showLoading {#myshowloading}

Last updated: 2022-07-03

Path: miniprogram_gcash

my.showLoading

2022-07-03 18:44

Show the loading dialog.

Sample Code

copy

```
my.showLoading({
  content: 'loading...',
  delay: 1000,
});
```

Parameters

Property	**Type**	**Required**	**Description**
content	String	No	Text contents of loading.
delay	Number	No	Displaying delay, in ms, 0 by default If my.hideLoading was called before this time, it is not displayed.
success	Function	No	Callback function upon call success.
fail	Function	No	Callback function upon call failure.
complete	Function	No	Callback function upon call completion (to be executed upon either call success or failure).

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev/api_ui_feedback_showloading

---

my.showLoading {#myshowloading}

Last updated: 2021-05-09

Path: miniprogram_gcash

my.showLoading

2021-05-09 18:43

Show the loading dialog.

Sample Code

copy

```
my.showLoading({
  content: 'loading...',
  delay: 1000,
});
```

Parameters

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Required** | **Description** | | content | String | No | Text contents of loading. | | delay | Number | No | Displaying delay, in ms, 0 by default If my.hideLoading was called before this time, it is not displayed. | | success | Function | No | Callback function upon call success. | | fail | Function | No | Callback function upon call failure. | | complete | Function | No | Callback function upon call completion (to be executed upon either call success or failure). |

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev-old/api_ui_feedback_showloading

---

my.showNavigationBarLoading {#myshownavigationbarloading}

Last updated: 2021-05-10

Path: miniprogram_gcash

my.showNavigationBarLoading

2021-05-10 03:43

Show the navigation bar loading.

Sample Code

copy

```
my.showNavigationBarLoading();
```

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev-old/api_ui_navigationbar_shownavigationbarloading

---

my.showNavigationBarLoading {#myshownavigationbarloading}

Last updated: 2022-07-03

Path: miniprogram_gcash

my.showNavigationBarLoading

2022-07-03 18:44

Show the navigation bar loading.

Sample Code

copy

```
my.showNavigationBarLoading();
```

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev/API_UI_NavigationBar_showNavigationBarLoading

---

my.showSharePanel {#myshowsharepanel}

Last updated: 2022-07-03

Path: miniprogram_gcash

my.showSharePanel

2022-07-03 18:44

Use this API to trigger the sharing.

Note:

When using this API to trigger the sharing, pass the value `code` to the request parameter `from` in page.onShareAppMessage.

Sample code

copy

```
// .js
my.showSharePanel();
```

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev/api_share_showsharepanel

---

my.showSharePanel {#myshowsharepanel}

Last updated: 2021-05-09

Path: miniprogram_gcash

my.showSharePanel

2021-05-09 18:43

Use this API to trigger the sharing.

Note:

When using this API to trigger the sharing, pass the value `code` to the request parameter `from` in page.onShareAppMessage.

Sample code

copy

```
// .js
my.showSharePanel();
```

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev-old/api_share_showsharepanel

---

my.showToast {#myshowtoast}

Last updated: 2021-05-09

Path: miniprogram_gcash

my.showToast

2021-05-09 18:43

Show the toast dialog, which disappears with the specified duration.

Sample Code

copy

```
my.showToast({
  type: 'success',
  content: 'Success',
  duration: 3000,
  success: () => {
    my.alert({
      title: 'toast is missing',
    });
  },
});
```

Parameters

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Required** | **Description** | | content | String | No | Text content. | | type | String | No | Toast type, showing the related icon, none by default, Supporting `success`/ `fail`/ `exception`/ `none` Here. If it is `exception`, `content` is mandatory. | | duration | Number | No | Displaying duration, in ms, 2000 by default. | | success | Function | No | Callback function upon call success. | | fail | Function | No | Callback function upon call failure. | | complete | Function | No | Callback function upon call completion (to be executed upon either call success or failure). |

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev-old/api_ui_feedback_showtoast

---

my.showToast {#myshowtoast}

Last updated: 2022-07-03

Path: miniprogram_gcash

my.showToast

2022-07-03 18:44

Show the toast dialog, which disappears with the specified duration.

Sample Code

copy

```
my.showToast({
  type: 'success',
  content: 'Success',
  duration: 3000,
  success: () => {
    my.alert({
      title: 'toast is missing',
    });
  },
});
```

Parameters

Property	**Type**	**Required**	**Description**
content	String	No	Text content.
type	String	No	Toast type, showing the related icon, none by default, Supporting `success`/ `fail`/ `exception`/ `none` Here. If it is `exception`, `content` is mandatory.
duration	Number	No	Displaying duration, in ms, 2000 by default.
success	Function	No	Callback function upon call success.
fail	Function	No	Callback function upon call failure.
complete	Function	No	Callback function upon call completion (to be executed upon either call success or failure).

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev/API_UI_Feedback_showToast

---

my.signContract {#mysigncontract}

Last updated: 2022-07-03

Path: miniprogram_gcash

my.signContract

2022-07-03 18:44

Use this API to redirect the user to the authorization page. After the user completes the authorization, the app will return the authorization code which can be used to obtain the access token for the agreement payment.

Note:

Please make sure you use the Appx with 1.24.6 or higher versions in order to use this API.

Below is a diagram that illustrates how the interaction works:

The merchant Mini Program server can call authorization consult API in step 2 to get the sign content with an authorization URL. Then the Mini Program will call the `my.signContract` JSAPI to invoke the authorization process. After the user completes the authorization, the Mini Program sends JSAPI result to its server so that the server can call apply token API to get accessToken.

Sample code

copy

```
my.signContract({
  signStr: 'https://openauth.xxx.com/authentication.htm?
authId=FBF16F91-28FB-47EC-B9BE-27B285C23CD3',
  success: (res) => {
    my.alert({
    content: JSON.stringify(res),
  });
  },
  fail: (res) => {
    my.alert({
    content: JSON.stringify(res),
  });
  }
});
```

Parameters

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Required** | **Description** | | signStr | String | Yes | This parameter is the authorization string returned by the app to further the authorization process. | | success | Function | No | Callback function upon call success. | |

fail | Function | No | Callback function upon call failure. | | complete | Function | No | Callback function upon call completion (to be executed upon either call success or failure). |

Success Callback Function

| | | | | --- | --- | --- | | **Property** | **Type** | **Description** | | authState | String | The authorization status. It is generated in Mini Program server and sent to app server. The maximum length is 256. Refer here for details. | | authCode | String | The authorization code assigned by app which can be used to obtain the access token for the agreement payment. The maximum length is 32. |

An example of a successfully returned message is as follows:

copy

```
{
    "authState":"663A8FA9-D836-48EE-8AA1-1FF682989DC7",
    "authCode":"663A8FA9D83648EE8AA11FF682989DC7"
}
```

Fail Callback Function

| | | | | --- | --- | --- | | **Property** | **Type** | **Description** | | error | String | The error code for the failure. | | errMessage | String | The error message. |

Error Code

When error happens, the fail callback function will be executed. The error code can refer to the following table.

| | | | --- | --- | | **Error Code** | **Description** | | 6001 | User cancels the sign process. | | 6002 | The sign fails because of network error. | | 7001 | The result of the sign is unknown, it may be successful. | | 7002 | The sign fails. |

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev/api_openapi_signcontract

---

my.signContract {#mysigncontract}

Last updated: 2021-05-09

Path: miniprogram_gcash

my.signContract

2021-05-09 18:43

Use this API to redirect the user to the authorization page. After the user completes the authorization, the app will return the authorization code which can be used to obtain the access token for the agreement payment.

Note:

Please make sure you use the Appx with 1.24.6 or higher versions in order to use this API.

Below is a diagram that illustrates how the interaction works:

The merchant Mini Program server can call authorization consult API in step 2 to get the sign content with an authorization URL. Then the Mini Program will call the `my.signContract` JSAPI to invoke the authorization process. After the user completes the authorization, the Mini Program sends JSAPI result to its server so that the server can call apply token API to get accessToken.

Sample code

copy

```
my.signContract({
  signStr: 'https://openauth.xxx.com/authentication.htm?
authId=FBF16F91-28FB-47EC-B9BE-27B285C23CD3',
  success: (res) => {
    my.alert({
    content: JSON.stringify(res),
  });
  },
  fail: (res) => {
    my.alert({
    content: JSON.stringify(res),
  });
  }
});
```

Parameters

Property	**Type**	**Required**	**Description**
signStr	String	Yes	This parameter is the authorization string returned by the app to further the authorization process.
success	Function	No	Callback function upon call success.
fail	Function	No	Callback function upon call failure.
complete	Function	No	Callback function upon call completion (to be executed upon either call success or failure).

Success Callback Function

| | | | | --- | --- | --- | | **Property** | **Type** | **Description** | | authState | String | The authorization status. It is generated in Mini Program server and sent to app server. The maximum length is 256. Refer here for details. | | authCode | String | The authorization code assigned by app which can be used to obtain the access token for the agreement payment. The maximum length is 32. |

An example of a successfully returned message is as follows:

copy

```
{
    "authState":"663A8FA9-D836-48EE-8AA1-1FF682989DC7",
    "authCode":"663A8FA9D83648EE8AA11FF682989DC7"
}
```

Fail Callback Function

| | | | | --- | --- | --- | | **Property** | **Type** | **Description** | | error | String | The error code for the failure. | | errMessage | String | The error message. |

Error Code

When error happens, the fail callback function will be executed. The error code can refer to the following table.

| | | | --- | --- | | **Error Code** | **Description** | | 6001 | User cancels the sign process. | | 6002 | The sign fails because of network error. | | 7001 | The result of the sign is unknown, it may be successful. | | 7002 | The sign fails. |

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev-old/api_openapi_signcontract

---

my.startBluetoothDevicesDiscovery {#mystartbluetoothdevicesdiscovery}

Last updated: 2022-07-03

Path: miniprogram_gcash

my.startBluetoothDevicesDiscovery

2022-07-03 18:44

Use this API to start discovering bluetooth devices. You can find the results in the API my.onBluetoothDeviceFound.

Note:

To avoid consuming too many system resources, stop the search by calling the API **my.stopBluetoothDevicesDiscovery** after you find and connect to the device.

Currently simulation in IDE is not supported. Please debug in the production environment.

Code Sample

copy

```
/* .acss */
.help-info {
  padding:10px;
  color:#000000;
}
.help-title {
  padding:10px;
  color:#FC0D1B;
}
```

copy

```
// .json
{
    "defaultTitle": "Bluetooth"
}
```

copy

```
<!-- .axml-->
<view class="page">
  <view class="page-description">Bluetooth API</view>
  <view class="page-section">
    <view class="page-section-title">The Bluetooth state</view>
    <view class="page-section-demo">
       <button type="primary" onTap="openBluetoothAdapter">Initialize
Bluetooth</button>
       <button type="primary" onTap="closeBluetoothAdapter">Close
Bluetooth</button>
       <button type="primary" onTap="getBluetoothAdapterState">Obtain
Bluetooth state</button>
    </view>
    <view class="page-section-title">Scan the Bluetooth device</view>
    <view class="page-section-demo">
       <button type="primary"
onTap="startBluetoothDevicesDiscovery">Start searching</button>
       <button type="primary" onTap="getBluetoothDevices">All devices
```

```
found</button>
       <button type="primary" onTap="getConnectedBluetoothDevices">All
devices connected</button>
       <button type="primary"
onTap="stopBluetoothDevicesDiscovery">Stop searching</button>
    </view>
    <view class="page-section-title">Connect the device</view>
    <view class="page-section-demo">
       <input class="input" onInput="bindKeyInput" type="{{text}}"
placeholder="Enter the device ID of the device to connect"></input>
       <button type="primary" onTap="connectBLEDevice">Connect the
device</button>
       <button type="primary" onTap="getBLEDeviceServices">Obtain
device services</button>
       <button type="primary"
onTap="getBLEDeviceCharacteristics">Obtain read and write
characteristics</button>
       <button type="primary" onTap="disconnectBLEDevice">Disconnect
the device</button>
    </view>
     <view class="page-section-title">Read and write data</view>
     <view class="page-section-demo">
       <button type="primary"
onTap="notifyBLECharacteristicValueChange">Listens to the
characteristic data change</button>
       <button type="primary" onTap="readBLECharacteristicValue">Read
data</button>
       <button type="primary"
onTap="writeBLECharacteristicValue">Write data</button>
       <button type="primary"
onTap="offBLECharacteristicValueChange">Un-listens to characteristic
value</button>
    </view>
     <view class="page-section-title">Other events</view>
     <view class="page-section-demo">
       <button type="primary"
onTap="bluetoothAdapterStateChange">Changes of the Bluetooth
state</button>
       <button type="primary"
onTap="offBluetoothAdapterStateChange">Un-listens to Bluetooth
state</button>
       <button type="primary"
onTap="BLEConnectionStateChanged">Changes of Bluetooth connection
state</button>
       <button type="primary" onTap="offBLEConnectionStateChanged">Un-
listens to Bluetooth connection state</button>

    </view>
  </view>
</view>
```

copy

```
// .js
Page({
  data: {
    devid: '0D9C82AD-1CC0-414D-9526-119E08D28124',
    serid: 'FEE7',
    notifyId: '36F6',
    writeId: '36F5',
    charid: '',
    alldev: [{ deviceId: '' }],
  },

  //Obtain the Bluetooth state
  openBluetoothAdapter() {
    my.openBluetoothAdapter({
      success: res => {
        if (!res.isSupportBLE) {
          my.alert({ content: 'Sorry, your mobile Bluetooth is
unavailable temporarily' });
          return;
        }
        my.alert({ content: 'Succeeded to initialize!' });
      },
      fail: error => {
        my.alert({ content: JSON.stringify(error) });
      },
    });
  },
  closeBluetoothAdapter() {
    my.closeBluetoothAdapter({
      success: () => {
        my.alert({ content: 'Bluetooth closed!' });
      },
      fail: error => {
        my.alert({ content: JSON.stringify(error) });
      },
    });
  },
  getBluetoothAdapterState() {
    my.getBluetoothAdapterState({
      success: res => {
        if (!res.available) {
          my.alert({ content: 'Sorry, your mobile Bluetooth is
unavailable temporarily' });
          return;
        }
        my.alert({ content: JSON.stringify(res) });
      },
      fail: error => {
        my.alert({ content: JSON.stringify(error) });
      },
    });
  },

  //Scan the Bluetooth device
  startBluetoothDevicesDiscovery() {
    my.startBluetoothDevicesDiscovery({
      allowDuplicatesKey: false,
      success: () => {
        my.onBluetoothDeviceFound({
          success: res => {
            // my.alert({content:'Listens to new device'+JSON.stringify(res)});
            var deviceArray = res.devices;
            for (var i = deviceArray.length - 1; i >= 0; i--) {
              var deviceObj = deviceArray[i];
              //Pair the target device with the device name or broadcast data, and then record the device ID for later use.
              if (deviceObj.name == this.data.name) {
                my.alert({ content: 'Target device is found' });
                my.offBluetoothDeviceFound();
                this.setData({
                  deviceId: deviceObj.deviceId,
                });
                break;
              }
            }
          },
          fail: error => {
            my.alert({ content: 'Failed to listen to new device' + JSON.stringify(error) });
          },
        });
      },
      fail: error => {
        my.alert({ content: 'Failed to start scanning' + JSON.stringify(error) });
      },
    });
  },

  //Stop scanning
  stopBluetoothDevicesDiscovery() {
    my.stopBluetoothDevicesDiscovery({
      success: res => {
        my.offBluetoothDeviceFound();
        my.alert({ content: 'Succeeded!' });
      },
      fail: error => {
        my.alert({ content: JSON.stringify(error) });
      },
    });
  },

  //Obtain the connected device
  getConnectedBluetoothDevices() {
    my.getConnectedBluetoothDevices({
      success: res => {
        if (res.devices.length === 0) {
          my.alert({ content: 'No connecting devices!' });
          return;
        }
        my.alert({ content: JSON.stringify(res) });
        devid = res.devices[0].deviceId;
      },
      fail: error => {
        my.alert({ content: JSON.stringify(error) });
      },
    });
  },

  //Obtain all searched devices
  getBluetoothDevices() {
    my.getBluetoothDevices({
      success: res => {
        my.alert({ content: JSON.stringify(res) });
      },
      fail: error => {
        my.alert({ content: JSON.stringify(error) });
      },
    });
  },
  bindKeyInput(e) {
    this.setData({
      devid: e.detail.value,
    });
  },

  //Connect the device
  connectBLEDevice() {
    my.connectBLEDevice({
      deviceId: this.data.devid,
      success: res => {
        my.alert({ content: 'Succeeded to connect!' });
      },
      fail: error => {
        my.alert({ content: JSON.stringify(error) });
      },
    });
  },

  //Disconnect the device
  disconnectBLEDevice() {
    my.disconnectBLEDevice({
      deviceId: this.data.devid,
      success: () => {
        my.alert({ content: 'Succeeded to disconnect!' });
      },
      fail: error => {
        my.alert({ content: JSON.stringify(error) });
      },
    });
  },

  //Obtain the services of the connected device
  getBLEDeviceServices() {
    my.getConnectedBluetoothDevices({
      success: res => {
        if (res.devices.length === 0) {
          my.alert({ content: 'No connected devices' });
          return;
        }
        my.getBLEDeviceServices({
          deviceId: this.data.devid,
          success: res => {
            my.alert({ content: JSON.stringify(res) });
            this.setData({
              serid: res.services[0].serviceId,
            });
          },
          fail: error => {
            my.alert({ content: JSON.stringify(error) });
          },
        });
      },
    });
  },

  //Obtain the char ID of the connected device, read and write
characteristics are respectively screened out.
  getBLEDeviceCharacteristics() {
    my.getConnectedBluetoothDevices({
      success: res => {
        if (res.devices.length === 0) {
          my.alert({ content: 'No connected devices' });
          return;
        }
        this.setData({
          devid: res.devices[0].deviceId,
        });
        my.getBLEDeviceCharacteristics({
          deviceId: this.data.devid,
          serviceId: this.data.serid,
          success: res => {
            my.alert({ content: JSON.stringify(res) });
            //See the related document for more information of the
properties of the characteristics. Pair the characteristics according
to the properties and record the value for later use.
            this.setData({
              charid: res.characteristics[0].characteristicId,
            });
          },
          fail: error => {
            my.alert({ content: JSON.stringify(error) });
          },
        });
      },
    });
  },

  //Read and write data
  readBLECharacteristicValue() {
    my.getConnectedBluetoothDevices({
      success: res => {
        if (res.devices.length === 0) {
          my.alert({ content: 'No connected devices' });
          return;
        }
        this.setData({
          devid: res.devices[0].deviceId,
        });
        my.readBLECharacteristicValue({
          deviceId: this.data.devid,
          serviceId: this.data.serid,
          characteristicId: this.data.notifyId,
          //1  Android reading service
          // serviceId:'0000180d-0000-1000-8000-00805f9b34fb',
          // characteristicId:'00002a38-0000-1000-8000-00805f9b34fb',
          success: res => {
            my.alert({ content: JSON.stringify(res) });
          },
          fail: error => {
            my.alert({ content: 'Failed to read' +
JSON.stringify(error) });
          },
        });
      },
    });
  },
  writeBLECharacteristicValue() {
    my.getConnectedBluetoothDevices({
      success: res => {
        if (res.devices.length === 0) {
          my.alert({ content: 'No connected devices' });
          return;
        }
        this.setData({
          devid: res.devices[0].deviceId,
        });
        my.writeBLECharacteristicValue({
          deviceId: this.data.devid,
          serviceId: this.data.serid,
          characteristicId: this.data.charid,
          //Android writing service
          //serviceId:'0000180d-0000-1000-8000-00805f9b34fb',
          //characteristicId:'00002a39-0000-1000-8000-00805f9b34fb',
          value: 'ABCD',
          success: res => {
            my.alert({ content: 'Succeeded to write data!' });
          },
          fail: error => {
            my.alert({ content: JSON.stringify(error) });
          },
        });
      },
    });
  },
  notifyBLECharacteristicValueChange() {
    my.getConnectedBluetoothDevices({
      success: res => {
        if (res.devices.length === 0) {
          my.alert({ content: 'No connected devices' });
          return;
        }
        this.setData({
          devid: res.devices[0].deviceId,
        });
        my.notifyBLECharacteristicValueChange({
          state: true,
          deviceId: this.data.devid,
          serviceId: this.data.serid,
          characteristicId: this.data.notifyId,
          success: () => {
            //Listens to characteristic change events
            my.onBLECharacteristicValueChange({
              success: res => {
                //  my.alert({content: 'Changes of
characteristics  '+JSON.stringify(res)});
                my.alert({ content: 'Obtain the response data = ' +
res.value });
              },
            });
            my.alert({ content: 'Succeeded to listen' });
          },
          fail: error => {
            my.alert({ content: 'Failed to listen' +
JSON.stringify(error) });
          },
        });
      },
    });
  },
  offBLECharacteristicValueChange() {
    my.offBLECharacteristicValueChange();
  },

  //Other events
  bluetoothAdapterStateChange() {

my.onBluetoothAdapterStateChange(this.getBind('onBluetoothAdapterState(
  },
  onBluetoothAdapterStateChange() {
    if (res.error) {
      my.alert({ content: JSON.stringify(error) });
    } else {
      my.alert({ content: 'Changes of the Bluetooth state  ' +
JSON.stringify(res) });
    }
  },
  offBluetoothAdapterStateChange() {

my.offBluetoothAdapterStateChange(this.getBind('onBluetoothAdapterState
  },
  getBind(name) {
    if (!this[`bind${name}`]) {
      this[`bind${name}`] = this[name].bind(this);
    }
    return this[`bind${name}`];
  },
  BLEConnectionStateChanged() {

my.onBLEConnectionStateChanged(this.getBind('onBLEConnectionStateChange
  },
  onBLEConnectionStateChanged(res) {
    if (res.error) {
      my.alert({ content: JSON.stringify(error) });
    } else {
      my.alert({ content: 'Changes of connection state  ' +
JSON.stringify(res) });
    }
  },
  offBLEConnectionStateChanged() {

my.offBLEConnectionStateChanged(this.getBind('onBLEConnectionStateChang
  },
  onUnload() {
    this.offBLEConnectionStateChanged();
    this.offBLECharacteristicValueChange();
    this.offBluetoothAdapterStateChange();
    this.closeBluetoothAdapter();
  },
});
```

Parameters

he input parameters are displayed in the following table:

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Required** | **Description** | | services | Array | No | UUID list of the service in the bluetooth device. | | allowDuplicatesKey | Boolean | No | Determines whether to allow for duplicate keys. If yes, the API my.onBluetoothDeviceFound will report for this device, with different RSSI values. | | interval | Integer | No | By default, the interval is 0, which means once a new device is found, the device will be reported; otherwise, the reporting frequency is the interval value. | | success | Function | No | The callback function for a successful API call. | | fail | Function | No | The callback function for a failed API call. | | complete | Function | No | The callback function for a completed API call (regardless of whether the call is successful or not). |

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev/api_device_bluetooth_bluetooth_startbluetoothdevicesdiscovery

---

my.startBluetoothDevicesDiscovery {#mystartbluetoothdevicesdiscovery}

Last updated: 2021-05-09

Path: miniprogram_gcash

my.startBluetoothDevicesDiscovery

2021-05-09 18:43

Use this API to start discovering bluetooth devices. You can find the results in the API my.onBluetoothDeviceFound.

Note:

To avoid consuming too many system resources, stop the search by calling the API **my.stopBluetoothDevicesDiscovery** after you find and connect to the device.

Currently simulation in IDE is not supported. Please debug in the production environment.

Code Sample

copy

```
/* .acss */
.help-info {
  padding:10px;
  color:#000000;
}
.help-title {
  padding:10px;
  color:#FC0D1B;
}
```

copy

```
// .json
{
    "defaultTitle": "Bluetooth"
}
```

copy

```
<!-- .axml-->
<view class="page">
  <view class="page-description">Bluetooth API</view>
  <view class="page-section">
    <view class="page-section-title">The Bluetooth state</view>
    <view class="page-section-demo">
       <button type="primary" onTap="openBluetoothAdapter">Initialize
Bluetooth</button>
       <button type="primary" onTap="closeBluetoothAdapter">Close
Bluetooth</button>
       <button type="primary" onTap="getBluetoothAdapterState">Obtain
Bluetooth state</button>
    </view>
    <view class="page-section-title">Scan the Bluetooth device</view>
    <view class="page-section-demo">
       <button type="primary"
onTap="startBluetoothDevicesDiscovery">Start searching</button>
       <button type="primary" onTap="getBluetoothDevices">All devices
found</button>
       <button type="primary" onTap="getConnectedBluetoothDevices">All
devices connected</button>
       <button type="primary"
onTap="stopBluetoothDevicesDiscovery">Stop searching</button>
    </view>
    <view class="page-section-title">Connect the device</view>
    <view class="page-section-demo">
       <input class="input" onInput="bindKeyInput" type="{{text}}"
placeholder="Enter the device ID of the device to connect"></input>
       <button type="primary" onTap="connectBLEDevice">Connect the
```

```
device</button>
       <button type="primary" onTap="getBLEDeviceServices">Obtain
device services</button>
       <button type="primary"
onTap="getBLEDeviceCharacteristics">Obtain read and write
characteristics</button>
       <button type="primary" onTap="disconnectBLEDevice">Disconnect
the device</button>
    </view>
     <view class="page-section-title">Read and write data</view>
     <view class="page-section-demo">
       <button type="primary"
onTap="notifyBLECharacteristicValueChange">Listens to the
characteristic data change</button>
       <button type="primary" onTap="readBLECharacteristicValue">Read
data</button>
       <button type="primary"
onTap="writeBLECharacteristicValue">Write data</button>
       <button type="primary"
onTap="offBLECharacteristicValueChange">Un-listens to characteristic
value</button>
    </view>
     <view class="page-section-title">Other events</view>
     <view class="page-section-demo">
       <button type="primary"
onTap="bluetoothAdapterStateChange">Changes of the Bluetooth
state</button>
       <button type="primary"
onTap="offBluetoothAdapterStateChange">Un-listens to Bluetooth
state</button>
       <button type="primary"
onTap="BLEConnectionStateChanged">Changes of Bluetooth connection
state</button>
       <button type="primary" onTap="offBLEConnectionStateChanged">Un-
listens to Bluetooth connection state</button>

    </view>
  </view>
</view>
```

copy

```
// .js
Page({
  data: {
    devid: '0D9C82AD-1CC0-414D-9526-119E08D28124',
    serid: 'FEE7',
    notifyId: '36F6',
    writeId: '36F5',
    charid: '',
    alldev: [{ deviceId: '' }],
  },
  //Obtain the Bluetooth state
  openBluetoothAdapter() {
    my.openBluetoothAdapter({
      success: res => {
        if (!res.isSupportBLE) {
          my.alert({ content: 'Sorry, your mobile Bluetooth is
unavailable temporarily' });
          return;
        }
        my.alert({ content: 'Succeeded to initialize!' });
      },
      fail: error => {
        my.alert({ content: JSON.stringify(error) });
      },
    });
  },
  closeBluetoothAdapter() {
    my.closeBluetoothAdapter({
      success: () => {
        my.alert({ content: 'Bluetooth closed!' });
      },
      fail: error => {
        my.alert({ content: JSON.stringify(error) });
      },
    });
  },
  getBluetoothAdapterState() {
    my.getBluetoothAdapterState({
      success: res => {
        if (!res.available) {
          my.alert({ content: 'Sorry, your mobile Bluetooth is
unavailable temporarily' });
          return;
        }
        my.alert({ content: JSON.stringify(res) });
      },
      fail: error => {
        my.alert({ content: JSON.stringify(error) });
      },
    });
  },

  //Scan the Bluetooth device
  startBluetoothDevicesDiscovery() {
    my.startBluetoothDevicesDiscovery({
      allowDuplicatesKey: false,
      success: () => {
        my.onBluetoothDeviceFound({
          success: res => {
            // my.alert({content:'Listens to new
device'+JSON.stringify(res)});
            var deviceArray = res.devices;
            for (var i = deviceArray.length - 1; i >= 0; i--) {
              var deviceObj = deviceArray[i];
              //Pair the target device with the device name or
broadcast data, and then record the device ID for later use.
              if (deviceObj.name == this.data.name) {
                my.alert({ content: 'Target device is found' });
                my.offBluetoothDeviceFound();
                this.setData({
                  deviceId: deviceObj.deviceId,
                });
                break;
              }
            }
          },
          fail: error => {
            my.alert({ content: 'Failed to listen to new device' +
JSON.stringify(error) });
          },
        });
      },
      fail: error => {
        my.alert({ content: 'Failed to start scanning' +
JSON.stringify(error) });
      },
    });
  },

  //Stop scanning
  stopBluetoothDevicesDiscovery() {
    my.stopBluetoothDevicesDiscovery({
      success: res => {
        my.offBluetoothDeviceFound();
        my.alert({ content: 'Succeeded!' });
      },
      fail: error => {
        my.alert({ content: JSON.stringify(error) });
      },
    });
  },

  //Obtain the connected device
  getConnectedBluetoothDevices() {
    my.getConnectedBluetoothDevices({
      success: res => {
        if (res.devices.length === 0) {
          my.alert({ content: 'No connecting devices!' });
          return;
        }
        my.alert({ content: JSON.stringify(res) });
        devid = res.devices[0].deviceId;
      },
      fail: error => {
        my.alert({ content: JSON.stringify(error) });
      },
    });
  },

  //Obtain all searched devices
  getBluetoothDevices() {
    my.getBluetoothDevices({
      success: res => {
        my.alert({ content: JSON.stringify(res) });
      },
      fail: error => {
        my.alert({ content: JSON.stringify(error) });
      },
    });
  },
  bindKeyInput(e) {
    this.setData({
      devid: e.detail.value,
    });
  },

  //Connect the device
  connectBLEDevice() {
    my.connectBLEDevice({
      deviceId: this.data.devid,
      success: res => {
        my.alert({ content: 'Succeeded to connect!' });
      },
      fail: error => {
        my.alert({ content: JSON.stringify(error) });
      },
    });
  },

  //Disconnect the device
  disconnectBLEDevice() {
    my.disconnectBLEDevice({
      deviceId: this.data.devid,
      success: () => {
        my.alert({ content: 'Succeeded to disconnect!' });
      },
      fail: error => {
        my.alert({ content: JSON.stringify(error) });
      },
    });
  },
  //Obtain the services of the connected device
  getBLEDeviceServices() {
    my.getConnectedBluetoothDevices({
      success: res => {
        if (res.devices.length === 0) {
          my.alert({ content: 'No connected devices' });
          return;
        }
        my.getBLEDeviceServices({
          deviceId: this.data.devid,
          success: res => {
            my.alert({ content: JSON.stringify(res) });
            this.setData({
              serid: res.services[0].serviceId,
            });
          },
          fail: error => {
            my.alert({ content: JSON.stringify(error) });
          },
        });
      },
    });
  },

  //Obtain the char ID of the connected device, read and write
characteristics are respectively screened out.
  getBLEDeviceCharacteristics() {
    my.getConnectedBluetoothDevices({
      success: res => {
        if (res.devices.length === 0) {
          my.alert({ content: 'No connected devices' });
          return;
        }
        this.setData({
          devid: res.devices[0].deviceId,
        });
        my.getBLEDeviceCharacteristics({
          deviceId: this.data.devid,
          serviceId: this.data.serid,
          success: res => {
            my.alert({ content: JSON.stringify(res) });
            //See the related document for more information of the
properties of the characteristics. Pair the characteristics according
to the properties and record the value for later use.
            this.setData({
              charid: res.characteristics[0].characteristicId,
            });
          },
          fail: error => {
            my.alert({ content: JSON.stringify(error) });
          },
        });
      },
    });
  },

  //Read and write data
  readBLECharacteristicValue() {
    my.getConnectedBluetoothDevices({
      success: res => {
        if (res.devices.length === 0) {
          my.alert({ content: 'No connected devices' });
          return;
        }
        this.setData({
          devid: res.devices[0].deviceId,
        });
        my.readBLECharacteristicValue({
          deviceId: this.data.devid,
          serviceId: this.data.serid,
          characteristicId: this.data.notifyId,
          //1  Android reading service
          // serviceId:'0000180d-0000-1000-8000-00805f9b34fb',
          // characteristicId:'00002a38-0000-1000-8000-00805f9b34fb',
          success: res => {
            my.alert({ content: JSON.stringify(res) });
          },
          fail: error => {
            my.alert({ content: 'Failed to read' +
JSON.stringify(error) });
          },
        });
      },
    });
  },
  writeBLECharacteristicValue() {
    my.getConnectedBluetoothDevices({
      success: res => {
        if (res.devices.length === 0) {
          my.alert({ content: 'No connected devices' });
          return;
        }
        this.setData({
          devid: res.devices[0].deviceId,
        });
        my.writeBLECharacteristicValue({
          deviceId: this.data.devid,
          serviceId: this.data.serid,
          characteristicId: this.data.charid,
          //Android writing service
          //serviceId:'0000180d-0000-1000-8000-00805f9b34fb',
          //characteristicId:'00002a39-0000-1000-8000-00805f9b34fb',
          value: 'ABCD',
          success: res => {
            my.alert({ content: 'Succeeded to write data!' });
          },
          fail: error => {
            my.alert({ content: JSON.stringify(error) });
          },
        });
      },
    });
  },
  notifyBLECharacteristicValueChange() {
    my.getConnectedBluetoothDevices({
      success: res => {
        if (res.devices.length === 0) {
          my.alert({ content: 'No connected devices' });
          return;
        }
        this.setData({
          devid: res.devices[0].deviceId,
        });
        my.notifyBLECharacteristicValueChange({
          state: true,
          deviceId: this.data.devid,
          serviceId: this.data.serid,
          characteristicId: this.data.notifyId,
          success: () => {
            //Listens to characteristic change events
            my.onBLECharacteristicValueChange({
              success: res => {
                //  my.alert({content: 'Changes of
characteristics  '+JSON.stringify(res)});
                my.alert({ content: 'Obtain the response data = ' +
res.value });
              },
            });
            my.alert({ content: 'Succeeded to listen' });
          },
          fail: error => {
            my.alert({ content: 'Failed to listen' +
JSON.stringify(error) });
          },
        });
      },
    });
  },
  offBLECharacteristicValueChange() {
    my.offBLECharacteristicValueChange();
  },

  //Other events
  bluetoothAdapterStateChange() {

my.onBluetoothAdapterStateChange(this.getBind('onBluetoothAdapterStateC
  },
  onBluetoothAdapterStateChange() {
    if (res.error) {
      my.alert({ content: JSON.stringify(error) });
    } else {
      my.alert({ content: 'Changes of the Bluetooth state  ' +
JSON.stringify(res) });
    }
  },
  offBluetoothAdapterStateChange() {

my.offBluetoothAdapterStateChange(this.getBind('onBluetoothAdapterState
  },
  getBind(name) {
    if (!this[`bind${name}`]) {
      this[`bind${name}`] = this[name].bind(this);
    }
    return this[`bind${name}`];
  },
  BLEConnectionStateChanged() {

my.onBLEConnectionStateChanged(this.getBind('onBLEConnectionStateChange
  },
  onBLEConnectionStateChanged(res) {
    if (res.error) {
      my.alert({ content: JSON.stringify(error) });
    } else {
      my.alert({ content: 'Changes of connection state  ' +
JSON.stringify(res) });
    }
  },
  offBLEConnectionStateChanged() {

my.offBLEConnectionStateChanged(this.getBind('onBLEConnectionStateChang
  },
  onUnload() {
    this.offBLEConnectionStateChanged();
    this.offBLECharacteristicValueChange();
    this.offBluetoothAdapterStateChange();
    this.closeBluetoothAdapter();
  },
});
```

Parameters

he input parameters are displayed in the following table:

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Required** | **Description** | | services | Array | No | UUID list of the service in the bluetooth device. | | allowDuplicatesKey | Boolean | No | Determines whether to allow for duplicate keys. If yes, the API my.onBluetoothDeviceFound will report for this device, with different RSSI values. | | interval | Integer | No | By default, the interval is 0, which means once a new device is found, the device will be reported; otherwise, the reporting frequency is the interval value. | | success | Function | No | The callback function for a successful API call. | | fail | Function | No | The callback function for a failed API call. | | complete | Function | No | The callback function for a completed API call (regardless of whether the call is successful or not). |

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev-old/api_device_bluetooth_bluetooth_startbluetoothdevicesdiscovery

---

my.startPullDownRefresh {#mystartpulldownrefresh}

Last updated: 2021-05-09

Path: miniprogram_gcash

my.startPullDownRefresh

2021-05-09 18:43

Start the pull-to-refresh function. The pull-to-refresh animation is triggered upon the code execution, which is consistent with the manual pull-to-refresh effect.

Sample Code

copy

```
my.startPullDownRefresh()
```

Parameters

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Required** | **Description** | | success | Function | No | The callback function for a successful API call. | | fail | Function | No | The callback function for a failed API call. | | complete | Function | No | The callback function used when the API call is completed. This function is always executed no matter the call succeeds or fails. |

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev-old/api_ui_pulldown_startpulldownrefresh

---

my.startPullDownRefresh {#mystartpulldownrefresh}

Last updated: 2022-07-03

Path: miniprogram_gcash

my.startPullDownRefresh

2022-07-03 18:44

Start the pull-to-refresh function. The pull-to-refresh animation is triggered upon the code execution, which is consistent with the manual pull-to-refresh effect.

Sample Code

copy

```
my.startPullDownRefresh()
```

Parameters

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Required** | **Description** | | success | Function | No | The callback function for a successful API call. | | fail | Function | No | The callback function for a failed API call. | | complete | Function | No | The callback function used when the API call is completed. This function is always executed no matter the call succeeds or fails. |

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev/api_ui_pulldown_startpulldownrefresh

---

my.stopBluetoothDevicesDiscovery {#mystopbluetoothdevicesdiscovery}

Last updated: 2021-05-09

Path: miniprogram_gcash

my.stopBluetoothDevicesDiscovery

2021-05-09 18:43

Use this API to stop discovering bluetooth devices.

Note:

Currently simulation in IDE is not supported. Please debug in the production environment.

Code Sample

copy

```
/* .acss */
.help-info {
  padding:10px;
  color:#000000;
}
.help-title {
  padding:10px;
  color:#FC0D1B;
}
```

copy

```
// .json
{
    "defaultTitle": "Bluetooth"
}
```

copy

```
<!-- .axml-->
<view class="page">
  <view class="page-description">Bluetooth API</view>
  <view class="page-section">
    <view class="page-section-title">The Bluetooth state</view>
    <view class="page-section-demo">
       <button type="primary" onTap="openBluetoothAdapter">Initialize
Bluetooth</button>
       <button type="primary" onTap="closeBluetoothAdapter">Close
Bluetooth</button>
       <button type="primary" onTap="getBluetoothAdapterState">Obtain
Bluetooth state</button>
    </view>
    <view class="page-section-title">Scan the Bluetooth device</view>
    <view class="page-section-demo">
       <button type="primary"
onTap="startBluetoothDevicesDiscovery">Start searching</button>
       <button type="primary" onTap="getBluetoothDevices">All devices
found</button>
       <button type="primary" onTap="getConnectedBluetoothDevices">All
devices connected</button>
```

```
       <button type="primary"
onTap="stopBluetoothDevicesDiscovery">Stop searching</button>
    </view>
    <view class="page-section-title">Connect the device</view>
    <view class="page-section-demo">
       <input class="input" onInput="bindKeyInput" type="{{text}}"
placeholder="Enter the device ID of the device to connect"></input>
       <button type="primary" onTap="connectBLEDevice">Connect the
device</button>
       <button type="primary" onTap="getBLEDeviceServices">Obtain
device services</button>
       <button type="primary"
onTap="getBLEDeviceCharacteristics">Obtain read and write
characteristics</button>
       <button type="primary" onTap="disconnectBLEDevice">Disconnect
the device</button>
    </view>
     <view class="page-section-title">Read and write data</view>
     <view class="page-section-demo">
       <button type="primary"
onTap="notifyBLECharacteristicValueChange">Listens to the
characteristic data change</button>
       <button type="primary" onTap="readBLECharacteristicValue">Read
data</button>
       <button type="primary"
onTap="writeBLECharacteristicValue">Write data</button>
       <button type="primary"
onTap="offBLECharacteristicValueChange">Un-listens to characteristic
value</button>
    </view>
     <view class="page-section-title">Other events</view>
     <view class="page-section-demo">
       <button type="primary"
onTap="bluetoothAdapterStateChange">Changes of the Bluetooth
state</button>
       <button type="primary"
onTap="offBluetoothAdapterStateChange">Un-listens to Bluetooth
state</button>
       <button type="primary"
onTap="BLEConnectionStateChanged">Changes of Bluetooth connection
state</button>
       <button type="primary" onTap="offBLEConnectionStateChanged">Un-
listens to Bluetooth connection state</button>

    </view>
  </view>
</view>
```

copy

```
// .js
Page({
  data: {
    devid: '0D9C82AD-1CC0-414D-9526-119E08D28124',
    serid: 'FEE7',
    notifyId: '36F6',
    writeId: '36F5',
    charid: '',
    alldev: [{ deviceId: '' }],
  },

  //Obtain the Bluetooth state
  openBluetoothAdapter() {
    my.openBluetoothAdapter({
      success: res => {
        if (!res.isSupportBLE) {
          my.alert({ content: 'Sorry, your mobile Bluetooth is
unavailable temporarily' });
          return;
        }
        my.alert({ content: 'Succeeded to initialize!' });
      },
      fail: error => {
        my.alert({ content: JSON.stringify(error) });
      },
    });
  },
  closeBluetoothAdapter() {
    my.closeBluetoothAdapter({
      success: () => {
        my.alert({ content: 'Bluetooth closed!' });
      },
      fail: error => {
        my.alert({ content: JSON.stringify(error) });
      },
    });
  },
  getBluetoothAdapterState() {
    my.getBluetoothAdapterState({
      success: res => {
        if (!res.available) {
          my.alert({ content: 'Sorry, your mobile Bluetooth is
unavailable temporarily' });
          return;
        }
        my.alert({ content: JSON.stringify(res) });
      },
      fail: error => {
        my.alert({ content: JSON.stringify(error) });
      },
    });
  },
  //Scan the Bluetooth device
  startBluetoothDevicesDiscovery() {
    my.startBluetoothDevicesDiscovery({
      allowDuplicatesKey: false,
      success: () => {
        my.onBluetoothDeviceFound({
          success: res => {
            // my.alert({content:'Listens to new
device'+JSON.stringify(res)});
            var deviceArray = res.devices;
            for (var i = deviceArray.length - 1; i >= 0; i--) {
              var deviceObj = deviceArray[i];
              //Pair the target device with the device name or
broadcast data, and then record the device ID for later use.
              if (deviceObj.name == this.data.name) {
                my.alert({ content: 'Target device is found' });
                my.offBluetoothDeviceFound();
                this.setData({
                  deviceId: deviceObj.deviceId,
                });
                break;
              }
            }
          },
          fail: error => {
            my.alert({ content: 'Failed to listen to new device' +
JSON.stringify(error) });
          },
        });
      },
      fail: error => {
        my.alert({ content: 'Failed to start scanning' +
JSON.stringify(error) });
      },
    });
  },

  //Stop scanning
  stopBluetoothDevicesDiscovery() {
    my.stopBluetoothDevicesDiscovery({
      success: res => {
        my.offBluetoothDeviceFound();
        my.alert({ content: 'Succeeded!' });
      },
      fail: error => {
        my.alert({ content: JSON.stringify(error) });
      },
    });
  },

  //Obtain the connected device
```

```
getConnectedBluetoothDevices() {
  my.getConnectedBluetoothDevices({
    success: res => {
      if (res.devices.length === 0) {
        my.alert({ content: 'No connecting devices!' });
        return;
      }
      my.alert({ content: JSON.stringify(res) });
      devid = res.devices[0].deviceId;
    },
    fail: error => {
      my.alert({ content: JSON.stringify(error) });
    },
  });
},

//Obtain all searched devices
getBluetoothDevices() {
  my.getBluetoothDevices({
    success: res => {
      my.alert({ content: JSON.stringify(res) });
    },
    fail: error => {
      my.alert({ content: JSON.stringify(error) });
    },
  });
},
bindKeyInput(e) {
  this.setData({
    devid: e.detail.value,
  });
},

//Connect the device
connectBLEDevice() {
  my.connectBLEDevice({
    deviceId: this.data.devid,
    success: res => {
      my.alert({ content: 'Succeeded to connect!' });
    },
    fail: error => {
      my.alert({ content: JSON.stringify(error) });
    },
  });
},

//Disconnect the device
disconnectBLEDevice() {
  my.disconnectBLEDevice({
    deviceId: this.data.devid,
    success: () => {
        my.alert({ content: 'Succeeded to disconnect!' });
      },
      fail: error => {
        my.alert({ content: JSON.stringify(error) });
      },
    });
  },

  //Obtain the services of the connected device
  getBLEDeviceServices() {
    my.getConnectedBluetoothDevices({
      success: res => {
        if (res.devices.length === 0) {
          my.alert({ content: 'No connected devices' });
          return;
        }
        my.getBLEDeviceServices({
          deviceId: this.data.devid,
          success: res => {
            my.alert({ content: JSON.stringify(res) });
            this.setData({
              serid: res.services[0].serviceId,
            });
          },
          fail: error => {
            my.alert({ content: JSON.stringify(error) });
          },
        });
      },
    });
  },

  //Obtain the char ID of the connected device, read and write
characteristics are respectively screened out.
  getBLEDeviceCharacteristics() {
    my.getConnectedBluetoothDevices({
      success: res => {
        if (res.devices.length === 0) {
          my.alert({ content: 'No connected devices' });
          return;
        }
        this.setData({
          devid: res.devices[0].deviceId,
        });
        my.getBLEDeviceCharacteristics({
          deviceId: this.data.devid,
          serviceId: this.data.serid,
          success: res => {
            my.alert({ content: JSON.stringify(res) });
            //See the related document for more information of the
properties of the characteristics. Pair the characteristics according
to the properties and record the value for later use.
            this.setData({
              charid: res.characteristics[0].characteristicId,
            });
          },
          fail: error => {
            my.alert({ content: JSON.stringify(error) });
          },
        });
      },
    });
  },

  //Read and write data
  readBLECharacteristicValue() {
    my.getConnectedBluetoothDevices({
      success: res => {
        if (res.devices.length === 0) {
          my.alert({ content: 'No connected devices' });
          return;
        }
        this.setData({
          devid: res.devices[0].deviceId,
        });
        my.readBLECharacteristicValue({
          deviceId: this.data.devid,
          serviceId: this.data.serid,
          characteristicId: this.data.notifyId,
          //1  Android reading service
          // serviceId:'0000180d-0000-1000-8000-00805f9b34fb',
          // characteristicId:'00002a38-0000-1000-8000-00805f9b34fb',
          success: res => {
            my.alert({ content: JSON.stringify(res) });
          },
          fail: error => {
            my.alert({ content: 'Failed to read' +
JSON.stringify(error) });
          },
        });
      },
    });
  },
  writeBLECharacteristicValue() {
    my.getConnectedBluetoothDevices({
      success: res => {
        if (res.devices.length === 0) {
          my.alert({ content: 'No connected devices' });
          return;
        }
        this.setData({
          devid: res.devices[0].deviceId,
        });
        my.writeBLECharacteristicValue({
          deviceId: this.data.devid,
          serviceId: this.data.serid,
          characteristicId: this.data.charid,
          //Android writing service
          //serviceId:'0000180d-0000-1000-8000-00805f9b34fb',
          //characteristicId:'00002a39-0000-1000-8000-00805f9b34fb',
          value: 'ABCD',
          success: res => {
            my.alert({ content: 'Succeeded to write data!' });
          },
          fail: error => {
            my.alert({ content: JSON.stringify(error) });
          },
        });
      },
    });
  },
  notifyBLECharacteristicValueChange() {
    my.getConnectedBluetoothDevices({
      success: res => {
        if (res.devices.length === 0) {
          my.alert({ content: 'No connected devices' });
          return;
        }
        this.setData({
          devid: res.devices[0].deviceId,
        });
        my.notifyBLECharacteristicValueChange({
          state: true,
          deviceId: this.data.devid,
          serviceId: this.data.serid,
          characteristicId: this.data.notifyId,
          success: () => {
            //Listens to characteristic change events
            my.onBLECharacteristicValueChange({
              success: res => {
                //  my.alert({content: 'Changes of
characteristics  '+JSON.stringify(res)});
                my.alert({ content: 'Obtain the response data = ' +
res.value });
              },
            });
            my.alert({ content: 'Succeeded to listen' });
          },
          fail: error => {
            my.alert({ content: 'Failed to listen' +
JSON.stringify(error) });
          },
        });
      },
    });
  },
  offBLECharacteristicValueChange() {
    my.offBLECharacteristicValueChange();
  },

  //Other events
  bluetoothAdapterStateChange() {

my.onBluetoothAdapterStateChange(this.getBind('onBluetoothAdapterStateC
  },
  onBluetoothAdapterStateChange() {
    if (res.error) {
      my.alert({ content: JSON.stringify(error) });
    } else {
      my.alert({ content: 'Changes of the Bluetooth state  ' +
JSON.stringify(res) });
    }
  },
  offBluetoothAdapterStateChange() {

my.offBluetoothAdapterStateChange(this.getBind('onBluetoothAdapterState
  },
  getBind(name) {
    if (!this[`bind${name}`]) {
      this[`bind${name}`] = this[name].bind(this);
    }
    return this[`bind${name}`];
  },
  BLEConnectionStateChanged() {

my.onBLEConnectionStateChanged(this.getBind('onBLEConnectionStateChange
  },
  onBLEConnectionStateChanged(res) {
    if (res.error) {
      my.alert({ content: JSON.stringify(error) });
    } else {
      my.alert({ content: 'Changes of connection state  ' +
JSON.stringify(res) });
    }
  },
  offBLEConnectionStateChanged() {

my.offBLEConnectionStateChanged(this.getBind('onBLEConnectionStateChang
  },
  onUnload() {
    this.offBLEConnectionStateChanged();
    this.offBLECharacteristicValueChange();
    this.offBluetoothAdapterStateChange();
    this.closeBluetoothAdapter();
  },
});
```

Parameters

The input parameters are displayed in the following table:

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Required** | **Description** | | success | Function | No | The callback function for a successful API call. | | fail | Function | No | The callback function for a failed API call. | | complete | Function | No | The callback function for a completed API call (Regardless of whether the call is successful or not). |

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev-old/api_device_bluetooth_bluetooth_stopbluetoothdevicesdiscovery

---

my.stopBluetoothDevicesDiscovery {#mystopbluetoothdevicesdiscovery}

Last updated: 2022-07-03

Path: miniprogram_gcash

my.stopBluetoothDevicesDiscovery

2022-07-03 18:44

Use this API to stop discovering bluetooth devices.

Note:

Currently simulation in IDE is not supported. Please debug in the production environment.

Code Sample

copy

```
/* .acss */
.help-info {
  padding:10px;
  color:#000000;
}
.help-title {
  padding:10px;
  color:#FC0D1B;
}
```

copy

```
// .json
{
    "defaultTitle": "Bluetooth"
}
```

copy

```
<!-- .axml-->
<view class="page">
  <view class="page-description">Bluetooth API</view>
  <view class="page-section">
    <view class="page-section-title">The Bluetooth state</view>
    <view class="page-section-demo">
       <button type="primary" onTap="openBluetoothAdapter">Initialize
Bluetooth</button>
       <button type="primary" onTap="closeBluetoothAdapter">Close
Bluetooth</button>
       <button type="primary" onTap="getBluetoothAdapterState">Obtain
Bluetooth state</button>
    </view>
    <view class="page-section-title">Scan the Bluetooth device</view>
    <view class="page-section-demo">
       <button type="primary"
onTap="startBluetoothDevicesDiscovery">Start searching</button>
       <button type="primary" onTap="getBluetoothDevices">All devices
found</button>
       <button type="primary" onTap="getConnectedBluetoothDevices">All
devices connected</button>
       <button type="primary"
onTap="stopBluetoothDevicesDiscovery">Stop searching</button>
    </view>
    <view class="page-section-title">Connect the device</view>
    <view class="page-section-demo">
       <input class="input" onInput="bindKeyInput" type="{{text}}"
placeholder="Enter the device ID of the device to connect"></input>
       <button type="primary" onTap="connectBLEDevice">Connect the
device</button>
       <button type="primary" onTap="getBLEDeviceServices">Obtain
device services</button>
       <button type="primary"
onTap="getBLEDeviceCharacteristics">Obtain read and write
characteristics</button>
       <button type="primary" onTap="disconnectBLEDevice">Disconnect
the device</button>
    </view>
     <view class="page-section-title">Read and write data</view>
     <view class="page-section-demo">
```

```
       <button type="primary"
onTap="notifyBLECharacteristicValueChange">Listens to the
characteristic data change</button>
       <button type="primary" onTap="readBLECharacteristicValue">Read
data</button>
       <button type="primary"
onTap="writeBLECharacteristicValue">Write data</button>
       <button type="primary"
onTap="offBLECharacteristicValueChange">Un-listens to characteristic
value</button>
    </view>
     <view class="page-section-title">Other events</view>
     <view class="page-section-demo">
       <button type="primary"
onTap="bluetoothAdapterStateChange">Changes of the Bluetooth
state</button>
       <button type="primary"
onTap="offBluetoothAdapterStateChange">Un-listens to Bluetooth
state</button>
       <button type="primary"
onTap="BLEConnectionStateChanged">Changes of Bluetooth connection
state</button>
       <button type="primary" onTap="offBLEConnectionStateChanged">Un-
listens to Bluetooth connection state</button>

    </view>
  </view>
</view>
```

copy

```
// .js
Page({
  data: {
    devid: '0D9C82AD-1CC0-414D-9526-119E08D28124',
    serid: 'FEE7',
    notifyId: '36F6',
    writeId: '36F5',
    charid: '',
    alldev: [{ deviceId: '' }],
  },

  //Obtain the Bluetooth state
  openBluetoothAdapter() {
    my.openBluetoothAdapter({
      success: res => {
        if (!res.isSupportBLE) {
          my.alert({ content: 'Sorry, your mobile Bluetooth is
unavailable temporarily' });
          return;
        }
        my.alert({ content: 'Succeeded to initialize!' });
      },
      fail: error => {
        my.alert({ content: JSON.stringify(error) });
      },
    });
  },
  closeBluetoothAdapter() {
    my.closeBluetoothAdapter({
      success: () => {
        my.alert({ content: 'Bluetooth closed!' });
      },
      fail: error => {
        my.alert({ content: JSON.stringify(error) });
      },
    });
  },
  getBluetoothAdapterState() {
    my.getBluetoothAdapterState({
      success: res => {
        if (!res.available) {
          my.alert({ content: 'Sorry, your mobile Bluetooth is
unavailable temporarily' });
          return;
        }
        my.alert({ content: JSON.stringify(res) });
      },
      fail: error => {
        my.alert({ content: JSON.stringify(error) });
      },
    });
  },

  //Scan the Bluetooth device
  startBluetoothDevicesDiscovery() {
    my.startBluetoothDevicesDiscovery({
      allowDuplicatesKey: false,
      success: () => {
        my.onBluetoothDeviceFound({
          success: res => {
            // my.alert({content:'Listens to new
device'+JSON.stringify(res)});
            var deviceArray = res.devices;
            for (var i = deviceArray.length - 1; i >= 0; i--) {
              var deviceObj = deviceArray[i];
              //Pair the target device with the device name or
broadcast data, and then record the device ID for later use.
              if (deviceObj.name == this.data.name) {
                my.alert({ content: 'Target device is found' });
                my.offBluetoothDeviceFound();
                this.setData({
                  deviceId: deviceObj.deviceId,
                });
                break;
              }
            }
          },
          fail: error => {
            my.alert({ content: 'Failed to listen to new device' +
JSON.stringify(error) });
          },
        });
      },
      fail: error => {
        my.alert({ content: 'Failed to start scanning' +
JSON.stringify(error) });
      },
    });
  },

  //Stop scanning
  stopBluetoothDevicesDiscovery() {
    my.stopBluetoothDevicesDiscovery({
      success: res => {
        my.offBluetoothDeviceFound();
        my.alert({ content: 'Succeeded!' });
      },
      fail: error => {
        my.alert({ content: JSON.stringify(error) });
      },
    });
  },

  //Obtain the connected device
  getConnectedBluetoothDevices() {
    my.getConnectedBluetoothDevices({
      success: res => {
        if (res.devices.length === 0) {
          my.alert({ content: 'No connecting devices!' });
          return;
        }
        my.alert({ content: JSON.stringify(res) });
        devid = res.devices[0].deviceId;
      },
      fail: error => {
        my.alert({ content: JSON.stringify(error) });
      },
    });
  },

  //Obtain all searched devices
  getBluetoothDevices() {
    my.getBluetoothDevices({
```

```
    success: res => {
      my.alert({ content: JSON.stringify(res) });
    },
    fail: error => {
      my.alert({ content: JSON.stringify(error) });
    },
  });
},
bindKeyInput(e) {
  this.setData({
    devid: e.detail.value,
  });
},

//Connect the device
connectBLEDevice() {
  my.connectBLEDevice({
    deviceId: this.data.devid,
    success: res => {
      my.alert({ content: 'Succeeded to connect!' });
    },
    fail: error => {
      my.alert({ content: JSON.stringify(error) });
    },
  });
},

//Disconnect the device
disconnectBLEDevice() {
  my.disconnectBLEDevice({
    deviceId: this.data.devid,
    success: () => {
      my.alert({ content: 'Succeeded to disconnect!' });
    },
    fail: error => {
      my.alert({ content: JSON.stringify(error) });
    },
  });
},

//Obtain the services of the connected device
getBLEDeviceServices() {
  my.getConnectedBluetoothDevices({
    success: res => {
      if (res.devices.length === 0) {
        my.alert({ content: 'No connected devices' });
        return;
      }
      my.getBLEDeviceServices({
        deviceId: this.data.devid,
        success: res => {
            my.alert({ content: JSON.stringify(res) });
            this.setData({
              serid: res.services[0].serviceId,
            });
          },
          fail: error => {
            my.alert({ content: JSON.stringify(error) });
          },
        });
      },
    });
  },

  //Obtain the char ID of the connected device, read and write
characteristics are respectively screened out.
  getBLEDeviceCharacteristics() {
    my.getConnectedBluetoothDevices({
      success: res => {
        if (res.devices.length === 0) {
          my.alert({ content: 'No connected devices' });
          return;
        }
        this.setData({
          devid: res.devices[0].deviceId,
        });
        my.getBLEDeviceCharacteristics({
          deviceId: this.data.devid,
          serviceId: this.data.serid,
          success: res => {
            my.alert({ content: JSON.stringify(res) });
            //See the related document for more information of the
properties of the characteristics. Pair the characteristics according
to the properties and record the value for later use.
            this.setData({
              charid: res.characteristics[0].characteristicId,
            });
          },
          fail: error => {
            my.alert({ content: JSON.stringify(error) });
          },
        });
      },
    });
  },

  //Read and write data
  readBLECharacteristicValue() {
    my.getConnectedBluetoothDevices({
      success: res => {
        if (res.devices.length === 0) {
          my.alert({ content: 'No connected devices' });
          return;
        }
        this.setData({
          devid: res.devices[0].deviceId,
        });
        my.readBLECharacteristicValue({
          deviceId: this.data.devid,
          serviceId: this.data.serid,
          characteristicId: this.data.notifyId,
          //1  Android reading service
          // serviceId:'0000180d-0000-1000-8000-00805f9b34fb',
          // characteristicId:'00002a38-0000-1000-8000-00805f9b34fb',
          success: res => {
            my.alert({ content: JSON.stringify(res) });
          },
          fail: error => {
            my.alert({ content: 'Failed to read' +
JSON.stringify(error) });
          },
        });
      },
    });
  },
  writeBLECharacteristicValue() {
    my.getConnectedBluetoothDevices({
      success: res => {
        if (res.devices.length === 0) {
          my.alert({ content: 'No connected devices' });
          return;
        }
        this.setData({
          devid: res.devices[0].deviceId,
        });
        my.writeBLECharacteristicValue({
          deviceId: this.data.devid,
          serviceId: this.data.serid,
          characteristicId: this.data.charid,
          //Android writing service
          //serviceId:'0000180d-0000-1000-8000-00805f9b34fb',
          //characteristicId:'00002a39-0000-1000-8000-00805f9b34fb',
          value: 'ABCD',
          success: res => {
            my.alert({ content: 'Succeeded to write data!' });
          },
          fail: error => {
            my.alert({ content: JSON.stringify(error) });
          },
        });
      },
    });
  },
  notifyBLECharacteristicValueChange() {
    my.getConnectedBluetoothDevices({
      success: res => {
        if (res.devices.length === 0) {
          my.alert({ content: 'No connected devices' });
          return;
        }
        this.setData({
          devid: res.devices[0].deviceId,
        });
        my.notifyBLECharacteristicValueChange({
          state: true,
          deviceId: this.data.devid,
          serviceId: this.data.serid,
          characteristicId: this.data.notifyId,
          success: () => {
            //Listens to characteristic change events
            my.onBLECharacteristicValueChange({
              success: res => {
                //  my.alert({content: 'Changes of
characteristics  '+JSON.stringify(res)});
                my.alert({ content: 'Obtain the response data = ' +
res.value });
              },
            });
            my.alert({ content: 'Succeeded to listen' });
          },
          fail: error => {
            my.alert({ content: 'Failed to listen' +
JSON.stringify(error) });
          },
        });
      },
    });
  },
  offBLECharacteristicValueChange() {
    my.offBLECharacteristicValueChange();
  },

  //Other events
  bluetoothAdapterStateChange() {

my.onBluetoothAdapterStateChange(this.getBind('onBluetoothAdapterState
  },
  onBluetoothAdapterStateChange() {
    if (res.error) {
      my.alert({ content: JSON.stringify(error) });
    } else {
      my.alert({ content: 'Changes of the Bluetooth state  ' +
JSON.stringify(res) });
    }
  },
  offBluetoothAdapterStateChange() {

my.offBluetoothAdapterStateChange(this.getBind('onBluetoothAdapterState
  },
  getBind(name) {
    if (!this[`bind${name}`]) {
      this[`bind${name}`] = this[name].bind(this);
    }
    return this[`bind${name}`];
  },
  BLEConnectionStateChanged() {

my.onBLEConnectionStateChanged(this.getBind('onBLEConnectionStateChange
  },
  onBLEConnectionStateChanged(res) {
    if (res.error) {
      my.alert({ content: JSON.stringify(error) });
    } else {
      my.alert({ content: 'Changes of connection state  ' +
JSON.stringify(res) });
    }
  },
  offBLEConnectionStateChanged() {

my.offBLEConnectionStateChanged(this.getBind('onBLEConnectionStateChang
  },
  onUnload() {
    this.offBLEConnectionStateChanged();
    this.offBLECharacteristicValueChange();
    this.offBluetoothAdapterStateChange();
    this.closeBluetoothAdapter();
  },
});
```

Parameters

The input parameters are displayed in the following table:

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Required** | **Description** | | success | Function | No | The callback function for a successful API call. | | fail | Function | No | The callback function for a failed API call. | | complete | Function | No | The callback function for a completed API call (Regardless of whether the call is successful or not). |

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev/api_device_bluetooth_bluetooth_stopbluetoothdevicesdiscovery

---

my.stopPullDownRefresh {#mystoppulldownrefresh}

Last updated: 2022-07-03

Path: miniprogram_gcash

my.stopPullDownRefresh

2022-07-03 18:44

Stop the pull-to-refresh for the current page.

Sample Code:

copy

```
Page({
  onPullDownRefresh(){
    my.stopPullDownRefresh()
  }
})
```

Parameters

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Required** | **Description** | | success | Function | No | The callback function for a successful API call. | | fail | Function | No | The callback function for a failed API call. | | complete | Function | No | The callback function used when the API call is completed. This function is always executed no matter the call succeeds or fails. |

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev/api_ui_pulldown_stoppulldownrefresh

---

my.stopPullDownRefresh {#mystoppulldownrefresh}

Last updated: 2021-05-09

Path: miniprogram_gcash

my.stopPullDownRefresh

2021-05-09 18:43

Stop the pull-to-refresh for the current page.

Sample Code:

copy

```
Page({
  onPullDownRefresh(){
    my.stopPullDownRefresh()
  }
})
```

Parameters

| | | | | --- | --- | --- | --- | | **Property** | **Type** | **Required** | **Description** | | success | Function | No | The callback function for a successful API call. | | fail | Function | No | The callback function for a failed API call. | | complete | Function | No | The callback function used when the API call is completed. This function is always executed no matter the call succeeds or fails. |

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev-old/api_ui_pulldown_stoppulldownrefresh

---

my.switchTab {#myswitchtab}

Last updated: 2022-07-03

Path: miniprogram_gcash

my.switchTab

2022-07-03 18:44

Introduction

Jump to the specified tabBar page, and then close all other pages that are not tabBar.

If the Mini Program has multiple tabs, you can use `tabBar` to configure the behavior of the specified tab, and the page to display during the tab switchover (You can switch tabs at the bottom of the client window).

Note: You can jump to the page via page jumping (`my.navigateTo`) or page redirecting (`my.redirectTo`) . Even the page is defined in the tabBar configuration, the bottom tab bar is not displayed. In addition, the first page of the `tabBar` must be the homepage.

Sample Code

copy

```
// app.json
{
  "tabBar": {
    "items": [{\
      "pagePath": "page/home/index",\
      "name": "Home"\
    },{\
      "pagePath": "page/user/index",\
      "name": "User"\
    }]
  }
}
```

copy

```
my.switchTab({
  url: 'page/home/index'
})
```

Parameters

Property	**Type**	**Required**	**Description**
url	String	Yes	Path of the jumping tabBar page (page to be defined in the tabBar field in the app.json). **Note:** The path cannot be followed by parameters.
success	Function	No	Callback function upon call success.
fail	Function	No	Callback function upon call failure.
complete	Function	No	Callback function upon call completion (to be executed upon either call success or failure).

tabBar Configuration

Property	**Type**	**Required**	**Description**
textColor	HexColor	No	Text color.
selectedColor	HexColor	No	Color of highlighted text.
backgroundColor	HexColor	No	Background color.
items	Array	Yes	Configured for each tab.

Configured for each item:

Property	**Type**	**Required**	**Description**
pagePath	String	Yes	Set page path.
name	String	Yes	Name.
icon	String	No	Normal icon path.
activeIcon	String	No	Highlighted icon path.

The recommended icon size is 60*60px. The system performs unequal stretching and scaling for any incoming picture.

Example

copy

```
{
  "tabBar": {
    "textColor": "#dddddd",
    "selectedColor": "#49a9ee",
    "backgroundColor": "#ffffff",
    "items": [\
      {\
        "pagePath": "pages/index/index",\
        "name": "Home"\
      },\
      {\
        "pagePath": "pages/logs/logs",\
        "name": "Log"\
      }\
    ]
  }
}
```

Source:
https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev/API_UI_Route_switchTab

---

my.switchTab {#myswitchtab}

Last updated: 2021-05-09

Path: miniprogram_gcash

my.switchTab

2021-05-09 18:43

Introduction

Jump to the specified tabBar page, and then close all other pages that are not tabBar.

If the Mini Program has multiple tabs, you can use `tabBar` to configure the behavior of the specified tab, and the page to display during the tab switchover (You can switch tabs at the bottom of the client window).

Note: You can jump to the page via page jumping (`my.navigateTo`) or page redirecting (`my.redirectTo`) . Even the page is defined in the tabBar configuration, the bottom tab bar is not displayed. In addition, the first page of the `tabBar` must be the homepage.

Sample Code

copy

```
// app.json
{
  "tabBar": {
    "items": [{\
      "pagePath": "page/home/index",\
      "name": "Home"\
    },{\
      "pagePath": "page/user/index",\
      "name": "User"\
    }]
  }
}
```

copy

```
my.switchTab({
  url: 'page/home/index'
})
```

Parameters

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Required** | **Description** | | url | String | Yes | Path of the jumping tabBar page (page to be defined in the tabBar field in the app.json).
Note:
The path cannot be followed by parameters. | | success | Function | No | Callback function upon call success. | | fail | Function | No | Callback function upon call failure. | | complete | Function | No | Callback function upon call completion (to be executed upon either call success or failure). |

tabBar Configuration

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Required** | **Description** | | textColor | HexColor | No | Text color. | | selectedColor | HexColor | No | Color of highlighted text. | | backgroundColor | HexColor | No | Background color. | | items | Array | Yes | Configured for each tab. |

Configured for each item:

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Required** | **Description** | | pagePath | String | Yes | Set page path. | | name | String | Yes | Name. | | icon | String | No | Normal icon path. | | activeIcon | String | No | Highlighted icon path. |

The recommended icon size is 60*60px. The system performs unequal stretching and scaling for any incoming picture.

Example

copy

```
{
  "tabBar": {
    "textColor": "#dddddd",
    "selectedColor": "#49a9ee",
    "backgroundColor": "#ffffff",
    "items": [\
      {\
        "pagePath": "pages/index/index",\
        "name": "Home"\
      },\
      {\
        "pagePath": "pages/logs/logs",\
        "name": "Log"\
      }\
    ]
  }
}
```

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev-old/api_ui_route_switchtab

---

my.tradePay {#mytradepay}

Last updated: 2022-07-04

Path: miniprogram_gcash

my.tradePay

2022-07-04 03:44

Start a payment transaction.

Sample Code

copy

```
my.tradePay({
  tradeNO: '201711152100110410533667792', // get the tradeNo from the
server first
  success: (res) => {
    my.alert({
      content: JSON.stringify(res),
    });
  },
  fail: (res) => {
    my.alert({
      content: JSON.stringify(res),
    });
  }
});
```

Parameters

Property	**Type**	**Required**	**Description**
tradeNO	String	No	The trade number. Its maximum length is 64. Refer here for details.
orderStr	String	No	A string of complete payment parameters, which is recommended to be obtained from the server. Refer here for details.
paymentUrl	String	No	The url of payment page. Refer here for details.
success	Function	No	Callback function upon call success.
fail	Function	No	Callback function upon call failure.
complete	Function	No	Callback function upon call completion (to be executed upon either call success or failure).

The `tradeNO`, `orderStr` and `paymentUrl` are different ways to start the payment transaction. **The Mini Program should use either one of them to start a payment transaction according to the payment service provided by the host app**.

Success Callback Function

The incoming parameter is of the Object type with the following attributes:

Property	**Type**	**Required**	**Description**
resultCode	String	Yes	The result code of the pay process.

An example of a successfully returned message is as follows:

copy

```
{
    "resultCode":"9000"
}
```

Result Code

resultCode	**Description**
9000	Payment is successful.
8000	Trade is processing.
4000	Payment is failed.
6001	User cancels to pay.
6002	Network exception.
6004	Unknown pay result, may be success.

Source:
https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev/API_OpenAPI_tradePay

my.tradePay {#mytradepay}

Last updated: 2021-05-09

Path: miniprogram_gcash

my.tradePay

2021-05-09 18:43

Start a payment transaction.

Sample Code

copy

```
my.tradePay({
  tradeNO: '201711152100110410533667792', // get the tradeNo from the
server first
  success: (res) => {
    my.alert({
      content: JSON.stringify(res),
    });
  },
  fail: (res) => {
    my.alert({
      content: JSON.stringify(res),
    });
  }
});
```

Parameters

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Required** | **Description** | | tradeNO | String | No | The trade number. Its maximum length is 64. Refer
here
for details. | | orderStr | String | No | A string of complete payment parameters, which is recommended to be obtained from the server. Refer
here
for details. | | paymentUrl | String | No | The url of payment page. Refer
here
for details. | | success | Function | No | Callback function upon call success. | | fail | Function | No | Callback function upon call failure. | | complete | Function | No | Callback function upon call completion (to be executed upon either call success or failure). |

The `tradeNO`, `orderStr` and `paymentUrl` are different ways to start the payment transaction. **The Mini Program should use either one of them to start a payment transaction according to the payment service provided by the host app**.

Success Callback Function

The incoming parameter is of the Object type with the following attributes:

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Required** | **Description** | | resultCode | String | Yes | The result code of the pay process. |

An example of a successfully returned message is as follows:

copy

```
{
    "resultCode":"9000"
}
```

Result Code

| | | | --- | --- | | **resultCode** | **Description** | | 9000 | Payment is successful. | | 8000 | Trade is processing. | | 4000 | Payment is failed. | | 6001 | User cancels to pay. | | 6002 | Network exception. | | 6004 | Unknown pay result, may be success. |

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev-old/api_openapi_tradepay

---

my.uploadFile {#myuploadfile}

Last updated: 2021-05-09

Path: miniprogram_gcash

my.uploadFile

2021-05-09 18:43

Upload the local resource to the server.

Sample Code

copy

```
my.uploadFile({
  url: 'Please use your own server address',
  fileType: 'image',
  fileName: 'file',
  filePath: '...',
  success: (res) => {
    my.alert({
      content: 'Upload success'
    });
  },
});
```

Parameters

| | | | | --- | --- | --- | --- | | **Property** | **Type** | **Required** | **Description** | | url | String | Yes | Address of the developer server. | | filePath | String | Yes | Local locator of the file resource to be uploaded. | | fileName | String | Yes | Filename, also the corresponding key. The developer uses this key at the server side to get the file binary contents. | | fileType | String | Yes | File type. The following file types are supported:
- PDF
- DOC
- DOCX
- XLS
- XLSX
- PPT
- PPTX | | header | Object | No | HTTP request Header. | | formData | Object | No | Other additional form data in the HTTP request. | | success | Function | No | Callback function upon call success. | | fail | Function | No | Callback function upon call failure. | | complete | Function | No | Callback function upon call completion (to be executed upon either call success or failure). |

Success Callback Function

The incoming parameter is of the Object type with the following attributes:

| | | | --- | --- | --- | | **Property** | **Type** | **Description** | | data | String | Data returned from the server. | | statusCode | String | HTTP status code. | | header | Object | Header returned from the server. |

Error Code

| | | --- | --- | | **Error** | **Description** | | 11 | File nonexistent. | | 12 | File uploading failed. | | 13 | No right. |

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev-old/api_network_uploadfile

---

my.uploadFile {#myuploadfile}

Last updated: 2022-07-03

Path: miniprogram_gcash

my.uploadFile

2022-07-03 18:44

Upload the local resource to the server.

Sample Code

copy

```
my.uploadFile({
  url: 'Please use your own server address',
  fileType: 'image',
  fileName: 'file',
  filePath: '...',
  success: (res) => {
    my.alert({
      content: 'Upload success'
    });
  },
});
```

Parameters

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Required** | **Description** | | url | String | Yes | Address of the developer server. | | filePath | String | Yes | Local locator of the file resource to be uploaded. | | fileName | String | Yes | Filename, also the corresponding key. The developer uses this key at the server side to get the file binary contents. | | fileType | String | Yes | File type. The following file types are supported:
- PDF
- DOC
- DOCX
- XLS
- XLSX
- PPT
- PPTX | | header | Object | No | HTTP request Header. | | formData | Object | No | Other additional form data in the HTTP request. | | success | Function | No | Callback function upon call success. | | fail | Function | No | Callback function upon call failure. | | complete | Function | No | Callback function upon call completion (to be executed upon either call success or failure). |

Success Callback Function

The incoming parameter is of the Object type with the following attributes:

| | | | | --- | --- | --- | | **Property** | **Type** | **Description** | | data | String | Data returned from the server. | | statusCode | String | HTTP status code. | | header | Object | Header returned from the server. |

Error Code

| | | | --- | --- | | **Error** | **Description** | | 11 | File nonexistent. | | 12 | File uploading failed. | | 13 | No right. |

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev/api_network_uploadfile

---

my.vibrate {#myvibrate}

Last updated: 2022-07-03

Path: miniprogram_gcash

my.vibrate

2022-07-03 18:44

Call the device vibrate function.

Sample Code

copy

```
// API-DEMO page/API/vibrate/vibrate.json
{
    "defaultTitle": "Vibrate"
}
```

copy

```
<!-- API-DEMO page/API/vibrate/vibrate.axml-->
<view class="page">

  <button type="primary" onTap="vibrate">
    Start vibration
  </button>
```

```
</view>
```

copy

```
// API-DEMO page/API/vibrate/vibrate.js
Page({
  vibrate() {
    my.vibrate({
      success: () => {
        my.alert({ title: 'Vibration on'});
      }
    });
  },
});
```

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev/api_device_vibrate_vibrate

---

my.vibrate {#myvibrate}

Last updated: 2021-05-09

Path: miniprogram_gcash

my.vibrate

2021-05-09 18:43

Call the device vibrate function.

Sample Code

copy

```
// API-DEMO page/API/vibrate/vibrate.json
{
    "defaultTitle": "Vibrate"
}
```

copy

```
<!-- API-DEMO page/API/vibrate/vibrate.axml-->
<view class="page">

  <button type="primary" onTap="vibrate">
```

```
    Start vibration
  </button>

</view>
```

copy

```
// API-DEMO page/API/vibrate/vibrate.js
Page({
  vibrate() {
    my.vibrate({
      success: () => {
        my.alert({ title: 'Vibration on'});
      }
    });
  },
});
```

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev-old/api_device_vibrate_vibrate

---

my.watchShake {#mywatchshake}

Last updated: 2021-05-10

Path: miniprogram_gcash

my.watchShake

2021-05-10 03:43

The watchshake function. Every time this API is called, a callback is triggered after the mobile phone is shaken. You must call this API again to continue to listen to this event.

Sample Code

copy

```
// API-DEMO page/API/watch-shake/watch-shake.json
{
    "defaultTitle": "Shake"
}
```

copy

```
<!-- API-DEMO page/API/watch-shake/watch-shake.axml-->
<view class="page">
```

```
  <button type="primary" onTap="watchShake">
    Bind the shake function and click the shake button to see the
effect.
  </button>
</view>
```

copy

```
// API-DEMO page/API/watch-shake/watch-shake.js
Page({
  watchShake() {
    my.watchShake({
      success: function() {
        console.log('It moved')
        my.alert({ title:'It moved o.o'});
      }
    });
  },
});
```

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev-old/api_device_watch-shake_watchshake

---

my.watchShake {#mywatchshake}

Last updated: 2022-07-03

Path: miniprogram_gcash

my.watchShake

2022-07-03 18:44

The watchshake function. Every time this API is called, a callback is triggered after the mobile phone is shaken. You must call this API again to continue to listen to this event.

Sample Code

copy

```
// API-DEMO page/API/watch-shake/watch-shake.json
{
    "defaultTitle": "Shake"
}
```

copy

```
<!-- API-DEMO page/API/watch-shake/watch-shake.axml-->
<view class="page">
  <button type="primary" onTap="watchShake">
    Bind the shake function and click the shake button to see the
effect.
  </button>
</view>
```

copy

```
// API-DEMO page/API/watch-shake/watch-shake.js
Page({
  watchShake() {
    my.watchShake({
      success: function() {
        console.log('It moved')
        my.alert({ title:'It moved o.o'});
      }
    });
  },
});
```

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev/api_device_watch-shake_watchshake

---

my.writeBLECharacteristicValue {#mywriteblecharacteristicvalue}

Last updated: 2022-07-03

Path: miniprogram_gcash

my.writeBLECharacteristicValue

2022-07-03 18:44

Use this API to write data to Bluetooth Low Energy (BLE) device characteristics.

Instructions:

- The device characteristics must support writing to use this API. See properties in my.getBLEDeviceCharacteristics for details.
- The written binary data needs Hex coding.

Note:

Currently simulation in IDE is not supported. Please debug in production environment.

Sample Code

copy

```
my.writeBLECharacteristicValue({
  deviceId: deviceId,
  serviceId: serviceId,
  characteristicId: characteristicId,
  value: 'fffe',
  success: (res) => {
    console.log(res)
  },
  fail:(res) => {
  },
  complete: (res)=>{
  }
});
```

Parameters

Property	**Type**	**Required**	**Description**
deviceId	String	Yes	The Bluetooth device ID.
serviceId	String	Yes	The UUID of the service corresponding to a Bluetooth characteristic.
characteristicId	String	Yes	The Bluetooth characteristic UUID.
value	Hex String	Yes	The value corresponding to the Bluetooth device characteristic, which is a hexadecimal string, limited to 20 bytes.
success	Function	No	The callback function for a successful API call.
fail	Function	No	The callback function for a failed API call.
complete	Function	No	The callback function used when the API call is completed. This function is always executed no matter the call succeeds or fails.

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev/API_Device_Bluetooth_BLE_writeBLECharacteristicValue

---

my.writeBLECharacteristicValue {#mywriteblecharacteristicvalue}

Last updated: 2021-05-09

Path: miniprogram_gcash

my.writeBLECharacteristicValue

2021-05-09 18:43

Use this API to write data to Bluetooth Low Energy (BLE) device characteristics.

Instructions:

- The device characteristics must support writing to use this API. See properties in my.getBLEDeviceCharacteristics for details.
- The written binary data needs Hex coding.

Note:

Currently simulation in IDE is not supported. Please debug in production environment.

Sample Code

copy

```
my.writeBLECharacteristicValue({
  deviceId: deviceId,
  serviceId: serviceId,
  characteristicId: characteristicId,
  value: 'fffe',
  success: (res) => {
    console.log(res)
  },
  fail:(res) => {
  },
  complete: (res)=>{
  }
});
```

Parameters

Property	**Type**	**Required**	**Description**
deviceId	String	Yes	The Bluetooth device ID.
serviceId	String	Yes	The UUID of the service corresponding to a Bluetooth characteristic.
characteristicId	String	Yes	The Bluetooth characteristic UUID.
value	Hex String	Yes	The value corresponding to the Bluetooth device characteristic, which is a hexadecimal string, limited to 20 bytes.
success	Function	No	The callback function for a successful API call.
fail	Function	No	The callback function for a failed API call.
complete	Function	No	The callback function used when the API call is completed. This function is always executed no matter the call succeeds or fails.

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev-old/api_device_bluetooth_ble_writeblecharacteristicvalue

---

navigator {#navigator}

Last updated: 2021-05-09

Path: miniprogram_gcash

navigator

2021-05-09 18:43

Page link

Scan QR code to try:

| | | | | --- | --- | --- | --- | | **Property** | **Type** | **Default** | **Description** | | open-type | String | navigate | Jump mode. | | hover-class | String | none | Added class upon click. | | hover-start-time | Number | | Click status shown in certain time after clicking, in milliseconds. | | hover-stay-time | Number | | Clicking status retention time after release, in milliseconds. | | url | String | | Jump link within application. |

open-type Effective Value

| | | --- | --- | | **Property** | **Description** | | navigate | Corresponding to my.navigateTo function. | | redirect | Corresponding to my.redirectTo function. | | switchTab | Corresponding to my.switchTab function. | | navigateBack | Corresponding to my.navigateBack function. |

Sample Code

copy

```
<!-- sample.axml -->
<view class="btn-area">
  <navigator url="/page/navigate/navigate?title=navigate" hover-class="navigator-hover">Jump to new page</navigator>
  <navigator url="../../redirect/redirect/redirect?title=redirect" open-type="redirect" hover-class="other-navigator-hover">Open in current page</navigator>
  <navigator url="/page/index/index" open-type="switchTab" hover-class="other-navigator-hover">Switch Tab</navigator>
</view>
```

九色鹿

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev-old/component_navigator_navigator

---

navigator {#navigator}

Last updated: 2022-07-03

Path: miniprogram_gcash

navigator

2022-07-03 18:44

Page link

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Default** | **Description** | | open-type | String | navigate | Jump mode. | | hover-class | String | none | Added class upon click. | | hover-start-time | Number | | Click status shown in certain time after clicking, in milliseconds. | | hover-stay-time | Number | | Clicking status retention time after release, in milliseconds. | | url | String | | Jump link within application. |

open-type Effective Value

| | | | --- | --- | | **Property** | **Description** | | navigate | Corresponding to my.navigateTo function. | | redirect | Corresponding to my.redirectTo function. | | switchTab | Corresponding to my.switchTab function. | | navigateBack | Corresponding to my.navigateBack function. |

Sample Code

copy

```
<!-- sample.axml -->
<view class="btn-area">
  <navigator url="/page/navigate/navigate?title=navigate" hover-class="navigator-hover">Jump to new page</navigator>
  <navigator url="../../redirect/redirect/redirect?title=redirect" open-type="redirect" hover-class="other-navigator-hover">Open in current page</navigator>
  <navigator url="/page/index/index" open-type="switchTab" hover-class="other-navigator-hover">Switch Tab</navigator>
</view>
```

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev/component_navigator_navigator

---

onPullDownRefresh {#onpulldownrefresh}

Last updated: 2022-07-03

Path: miniprogram_gcash

onPullDownRefresh

2022-07-03 18:44

On the Page, customize the onPullDownRefresh function to listen to the pull-to-refresh event from the user.

- The `"pullRefresh": true` options needs to be configured in the `.json` configuration file of the related page to enable the pull-to-refresh event.
- when the data refresh is processed completely, call `my.stopPullDownRefresh` to stop the pull-to-refresh for that page.

Sample Code:

In the `pull-down-refresh.json` configuration file, the code configuration is as below:

copy

```
{
    "pullRefresh": true
}
```

In the Page, define the onPullDownRefresh processing function:

copy

```
onPullDownRefresh() {
    console.log('onPullDownRefresh', new Date())
}
```

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev/API_UI_Pulldown_onPullDownRefresh

---

onPullDownRefresh {#onpulldownrefresh}

Last updated: 2021-05-09

Path: miniprogram_gcash

onPullDownRefresh

2021-05-09 18:43

On the Page, customize the onPullDownRefresh function to listen to the pull-to-refresh event from the user.

- The `"pullRefresh": true` options needs to be configured in the `.json` configuration file of the related page to enable the pull-to-refresh event.
- when the data refresh is processed completely, call `my.stopPullDownRefresh` to stop the pull-to-refresh for that page.

Sample Code:

In the `pull-down-refresh.json` configuration file, the code configuration is as below:

copy

```
{
    "pullRefresh": true
}
```

In the Page, define the onPullDownRefresh processing function:

copy

```
onPullDownRefresh() {
    console.log('onPullDownRefresh', new Date())
}
```

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev-old/api_ui_pulldown_onpulldownrefresh

---

picker {#picker}

Last updated: 2022-07-03

Path: miniprogram_gcash

picker

2022-07-03 18:44

Scroll picker pop-up from bottom

Property	**Type**	**Default**	**Description**
range	String[] / Object[]	[]	For String[], indicating selectable string list; for Object[], specifying the range-key to indicate the selectable fields.
range-key	String		When the range is an Object[], the range-key is used to specify the key value in the Object as the picker displaying contents.
value	Number		Indicating which one is selected in the range (subscript starting from 0).
onChange	EventHandle		Trigger on value change, event.detail = {value: value}.
disabled	Boolean	false	Disable or not.

Screenshot

Sample Code

copy

```
<view class="section">
  <view class="section-title"> region picker</view>
  <picker onChange="bindPickerChange" value="{{index}}" range="
{{array}}">
    <view class="picker">
      Current selection{{array[index]}}
    </view>
  </picker>

  <picker onChange="bindObjPickerChange" value="{{arrIndex}}" range="
{{objectArray}}" range-key="name">
    <view class="row">
      <view class="row-title">ObjectArray</view>
      <view class="row-extra">Current selection:
{{objectArray[arrIndex].name}}</view>
      <image class="row-arrow" src="/image/arrowright.png"
mode="aspectFill" />
    </view>
  </picker>
</view>
```

copy

```
Page({
  data: {
    array: ['Country1', 'Country2', 'Country3', 'Country4'],
    objectArray: [\
      {\
        id: 0,\
        name: 'Country1',\
      },\
      {\
        id: 1,\
        name: 'Country2',\
      },\
      {\
        id: 2,\
        name: 'Country3',\
      },\
      {\
        id: 3,\
        name: 'Country4',\
      },\
    ],
    arrIndex: 0,
    index: 0
  },
  bindPickerChange(e) {
    console.log('picker sends selection change, carried value ',
e.detail.value);
    this.setData({
      index: e.detail.value,
    });
  },
  bindObjPickerChange(e) {
    console.log('picker sends selection change, carried value ',
e.detail.value);
    this.setData({
      arrIndex: e.detail.value,
    });
  },
});
```

Source:
https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev/component_form-component_picker

---

picker {#picker}

Last updated: 2021-05-09

Path: miniprogram_gcash

picker

2021-05-09 18:43

Scroll picker pop-up from bottom

Scan QR code to try:

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Default** | **Description** | | range | String[] / Object[] | [] | For String[], indicating selectable string list; for Object[], specifying the range-key to indicate the selectable fields. | | range-key | String | | When the range is an Object[], the range-key is used to specify the key value in the Object as the picker displaying contents. | | value | Number | | Indicating which one is selected in the range (subscript starting from 0). | | onChange | EventHandle | | Trigger on value change, event.detail = {value: value}. | | disabled | Boolean | false | Disable or not. |

Screenshot

Sample Code

copy

```
<view class="section">
  <view class="section-title"> region picker</view>
  <picker onChange="bindPickerChange" value="{{index}}" range="
{{array}}">
    <view class="picker">
      Current selection{{array[index]}}
    </view>
  </picker>

  <picker onChange="bindObjPickerChange" value="{{arrIndex}}" range="
{{objectArray}}" range-key="name">
    <view class="row">
      <view class="row-title">ObjectArray</view>
      <view class="row-extra">Current selection:
{{objectArray[arrIndex].name}}</view>
      <image class="row-arrow" src="/image/arrowright.png"
mode="aspectFill" />
    </view>
  </picker>
</view>
```

copy

```
Page({
  data: {
    array: ['Country1', 'Country2', 'Country3', 'Country4'],
    objectArray: [\
      {\
        id: 0,\
        name: 'Country1',\
      },\
      {\
        id: 1,\
        name: 'Country2',\
      },\
      {\
        id: 2,\
        name: 'Country3',\
      },\
      {\
        id: 3,\
        name: 'Country4',\
      },\
    ],
    arrIndex: 0,
    index: 0
  },
  bindPickerChange(e) {
    console.log('picker sends selection change, carried value ',
e.detail.value);
    this.setData({
      index: e.detail.value,
    });
  },
  bindObjPickerChange(e) {
    console.log('picker sends selection change, carried value ',
e.detail.value);
    this.setData({
      arrIndex: e.detail.value,
    });
  },
});
```

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev-old/component_form-component_picker

---

picker-view {#picker-view}

Last updated: 2022-07-03

Path: miniprogram_gcash

picker-view

2022-07-03 18:44

Scroll picker embedded in page.

| | | | | --- | --- | --- | | **Property** | **Type** | **Description** | | value | Number Array | The number indicates the index corresponding to the picker-view-column (starting from 0) . | | indicator-style | String | Selected box style. | | onChange | EventHandle | Trigger on scroll selection value change, event.detail = {value: Value}; value is an array, indicating the picker-view-column index in picker-view, starting from 0. |

> Note: Only component can be placed inside. The other nodes will not be displayed. Do not place the component in the hidden or display none node. For the hiding requirement, use a:if to switch.

Do not:

copy

```
<view hidden><picker-view/></view>
```

Recommend:

copy

```
<view a:if="{{xx}}"><picker-view/></view>
```

Screenshot

Sample Code

copy

```
<view class="pv-container">
  <view class="pv-left">
    <picker-view value="{{value}}" onChange="onChange">
      <picker-view-column>
        <view>2013</view>
        <view>2014</view>
      </picker-view-column>
      <picker-view-column>
        <view>Spring</view>
        <view>Summer</view>
      </picker-view-column>
    </picker-view>
  </view>
  <view class="pv-right">
    {{value}}
  </view>
</view>
```

copy

```
Page({
  data: {},
  onChange(e) {
    console.log(e.detail.value);
    this.setData({
      value: e.detail.value,
    });
  },
});
```

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev/component_form-component_picker-view

---

picker-view {#picker-view}

Last updated: 2021-05-09

Path: miniprogram_gcash

picker-view

2021-05-09 18:43

Scroll picker embedded in page.

Scan QR code to try:

| | | | | --- | --- | --- | | **Property** | **Type** | **Description** | | value | Number Array | The number indicates the index corresponding to the picker-view-column (starting from 0) . | | indicator-style | String | Selected box style. | | onChange | EventHandle | Trigger on scroll selection value change, event.detail = {value: Value}; value is an array, indicating the picker-view-column index in picker-view, starting from 0. |

> Note: Only component can be placed inside. The other nodes will not be displayed. Do not place the component in the hidden or display none node. For the hiding requirement, use a:if to switch.

Do not:

copy

```
<view hidden><picker-view/></view>
```

Recommend:

copy

```
<view a:if="{{xx}}"><picker-view/></view>
```

Screenshot

Sample Code

copy

```
<view class="pv-container">
  <view class="pv-left">
    <picker-view value="{{value}}" onChange="onChange">
      <picker-view-column>
        <view>2013</view>
        <view>2014</view>
      </picker-view-column>
      <picker-view-column>
        <view>Spring</view>
        <view>Summer</view>
      </picker-view-column>
    </picker-view>
  </view>
  <view class="pv-right">
    {{value}}
  </view>
</view>
```

copy

```
Page({
  data: {},
  onChange(e) {
    console.log(e.detail.value);
    this.setData({
      value: e.detail.value,
    });
  },
});
```

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev-old/component_form-component_picker-view

---

progress {#progress}

Last updated: 2022-07-03

Path: miniprogram_gcash

progress

2022-07-03 18:44

Progress bar.

Property	**Type**	**Default**	**Description**
percent	Float		Percentage (0~100).
show-info	Boolean	false	Show percentage value at right side.
stroke-width	Number	6	Line weight, in px.
active-color	Color	#09BB07	Selected progress bar color.
background-color	Color		Unselected progress bar color.
active	Boolean	false	Loading animation from left to right or not.

Screenshot

Sample Code

copy

```
<progress percent="20" show-info/>
<progress percent="40" active/>
```

```
<progress percent="60" stroke-width="10"/>
<progress percent="80" active/>
<progress percent="80" color="#10AEFF"/>
```

九色鹿

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev/component_basic-content_progress

---

progress {#progress}

Last updated: 2021-05-09

Path: miniprogram_gcash

progress

2021-05-09 18:43

Progress bar.

Scan QR code to try:

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Default** | **Description** | | percent | Float | | Percentage (0~100). | | show-info | Boolean | false | Show percentage value at right side. | | stroke-width | Number | 6 | Line weight, in px. | | active-color | Color | #09BB07 | Selected progress bar color. | | background-color | Color | | Unselected progress bar color. | | active | Boolean | false | Loading animation from left to right or not. |

Screenshot

Sample Code

copy

```
<progress percent="20" show-info/>
<progress percent="40" active/>
<progress percent="60" stroke-width="10"/>
<progress percent="80" active/>
<progress percent="80" color="#10AEFF"/>
```

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev-old/component_basic-content_progress

---

radio {#radio}

Last updated: 2022-07-03

Path: miniprogram_gcash

radio

2022-07-03 18:44

radio-group

Radio selector group.

Property	**Type**	**Description**
onChange	EventHandle	Trigger on change of selected item, event.detail = {value: Selected radio value}.
name	String	Component name, used for form submission of obtained data.

radio

Radio item

Property	**Type**	**Default**	**Description**
value	String		Component value, value carried in change event when selected.
checked	Boolean	false	Selected or not currently.
disabled	Boolean	false	Disable or not.
color	Color		Radio color.

Screenshot

Sample Code

copy

```
<radio-group class="radio-group" onChange="radioChange">
  <label class="radio" a:for="{{items}}">
    <radio value="{{item.name}}" checked="{{item.checked}}"/>
{{item.value}}
  </label>
</radio-group>
```

copy

```
Page({
  data: {
    items: [\
      {name: 'angular', value: 'AngularJS'},\
      {name: 'react', value: 'React', checked: true},\
      {name: 'polymer', value: 'Polymer'},\
      {name: 'vue', value: 'Vue.js'},\
      {name: 'ember', value: 'Ember.js'},\
      {name: 'backbone', value: 'Backbone.js'},\
    ]
  },
  radioChange: function(e) {
    console.log('you are selecting the framework:', e.detail.value)
  }
})
```

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev/component_form-component_radio

---

radio {#radio}

Last updated: 2021-05-09

Path: miniprogram_gcash

radio

2021-05-09 18:43

Scan QR code to try:

radio-group

Radio selector group.

Property	**Type**	**Description**
onChange	EventHandle	Trigger on change of selected item, event.detail = {value: Selected radio value}.
name	String	Component name, used for form submission of obtained data.

radio

Radio item

Property	**Type**	**Default**	**Description**
value	String		Component value, value carried in change event when selected.
checked	Boolean	false	Selected or not currently.
disabled	Boolean	false	Disable or not.
color	Color		Radio color.

Screenshot

Sample Code

copy

```
<radio-group class="radio-group" onChange="radioChange">
  <label class="radio" a:for="{{items}}">
    <radio value="{{item.name}}" checked="{{item.checked}}"/>
{{item.value}}
  </label>
</radio-group>
```

copy

```
Page({
  data: {
    items: [\
      {name: 'angular', value: 'AngularJS'},\
      {name: 'react', value: 'React', checked: true},\
      {name: 'polymer', value: 'Polymer'},\
      {name: 'vue', value: 'Vue.js'},\
      {name: 'ember', value: 'Ember.js'},\
      {name: 'backbone', value: 'Backbone.js'},\
    ]
  },
  radioChange: function(e) {
    console.log('you are selecting the framework:', e.detail.value)
  }
})
```

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev-old/component_form-component_radio

---

scroll-view {#scroll-view}

Last updated: 2021-05-09

Path: miniprogram_gcash

scroll-view

2021-05-09 18:43

Scroll view region

Scan QR code to try:

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Default** | **Description** | | class | String | | External style name. | | style | String | | Inline style name. | | scroll-x | Boolean | false | Allow horizontal scroll. | | scroll-y | Boolean | false | Allow vertical scroll. | | upper-threshold | Number | 50 | How far it is to top/left (in px) to trigger the scrolltoupper event. | | lower-threshold | Number | 50 | How far it is to bottom/right (in px) to trigger the scrolltolower event. | | scroll-top | Number | | Set location of virtical scroll bar. | | scroll-left | Number | | Set location of horizontal scroll bar. | | scroll-into-view | String | | Value is an element ID, scrolling to that element, element top aligning with scroll region top. | | onScrollToUpper | EventHandle | | Scroll to top/left triggers the scrolltoupper event. | | onScrollToLower | EventHandle | | Scroll to bottom/right triggers the scrolltolower event. | | onScroll | EventHandle | | Trigger on scroll, event.detail = {scrollLeft, scrollTop, scrollHeight, scrollWidth}. |

> When vertical scroll is used, a fixed height is required, which is set by using acss.

Sample Code

copy

```
<view class="page">
  <view class="page-description">Scroll view region</view>
  <view class="page-section">
    <view class="page-section-title">vertical scroll</view>
    <view class="page-section-demo">
      <scroll-view scroll-y="{{true}}" style="height: 200px;"
onScrollToUpper="upper" onScrollToLower="lower" onScroll="scroll"
scroll-into-view="{{toView}}" scroll-top="{{scrollTop}}">
        <view id="blue" class="scroll-view-item bc_blue"></view>
        <view id="red"  class="scroll-view-item bc_red"></view>
        <view id="yellow" class="scroll-view-item bc_yellow"></view>
        <view id="green" class="scroll-view-item bc_green"></view>
      </scroll-view>
    </view>
    <view class="page-section-btns">
      <view onTap="tap">next</view>
      <view onTap="tapMove">move</view>
      <view onTap="scrollToTop">scrollToTop</view>
    </view>
  </view>
  <view class="page-section">
    <view class="page-section-title">horizontal scroll</view>
    <view class="page-section-demo">
      <scroll-view class="scroll-view_H" scroll-x="{{true}}"
style="width: 100%" >
        <view id="blue2" class="scroll-view-item_H bc_blue"></view>
        <view id="red2"  class="scroll-view-item_H bc_red"></view>
        <view id="yellow2" class="scroll-view-item_H bc_yellow">
</view>
        <view id="green2" class="scroll-view-item_H bc_green"></view>
      </scroll-view>
    </view>
  </view>
</view>
```

copy

```
const order = ['blue', 'red', 'green', 'yellow'];
Page({
  data: {
    toView: 'red',
    scrollTop: 100,
  },
  upper(e) {
    console.log(e);
  },
  lower(e) {
    console.log(e);
  },
  scroll(e) {
    console.log(e.detail.scrollTop);
  },
  scrollToTop(e) {
    console.log(e);
    this.setData({
      scrollTop: 0,
    });
  },
});
```

Tips

- scroll-into-view has a higher priority than scroll-top
- Prevent page kickback in scroll scroll-view, so the scroll in scroll-view will not trigger onPullDownRefresh

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev-old/component_view-container_scroll-view

---

scroll-view {#scroll-view}

Last updated: 2022-07-03

Path: miniprogram_gcash

scroll-view

2022-07-03 18:44

Scroll view region

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Default** | **Description** | | class | String | | External style name. | | style | String | | Inline style name. | | scroll-x | Boolean | false | Allow horizontal scroll. | | scroll-y | Boolean | false | Allow vertical scroll. | | upper-threshold | Number | 50 | How far it is to top/left (in px) to trigger the scrolltoupper event. | | lower-threshold | Number | 50 | How far it is to bottom/right (in px) to trigger the scrolltolower event. | | scroll-top | Number | | Set location of virtical scroll bar. | | scroll-left | Number | | Set location of horizontal scroll bar. | | scroll-into-view | String | | Value is an element ID, scrolling to that element, element top aligning with scroll region top. | | onScrollToUpper | EventHandle | | Scroll to top/left triggers the scrolltoupper event. | | onScrollToLower | EventHandle | | Scroll to bottom/right triggers the scrolltolower event. | | onScroll | EventHandle | | Trigger on scroll, event.detail = {scrollLeft, scrollTop, scrollHeight, scrollWidth}. |

> When vertical scroll is used, a fixed height is required, which is set by using acss.

Sample Code

copy

```
<view class="page">
  <view class="page-description">Scroll view region</view>
  <view class="page-section">
    <view class="page-section-title">vertical scroll</view>
    <view class="page-section-demo">
      <scroll-view scroll-y="{{true}}" style="height: 200px;"
onScrollToUpper="upper" onScrollToLower="lower" onScroll="scroll"
scroll-into-view="{{toView}}" scroll-top="{{scrollTop}}">
        <view id="blue" class="scroll-view-item bc_blue"></view>
        <view id="red"  class="scroll-view-item bc_red"></view>
        <view id="yellow" class="scroll-view-item bc_yellow"></view>
        <view id="green" class="scroll-view-item bc_green"></view>
      </scroll-view>
    </view>
    <view class="page-section-btns">
      <view onTap="tap">next</view>
      <view onTap="tapMove">move</view>
      <view onTap="scrollToTop">scrollToTop</view>
    </view>
  </view>
  <view class="page-section">
    <view class="page-section-title">horizontal scroll</view>
    <view class="page-section-demo">
      <scroll-view class="scroll-view_H" scroll-x="{{true}}"
style="width: 100%" >
        <view id="blue2" class="scroll-view-item_H bc_blue"></view>
        <view id="red2"  class="scroll-view-item_H bc_red"></view>
        <view id="yellow2" class="scroll-view-item_H bc_yellow">
</view>
```

```
        <view id="green2" class="scroll-view-item_H bc_green"></view>
      </scroll-view>
    </view>
  </view>
</view>
```

copy

```
const order = ['blue', 'red', 'green', 'yellow'];
Page({
  data: {
    toView: 'red',
    scrollTop: 100,
  },
  upper(e) {
    console.log(e);
  },
  lower(e) {
    console.log(e);
  },
  scroll(e) {
    console.log(e.detail.scrollTop);
  },
  scrollToTop(e) {
    console.log(e);
    this.setData({
      scrollTop: 0,
    });
  },
});
```

Tips

- scroll-into-view has a higher priority than scroll-top
- Prevent page kickback in scroll scroll-view, so the scroll in scroll-view will not trigger onPullDownRefresh

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev/component_view-container_scroll-view

---

slider {#slider}

Last updated: 2021-05-09

Path: miniprogram_gcash

slider

2021-05-09 18:43

Slide selector

Scan QR code to try:

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Default** | **Description** | | name | String | | Component name, used for form submission to obtain data. | | min | Number | 0 | Minimum. | | max | Number | 100 | Maximum. | | step | Number | 1 | Step, must be greater than 0 and can be exactly divided by (max - min). | | disabled | Boolean | false | Disable or not. | | value | Number | 0 | Current value. | | show-value | Boolean | false | Show current value or not. | | active-color | String | #108ee9 | Selected color. | | background-color | String | #ddd | Background bar color. | | track-size | Number | 4 | Track line height. | | handle-size | Number | 22 | Slider size. | | handle-color | String | #fff | Slider fill color. | | onChange | EventHandle | | Trigger on completion of a drag, event.detail = {value: value}. | | onChanging | EventHandle | | Event triggered in process of drag, event.detail = {value: value}. |

Screenshot

Sample Code

copy

```
<view class="section section-gap">
  <text class="section-title">Set step</text>
  <view class="body-view">
    <slider value="60" onChange="sliderChange" step="5"/>
  </view>
</view>
<view class="section section-gap">
  <text class="section-title">Shown current value</text>
  <view class="body-view">
    <slider value="50" show-value/>
  </view>
</view>
<view class="section section-gap">
  <text class="section-title">Set mininum/maximum</text>
  <view class="body-view">
    <slider value="100" min="50" max="200" show-value/>
  </view>
</view>
<view class="page-section">
  <view class="page-section-title">Custom style</view>
  <view class="page-section-demo">
    <slider value="33" onChange="slider4change" min="25" max="50"
show-value
    backgroundColor="#FFAA00" activeColor="#00aaee" trackSize="2"
```

```
handleSize="6" handleColor="blue" />
  </view>
</view>
```

copy

```
Page({
  sliderChange(e)
    console.log('slider changed value:', e.detail.value)
})
```

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev-old/component_form-component_slider

---

slider {#slider}

Last updated: 2022-07-03

Path: miniprogram_gcash

slider

2022-07-03 18:44

Slide selector

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Default** | **Description** | | name | String | | Component name, used for form submission to obtain data. | | min | Number | 0 | Minimum. | | max | Number | 100 | Maximum. | | step | Number | 1 | Step, must be greater than 0 and can be exactly divided by (max - min). | | disabled | Boolean | false | Disable or not. | | value | Number | 0 | Current value. | | show-value | Boolean | false | Show current value or not. | | active-color | String | #108ee9 | Selected color. | | background-color | String | #ddd | Background bar color. | | track-size | Number | 4 | Track line height. | | handle-size | Number | 22 | Slider size. | | handle-color | String | #fff | Slider fill color. | | onChange | EventHandle | | Trigger on completion of a drag, event.detail = {value: value}. | | onChanging | EventHandle | | Event triggered in process of drag, event.detail = {value: value}. |

Screenshot

Sample Code

copy

```
<view class="section section-gap">
  <text class="section-title">Set step</text>
  <view class="body-view">
    <slider value="60" onChange="sliderChange" step="5"/>
  </view>
</view>
<view class="section section-gap">
  <text class="section-title">Shown current value</text>
  <view class="body-view">
    <slider value="50" show-value/>
  </view>
</view>
<view class="section section-gap">
  <text class="section-title">Set mininum/maximum</text>
  <view class="body-view">
    <slider value="100" min="50" max="200" show-value/>
  </view>
</view>
<view class="page-section">
  <view class="page-section-title">Custom style</view>
  <view class="page-section-demo">
    <slider value="33" onChange="slider4change" min="25" max="50"
show-value
    backgroundColor="#FFAA00" activeColor="#00aaee" trackSize="2"
handleSize="6" handleColor="blue" />
  </view>
</view>
```

copy

```
Page({
  sliderChange(e)
    console.log('slider changed value:', e.detail.value)
})
```

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev/component_form-component_slider

---

swiper {#swiper}

Last updated: 2022-07-03

Path: miniprogram_gcash

swiper

2022-07-03 18:44

Swiper view container

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Default** | **Description** | | indicator-dots | Boolean | false | Show indicator or not. | | indicator-color | Color | rgba(0, 0, 0, .3) | Indicator color. | | indicator-active-color | Color | #000 | Color of currently selected indicator. | | autoplay | Boolean | false | Auto switch or not. | | current | Number | 0 | Current page index. | | duration | Number | 500(ms) | Swipe animation duration. | | interval | Number | 5000(ms) | Auto switch interval. | | circular | Boolean | false | Enable infinite swipe or not. | | vertical | Boolean | false | Is swipe direction vertical or not. | | onChange | EventHandle | No | Trigger on current change, event.detail = {current, current}. |

Swiper-item

Can place in component or not; width and height are automatically set as 100%.

Sceenshot

Sample Code

copy

```
<swiper
  indicator-dots="{{indicatorDots}}"
  autoplay="{{autoplay}}"
  interval="{{interval}}"
>
  <block a:for="{{background}}">
    <swiper-item>
      <view class="swiper-item bc_{{item}}"></view>
    </swiper-item>
  </block>
</swiper>
<view class="btn-area">
  <button class="btn-area-button" type="default"
onTap="changeIndicatorDots">indicator-dots</button>
  <button class="btn-area-button" type="default"
onTap="changeAutoplay">autoplay</button>
</view>
<slider onChange="intervalChange" value="{{interval}}" show-value
min="2000" max="10000"/>
<view class="section__title">interval</view>
```

copy

```
Page({
  data: {
    background: ['green', 'red', 'yellow'],
    indicatorDots: true,
    autoplay: false,
    interval: 3000,
  },
  changeIndicatorDots(e) {
    this.setData({
      indicatorDots: !this.data.indicatorDots
    })
  },
  changeAutoplay(e) {
    this.setData({
      autoplay: !this.data.autoplay
    })
  },
  intervalChange(e) {
    this.setData({
      interval: e.detail.value
    })
  },
})
```

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev/component_view-container_swiper

---

swiper {#swiper}

Last updated: 2021-05-10

Path: miniprogram_gcash

swiper

2021-05-10 03:43

Swiper view container

Scan QR code to try:

Property	**Type**	**Default**	**Description**
indicator-dots	Boolean	false	Show indicator or not.
indicator-color	Color	rgba(0, 0, 0, .3)	Indicator color.
indicator-active-color	Color	#000	Color of currently selected indicator.
autoplay	Boolean	false	Auto switch or not.
current	Number	0	Current page index.
duration	Number	500(ms)	Swipe animation duration.
interval	Number	5000(ms)	Auto switch interval.
circular	Boolean	false	Enable infinite swipe or not.
vertical	Boolean	false	Is swipe direction vertical or not.
onChange	EventHandle	No	Trigger on current change, event.detail = {current, current}.

Swiper-item

Can place in component or not; width and height are automatically set as 100%.

Sceenshot

Sample Code

copy

```
<swiper
  indicator-dots="{{indicatorDots}}"
  autoplay="{{autoplay}}"
  interval="{{interval}}"
>
  <block a:for="{{background}}">
    <swiper-item>
      <view class="swiper-item bc_{{item}}"></view>
    </swiper-item>
  </block>
</swiper>
<view class="btn-area">
  <button class="btn-area-button" type="default"
onTap="changeIndicatorDots">indicator-dots</button>
  <button class="btn-area-button" type="default"
onTap="changeAutoplay">autoplay</button>
</view>
<slider onChange="intervalChange" value="{{interval}}" show-value
min="2000" max="10000"/>
<view class="section__title">interval</view>
```

copy

```
Page({
  data: {
    background: ['green', 'red', 'yellow'],
    indicatorDots: true,
    autoplay: false,
    interval: 3000,
  },
  changeIndicatorDots(e) {
    this.setData({
      indicatorDots: !this.data.indicatorDots
    })
  },
  changeAutoplay(e) {
    this.setData({
      autoplay: !this.data.autoplay
    })
  },
  intervalChange(e) {
    this.setData({
      interval: e.detail.value
    })
  },
})
```

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev-old/component_view-container_swiper

---

switch {#switch}

Last updated: 2021-05-09

Path: miniprogram_gcash

switch

2021-05-09 18:43

Switch.

Scan QR code to try:

| | | | | --- | --- | --- | | **Property** | **Type** | **Description** | | name | String | Component name, used for form submission to get data. | | checked | Boolean | Checked or not. | | disabled | Boolean | Disable or not. | | color | String | Component color. | | onChange | EventHandle | Trigger on checked change, event.detail={ value:checked}. | | color | Color | Switch color. |

Screenshot

Sample Code

copy

```
<view class="page">
  <view class="switch-list">
    <view class="switch-item">
      <switch checked onChange="switchChange"/>
    </view>
  </view>
</view>
```

copy

```
Page({
  switchChange (e){
    console.log('switchChange event, value:', e.detail.value)
  },
})
```

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev-old/component_form-component_switch

---

switch {#switch}

Last updated: 2022-07-04

Path: miniprogram_gcash

switch

2022-07-04 03:44

Switch.

| | | | | --- | --- | --- | | **Property** | **Type** | **Description** | | name | String | Component name, used for form submission to get data. | | checked | Boolean | Checked or not. | | disabled | Boolean | Disable or not. | | color | String | Component color. | | onChange | EventHandle | Trigger on checked change, event.detail={ value:checked}. | | color | Color | Switch color. |

Screenshot

Sample Code

copy

```
<view class="page">
  <view class="switch-list">
    <view class="switch-item">
      <switch checked onChange="switchChange"/>
    </view>
  </view>
</view>
```

copy

```
Page({
  switchChange (e){
    console.log('switchChange event, value:', e.detail.value)
  },
})
```

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev/component_form-component_switch

---

text {#text}

Last updated: 2022-07-03

Path: miniprogram_gcash

text

2022-07-03 18:44

Text, supporting only nesting within component.

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Default** | **Description** | | selectable | Boolean | false | Selectable or not. | | space | String | | Show continuous spaces. | | decode | Boolean | false | Decode or not. | | number-of-lines | number | | Multi-line omission, value must be greater than or equal to 1, same behavior as the -webkit-line-clamp attribute of css. |

Space Effective Value:

| | | | --- | --- | | **Value** | **Description** | | nbsp | Space size as per font setting. | | ensp | Half-size space of character. | | emsp | Space size of character. |

Sample Code

copy

```
<view class="page">
  <view class="text-view">
    <text>{{text}}</text>
  </view>
</view>
```

copy

```
Page({
  data: {
    text: 'Hello Mini Program',
  },
})
```

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev/component_basic-content_text

---

text {#text}

Last updated: 2021-05-09

Path: miniprogram_gcash

text

2021-05-09 18:43

Text, supporting only nesting within component.

Scan QR code to try:

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Default** | **Description** | | selectable | Boolean | false | Selectable or not. | | space | String | | Show continuous spaces. | | decode | Boolean | false | Decode or not. | | number-of-lines | number | | Multi-line omission, value must be greater than or equal to 1, same behavior as the -webkit-line-clamp attribute of css. |

Space Effective Value:

| | | | --- | --- | | **Value** | **Description** | | nbsp | Space size as per font setting. | | ensp | Half-size space of character. | | emsp | Space size of character. |

Sample Code

copy

```
<view class="page">
  <view class="text-view">
    <text>{{text}}</text>
  </view>
</view>
```

copy

```
Page({
  data: {
    text: 'Hello Mini Program',
  },
})
```

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev-old/component_basic-content_text

---

textarea {#textarea}

Last updated: 2021-05-09

Path: miniprogram_gcash

textarea

2021-05-09 18:43

Multi-row entry box

Scan QR code to try:

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Default** | **Description** | | name | String | | Component name, used for form submission to get data. | | value | String | | Initial contents. | | placeholder | String | | Placeholder. | | placeholder-style | String | | Specify placeholder style. | | placeholder-class | String | | Specify placeholder style class. | | disabled | Boolean | false | Disable or not. | | maxlength | Number | 140 | Maximum length, no length limit when setting is -1. | | focus | Boolean | false | Get focus. | | auto-height | Boolean | false | Use auto height or not. | | show-count | Boolean | true | Render wordcount statistics or not. | | controlled | Boolean | false | Is controlled component or not When it is true, the value content is fulled controlled by setData. | | onInput | EventHandle | | Trigger on keyboard entry, event.detail = {value: value}, can return directly a string to replace the contents in the entry box. | | onFocus | EventHandle | | Trigger on entry box getting focus, event.detail = {value: value}. | | onBlur | EventHandle | | Rigger on entry box losing focus, event.detail = {value: value}. | | onConfirm | EventHandle | | Rrigger on clicking completion, event.detail = {value: value}. |

Screenshot

Sample Code

copy

```
<view class="section">
  <textarea onBlur="bindTextAreaBlur" auto-height placeholder="Auto height" />
</view>
<view class="section">
  <textarea placeholder="Get focus only when this button is clicked" focus="{{focus}}" />
  <view class="btn-area">
    <button onTap="bindButtonTap">Have entry box get focus</button>
  </view>
</view>
<view class="section">
  <form onSubmit="bindFormSubmit">
    <textarea placeholder="textarea in the form" name="textarea"/>
    <button form-type="submit"> Submit </button>
  </form>
</view>
```

copy

```
Page({
  data: {
    focus: false,
    inputValue: ''
  },
  bindButtonTap() {
    this.setData({
      focus: true
    })
  },
  bindTextAreaBlur: function(e) {
    console.log(e.detail.value)
  },
  bindFormSubmit: function(e) {
    console.log(e.detail.value.textarea)
  }
})
```

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev-old/component_form-component_textarea

---

textarea {#textarea}

Last updated: 2022-07-03

Path: miniprogram_gcash

textarea

2022-07-03 18:44

Multi-row entry box

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Default** | **Description** | | name | String | | Component name, used for form submission to get data. | | value | String | | Initial contents. | | placeholder | String | | Placeholder. | | placeholder-style | String | | Specify placeholder style. | | placeholder-class | String | | Specify placeholder style class. | | disabled | Boolean | false | Disable or not. | | maxlength | Number | 140 | Maximum length, no length limit when setting is -1. | | focus | Boolean | false | Get focus. | | auto-height | Boolean | false | Use auto height or not. | | show-count | Boolean | true | Render wordcount statistics or not. | | controlled | Boolean | false | Is controlled component or not When it is true, the value content is fulled controlled by setData. | | onInput | EventHandle | | Trigger on keyboard entry, event.detail = {value: value}, can return directly a string to replace the contents in the entry box. | | onFocus | EventHandle | | Trigger on entry box getting focus, event.detail = {value: value}. | | onBlur | EventHandle | | Rigger on entry box losing focus, event.detail = {value: value}. | | onConfirm | EventHandle | | Rrigger on clicking completion, event.detail = {value: value}. |

Screenshot

Sample Code

copy

```
<view class="section">
  <textarea onBlur="bindTextAreaBlur" auto-height placeholder="Auto
height" />
</view>
<view class="section">
  <textarea placeholder="Get focus only when this button is clicked"
focus="{{focus}}" />
  <view class="btn-area">
    <button onTap="bindButtonTap">Have entry box get focus</button>
  </view>
</view>
<view class="section">
  <form onSubmit="bindFormSubmit">
    <textarea placeholder="textarea in the form" name="textarea"/>
    <button form-type="submit"> Submit </button>
  </form>
</view>
```

copy

```
Page({
  data: {
    focus: false,
    inputValue: ''
  },
  bindButtonTap() {
    this.setData({
      focus: true
    })
  },
  bindTextAreaBlur: function(e) {
    console.log(e.detail.value)
  },
  bindFormSubmit: function(e) {
    console.log(e.detail.value.textarea)
  }
})
```

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev/component_form-component_textarea

view {#view}

Last updated: 2022-07-03

Path: miniprogram_gcash

view

2022-07-03 18:44

View container Equivalent with the div of web or the view of react-native.

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Default** | **Description** | | disable-scroll | Boolean | false | Forbid scroll page within region. | | hover-class | String | | Style class to be added upon clicking. | | hover-start-time | Number | | Start the clicking status after a period of holding, in millisecond. | | hover-stay-time | Number | | Hold the clicking status after release, in millisecond. | | hidden | boolean | false | Hide or not. | | class | String | | Custom style name. | | style | String | | Inline style. | | onTap | EventHandle | | Click. | | onTouchStart | EventHandle | | Start of touch action. | | onTouchMove | EventHandle | | Move after touch. | | onTouchEnd | EventHandle | | End of touch action. | | onTouchCancel | EventHandle | | Touch action interrupted, such as incoming call and pop-up. | | onLongTap | EventHandle | | Trigger on 500ms-duration pressing. After long-press event, moving does not trigger screen scroll. |

> Note: The animation generated with my.createAnimation is realized via transition and triggers only onTransitionEnd, but does not trigger onAnimationStart, onAnimationIteration or onAnimationEnd.

Sample Code

copy

```
<view class="post">
  <!-- hidden -->
  <view class="postUser" hidden>
    <view class="postUser__name">Jessie</view>
  </view>
  <!-- hover class -->
  <view class="postBody" hover-class="red">
    <view class="postBody__content">
      Good!
    </view>
    <view class="postBody__date">
      June 1
    </view>
  </view>
</view>
```

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev/component_view-container_view

---

view {#view}

Last updated: 2021-05-09

Path: miniprogram_gcash

view

2021-05-09 18:43

View container Equivalent with the div of web or the view of react-native.

Scan QR code to try:

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Default** | **Description** | | disable-scroll | Boolean | false | Forbid scroll page within region. | | hover-class | String | | Style class to be added upon clicking. | | hover-start-time | Number | | Start the clicking status after a period of holding, in millisecond. | | hover-stay-time | Number | | Hold the clicking status after release, in millisecond. | | hidden | boolean | false | Hide or not. | | class | String | | Custom style name. | | style | String | | Inline style. | | onTap | EventHandle | | Click. | | onTouchStart | EventHandle | | Start of touch action. | | onTouchMove | EventHandle | | Move after touch. | | onTouchEnd | EventHandle | | End of touch action. | | onTouchCancel | EventHandle | | Touch action interrupted, such as incoming call and pop-up. | | onLongTap | EventHandle | | Trigger on 500ms-duration pressing. After long-press event, moving does not trigger screen scroll. |

> Note: The animation generated with my.createAnimation is realized via transition and triggers only onTransitionEnd, but does not trigger onAnimationStart, onAnimationIteration or onAnimationEnd.

Sample Code

copy

```
<view class="post">
  <!-- hidden -->
  <view class="postUser" hidden>
    <view class="postUser__name">Jessie</view>
  </view>
  <!-- hover class -->
  <view class="postBody" hover-class="red">
    <view class="postBody__content">
      Good!
    </view>
    <view class="postBody__date">
      June 1
    </view>
  </view>
</view>
```

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev-old/component_view-container_view

---

web-view {#web-view}

Last updated: 2022-07-03

Path: miniprogram_gcash

web-view

2022-07-03 18:44

Webview

> This component does not support personal Mini Program temporarily.

`<web-view/>` The component is a component that is used to carry H5 webpage and automatically bespreads the whole Mini Program page.

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Default** | **Description** | | src | String | No | H5 webpage URL to be rendered in web-view H5 webpage URL needs login Mini Program management background- In Mini Program details - Setting, configure H5 domain whitelist. | | onMessage | EventHandle | No | postMessage message from webpage to Mini Program e.detail = { data }. |

> Each page can have only one . Do not render multiple . It bespreads whole page and overlaps other components.

Sample Code

copy

```
<!-- axml -->
<!-- web-view pointing to google -->
<web-view src="https://google.com/" onMessage="test"></web-view>
```

APIs Available

| | | | | --- | --- | --- | | **API Type** | **Name** | **Description** | | Navigation | my.navigateTo | Navigate to another page of the app while keeping current one. | | Navigation | my.navigateBack | Close current page and return to one of the pages before. | | Navigation

| my.switchTab | Navigate to a page on tabBar and close any other pages that are not on the tabBar. | | Navigation | my.reLaunch | Close all pages and navigate to a page from the app. | | Navigation | my.redirectTo | Close current page and navigate to a page from the app. | | Image | my.chooseImage | Take a photo or choose one from the album. (The file path of the obtained photo can be sent to the current mini program via `my.postMessage()`, and then be uploaded if needed.) | | Image | my.previewImage | Preview the image. | | Location | my.getLocation | Get location information of current user. | | Popups | my.alert | Show alert window. | | Popups | my.showLoading | Show loading indicator. | | Popups | my.hideLoading | Hide loading indicator. | | Storage | my.setStorage | Store some data in local storage with a key. Will overwrite if the key already exists. | | Storage | my.getStorage | Retrieve stored data. | | Storage | my.removeStorage | Delete stored data. | | Storage | my.clearStorage | Clear local storage. | | Storage | my.getStorageInfo | Get information about local storage asynchronously. | | Network | my.getNetworkType | Get information about current network status. | | Payment | my.tradePay | Invoke payment procedures. (Don't process payment in H5 environment, and always call this API for payments.) | | Message | my.postMessage | Send message to the current mini program, in JSON format. | | Message | my.onMessage | Listen to messages from the current mini program. | | Environment | my.getEnv | Get information about current environment. |

Demo Code

Code for `web-view` page for H5.

copy

```
<script type="text/javascript" src="https://appx/web-view.min.js">
</script>
<script>
  my.navigateTo({url: '../get-user-info/get-user-info'});

  // Send message to Mini Program.
  my.postMessage({name:"test web-view"});

  // Did receive message from Mini Program.
  my.onMessage = function(e) {
    console.log(e); // {'sendToWebView': '1'}
  }
  // Check if is run in Mini Program environment
  my.getEnv(function(res) {
    console.log(res.miniprogram) // true
  });
</script>
```

After `my.postMessage` is called, `onMessage` will be executed in this mini program page.

copy

```
<!-- .axml -->
<view>
  <web-view id="web-view-1" src="..." onMessage="test"></web-view>
</view>
```

copy

```
// A 'test' method is declared in this page.
// As web-view in page.axml has set 'test' for the call of
'onMessage',
// after my.postMessage is executed in the web-view, test will be
called.
Page({
  onLoad(e){
    this.webViewContext = my.createWebViewContext('web-view-1');
  },
  test(e){
    my.alert({
      content:JSON.stringify(e.detail),
    });
    this.webViewContext.postMessage({'sendToWebView': '1'});
  },
});
```

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev/component_open_web-view

---

web-view {#web-view}

Last updated: 2021-05-09

Path: miniprogram_gcash

web-view

2021-05-09 18:43

Webview

Scan QR code to try:

> This component does not support personal Mini Program temporarily.

`<web-view/>` The component is a component that is used to carry H5 webpage and automatically bespreads the whole Mini Program page.

| | | | | | --- | --- | --- | --- | | **Property** | **Type** | **Default** | **Description** | | src | String | No | H5 webpage URL to be rendered in web-view H5 webpage URL needs login Mini Program management background- In Mini Program details - Setting, configure H5 domain whitelist. | | onMessage | EventHandle | No | postMessage message from webpage to Mini Program e.detail = { data }. |

> Each page can have only one . Do not render multiple . It bespreads whole page and overlaps other components.

Sample Code

copy

```
<!-- axml -->
<!-- web-view pointing to google -->
<web-view src="https://google.com/" onMessage="test"></web-view>
```

APIs Available

| | | | | --- | --- | --- | | **API Type** | **Name** | **Description** | | Navigation | my.navigateTo | Navigate to another page of the app while keeping current one. | | Navigation | my.navigateBack | Close current page and return to one of the pages before. | | Navigation | my.switchTab | Navigate to a page on tabBar and close any other pages that are not on the tabBar. | | Navigation | my.reLaunch | Close all pages and navigate to a page from the app. | | Navigation | my.redirectTo | Close current page and navigate to a page from the app. | | Image | my.chooseImage | Take a photo or choose one from the album. (The file path of the obtained photo can be sent to the current mini program via `my.postMessage()`, and then be uploaded if needed.) | | Image | my.previewImage | Preview the image. | | Location | my.getLocation | Get location information of current user. | | Popups | my.alert | Show alert window. | | Popups | my.showLoading | Show loading indicator. | | Popups | my.hideLoading | Hide loading indicator. | | Storage | my.setStorage | Store some data in local storage with a key. Will overwrite if the key already exists. | | Storage | my.getStorage | Retrieve stored data. | | Storage | my.removeStorage | Delete stored data. | | Storage | my.clearStorage | Clear local storage. | | Storage | my.getStorageInfo | Get information about local storage asynchronously. | | Network | my.getNetworkType | Get information about current network status. | | Payment | my.tradePay | Invoke payment procedures. (Don't process payment in H5 environment, and always call this API for payments.) | | Message | my.postMessage | Send message to the current mini program, in JSON format. | | Message | my.onMessage | Listen to messages from the current mini program. | | Environment | my.getEnv | Get information about current environment. |

Demo Code

Code for `web-view` page for H5.

copy

```
<script type="text/javascript" src="https://appx/web-view.min.js">
</script>
<script>
  my.navigateTo({url: '../get-user-info/get-user-info'});

  // Send message to Mini Program.
  my.postMessage({name:"test web-view"});

  // Did receive message from Mini Program.
  my.onMessage = function(e) {
    console.log(e); // {'sendToWebView': '1'}
  }
  // Check if is run in Mini Program environment
  my.getEnv(function(res) {
    console.log(res.miniprogram) // true
  });
</script>
```

After `my.postMessage` is called, `onMessage` will be executed in this mini program page.

copy

```
<!-- .axml -->
<view>
  <web-view id="web-view-1" src="..." onMessage="test"></web-view>
</view>
```

copy

```
// A 'test' method is declared in this page.
// As web-view in page.axml has set 'test' for the call of
'onMessage',
// after my.postMessage is executed in the web-view, test will be
called.
Page({
  onLoad(e){
    this.webViewContext = my.createWebViewContext('web-view-1');
  },
  test(e){
    my.alert({
      content:JSON.stringify(e.detail),
    });
    this.webViewContext.postMessage({'sendToWebView': '1'});
  },
});
```

九色鹿

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev-old/component_open_web-view

---

xxpbkg {#xxpbkg}

Path: miniprogram_gcash

404 Not Found

Sorry, the page you visited does not exist.

traceId: 21b85ba617474850310774677eb95e

Go Back

Source: https://miniprogram.gcash.com/docs/miniprogram_gcash/mpdev/xxpbkg

---