

uniapp 安卓webview操作UTS原生组件

目 录

使用说明	1
使用方法	1.1
更新日志	1.2
插件方法	2
设置cookie	2.1
获取cookie	2.2
删除cookie	2.3
加载网页	2.4
获取网页属性	2.5
刷新页面	2.6
停止加载	2.7
是否能够前进	2.8
是否能够后退	2.9
后退	2.10
前进	2.11
执行JavaScript	2.12
清除缓存	2.13
获取LocalStorage v1.1.0	2.14
插件属性	3
config	3.1
插件事件	4
错误事件	4.1
开始加载网页	4.2
加载网页完成	4.3
网页资源加载	4.4
网页加载进度	4.5

使用方法

介绍

安卓WebView操作UTS原生插件，支持获取、设置、移除cookie，重载url，支持设置userAgent等，插件UTS开发，支持uniapp x

插件只能在nvue或uvue中使用

联系作者

关注微信公众号可联系作者



插件地址

<https://ext.dcloud.net.cn/plugin?id=20253>

用法

在需要使用插件的页面加载以下代码

```
<leven-uts-webview ref="refLevenUtsWebview" style="flex:1; height: 400px;" :config="{
  @onPageFinished:"onPageFinished" @onLoadResource:"onLoadResource" @onProgress:
}</leven-uts-webview>
```

页面内容

```

<template>
  <view>
    <uni-card title="安卓webview操作UTS原生插件">
      <leven-uts-webview ref="refLevenUtsWebview" style="flex:1; height: 400px;"
        @onPageFinished="onPageFinished" @onLoadResource="onLoadResource" @onProg
      </leven-uts-webview>
    </uni-card>
    <uni-card title="操作">
      <button type="primary" @click="setCookie">设置cookie</button>
      <button type="primary" @click="getCookie">获取cookie</button>
      <button type="primary" @click="removeCookie">删除cookie</button>
      <button type="primary" @click="loadUrl">加载网页</button>
      <button type="primary" @click="getWebData">获取网页属性</button>
      <button type="primary" @click="reload">刷新页面</button>
      <button type="primary" @click="stopLoading">停止加载</button>
      <button type="primary" @click="canGoForward">是否能够前进</button>
      <button type="primary" @click="canGoBack">是否能够后退</button>
      <button type="primary" @click="goBack">后退</button>
      <button type="primary" @click="goForward">前进</button>
      <button type="primary" @click="evaluateJavascript">执行JavaScript</button>
      <button type="primary" @click="clearCache">清除缓存</button>
      <button type="primary" @click="logStr = ''">清空日志</button>
    </uni-card>
    <uni-card title="url设置">
      <uni-section title="网页地址">
        <uni-easyinput v-model="config.url" :maxlength="-1" placeholder="请输入网页
      </uni-section>
    </uni-card>
    <uni-card title="cookie设置">
      <uni-section title="cookie名称">
        <uni-easyinput v-model="form.name" :maxlength="-1" placeholder="请输入cook
      </uni-section>
      <uni-section title="cookie值">
        <uni-easyinput v-model="form.value" :maxlength="-1" placeholder="请输入cool
      </uni-section>
      <uni-section title="存活时间">
        <uni-easyinput v-model="form.maxAge" :maxlength="-1" placeholder="请输入存活
      </uni-section>
      <uni-section title="域名">
        <uni-easyinput v-model="form.domain" :maxlength="-1" placeholder="请输入域
      </uni-section>
      <uni-section title="路径">
        <uni-easyinput v-model="form.path" :maxlength="-1" placeholder="请输入路径"
      </uni-section>
    </uni-card>
    <uni-card class="uni-card-box" title="日志">

```

```

    <view><text style="font-size: 14px; flex-wrap: wrap;">{{logStr}}</text></view>
  </uni-card>
</view>
</template>

<script>
  export default {
    data() {
      return {
        //webview配置
        config: {
          //默认加载的url
          url: "https://www.baidu.com",
          //设置JavaScript是否可用，默认：true
          javaScriptEnabled: true,
          //媒体播放是否需要用户主动操作，默认：true
          mediaPlaybackRequiresUserGesture: true,
          //设置浏览器标识，默认：空
          userAgent: "",
          //浏览器缓存模式：default.默认处理方式，noCache:不走缓存，直接从网络获取，cacheOnDisk:走缓存
          cacheMode: "noCache"
        },
        //表单内容
        form: {
          //cookie名称
          name: "c1",
          //cookie的值
          value: "1",
          //存活时间
          maxAge: 86400,
          //域名
          domain: "baidu.com",
          //路径
          path: "/"
        },
        //日志
        logStr: ""
      }
    },
    onLoad() {

    },
    methods: {
      //清除缓存
      clearCache() {
        this.$refs.refLevenUtsWebview.clearCache(res => {
          console.log("clearCache:" + JSON.stringify(res))
        })
      },
      //执行JavaScript

```

```

evaluateJavascript() {
  this.$refs.refLevenUtsWebview.evaluateJavascript({
    script: "javascript:alert('Hello from JavaScript!');"
  }, res => {
    console.log("evaluateJavascript:" + JSON.stringify(res))
  })
},
//前进
goForward() {
  this.$refs.refLevenUtsWebview.goForward(res => {
    console.log("goForward:" + JSON.stringify(res))
  })
},
//后退
goBack() {
  this.$refs.refLevenUtsWebview.goBack(res => {
    console.log("goBack:" + JSON.stringify(res))
  })
},
//是否能够后退
canGoBack() {
  this.$refs.refLevenUtsWebview.canGoBack(res => {
    console.log("canGoBack:" + JSON.stringify(res))
  })
},
//是否能够前进
canGoForward() {
  this.$refs.refLevenUtsWebview.canGoForward(res => {
    console.log("canGoForward:" + JSON.stringify(res))
  })
},
//停止加载
stopLoading() {
  this.$refs.refLevenUtsWebview.stopLoading(res => {
    console.log("stopLoading:" + JSON.stringify(res))
  })
},
//刷新页面
reload() {
  this.$refs.refLevenUtsWebview.reload(res => {
    console.log("reload:" + JSON.stringify(res))
  })
},
//获取当前网页属性
getWebData() {
  this.$refs.refLevenUtsWebview.getWebData(res => {
    console.log("getWebData:" + JSON.stringify(res))
  })
},
//加载网页

```

```

loadUrl() {
  this.$refs.refLevenUtsWebview.loadUrl({
    //加载的url
    url: this.config.url,
    //设置的headers
    headers: {
      token: "123"
    }
  }, res => {
    console.log("loadUrl:" + JSON.stringify(res))
  })
},
//删除cookie
removeCookie() {
  if (this.$refs.refLevenUtsWebview) {
    this.$refs.refLevenUtsWebview.removeCookie({
      //名称, 必须
      name: this.form.name,
      //域名, 必须
      domain: this.form.domain,
      //路径, 默认: /
      path: this.form.path
    }, res => {
      console.log("removeCookie:" + JSON.stringify(res))
    })
  }
},
//获取cookie
getCookie() {
  if (this.$refs.refLevenUtsWebview) {
    this.$refs.refLevenUtsWebview.getCookie({
      //名称, 为空则获取域名下所有的cookie
      name: this.form.name,
      //域名, 为空则获取当前网页下的所有cookie
      domain: this.form.domain,
      //路径
      path: this.form.path
    }, res => {
      console.log("获取cookie:" + JSON.stringify(res))
    })
  }
},
//设置cookie
setCookie() {
  if (this.$refs.refLevenUtsWebview) {
    this.$refs.refLevenUtsWebview.setCookie({
      //名称, 必须
      name: this.form.name,
      //值, 必须
      value: this.form.value,

```

```
        //域名,必须
        domain: this.form.domain,
        //存活时间,单位:秒,默认:86400
        maxAge: this.form.maxAge,
        //路径,默认:/
        path: this.form.path
    }, res => {
        console.log("设置cookie:" + JSON.stringify(res))
    })
}
},
//网页加载进度
onProgressChanged(e) {
    console.log("onProgressChanged:" + JSON.stringify(e.detail))
},
//网页资源加载
onLoadResource(e) {
    console.log("onLoadResource:" + JSON.stringify(e.detail))
},
//结束加载网页
onPageFinished(e) {
    console.log("onPageFinished:" + JSON.stringify(e.detail))
},
//开始加载网页
onPageStarted(e) {
    console.log("onPageStarted:" + JSON.stringify(e.detail))
},
//错误事件
onError(e) {
    console.log("onError:" + JSON.stringify(e.detail))
}
}
}
</script>

<style>
</style>
```

更新日志

2025-01-21

- [优化] 修复异常闪退的bug
- [新增] 新增接口【[获取LocalStorage](#)】

2024-09-15

首次发布

设置cookie

方法名

setCookie

用法

- 用法如下：

```
this.$refs.refLevenUtsWebview.setCookie({
  //名称，必须
  name: this.form.name,
  //值，必须
  value: this.form.value,
  //域名，必须
  domain: this.form.domain,
  //存活时间，单位：秒，默认：86400
  maxAge: this.form.maxAge,
  //路径，默认：/
  path: this.form.path
}, res => {
  console.log("设置cookie：" + JSON.stringify(res))
})
```

- 参数说明

参数名	参数类型	是否必填	默认值	参数描述
name	String	是	无	cookie名称
value	String	是	无	cookie的值
domain	String	是	无	域名
maxAge	Integer	否	86400	cookie存活时间，单位：秒
path	String	否	/	域名路径

回调

- 示例

设置cookie

```
{  
    "message": "",  
    "code": 0,  
    "data": {}  
}
```

- 回调说明：

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
code	Integer	返回类型，0.成功，其他：失败

获取cookie

方法名

getCookie

用法

- 用法如下：

```
this.$refs.refLevenUtsWebview.getCookie({
  //名称，为空则获取域名下所有的cookie
  name: this.form.name,
  //域名，为空则获取当前网页下的所有cookie
  domain: this.form.domain,
  //路径
  path: this.form.path
}, res => {
  console.log("获取cookie：" + JSON.stringify(res))
})
```

- 参数说明

参数名	参数类型	是否必填	默认值	参数描述
name	String	否	无	cookie名称，为空则获取域名下所有的cookie
domain	String	否	无	域名，为空则获取当前网页下的所有cookie
path	String	否	/	域名路径

回调

- 示例

```
{
  "data": {
    "list": {
```

获取cookie

```
        "c1": "1"
    },
    "message": "",
    "code": 0
}
```

- 回调说明：

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
data.list	Object	cookie集合
code	Integer	返回类型，0.成功，其他：失败

删除cookie

方法名

removeCookie

用法

- 用法如下：

```
this.$refs.refLevenUtsWebview.removeCookie({
  //名称，必须
  name: this.form.name,
  //域名，必须
  domain: this.form.domain,
  //路径，默认：/
  path: this.form.path
}, res => {
  console.log("removeCookie:" + JSON.stringify(res))
})
```

- 参数说明

参数名	参数类型	是否必填	默认值	参数描述
name	String	是	无	cookie名称
domain	String	是	无	域名
path	String	否	/	域名路径

回调

- 示例

```
{
  "data": {},
  "message": "",
  "code": 0
}
```

- 回调说明：

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
code	Integer	返回类型，0.成功，其他：失败

加载网页

方法名

用法

- 用法如下：

- 参数说明

参数名	参数类型	是否必填	默认值	参数描述
url	String	是	无	加载的url
headers	Object	否	无	附加头信息

回调

- 示例

- 回调说明：

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
code	Integer	返回类型，0.成功，其他：失败

获取网页属性

方法名

getWebData

用法

- 用法如下：

```
this.$refs.refLevenUtsWebview.getWebData(res => {
  console.log("getWebData:" + JSON.stringify(res))
})
```

- 参数说明

无

回调

- 示例

```
{
  "data": {
    "url": "https://www.baidu.com/",
    "title": "百度一下",
    "originalUrl": "https://www.baidu.com/",
    "favicon": "iVBORw0KGgoAAAANSUhEUgAAEAAAABACAYAAACqaXHeAAAAAXNSR0IArs4c6Q",
  },
  "message": "",
  "code": 0
}
```

- 回调说明：

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
data.url	String	网页地址

参数名	参数类型	参数描述
data.title	String	标题
data.originalUrl	String	原始网页地址（重定向后的原始地址）
data.favicon	String	网页图标，base64数据
code	Integer	返回类型，0.成功，其他：失败

刷新页面

方法名

reload

用法

- 用法如下：

```
this.$refs.refLevenUtsWebview.reload(res => {
  console.log("reload:" + JSON.stringify(res))
})
```

- 参数说明

无

回调

- 示例

```
{
  "data": {},
  "message": "",
  "code": 0
}
```

- 回调说明：

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
code	Integer	返回类型，0.成功，其他：失败

停止加载

方法名

stopLoading

用法

- 用法如下：

```
this.$refs.refLevenUtsWebview.stopLoading(res => {
  console.log("stopLoading:" + JSON.stringify(res))
})
```

- 参数说明

无

回调

- 示例

```
{
  "data": {},
  "message": "",
  "code": 0
}
```

- 回调说明：

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
code	Integer	返回类型，0.成功，其他：失败

是否能够前进

方法名

canGoForward

用法

- 用法如下：

```
this.$refs.refLevenUtsWebview.canGoForward(res => {
  console.log("canGoForward:" + JSON.stringify(res))
})
```

- 参数说明

无

回调

- 示例

```
{
  "data": {
    "canGoForward": false
  },
  "message": "",
  "code": 0
}
```

- 回调说明：

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
data.canGoForward	Boolean	是否能够前进
code	Integer	返回类型，0.成功，其他：失败

是否能够后退

方法名

canGoBack

用法

- 用法如下：

```
this.$refs.refLevenUtsWebview.canGoBack(res => {
  console.log("canGoBack:" + JSON.stringify(res))
})
```

- 参数说明

无

回调

- 示例

```
{
  "data": {
    "canGoBack": false
  },
  "message": "",
  "code": 0
}
```

- 回调说明：

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
data.canGoBack	Boolean	是否能够后退
code	Integer	返回类型，0.成功，其他：失败

后退

方法名

goBack

用法

- 用法如下：

```
this.$refs.refLevenUtsWebview.goBack(res => {
  console.log("goBack:" + JSON.stringify(res))
})
```

- 参数说明

无

回调

- 示例

```
{
  "data": {},
  "message": "",
  "code": 0
}
```

- 回调说明：

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
code	Integer	返回类型，0.成功，其他：失败

前进

方法名

goForward

用法

- 用法如下：

```
this.$refs.refLevenUtsWebview.goForward(res => {
  console.log("goForward:" + JSON.stringify(res))
})
```

- 参数说明

无

回调

- 示例

```
{
  "data": {},
  "message": "",
  "code": 0
}
```

- 回调说明：

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
code	Integer	返回类型，0.成功，其他：失败

执行JavaScript

方法名

evaluateJavascript

用法

- 用法如下：

```
this.$refs.refLevenUtsWebview.evaluateJavascript({
  script: "javascript:alert('Hello from JavaScript!');"
}, res => {
  console.log("evaluateJavascript:" + JSON.stringify(res))
})
```

- 参数说明

参数名	参数类型	是否必填	默认值	参数描述
script	String	是	无	要执行的js代码

回调

- 示例

```
{
  "data": {},
  "message": "",
  "code": 0
}
```

- 回调说明：

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
code	Integer	返回类型，0.成功，其他：失败

清除缓存

方法名

clearCache

用法

- 用法如下：

```
this.$refs.refLevenUtsWebview.clearCache(res => {
  console.log("clearCache:" + JSON.stringify(res))
})
```

- 参数说明
无

回调

- 示例

```
{
  "data": {},
  "message": "",
  "code": 0
}
```

- 回调说明：

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
code	Integer	返回类型，0.成功，其他：失败

获取LocalStorage v1.1.0

方法名

getLocalStorage

v1.1.0

用法

- 用法如下：

```
this.$refs.refLevenUtsWebview.getLocalStorage(res => {  
  console.log("getLocalStorage:" + JSON.stringify(res))  
})
```

- 参数说明
无

回调

- 示例

```
{  
  "data": {  
    "__tea_cache_tokens_2906": "{\\"web_id\\":\\"7462181361303012904\\",\\"user_uni"  
    "security-sdk/s_sdk_server_cert_key": "{\\"cert\\":\\"-----BEGIN CERTIFICATE-"  
    "security-sdk/s_sdk_crypt_sdk": "{\\"data\\":\\"{\\"ec_privateKey\\":\\"-----BEG"  
    "dtrait-sdk/s_sdk_server_cert_key": "eyJjZW50cmFsUnNhUHVhIjoiTFMwdExTMUNSV  
    "CREATOR_LAYOUT_CONFIG": "{\\"headerLogo\\":{\\"douyin\\":{\\"src\\":\\"//lf3-sta  
    "SLARDARuc_dtrait_sdk": "JTdCJTlYdXNlcklkJTlY0iUyMjQ2MWFhYTJkLWUzNzYtNGEyM  
    "SLARDARuc_secure_sdk": "JTdCJTlYdXNlcklkJTlY0iUyMjM0MzI4NTA2LTI4YmQtNDk1Z  
    "xmst": "w7UGB0t1U06w5aBNgSckJ0rD8kIP_KPftSb5ZbyyABGnYqKHDozakK2MEVU0mMckp  
    "LOGIN_STATUS": "{\\"logintype\\":\\"user\\",\\"loginapp\\":\\"douyin\\"}",  
    "SLARDARdouyin_creator": "JTdCJTlYdXNlcklkJTlY0iUyMjVkZGU5MjBhLWE3ZDctNGNj  
    "ztsdk_tcc_config": "{\\"value\\":{\\"ztsdk_config\\":{\\"6383\\":[{\\"aid\\":6383  
    "__tea_sdk_ab_version_2906": "{\\"data\\":{\\"assistantImage\\":{\\"val\\":1,\\"v  
    "__msuuid__": "a9bc813a-2539-462f-9dcd-6645ba5ccd91",  
    "__tea_cache_first_2906": "1"  
  },  
  "message": "",  
  "code": 0  
}
```

- 回调说明：

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
code	Integer	返回类型，0.成功，其他：失败

config

属性配置 config

config为组件启动时默认设置

- 示例

```
//webview配置
config: {
  //默认加载的url
  url: "https://www.baidu.com",
  //设置JavaScript是否可用，默认：true
  javaScriptEnabled: true,
  //媒体播放是否需要用户主动操作，默认：true
  mediaPlaybackRequiresUserGesture: true,
  //设置浏览器标识，默认：空
  userAgent: "",
  //浏览器缓存模式：default.默认处理方式，noCache:不走缓存，直接从网络获取，cacheOnly:不
  cacheMode: "noCache"
},
```

- 参数说明

参数名	参数类型	是否必填	默认值	参数描述
url	String	是	无	默认加载的url
javaScriptEnabled	Boolean	否	true	设置JavaScript是否可用
mediaPlaybackRequiresUserGesture	Boolean	否	true	媒体播放是否需要用户主动操作
userAgent	String	否	无	设置浏览器标识

参数名	参数类型	是否必填	默认值	参数描述
cacheMode	String	是	无	网页缓存加载模式：default.默认处理方式， noCache:不走缓存，直接从网络获取， cacheOnly:不使用网络，仅从缓存加载， cacheElseNetwork:先取缓存（即使已过期），不存在则再网络获取

错误事件

方法名

onError

用法

- 用法如下：

```
onError(e) {  
  console.log("onError: " + JSON.stringify(e.detail))  
},
```

- 参数说明

无

回调

- 示例

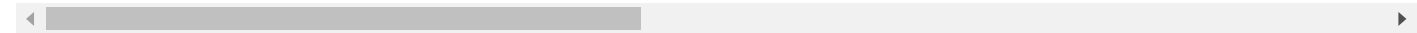
```
{  
  "response": {  
    "statusCode": 403,  
    "encoding": "utf-8",  
    "reasonPhrase": "Forbidden",  
    "mimeType": "text/plain",  
    "responseHeaders": {  
      "Content-Type": "text/plain; charset=utf-8",  
      "Date": "Sun, 15 Sep 2024 05:22:55 GMT",  
      "Content-Length": "0",  
      "Server": "bfe",  
      "X-Bfe-Svbbrrers": "MTExLjE3NS4xMTkuMTYw,1.0"  
    }  
  },  
  "request": {  
    "isMainFrame": false,  
    "hasGesture": false,  
    "method": "GET",  
    "requestHeaders": {  
      "Referer": "https://www.baidu.com/",
```

错误事件

```
        "Accept": "*/*",
        "User-Agent": "Mozilla/5.0 (Linux; Android 12; EBG-AN00 Build/HUAWEIEBG-AN00)",
    },
    "isRedirect": false
}
```

- 回调说明：

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
code	Integer	返回类型，0.成功，其他：失败



开始加载网页

方法名

onPageStarted

用法

- 用法如下：

```
onPageStarted(e) {  
  console.log("onPageStarted: " + JSON.stringify(e.detail))  
},
```

- 参数说明

无

回调

- 示例

```
{  
  "type": "onPageStarted",  
  "timeStamp": 1723431417521,  
  "target": {  
    "id": "",  
    "dataset": {},  
    "offsetLeft": 0,  
    "offsetTop": 0  
  },  
  "currentTarget": {  
    "id": "",  
    "dataset": {},  
    "offsetLeft": 0,  
    "offsetTop": 0  
  },  
  "detail": {  
    "url": "https://www.baidu.com/"  
  }  
}
```

- 回调说明：

开始加载网页

参数名	参数类型	参数描述
url	String	当前加载的地址

加载网页完成

方法名

onPageFinished

用法

- 用法如下：

```
onPageFinished(e) {  
  console.log("onPageFinished: " + JSON.stringify(e.detail))  
},
```

- 参数说明

无

回调

- 示例

```
{  
  "type": "onPageFinished",  
  "timeStamp": 1723431418901,  
  "target": {  
    "id": "",  
    "dataset": {},  
    "offsetLeft": 0,  
    "offsetTop": 0  
  },  
  "currentTarget": {  
    "id": "",  
    "dataset": {},  
    "offsetLeft": 0,  
    "offsetTop": 0  
  },  
  "detail": {  
    "url": "https://www.baidu.com/"  
  }  
}
```

- 回调说明：

加载网页完成

参数名	参数类型	参数描述
url	String	当前加载的地址

网页资源加载

方法名

onLoadResource

用法

- 用法如下：

```
onLoadResource(e) {  
  console.log("onLoadResource: " + JSON.stringify(e.detail))  
},
```

- 参数说明

无

回调

- 示例

```
{  
  "type": "onLoadResource",  
  "timeStamp": 1723431418269,  
  "target": {  
    "id": "",  
    "dataset": {},  
    "offsetLeft": 0,  
    "offsetTop": 0  
  },  
  "currentTarget": {  
    "id": "",  
    "dataset": {},  
    "offsetLeft": 0,  
    "offsetTop": 0  
  },  
  "detail": {  
    "url": "https://s.bdstatic.com/common/openjs/amd/eslx.js"  
  }  
}
```

- 回调说明：

参数名	参数类型	参数描述
url	String	当前加载的地址

网页加载进度

方法名

onProgressChanged

用法

- 用法如下：

```
onProgressChanged(e) {  
  console.log("onProgressChanged : " + JSON.stringify(e.detail))  
},
```

- 参数说明

无

回调

- 示例

```
{  
  "progress": 70  
}
```

- 回调说明：

参数名	参数类型	参数描述
progress	Integer	当前网页加载进度