## به نام خدا

مستند راهنمای نحوه ارتباط با اپلیکیشن موبایل از طریق صفحه وب

نسخه 2.0.0

تهیه کننده: تیم وب آسان پرداخت پرشین

|  |  |
| --- | --- |
| تغییرات | نسخه |
| نسخه اولیه | 1.0 |
| پارامترهای asanPardakht.application.doPaymentاضافه شد | 1.1 |
| روش جدید http post جهت اجرای وب اپ اضافه شد | 1.2 |
| روش جدید برقراری ارتباط اضافه شد | 1.3 |
| حذف متد doPayment  افزودن متد startPayment | 1.4 |
| Bug FIx | 1.4.1 |
| openExternalUrl | 1.5 |
| متد file.Upload | 1.6 |
| حذف وابستگی به jquery و افزودن pwa و حذف فایل تنظیمات | 2.0 |

سابقه تغییرات

## تعاریف

اپلیکیشن یا اپ: منظور اپلیکیشن آسان پرداخت می باشد.

وب اپ: نرم‏افزار خارجی بر روی پلفرم وب را وب-اپ می‏نامیم.

## روند اجرای وب اپ

1. کاربر آیکون مربوط به وب اپ را انتخاب می نماید.
2. اپ یک html form حاوی تمامی اطلاعات مورد نیاز جهت اجرای وب اپ با متد post به url وب اپ ارسال می کند.
3. اپ، پاسخ بازگشتی از سرور وب اپ که صفحه html است را در صفحه به کاربر نمایش می دهد.
4. صفحه html پس از لود شدن ابتدا باید متد asanPardakht.application.beInformedWebAppIsReady() را فراخوانی کند و فیلد config که در startupData موجود میباشد را به تابع پاس دهد.
5. پس از این مرحله کنترل صفحه به وب اپ واگذار می شود تا سرویس مربوطه را به کاربر ارائه دهد.
6. در طول زمان ارائه سرویس، وب اپ می تواند در صورت نیاز از اپ سرویس هایی را دریافت نماید که در ادامه مستند لیست آنها به همراه توضیحات مربوطه آورده شده است.
7. پس از آنکه ارائه سرویس به اتمام رسید (یا کاربر تصمیم به خروج گرفت) با استفاده از متد asanPardakht.application.goBack() کاربر به صفحه اصلی اپ منتقل خواهد شد.

توجه: آدرس url وب اپ **حتما** **باید** از پروتکل https با certificate معتبر استفاده نماید. البته این موضوع در زمان و محیط تستی اجباری نبوده و میتوان از پروتکلhttp نیز استفاده کرد.

## محتوای فیلد های ارسالی جهت اجرای وب اپ:

اطلاعات مورد نیاز جهت اجرای وب اپ در یک فیلد با عنوان startupData با متد post ارسال می گردد. محتوای این فیلد یک string با فرمت json است که حاوی فیلد های ذیل می باشد:

|  |  |  |
| --- | --- | --- |
| نام فیلد | نوع | توضیحات |
| authToken | string | توکن مربوط جهت انجام authentication و دریافت شماره موبایل کاربر جاری در اپ |
| platform | string | ios یا android یا pwa |
| appVersion | string | ورژن اپ |
| lang | string | fa یا en |
| additionalData | string | اطلاعات اضافی(در صورت وجود)، که یک string با فرمت json است. |
| config | string | اطلاعاتی که در هنگام صدا زدن متد beInformedWebAppIsReady باید به تابع پاس داده شود. |

## الزامات اولیه جهت برقراری ارتباط

جهت برقراری ارتباط، باید فایل های اسکریپت که به همراه این مستند ارسال شده اند به صفحه اضافه شوند:

1. آخرین ورژن از فایل asanpardakh.application.bridge-x.x.x.js

* منظور از x.x.x ورژن فایل می باشد. بطور مثال: asanpardakh.application.bridge-1.3.0.js

## برقراری ارتباط بین وب اپ و اپلیکیشن

پس از اضافه کردن فایل های فوق شی asanPardakht.application در دسترس خواهد بود. لیست ذیل شامل متدهای ارائه شده توسط این شی جهت برقراری ارتباط با اپ به همراه توضیحات مربوطه می باشد:

|  |
| --- |
| asanPardakht.application.beInformedWebAppIsReady(); |
| مطلع نمودن اپ از اینکه وب اپ با صحت لود شده است |
| این متد را وب اپ پس از آماده شدن فراخوانی میکند تا اپ مطلع شده و عملیات مورد نیاز خود را انجام دهد و میبایستی مقدار فیلد config که در startupData به صفحه post شده است را به تابع پاس دهد.  توجه: این متد تنها 1 بار باید فراخوانی گردد.  به عنوان مثال از جمله کارهایی که اپ در زمان لود کردن وب اپ انجام میدهد این است که یک صفحه loading نمایش میدهد که با فراخوانی این متد آن صفحه را حذف می کند. |

|  |
| --- |
| asanPardakht.application.setPageTitle(title); |
| ثبت عنوان صفحه |
| title: عنوان صفحه |

|  |
| --- |
| asanPardakht.application.goBack(); |
| بازگشت(خروج) به صفحه اصلی اپلیکیشن |
| با فراخوانی این متد کاربر از محیط وب اپ خارج و به صفحه اصلی اپلیکیشن هدایت می شود.  مثال: هنگامی که کاربر کلید back را لمس می‏کند، اگر در وب اپ مرحله‏ایی برای بازگشت وجود نداشت یعنی کاربر در root وب اپ بود، باید این متد فراخوانی گردد تا کاربر به صفحه اصلی اپلیکیشن بازگردد. |

|  |
| --- |
| asanPardakht.application.showLoading(); |
| نمایش علامت loading توسط اپلیکیشن |
| هنگامی که وب-اپ عملیاتی دارای زمان انجام می‏دهد باید این متد فراخوانی گردد مانند ajax |

|  |
| --- |
| asanPardakht.application.hideLoading(); |
| حذف علامت loading توسط اپلیکیشن |
|  |

|  |
| --- |
| asanPardakht.application.showMessageBox(title, message, type); |
| نمایش پیام توسط اپ |
| title: عنوان  message: پیام  type:  مقدار عددی است که مقادیر قابل استفاده از متغیر asanPardakht.application.deviceMessageBoxTypes بدست می آید.  deviceMessageBoxTypes:  {  General: 0,  Info: 1,  Success: 2,  Error: 3,  FetalError: 4  } |

|  |
| --- |
| asanPardakht.application.showConfirmBox(title, message, okTitle, cancelTitle, onOk, onCancel); |
| نمایش پنجره پرسش از کاربر توسط اپ |
| title: عنوان  message: پیام  okTitle: عنوان دکمه تائید  cancelTitle: عنوان دکمه انصراف  onOk:یک متد است که در زمانی کاربر دکمه تائید را انتخاب کند فراخوانی خواهد شد  onCancel:یک متد است که در زمانی کاربر دکمه انصراف را انتخاب کند فراخوانی خواهد شد |

|  |
| --- |
| asanPardakht.application.showError(errorMessage); |
| نمایش خطا |
| errorMessage: متن خطا |

|  |
| --- |
| asanPardakht.application.showFetalError(errorMessage); |
| نمایش خطای بحرانی. این متد پس از بسته شدن کادر خطا توسط کاربر، کل صفحه را refresh می کند. |
| errorMessage: متن خطا |

|  |
| --- |
| asanPardakht.application.showPlusButton(pressEventHandler); |
| دکمه ای با شکل + با مفهوم اضافه کردن در Actionbar اپ نمایش داده می شود. |
| پارامتر pressEventHandler یک function است که در زمان زدن دکمه plus توسط کاربر، فراخوانی خواهد شد. |

|  |
| --- |
| asanPardakht.application.hidePlusButton(); |
| دکمه plus را از Actionbar اپ حذف می کند. |

|  |
| --- |
| asanPardakht.application.callMeOnBackButtonPress(pressEventHandler); |
| هنگامی که کاربر کلید بازگشت را در اپلیکیشن لمس میکند |
| پارامتر pressEventHandler یک function است که در زمان زدن دکمه بازگشت توسط کاربر، فراخوانی خواهد شد. |

|  |
| --- |
| asanPardakht.application.openExternalUrl(url, androidPreferredApp, iosPreferredApp); |
| url را در PreferredApp مربوطه باز میکند. پیش فرض browser است. |
| مثال:  asanPardakht.application.openExternalUrl(“www.google.com”, null, null);  asanPardakht.application.openExternalUrl('https://instagram.com/\_u/asanpardakht/', 'com.instagram.android', 'instagram://user?username=asanpardakht');  asanPardakht.application.openExternalUrl('https://instagram.com/\_u/asanpardakht/', 'com.instagram.android', 'instagram://user?username=asanpardakht');  asanPardakht.application.openExternalUrl(‘tel:09121234567’, null, null);  asanPardakht.application.openExternalUrl(‘sms:09121234567’, null, null); |

|  |
| --- |
| asanPardakht.application.startPayment(paymentRequest, paymentCompletedEventHandler); |
| شروع فرآیند خرید در اپلیکیشن |
| پارامتر paymentRequest باید دارای مشخصات ذیل باشد:  Amount,  Cardnumber,  DescriptionEn,  DescriptionFa,  HostRequestData {  hreq,  hsign,  ver  },  PaymentChanelID,  PaymentId,  PointEnabled,  ServerData,  TitleEn,  TitleFa  Amount : مبلغ تراکنش  CardNumber: شماره کارتی که جهت پرداخت باید باید از آن استفاده شود  PaymentChanelID: کانال پرداخت(1: اینترنتی، 2 موبایلی)  PaymentId: شماره فاکتور  PointEnabled: امکان پرداخت از طریق امتیاز ها وجود داشته باشد یا خیر.  پارامتر paymentCompletedEventHandler یک تابع است که پس از اتمام پرداخت فراخوانی میشود. این متد یک پارامتر به صورت زیر دریافت میکند که جهت تکمیل عملیات پرداخت به سرور باید ارسال شود (جهت انجام عملیات settle/reverse).  {  paymentId,  paymentData }  مقدار paymentId همان شناسه ی paymentRequest بوده و paymentData جواب تراکنش میباشد که به صورت یک string ذخیره شده است. توجه داشته باشید که این string باید توسط برنامه deserialize شود و برای عملیات settle/reverse مورد استفاده قرار گیرد. نکته مهم این است که عملیات deserialize نباید در سمت کلاینت و کد javascript انجام شود چرا که مقدار ارسالی حاوی شناسه ای بوده که بدلیل محدودیت در طول ارقام جاوااسکریپت، امکان تبدیل آن به یک عدد جاوااسکریپت وجود ندارد.  فرمت فیلد paymentDate به صورت ذیل میباشد.  {  string message,  string host\_response,  string host\_response\_sign,  int status\_code,  long unique\_tran\_id,  long payment\_id,  } |

|  |
| --- |
| asanPardakht.application.fileManager.upload(destinationUrl, httpHeader, httpBody, title, desc, type, allowedExtensions, fileSize, duration, maxWidth, maxHeight, minWidth, minHeight, successCallback, errorCallback) |
| آپلود فایل |
| این متد فقط در اندروید استفاده دارد و آپلود فایل در ios باید بصورت مستقل پیاده سازی شود.   * destinationUrl: آدرسی که فایلها جهت آپلود به آن فرستاده میشوند * httpHeader: یک آبجکت شامل مقادیری که در header نوشته خواهد شد. * httpBody: یک آبجکت شامل مقادیری که در body نوشته خواهد شد. * MaxFileSize بر حسب کیلوبایت * MaxDuration بر حسب ثانیه * FileType: مقادیر قابل قبول: * image * video * audio * other   کدهای خطا زمان اتمام آپلود فایل  -1: لطفاً ارتباط خود را چک کنید  -2: خطای داخلی سیستم  -3: لطفاً ارتباط خود را چک کنید (timeout)  -4: خطای نامشخص  -5: انصراف توسط کاربر |