تحلیل سفر مینی‌اپ

راهنمای تحلیل سفر کاربران در مینی‌اپ‌های بومی و وب، شامل حالت خودکار، ثبت دستی رویدادها و مشاهده نتایج در پنل اپسان.

- معرفی تحلیل سفر مینی‌اپ
- مینی‌اپ بومی: حالت خودکار و تنظیم سفر در پنل
- مینی‌اپ بومی: ثبت دستی مراحل و نتیجه سفر
- مینی‌اپ وب: تحلیل خودکار و ثبت دستی سفر

معرفی تحلیل سفر مینی‌اپ

تحلیل سفر مسیر واقعی کاربر را از زمان باز شدن مینی‌اپ تا رسیدن به نتیجه نهایی ثبت می‌کند. هر سفر مجموعه‌ای مرتب از رویدادها است؛ برای نمونه ورود به صفحه اصلی، نمایش یک گفت‌وگو، رفتن به صفحه پرداخت و در نهایت تکمیل یا خطای سفر.

با این قابلیت می‌توان مشخص کرد کاربران از کدام مسیر عبور می‌کنند، در کدام مرحله متوقف می‌شوند و نرخ تکمیل هر فرایند چقدر است. نتیجه در پنل اپسان، هم به‌صورت تجمیعی و هم در جزئیات هر سفر نمایش داده می‌شود.

چه اطلاعاتی ثبت می‌شود؟

- زمان شروع و پایان سفر و زمان وقوع هر رویداد
- مراحل مشاهده‌شده با نوع `page`، `dialog` یا `custom`
- نتیجه نهایی: تکمیل‌شده، خطا، لغوشده، اتمام زمان یا ناتمام
- نوع اجرا: مینی‌اپ بومی اندروید یا Web
- داده‌های فنی و غیرحساس که توسعه‌دهنده به سفر اضافه می‌کند
- لاگ‌های مجاز، فقط وقتی قانون جمع‌آوری لاگ در تعریف سفر فعال باشد

> **نکته امنیتی:** در رابط تحلیل سفر پنل، شناسه کاربر و شناسه نشست نمایش داده نمی‌شود. در داده‌های سفر نیز اطلاعات شخصی، توکن، شماره تماس و ورودی آزاد کاربر را ثبت نکنید.

نسخه‌های موردنیاز

بخش	حداقل نسخه	کاربرد
`Appsan Core`	`0.1.78`	چرخه اجرای مینی‌اپ و ارتباط با میزبان

بخش	حداقل نسخه	کاربرد
`Appsan Analytics`	`0.1.0`	ساخت، مرتب‌سازی و ارسال رویدادهای سفر
`Appsan Web Extension`	`0.0.8`	پل امن میان WebView و SDK بومی
`Appsan Mini Web SDK`	`0.1.0`	تشخیص مسیر و API عمومی مینی‌اپ وب

برای مینی‌اپ کاملاً بومی، دو مورد اول لازم است. برای مینی‌اپ وب داخل Android WebView، هر چهار مورد باید با نسخه‌های بالای نسخه سازگار جدیدتر استفاده شوند.

روش‌های ثبت سفر

حالت خودکار

شروع سفر، نمایش صفحات بومی و تغییر مسیرهای معمول وب بدون ثبت دستی توسعه‌دهنده شناسایی می‌شود. این حالت، انتخاب پیش‌فرض است.

حالت دستی

برای مراحل معنایی که از چرخه صفحه یا URL قابل تشخیص نیست، توسعه‌دهنده رویداد را صریحاً ثبت می‌کند؛ مانند نمایش پنجره تأیید، انتخاب روش پرداخت یا پایان موفق یک فرایند چندمرحله‌ای.

صفحات بعدی، نحوه استفاده از هر دو حالت در مینی‌اپ بومی و وب و همچنین تنظیم و مشاهده نتیجه در پنل را توضیح می‌دهند.

مینی‌اپ بومی: حالت خودکار و تنظیم سفر در پنل

در مینی‌اپ بومی، افزونه Analytics چرخه اجرای مینی‌اپ و نمایش عناصر `page` و `dialog` را به‌صورت خودکار به رویدادهای سفر تبدیل می‌کند.

چه چیزهایی خودکار ثبت می‌شوند؟

1. با رندر شدن مینی‌اپ، رویداد شروع سفر ساخته می‌شود.
2. هر عنصر `page` یا `dialog` هنگام نمایش، با شناسه فنی خود به‌عنوان مرحله ثبت می‌شود.
3. ترتیب رخدادها داخل هر سفر به‌صورت افزایشی نگهداری می‌شود.
4. اگر مینی‌اپ بدون نتیجه صریح بسته شود، سفر با نتیجه لغوشده و دلیل `app_closed` پایان می‌یابد.

> **نکته:** شناسه عناصر باید پایدار، کوتاه و غیرحساس باشد؛ مانند `checkout`، `payment-confirmation` یا `error-network`. متن قابل‌نمایش، شماره سفارش و اطلاعات کاربر را به‌عنوان ID استفاده نکنید.

تنظیم سفر در پنل اپسان

ثبت رویدادها مستقل از تعریف پنل انجام می‌شود؛ اما برای ساخت قیف، محاسبه نرخ عبور مراحل و تشخیص خطا باید یک تعریف فعال ایجاد شود.

1. از منوی **مینی‌اپ‌های شما**، مینی‌اپ موردنظر را باز کنید.
2. به زبانه **سفرها** بروید و **تنظیم سفرها** را انتخاب کنید.
3. دکمه **تعریف جدید** را بزنید.
4. نوع تعریف را انتخاب کنید: **سفر** برای مسیر انعطاف‌پذیر یا **قیف تبدیل** برای ترتیب مشخص مراحل.
5. نام و توضیح فارسی وارد کنید و گزینه **تعریف فعال باشد** را روشن نگه دارید.

6. مراحل را به‌ترتیب اضافه کنید. نوع و شناسه هر مرحله باید دقیقاً با `page` یا `dialog` در XML برابر باشد.
7. در صورت نیاز، قوانین خطا را با تطبیق دقیق یا Regex تعریف کنید و سپس **ذخیره تعریف** را بزنید.

۱. ورود به زبانه سفرها

زبانه سفرهای مینیاپ در پنل اپسان

زبانه «سفرها» و دکمه «تنظیم سفرها» در پنل محلی اپسان

۲. باز کردن تنظیمات سفر

صفحه تنظیم سفرهای مینیاپ در پنل اپسان

صفحه تنظیم سفرها، فیلتر نوع تعریف و دکمه «تعریف جدید»

۳. ایجاد تعریف جدید و افزودن مراحل

فرم تعریف سفر جدید در پنل اپسان

فرم تعریف سفر جدید با نام فارسی و شناسه فنی پایدار مرحله

قوانین خطا

مشاهده یک صفحه یا دیالوگ می‌تواند سفر را خودکار خطادار کند. در حالت تطبیق دقیق، ID باید کاملاً برابر باشد. در حالت Regex، الگو روی کل ID اجرا می‌شود؛ برای نمونه `error-(network|timeout)`.

Regex فقط روی شناسه فنی اجرا می‌شود، نه متن دیالوگ یا محتوای لاگ. موتور مشترک سرور و Android از قواعد RE2 استفاده می‌کند و Lookaround یا Backreference را پشتیبانی نمی‌کند.

تنظیم قانون خطا در تعریف سفر

نمونه عمومی قانون خطا با تطبیق دقیق و شناسه فنی `error-network`

مشاهده نتیجه

در زبانه **سفرها**، جدول سفرها شناسه سفر، زمان شروع، نتیجه و تعداد مراحل را نشان می‌دهد. با انتخاب آیکون مشاهده، جزئیات مرتب مراحل باز می‌شود. داشبورد نیز تعداد کل سفرها، نرخ تکمیل، مراحل مشاهده‌شده و تفکیک نتایج را نمایش می‌دهد.

مینی‌اپ بومی: ثبت دستی مراحل و نتیجه سفر

حالت خودکار برای ثبت نمایش صفحات و دیالوگ‌ها کافی است. هرجا یک مرحله از روی چرخه رابط قابل تشخیص نیست، یا نتیجه سفر باید صریح باشد، از اکشن‌های Analytics استفاده کنید.

چه زمانی ثبت دستی لازم است؟

- رویداد تجاری بدون تغییر صفحه رخ می‌دهد؛ مانند تأیید پرداخت یا انتخاب روش ارسال.
- یک صفحه چند حالت مهم دارد که باید به‌صورت مرحله‌های جداگانه دیده شود.
- پایان موفق، لغو، پایان زمان یا خطای سفر باید با دلیل مشخص ثبت شود.
- یک مقدار فنی و غیرحساس باید به جزئیات سفر اضافه شود.

اکشن‌های سفر

اکشن	کاربرد
`analytics/journey/start`	شروع صریح سفر
`analytics/journey/complete`	پایان موفق سفر
`analytics/journey/cancel`	لغو سفر
`analytics/journey/timeout`	پایان سفر به‌علت اتمام زمان
`analytics/journey/fail`	پایان ناموفق بدون کد دلیل
`analytics/journey/fail/{reason}`	پایان ناموفق با کد دلیل؛ مانند `payment_declined`
`analytics/journey/step/{type}/{id}`	ثبت مرحله دستی با نوع `page`، `dialog` یا `custom`
`analytics/journey/data/{key}/{value}`	افزودن داده فنی و غیرحساس به سفر

ثبت نتیجه همراه تغییر صفحه

اکشن‌ها مانند سایر اکشن‌های Appsan با `->` زنجیره می‌شوند. در مثال زیر ابتدا صفحه موفقیت باز و سپس سفر تکمیل می‌شود.

```
<button
    id="pay-button"
    text="پرداخت"
    onclick="page/success->analytics/journey/complete" />
```

ثبت مرحله سفارشی

```
<button
    id="confirm-button"
    text="تأیید"
    onclick="analytics/journey/step/custom/payment-authorized" />
```

ثبت خطا با دلیل

```
<button
    id="retry-button"
    text="تلاش دوباره"
    onclick="analytics/journey/fail/payment_declined" />
```

افزودن داده به سفر

```
<button
    id="wallet-button"
    text="کیف پول"
    onclick="analytics/journey/data/payment.method/wallet" />
```

قواعد شناسه و داده

- طول شناسه مرحله و کد دلیل حداکثر ۱۲۸ کاراکتر است.
- نوع مرحله فقط یکی از `page`، `dialog` یا `custom` است.
- در هر سفر حداکثر ۲۰ مقدار داده ثبت می‌شود.
- کلید داده حداکثر ۶۴ کاراکتر و شامل حروف، عدد، نقطه، زیرخط یا خط تیره است.
- مقدار داده حداکثر ۲۵۶ کاراکتر و مجموع JSON داده‌ها حداکثر ۲۰۴۸ بایت است.

>

> **نکته امنیتی:** نام کاربر، شماره تماس، توکن، شماره سفارش و متن آزاد را در شناسه، دلیل خطا یا داده سفر قرار ندهید. برای موارد متغیر از کدهای عمومی و پایدار استفاده کنید.

ارتباط با تعریف پنل

نوع و شناسه‌های که در اکشن `step` می‌فرستید باید با مرحله تعریف‌شده در پنل یکسان باشد. برای نمونه، اکشن `analytics/journey/step/custom/payment-authorized` در پنل به‌صورت نوع `custom` و شناسه `payment-authorized` تعریف می‌شود.

مینی‌اپ وب: تحلیل خودکار و ثبت دستی سفر

در مینی‌اپ وب، SDK مسیرهای داخل صفحه را شناسایی و از طریق سوپراپ به Analytics بومی Appsan ارسال می‌کند. همین کد در نسخه PWA نیز قابل استفاده خواهد بود.

نصب و فعال‌سازی

نسخه `0.1.0` یا جدیدتر بسته زیر را به پروژه وب اضافه کنید:

```
npm install @appsan-web/mini-web-sdk@^0.1.0
```

در پروژه Android میزبان نیز `Appsan Core 0.1.78`، `Appsan Analytics 0.1.0` و `Appsan Web Extension 0.0.8` یا نسخه سازگار جدیدتر لازم است.

```
import { AppsanWeb } from '@appsan-web/mini-web-sdk';

AppsanWeb.analytics.configure({
  autoTrackRoutes: true
});
```

حالت خودکار

گزینه `autoTrackRoutes` به‌صورت پیش‌فرض فعال است. SDK مسیر اولیه و تغییرات بعدی را از منابع زیر تشخیص می‌دهد:

نمونه کاربرد	رویداد یا API مرورگر
اولین صفحه‌ای که کاربر می‌بیند	بارگذاری مسیر اولیه
مسیریابی SPA به صفحه جدید	`history.pushState`
جایگزینی مسیر فعلی	`history.replaceState`

نمونه کاربرد	رویداد ای API مرورگر
دکمه برگشت ای جلو مرورگر	`popstate`
مسیریابی مبتنی بر `#/`	`hashchange`

مسیرهای تکراری ِپشت‌سرهم دوباره ثبت نمی‌شوند. در حالت پیش‌فرض Query String و Fragment معمولی از شناسه مرحله حذف می‌شوند؛ اما مسیر Hash Router مانند `#/checkout` نگهداری می‌شود.

مسیرهای پویا و حساس

اگر URL شامل شناسه سفارش، شناسه کاربر یا مقدار متغیر است، همیشه `routeIdResolver` تعریف کنید تا پیش از ثبت، مسیر عمومی و پایدار شود.

```
AppsanWeb.analytics.configure({
  autoTrackRoutes: true,
  routeIdResolver: url =>
    url.pathname.replace(//orders/[^/]+/, '/orders/:id')
});
```

برای نادیده گرفتن یک مسیر، از Resolver مقدار `null` برگردانید.

چه زمانی حالت دستی لازم است؟

- نمای مهمی باز می‌شود ولی URL تغییری نمی‌کند.
- دیالوگ تأیید، پرداخت یا خطا باید در سفر دیده شود.
- یک رویداد تجاری مانند انتخاب روش پرداخت یا تأیید نهایی رخ می‌دهد.
- نتیجه نهایی سفر باید صریحاً تکمیل، لغو، Timeout یا خطا اعلام شود.
- باید یک مقدار فنی و غیرحساس به سفر اضافه شود.

متدهای دستی

متد	ورودی	کاربرد
`start`	-	شروع صریح سفر
`complete`	-	پایان موفق
`cancel`	-	لغو سفر

متد	ورودی	کاربرد
`timeout`	-	پایان به‌علت اتمام زمان
`fail`	`reasonCode?`	پایان ناموفق با دلیل اختیاری
`step`	`type, id`	ثبت مرحله `page`، `dialog` یا `custom`
`setData`	`key, value`	افزودن داده فنی به سفر
`log / warn / error`	`message`	ثبت پیام تشخیصی در صورت فعال بودن قانون متناظر در پنل

مثال کامل

```
import { AppsanWeb } from '@appsan-web/mini-web-sdk';

AppsanWeb.analytics.step('dialog', 'payment.confirmation');
AppsanWeb.analytics.setData('payment.method', 'wallet');

AppsanWeb.analytics.complete().subscribe({
  next: () => console.log('Journey completed'),
  error: error => console.error(error)
});
```

تمام متدهای سفر یک `Subject` از RxJS برمی‌گردانند و در صورت نیاز می‌توان نتیجه ارسال را با `subscribe` دریافت کرد.

خطا و لاگ

خطاهای Runtime و Promiseهای بدون Handler به‌صورت خودکار گزارش می‌شوند. SDK عمداً `console`، `fetch` و `XMLHttpRequest` را شنود نمی‌کند. برای ثبت پیام تشخیصی از `AppsanWeb.analytics.log`، `warn` یا `error` استفاده کنید. این پیام‌ها فقط در صورت وجود قانون فعال و منطبق در تعریف سفر پنل جمع‌آوری می‌شوند.

مسیر انتقال اطلاعات

1. SDK وب مسیر ای رویداد دستی را به پیام Analytics تبدیل می‌کند.
2. Web Extension پیام را از WebView به میزبان Android منتقل می‌کند.
3. Appsan Analytics رویداد را مرتب و به سرویس Analytics ارسال می‌کند.
4. تعریف فعال پنل، مراحل را به سفر ای قیف تبدیل می‌کند.

نمایش نتیجه در پنل

سفرهای ثبت‌شده از زبانه «سفرها» در پنل اپسان در دسترس هستند. برای مشاهده جزئیات هر سفر، آیکون مشاهده همان ردیف را انتخاب کنید.

زبانه سفرها در پنل اپسان

زبانه «سفرها» و محل نمایش سفرهای ثبت‌شده در پنل محلی اپسان

صفحه تنظیم سفرها در پنل اپسان

صفحه تنظیم سفرها و دکمه «تعریف جدید»

تنظیم مراحل وب در پنل

برای مسیرهای خودکار، نوع مرحله را `page` و شناسه را دقیقاً برابر خروجی Resolver وارد کنید؛ مانند `/checkout` یا `/orders/:id`. برای مراحل دستی نیز نوع و شناسه باید با آرگومان‌های متد `step` یکسان باشد؛ برای نمونه `dialog / payment.confirmation`.

تعریف مراحل سفر وب در پنل اپسان

نمونه عمومی تعریف مراحل وب با مسیرهای `/home` و `/checkout` و مرحله دستی `payment.confirmation`

> **پیش‌نهاد:** ابتدا خروجی Resolver و نام مرحله‌های دستی را به‌صورت یک قرارداد ثابت در پروژه تعریف کنید، سپس همان مقادیر را در پنل وارد کنید. این کار از شکسته شدن گزارش‌ها پس از تغییر مسیرهای رابط جلوگیری می‌کند.