مینی‌اپ‌ها در بله

با مینی‌اپ‌ها، می‌توانید قدرت و امکانات صفحات وب را به بازوهای خود بیاورید و رابط‌های کاربری پویا و انعطاف‌پذیری طراحی کنید که مستقیماً در بله اجرا شده و تجربه‌ای یکپارچه و روان برای کاربران فراهم می‌کنند. این مینی اپ‌ها می‌توانند به‌طور کامل جایگزین وب‌سایت‌ها شوند و امکان تعامل پیشرفته‌تری را در محیط بله ارائه دهند.

مثال‌ها

برای دانلود یک نمونه مثال از نحوه‌ی کار با sdk مینی‌اپ، اینجا کلیک کنید.

برای مشاهدهٔ نمونه‌ای زنده از قابلیت‌های مختلف sdk مینی‌اپ، مینی‌اپ نمونهٔ بله را در @sample_miniapp_bot باز کنید.

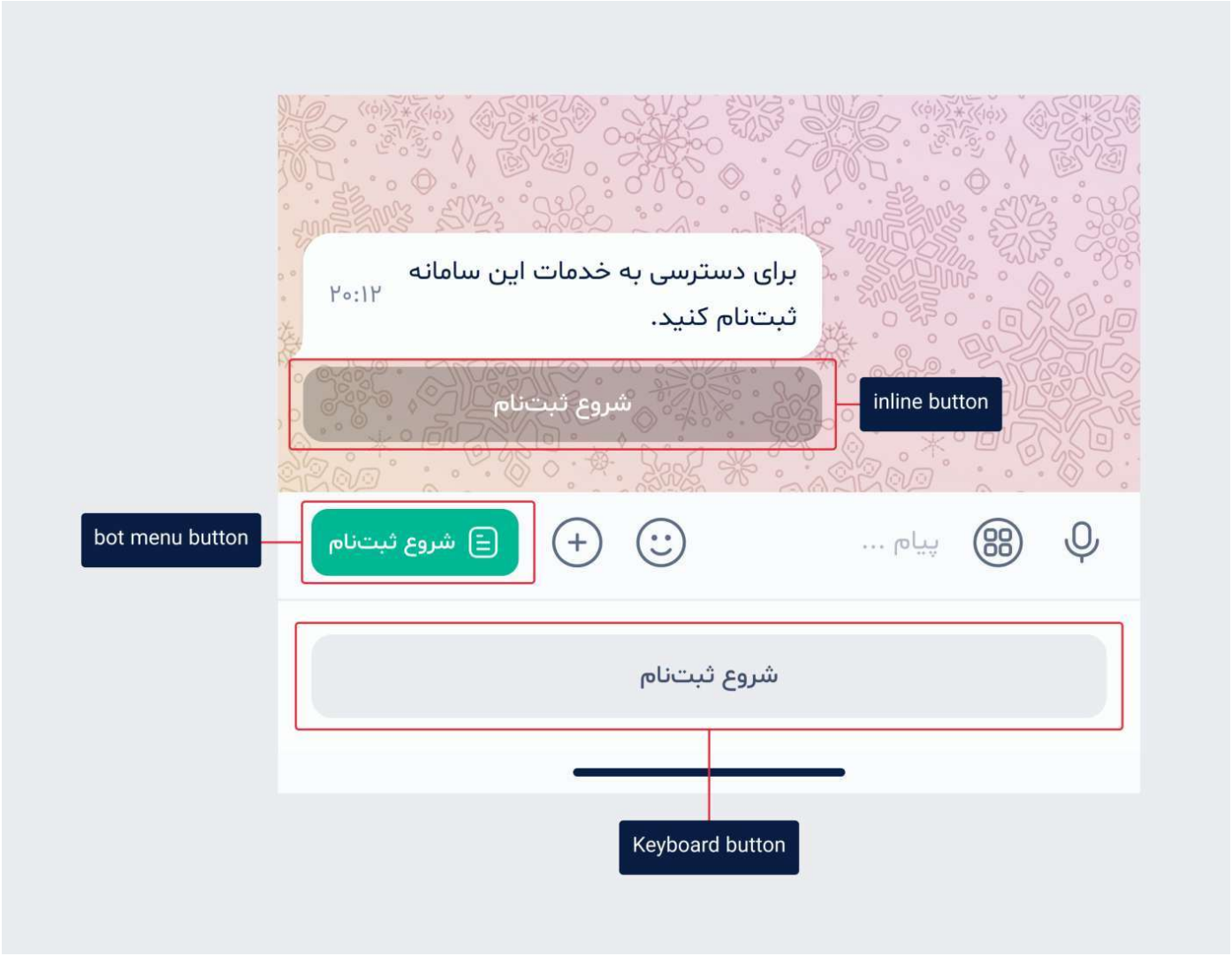
راهنمای طراحی

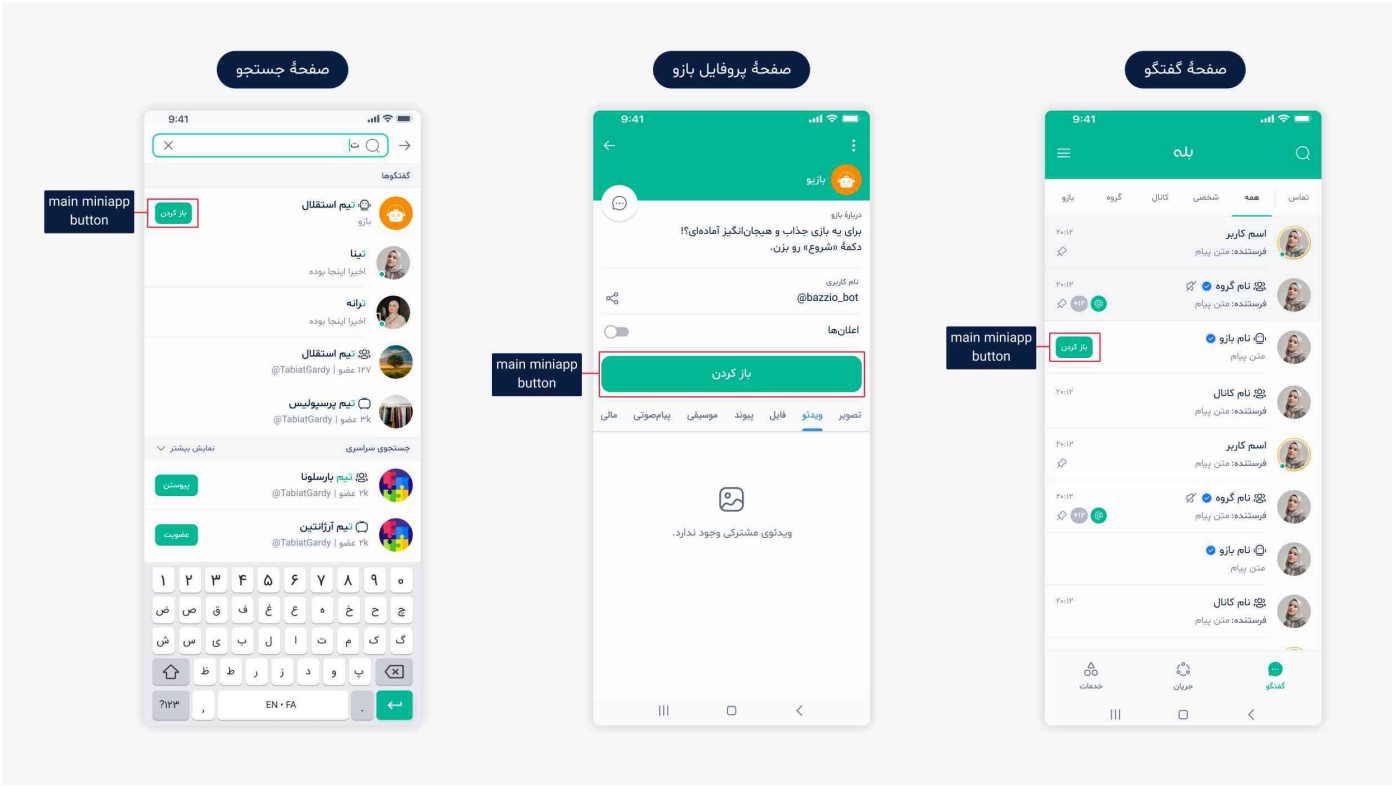
این راهنما، به هدف یکپارچگی بیشتر مینی‌اپ‌ها، ارائه یک تجربهٔ خوب به کاربران و در نتیجه بالا رفتن تعداد کاربر بازو در نظر گرفته شده است. در طراحی مینی‌اپ‌ها به این نکات توجه کنید:

- همه عناصر موجود در مینی‌اپ باید **پاسخگو (Responsive)** باشند. به این معنی که با ابعاد دستگاه‌های مختلف (موبایل، تبلت، لپ‌تاپ) هماهنگ بوده و کاربران بدون توجه به بزرگ یا کوچک بودن نمایشگر، بتوانند به‌راحتی و به‌طور کامل مینی‌اپ را ببینند.
- عناصر موجود در مینی‌اپ با رویکرد **موبایل محور (mobile-first)** طراحی شوند. به این معنی که ابتدا با در نظر گرفتن صفحهٔ نمایش کوچکتر (موبایل) طراحی کنید و توسعه دهید و بعد از آن برای صفحه‌های نمایش بزرگ‌تر طراحی کنید (وب و ...).
- در صورت استفاده از انیمیشن، باید روان و در حالت ایده‌آل ۶۰ فریم بر ثانیه باشند. انیمیشن‌ها و جلوه‌های بصری را در دستگاه‌های با کارایی پایین به حداقل برسانید تا از عملکرد روان مینی‌اپ در آن دستگاه‌ها اطمینان داشته باشید.
- در انتخاب رنگ عناصر مینی‌اپ به تم تیره و روشن دستگاه کاربر توجه شود.
- عرض دکمهٔ مینی‌اپ در اینپوت بار بازو برای اندروید حداکثر می‌تواند **۲۰ کاراکتر** باشد.
- خطاهایی که ممکن است به کاربر نشان داده شود، را در نظر بگیرید (Error Handling). متن خطا باید گویا و قابل فهم باشد و کاربر را برای اقدام بعدی راهنمایی کند.

دسترسی به مینی‌اپ‌ها

در حال حاضر از چهار روش برای اجرای مینی‌اپ‌ها در بله پشتیبانی می‌کنیم؛ مینی‌اپ اصلی، دکمه منوی بازو، دکمه اینلاین و دکمه صفحه‌کلید ریپلای.





مینی‌اپ اصلی

مینی‌اپ اصلی به شما اجازه می‌دهد که یک آدرس مشخص را به عنوان نقطه ورود اصلی برای کاربران خود تعیین کنید. هر بازو یک مینی‌اپ اصلی دارد که می‌توانید با مراجعه به @botfather آن را فعال کنید.

برای ورود به مینی‌اپ اصلی، کاربران نیازی به شروع گفتگو با بازوی شما ندارند و می‌توانند از طریق **پیوند مستقیم**، **صفحه جستجو** و **صفحه پروفایل بازو** به آن وارد شوند. علاوه بر این موارد، از **صفحه گفتگوها** نیز دکمه مینی‌اپ اصلی در دسترس است. این ویژگی‌هابه کاربران اجازه می‌دهند دسترسی به مراتب ساده و سریع‌تری به مینی‌اپ بازوی شما داشته باشند.

پیوند مستقیم

کاربران می‌توانند از هر جایی با کلیک روی پیوندی در قالب زیر، مستقیم وارد مینی‌اپ اصلی شود؛

```
https://ble.ir/<bot_username>?startapp
https://ble.ir/<bot_username>?startapp=<start_param>
```

اگر پارامتر `startapp` در پیوند به‌صورت غیرخالی مقداردهی شده باشد، این مقدار به مینی‌اپ منتقل شده و در قالب query با نام `tgWebAppStartParam` در دسترس خواهد بود.

مینی‌اپ در دکمه منو بازو

در گفتگوی شخصی با بازوها، همیشه دکمه‌ای کاربردی کنار نوار ورودی قرار دارد که دسترسی سریع به دستورهای بازو را فراهم می‌کند. با مراجعه به @botfather می‌توانید **عملکرد این دکمه را تغییر دهید** تا پیوند ثابتی را در قالب مینی‌اپ باز کند و **متن دلخواه** شما را نمایش دهد.

مینی‌اپ در پیام‌ها

با استفاده از فیلد `web_app` در InlineKeyboardButton می‌توانید **هر پیوندی** را در دکمه زیر پیام‌ها قرار دهید تا در قالب مینی‌اپ باز شود.

همچنین با استفاده از فیلد مشابه `web_app` در KeyboardButton می‌توانید **هر پیوندی** را در دکمه صفحه‌کلید، همراه پیام‌ها قرار دهید تا در قالب مینی‌اپ باز شود.

پیاده‌سازی مینی‌اپ‌ها

همانطور که گفته شد، مینی‌اپ‌ها قابلیت استفاده از بسیاری از امکانات صفحات وب را دارا هستند. اما ویژگی اصلی مینی‌اپ، ارتباط با اپلیکیشن بله و تعامل با کاربرها از طریق آن می‌باشد. برای برقراری ارتباط مینی‌اپ‌تان با کلاینت بله، `script` زیر را قبل از هر `script` دیگری در ابتدای `<head>` فایل اصلی `html` خود قرار دهید:

```
<script src="https://tapi.bale.ai/miniapp.js"></script>
```

با اجرای این کد، شیء `window.Bale.WebApp` با ویژگی‌هایی که در ادامه توضیح داده شدند، ایجاد خواهد شد.

WebApp

فیلد	نسخه	نوع	توضیحات
initData	۱.۰	String	یک رشته که حاوی داده خام ارسال شده به مینی‌اپ است و می‌توان از آن برای اعتبارسنجی داده‌ها استفاده کرد.

فیلد	نسخه	نوع	توضیحات
initDataUnsafe	۱.۰	WebAppInitData	شی‌ای شامل داده‌های ورودی منتقل‌شده به مینی اپ. **هشدار:** این داده‌ها را تنها پس از اعتبارسنجی در سرور بازو استفاده کنید، زیرا ممکن است دارای مقادیر قدیمی یا جعلی باشند.
version	۱.۰	String	نسخه‌ای از API مینی‌اپ که کلاینت کاربر از آن پشتیبانی می‌کند.
isIframe	۱.۰	Boolean	نشان‌دهنده پلتفرمی که از طریق آن مینی‌اپ باز شده است. اگر true باشد یعنی از وب باز شده است و اگر false باشد، یعنی از اندروید باز شده است
themeParams	۱.۰	ThemeParams	شی‌ای شامل اطلاعات تم استفاده‌شده فعلی در اپلیکیشن بله.
isClosingConfirmationEnabled	۱.۰	Boolean	مقدار `true` اگر دیالوگ تأیید برای بستن مینی اپ فعال باشد، مقدار `false` اگر غیرفعال باشد.
BackButton	۱.۰	BackButton	آبجکتی برای کنترل دکمه برگشت که می‌تواند در هدر مینی‌اپ در رابط کاربری بله نمایش داده شود.
SettingsButton	۱.۰	SettingsButton	آبجکتی برای کنترل گزینه‌ی تنظیمات در منوی زمینه‌ی مینی‌اپ در رابط کاربری بله.
colorScheme	۱.۰	String	نشان‌دهنده‌ی تم کاربر در اپلیکیشن بله است؛ مقدار `light` برای تم روشن و `dark` برای تم تاریک.
isMiniAppSupported	۱.۰	Boolean	نشان‌دهنده‌ی پشتیبانی شدن مینی‌اپ در اپلیکیشن بله کاربر است.

متدها

در این قسمت تمام متدهایی قابل استفاده شیء WebApp توضیح داده شده است. این متدها به شما این امکان را می‌دهند که بتوانید با اپلیکیشن بله و خارج از مینی‌اپ خود، ارتباط برقرار کنید. همچنین در نتیجه‌ی صدا زدن این متدها ممکن است برخی از Eventها رخ بدهند. اگر اپلیکیشن کاربر نسخه پایینتری از مینی‌اپ را داشته باشد، با صدا زدن متد مربوط به نسخه بالاتر، اتفاقی نخواهد افتاد.

نام	نسخه	توضیحات
openLink(url, options)	۱.۰	با این متد می‌توانید یک پیوند را در مرورگر داخلی بله یا مرورگر پیش‌فرض دستگاه کاربر باز کنید. اگر پارامتر اختیاری `options` دارای فیلد `try_instant_view` با مقدار `true` باشد، پیوند در مرورگر داخلی بله باز می‌شود.
()addToHomeScreen	۱.۰	متدی برای نمایش درخواست اضافه کردن میانبر مینی‌اپ به صفحه اصلی دستگاه.
checkHomeScreenStatus(callback)	۱.۰	این متد بررسی می‌کند که آیا امکان افزودن میانبر مینی‌اپ به صفحه اصلی وجود دارد یا نه. همچنین بررسی می‌کند که میانبر قبلاً به صفحه اصلی اضافه شده است یا خیر. اگر تابع `callback` به عنوان پارامتر اختیاری داده شده باشد، با یک آرگومان `status` فراخوانی می‌شود که یک رشته حاوی وضعیت صفحه اصلی است. مقادیر ممکن برای `status` عبارتند از: – **unsupported** – قابلیت افزودن میانبر به صفحه اصلی وجود ندارد. – **unknown** – این قابلیت پشتیبانی می‌شود و امکان افزودن میانبر وجود دارد، اما نمی‌توان تعیین کرد که آیا میانبر قبلاً اضافه شده است یا خیر. – **added** – میانبر قبلاً به صفحه اصلی اضافه شده است، – missed – میانبر به صفحه اصلی اضافه نشده است.

نام	نسخه	توضیحات
requestContact(callback)	۱.۰	فراخوانی این متد یک دیالوگ به کاربر نمایش می‌دهد و از او درخواست دسترسی به شماره تلفن می‌کند. اگر پارامتر اختیاری `callback` داده شده باشد، تابع `callback` هنگام بسته شدن دیالوگ، فراخوانی می‌شود؛ آرگومان اول یک مقدار *boolean* خواهد بود که نشان می‌دهد آیا کاربر شماره تلفن خود را به اشتراک گذاشته است یا خیر و آرگومان دوم، شماره تلفن همراه کاربر است **نکتهٔ امنیتی:** اعتبارسنجی شمارهٔ تلفن در سرور را انجام دهید.
()enableClosingConfirmation	۱.۰	متدی برای فعال‌سازی دیالوگ تأیید بستن مینی اپ.
()disableClosingConfirmation	۱.۰	متدی برای غیرفعال‌سازی دیالوگ تأیید بستن مینی اپ.
onEvent(eventType, eventHandler)	۱.۰	با استفاده از این متد می‌توانید برای دریافت رویدادهای مختلفی که مینی‌اپ دریافت می‌کند، یک یا چند تابع (`eventHandler`) ثبت کند. فهرست تمام رویدادها را می‌توانید اینجا مشاهده کنید.
offEvent(eventType, eventHandler)	۱.۰	متدی برای حذف یک event handler که قبلاً ثبت شده است.
()ready	۱.۰	این متد به بله اطلاع می‌دهد که مینی‌اپ آماده نمایش است. پس از فراخوانی این متد، حالت بارگذاری به اتمام رسیده و مینی‌اپ نمایش داده می‌شود. اگر این متد فراخوانی نشود، حالت بارگذاری تنها زمانی به اتمام می‌رسد که صفحه به طور کامل بارگیری شود. توصیه می‌کنیم این متد را در سریع‌ترین زمان ممکن و پس از بارگیری تمام عناصر ضروری رابط کاربری، فراخوانی کنید. همچنین می‌توانید پس از فراخوانی این متد، بارگذاری شخصی‌سازی شده مینی‌اپ خود را نمایش دهید.

نام	نسخه	توضیحات
expand()	۱.۰	با فراخوانی این متد، مینی‌اپ در حالت تمام‌صفحه نمایش داده می‌شود.
close()	۱.۰	متدی برای بستن مینی‌اپ.
setHeaderColor(color)	۱.۱	با این متد می‌توانید رنگ Header مینی‌اپ را با استفاده از **کلیدهای رنگی پیشفرض** (`bg_color` یا `secondary_bg_color`) یا رنگ دلخواه hex (در فرمت `#RRGGBB`) تنظیم کنید.
showScanQrPopup(data)	۱.۲	اسکنر QR را باز می‌کند. هنگامی که اسکنر QR را می‌خواند، ایونت `qrTextReceived` را ارسال می‌کند. هنگامی که اسکنر بسته می‌شود، ایونت `scanQrPopupClosed` را ارسال می‌کند.
sendData(data)	۱.۲	با فراخوانی این متد `data` در قالب آپدیت یک پیام با فیلد web_app_data به بازو ارسال می‌شود. دیتای ارسال شده حداکثر می‌تواند ۴۰۹۶ بایت باشد. ⚠️ **توجه:** بعد از فراخوانی این متد مینی‌اپ بسته می‌شود.
openInvoice(invoiceParams, callback)	۱.۳	با فراخوانی این متد، صفحه پرداخت در مینی‌اپ باز می‌شود. **پارامترها:** - پارامتر اول: `invoiceParams` - دقیقاً همان مقداری که از پاسخ createInvoiceLink دریافت می‌کنید را بدون تغییر پاس دهید - پارامتر دوم (اختیاری): `callback` `function` - تابعی که پس از تکمیل فرآیند پرداخت اجرا می‌شود و یک آبجکت با فیلد `status` که وضعیت نهایی پرداخت را نشان می‌دهد به آن ارسال می‌گردد **مقادیر ممکن برای وضعیت پرداخت (`status`):**

نام	نسخه	توضیحات
		– **paid**: پرداخت با موفقیت انجام شده است – **cancelled**: کاربر صفحه پرداخت را بسته است – **failed**: کاربر تلاش کرده است که پرداخت را انجام دهد ولی موفق نشده است – **pending**: پرداخت در حال پردازش است **رفتار مینی‌اپ:** هنگام بسته شدن صفحه پرداخت، مینی‌اپ ایونت `invoiceClosed` را دریافت می‌کند. **نکته مهم:** قبل از استفاده از این متد، ابتدا باید متد createInvoiceLink را فراخوانی کرده و مقدار `invoiceParams` را از پاسخ آن دریافت نمایید. **توجه – نسخه‌های قدیمی اندروید:** در برخی نسخه‌های قدیمی‌تر اندروید، کاربران ممکن است با مشکلات UX در انجام فرآیند پرداخت مواجه شوند. برای حل این مشکل، می‌توانید از Memory Router استفاده کنید
askReview(data)	۱.۴	از این متد مشابه متد askReview در API بازو، برای درخواست از کاربر جهت ثبت یا ویرایش نظر دربارهٔ بازوی خود استفاده کنند. پارامتر `data` دارای یک فیلد **الزامی** به نام `delay_seconds` است که میزان تأخیر را از زمان فراخوانی متد تا نمایش فرم رضایت‌سنجی مشخص می‌کند.

امنیت شماره تلفن در `requestContact`

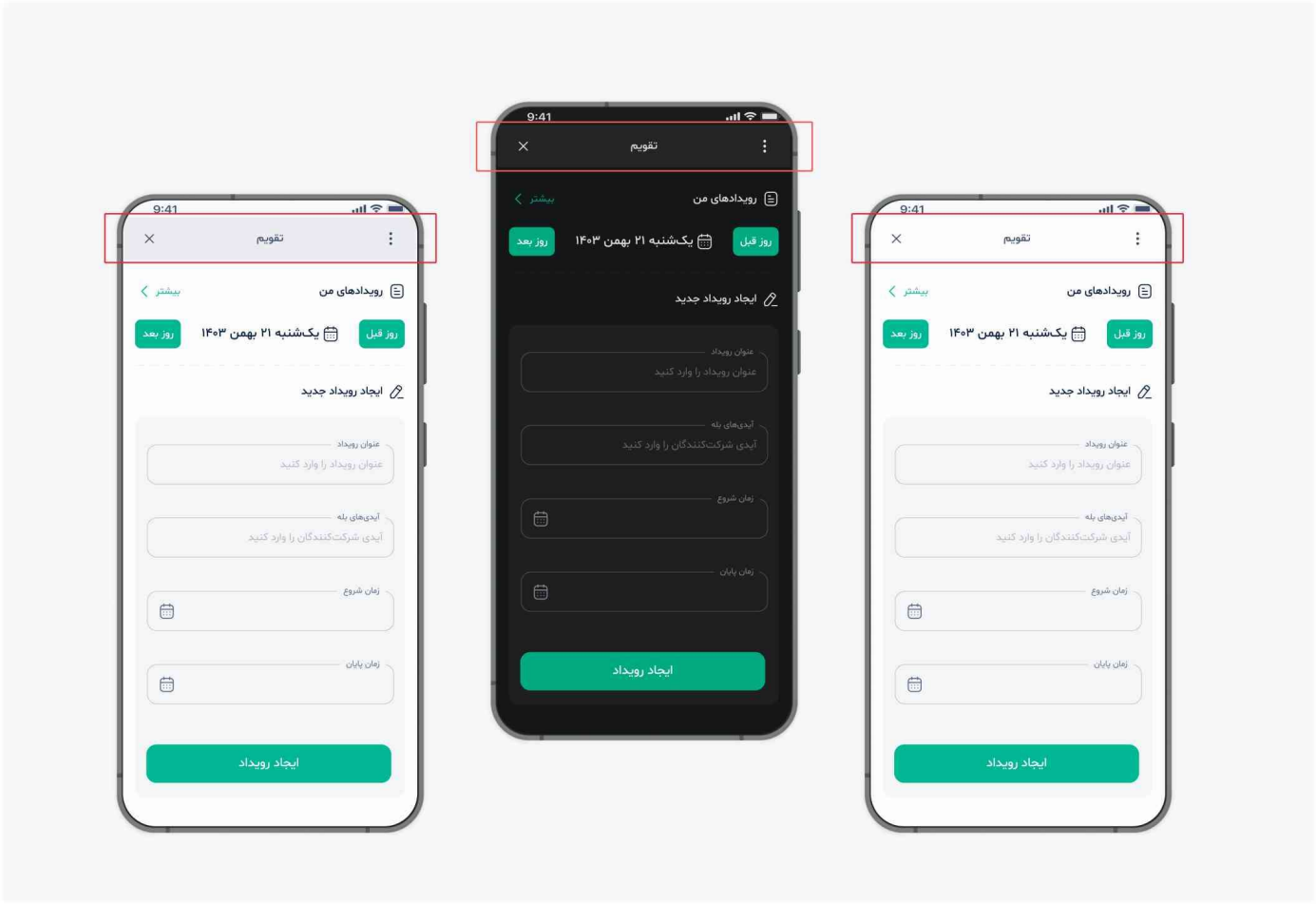
> ***هشدار:*** *شماره تلفنی که در callback متد `requestContact` یا رویداد `contactRequested` به مینی‌اپ می‌رسد، تنها برای نمایش به کاربر و ادامهٔ تجربهٔ کاربری است و نباید مبنای احراز هویت، اعطای دسترسی، ثبت شمارهٔ تأییدشده یا انجام هر عملیات حساس دیگری قرار گیرد.*

پس از آنکه کاربر شمارهٔ خود را به اشتراک گذاشت، بله یک پیام (آپدیت) از نوع `contact` برای بازوی شما ارسال می‌کند. برای تأیید شماره، سرور بازو باید شرط زیر را بررسی کند:

```
contact.user_id == from.id
```

فقط پس از برقرار بودن این شرط، شمارهٔ `contact.phone_number` را ذخیره کنید یا برای صحت‌سنجی و عملیات حساس به کار ببرید.

Header



WebAppInitData

این شیء حاوی داده‌ای است که هنگام باز شدن مینی‌اپ به آن ارسال می‌شود.

فیلد	نوع	توضیحات
query_id	String	شناسه‌ای یکتا برای نشست مینی اپ.
user	WebAppUser	شیءای شامل اطلاعات مربوط به کاربر فعلی.
auth_date	Integer	زمان باز شدن مینی‌اپ در قالب Unix
hash	String	مقدار hash شده کل داده ارسالی. می‌توانید از آن برای اعتبارسنجی داده در سرور بازو استفاده کنید.

WebAppUser

این شیء حاوی داده کاربری است که از مینی‌اپ استفاده می‌کند.

فیلد	نوع	توضیحات
id	Integer	شناسه‌ی یکتای کاربر.
first_name	String	نام کاربر در بله.
username	String	*اختیاری*. نام کاربری کاربر در بله.
allows_write_to_pm	Boolean	*اختیاری*. مقدار `true` اگر این کاربر اجازه‌ی ارسال پیام توسط بازو به خود را داده باشد.
photo_url	String	*اختیاری*. آدرس عکس پروفایل کاربر. این عکس می‌تواند در قالب `jpeg` یا `svg` باشد.

ThemeParams

مینی‌اپ‌ها می‌توانند ظاهر رابط کاربری خود را مطابق تنظیمات کاربر در اپلیکیشن بله قرار دهند. این شیء حاوی داده تم فعلی کاربر در بله است.

فیلد	نوع	توضیحات
bg_color	String	*اختیاری*. رنگ پس‌زمینه در قالب `#RRGGBB` .

فیلد	نوع	توضیحات
text_color	String	*اختیاری*. رنگ متن اصلی در قالب `#RRGGBB` . برای استفاده در متن‌های بدنه (تیترها، پاراگراف‌ها، فهرست‌ها)
hint_color	String	*اختیاری*. رنگ متن راهنما در قالب `#RRGGBB` . برای استفاده در متن‌های Placeholder (مثلاً در ورودی‌ها)، توضیحات فرعی و زیرنویس‌های کمرنگ.
link_color	String	*اختیاری*. رنگ پیوند در قالب `#RRGGBB` . برای استفاده در پیوندها و متن‌های قابل کلیک
button_color	String	*اختیاری*. رنگ دکمه در قالب `#RRGGBB` . برای استفاده در پس زمینه دکمه‌های اصلی مثل دکمه‌های تأیید
button_text_color	String	*اختیاری*. رنگ متن دکمه در قالب `#RRGGBB` . برای استفاده در متن روی دکمه‌های رنگی.
secondary_bg_color	String	*اختیاری*. رنگ پس‌زمینه‌ی ثانویه در قالب `#RRGGBB` . برای استفاده در پس‌زمینه کارت‌ها و بخش‌های جدا شده از پس‌زمینه اصلی.
header_bg_color	String	*اختیاری*. رنگ پس‌زمینه‌ی header در قالب `#RRGGBB` . برای استفاده در نوار بالایی صفحات.
bottom_bar_bg_color	String	*اختیاری*. رنگ پس‌زمینه‌ی نوار پایینی در قالب `#RRGGBB` . برای استفاده در بخش‌های ثابت پایین صفحه.
accent_text_color	String	*اختیاری*. رنگ متن تأکیدی در قالب `#RRGGBB` . برای استفاده در متن‌های هایلایت شده و عناصر متنی با اهمیت بالا.
section_bg_color	String	*اختیاری*. رنگ پس‌زمینه‌ی بخش در قالب `#RRGGBB` . برای استفاده همراه با `secondary_bg_color` .
section_header_text_color	String	*اختیاری*. رنگ متن header بخش در قالب `#RRGGBB` . برای استفاده در تیتر بخش‌ها (مثلاً *تنظیمات حساب*، *اطلاعات پرداخت*)
section_separator_color	String	*اختیاری*. رنگ جداکننده‌ی بخش در قالب `#RRGGBB` .

فیلد	نوع	توضیحات
subtitle_text_color	String	*اختیاری*. رنگ متن زیرعنوان در قالب `#RRGGBB` .
destructive_text_color	String	*اختیاری*. رنگ متن برای اقدامات غیرقابل بازگشت (مثلاً *حذف حساب*) در قالب `#RRGGBB` .



BackButton

این شیء، کنترل دکمه بازگشت در header رابط کاربری بله را بر عهده دارد و با استفاده از آن می‌توانید دکمه بستن را با دکمه بازگشت جایگزین کنید.

فیلد	نوع	توضیحات
isVisible	Boolean	نشان می‌دهد که آیا دکمه قابل مشاهده است یا نه. به‌طور پیش‌فرض برابر با `false` است.
onClick(callback)	Function	این متد یک `eventHandler` برای کلیک روی دکمه ثبت می‌کند. معادل `Bale.WebApp.onEvent('backButtonPressed', callback)`
offClick(callback)	Function	این متد `eventHandler` کلیک روی دکمه را غیر فعال می‌کند. معادل `Bale.WebApp.offEvent('backButtonPressed', callback)`
()show	Function	متدی برای فعال و قابل مشاهده‌کردن دکمه.
()hide	Function	متدی برای پنهان‌کردن دکمه.

⚠️ هشدار مهم

برای اطمینان از کارکرد صحیح این دکمه در کلاینت وب بله، از روتینگ درون‌برنامه‌ای (In-Memory Routing) استفاده کنید که تاریخچه مرورگر را تغییر نمی‌دهند.

React Example:

```
import { MemoryRouter } from "react-router-dom";

<MemoryRouter>{/* Routes */}</MemoryRouter>;
```

Vue Example:

```
import { createMemoryHistory, createRouter } from 'vue-router';

const router = createRouter({
 history: createMemoryHistory(),
 routes: [...]
});
```

SettingsButton

این شیء وظیفه افزودن و کنترل گزینه *تنظیمات* را در منوی زمینه header در رابط کاربری بله را بر عهده دارد.

فیلد	نوع	توضیحات
isVisible	Boolean	نشان می‌دهد که آیا گزینه در منو قابل مشاهده است یا نه. به‌طور پیش‌فرض برابر با `false` است.
onClick(callback)	Function	این متد یک `eventHandler` برای کلیک روی گزینه ثبت می‌کند. معادل `Bale.WebApp.onEvent('settingsButtonClicked', callback)`
offClick(callback)	Function	این متد `eventHandler` کلیک روی گزینه را غیر فعال می‌کند. معادل `Bale.WebApp.offEvent('settingsButtonClicked', callback)`
show()	Function	متدی برای فعال و قابل مشاهده‌کردن گزینه *تنظیمات* در منوی زمینه.
hide()	Function	متدی برای پنهان‌کردن گزینه *تنظیمات* در منوی زمینه.

اعتبارسنجی داده‌های دریافت‌شده از طریق مینی‌اپ

برای اعتبارسنجی داده‌های دریافت‌شده از طریق مینی‌اپ، باید داده موجود در فیلد `Bale.WebApp.initData` را به سرور بازو ارسال کنید. این داده‌ها به‌صورت یک رشته **url encoded** و شامل مجموعه‌ای از key-value ها هستند. انواع داده‌های پیچیده (مانند WebAppUser) به‌صورت **JSON-serialized** در این رشته قرار می‌گیرند.

برای اطمینان از صحت داده‌های دریافت‌شده، می‌توانید مقدار `hash` دریافت‌شده را با نمایش hexadecimal امضای **HMAC-SHA-۲۵۶** از **data-check-string** مقایسه کنید. کلید مورد استفاده در این امضا، امضای **HMAC-SHA-۲۵۶** از توکن بازو با رشته ثابت `"WebAppData"` است.

data-check-string شامل تمامی فیلدهای دریافت‌شده است که به ترتیب حروف الفبا مرتب شده‌اند و مانند مثال زیر در کنار یکدیگر قرار گرفته اند. از کاراکتر **خط جدید** (`'\n'` , `0x0A`) به‌عنوان جداکننده استفاده می‌شود.

```
auth_date=<auth_date>\nquery_id=<query_id>\nuser=<user>
```

شبه‌کد بررسی صحت داده‌ها:

```
data_check_string = ...
secret_key = HMAC_SHA۲۵۶(<bot_token>, "WebAppData")
if (hex(HMAC_SHA۲۵۶(data_check_string, secret_key)) == hash) {
  // data is sent from Bale
}
```

برای جلوگیری از استفاده از داده‌های قدیمی، می‌توانید مقدار **auth_date** را بررسی کنید. این مقدار زمان باز شدن مینی‌اپ را در قالب Unix نشان می‌دهد.

Events

مینی‌اپ می‌تواند رویدادهایی را از کلاینت بله دریافت کند و با استفاده از متد `Bale.WebApp.onEvent(eventType, eventHandler)` ، یک یا چند تابع را برای دریافت آن‌ها ثبت کند. داخل `eventHandler` ، کلمه کلیدی `this` به `Bale.WebApp` اشاره می‌کند و مجموعه پارامترهای ارسال‌شده به آن بستگی به نوع رویداد دارد. در ادامه فهرستی از رویدادهای ممکن آورده شده است:

اسم	نسخه	توضیح
backButtonPressed	۱	زمانی رخ می‌دهد که دکمه بازگشت فشرده شود. `eventHandler` هیچ پارامتری دریافت نمی‌کند.

اسم	نسخه	توضیح
settingsButtonPressed	۱	زمانی رخ می‌دهد که دکمه تنظیمات در منو فشرده شود. `eventHandler` هیچ پارامتری دریافت نمی‌کند.
contactRequested	۱	زمانی رخ می‌دهد که شماره تلفن کاربر درخواست شود. `eventHandler` یک آبجکت دریافت می‌کند که شامل فیلد `status` با یکی از مقادیر زیر است: – `sent` – کاربر شماره تلفن خود را با بات به اشتراک گذاشت. – `cancelled` – کاربر این درخواست را رد کرد.
homeScreenChecked	۱.۱	بعد از بررسی وضعیت بودن یا نبودن در home screen رخ می دهد. تابعی که به عنوان event handler پاس داده می‌شود، یک آبجکت با فیلد status دریافت می‌کند که مقادیر زیر را می‌تواند داشته باشد: – **unsupported**: این ویژگی پشتیبانی نمی شود، و امکان اضافه کردن مینی‌اپ به home screen وجود ندارد. – **unknown**: این ویژگی پشتیبانی می شود، و مینی‌اپ را می توان اضافه کرد، اما تعیین اینکه آیا مینی‌اپ قبلاً اضافه شده است، امکان پذیر نیست. – **added**: مینی‌اپ قبلاً به صفحه اصلی اضافه شده است. – **missed**: – مینی‌اپ به صفحه اصلی اضافه نشده است.
homeScreenAdded	۱.۲	زمانی رخ می دهد که مینی‌اپ با موفقیت به home screen اضافه شود. تابع `eventHandler` ای که پاس داده می‌شود، هیچی پارامتری دریافت نمی‌کند.
homeScreenFailed	۱.۲	زمانی رخ می دهد که مینی‌اپ به home screen نتوانست اضافه شود. تابع `eventHandler` ای که پاس داده می‌شود، هیچی پارامتری دریافت نمی‌کند
qrTextReceived	۱.۲	زمان رخ می‌دهد که اسکنر QR را می‌خواند. تابع `eventHandler` یک آبجکت با فیلد `text` دریافت می‌کند. متن اسکن شده درون این فیلد قرار دارد.
scanQrPopupClosed	۱.۲	زمان رخ می‌دهد که اسکنر QR بدون اسکن کردن بسته شود. `eventHandler` هیچ پارامتری دریافت نمی‌کند.
invoiceClosed	۱.۳	زمانی رخ می‌دهد که صفحه پرداخت بسته شود. تابع `eventHandler` به عنوان ورودی یک آبجکت دریافت

اسم	نسخه	توضیح
		می‌کند که ۲ فیلد دارد: – **invoiceId** – **status** مقادیری که status می‌تواند داشته باشد: – **paid**: پرداخت با موفقیت انجام شده است – **cancelled** – کاربر صفحه پرداخت را بسته است – **failed** – کاربر تلاش کرده است که پرداخت را انجام دهد ولی موفق نشده است – **pending** – پرداخت در حال پردازش است

رفع اشکال مینی‌اپ در اپلیکیشن اندروید

برای تسهیل فرآیند توسعه و شناسایی سریع‌تر اشکالات در مینی اپ می‌توانید با فعال کردن قابلیت رفع اشکال (Debugging) به لاگ‌ها و خطاهای اپلیکیشن دسترسی پیدا کرده و مشکلات را به سرعت شناسایی کنید.

مراحل فعال‌سازی به شرح زیر است:

- در تنظیمات اپلیکیشن بله، گزینه *نسخه* را نگه دارید و حالت ارسال لاگ را فعال کنید.
- از بین گزینه‌ها، گزینه "فعال‌سازی رفع اشکال وب ویو" را انتخاب کنید.
- در ادامه برای استفاده از این قابلیت این توضیحات را مطالعه کنید.

رفع اشکال باز نشدن مینی‌اپ در نسخه وب

مینی‌اپ‌ها در نسخه وب بله از طریق iframe باز می‌شوند و ممکن است که وجود هدرهای مشخصی، مانع باز شدن مینی‌اپ شما در نسخه وب شود. برای اطمینان از درست باز شدن مینی‌اپ در نسخه وب، از برقرار بودن موارد زیر مطمئن شوید:

- هدر *Content Security Policy (CSP)* با directiveای به اسم `frame-src` و مقدار `https://*.bale.ai` موجود باشد. برای مثال:

```
Content-Security-Policy: frame-src 'https://*.bale.ai';
```

- هدر `X-Frame-Options` وجود نداشته باشد.

نمایش هشدار آپدیت به کاربران با کلاینت قدیمی

ممکن است که ورژن اپلیکیشن بله بعضی از کاربران شما قدیمی باشد و از مینی‌اپ پشتیبانی نکند. با این حال اگر آن دسته از کاربران با کلاینت قدیمی، قصد باز کردن مینی‌اپ از طریق **Inline Button** یا **Reply Keyboard** **Markup** را داشته باشند، متوجه قدیمی بودن کلاینت خود نمی‌شوند و احتمالا فکر می‌کنند که مشکل از بازوی شماست. برای حل این مشکل می‌توانید از فیلد isMiniAppSupported در WebApp استفاده کنید تا بتوانید به آن‌ها هشدار دهید که اپلیکیشن خود را به‌روزرسانی کنند.

مشکل پرداخت در نسخه‌های قدیمی اندروید

در برخی نسخه‌های قدیمی‌تر اندروید، کاربران ممکن است با مشکلاتی در انجام فرآیند پرداخت درون مینی‌اپ مواجه شوند. برای حل این مشکل می‌توانید از Memory Router استفاده کنید

React Example:

```
import { MemoryRouter } from "react-router-dom";

<MemoryRouter>{/* Routes */}</MemoryRouter>;
```

Vue Example:

```
import { createMemoryHistory, createRouter } from 'vue-router';

const router = createRouter({
 history: createMemoryHistory(),
 routes: [...]
});
```

آخرین به‌روزرسانی در ۱۵ مهر ۱۴۰۵

© بله ۱۴۰۴