

BANXX — API Specification

32 endpoints · UAT · 27 August 2026 · app v1.7.0

Every HTTP call the BANXX app (banxx-app_v1) makes at v1.7.0, transcribed from the app source. Every endpoint here has a live caller, and each page names it under “Called from”.

Base URLs

Constant	Base URL	What lives there
banxxBaseUrl	https://uat.banxx.com:8000/in/home/app/api	Auth, profile, notifications, banners, vouchers, product list, lender hand-off
chatBaseUrl	https://uat.banxx.com:8000/in/home/app/api/chat/	Charles — AI chat, voice command, AI advice, FAQ and support chat
loanNetBaseUrl	https://loannet.in:8000/api	Partner credit APIs — dedupe, leads, offers, gold and housing applications
banxxWebBaseUrl	https://uat.banxx.com/in/home/app	Not an API. The lender journeys the hand-off calls redirect into, opened in a web view. Same host as banxxBaseUrl but on the default port — these paths 404 on :8000.

Declared in lib/network/api_constants.dart. Connect and receive timeouts are 50 seconds on every call.

Index

#	Section	Endpoints	Base
1	Login	POST /send-otp	banxxBaseUrl
2	OTP Verification	POST /verifyotp POST /resend-otp/{app_id}	banxxBaseUrl
3	Signup — Personal Details Registration	POST /v2/personal-data/{credit_uid}	banxxBaseUrl
4	Session management	POST /refresh-token	banxxBaseUrl
5	Dashboard	GET /v2/personal-data/{credit_uid} GET /user/{credit_uid} GET /product-list	banxxBaseUrl
6	Offers and Credit — partner APIs	POST /partner/dedupe POST /v2/partner/create-lead POST /v2/partner/update-lead/{lead_id} GET /partner/get-offers/{lead_id} POST /v2/partner/gold-loans POST /v2/partner/housing-loan	loanNetBaseUrl
7	Loan status and lender hand-off	GET /partner/get-summary/{creditprospect_uuid} POST /gold-loans/{creditapp_uuid} POST /housing-loan/{creditapp_uuid}	banxxBaseUrl
8	Notifications	GET /banxx/notifications GET /banxx/notifications/fetch/{notification_id}	banxxBaseUrl
9	Promotional banners	GET /banxx/banners	banxxBaseUrl
10	AI Chat and Voice Command	POST chat/start POST chat/message GET chat/questions	chatBaseUrl
11	AI Advice	POST chat/advice	chatBaseUrl
12	Deals and Offers (vouchers)	POST /vouchagram/getbrands	banxxBaseUrl
13	Support — Help and Chat	GET chat/faq POST chat/faq-feedback GET chat/conversations GET chat/history/{conversation_id}	chatBaseUrl
14	Settings	GET /banxx/notification-preferences POST /banxx/notification-preferences POST /banxx-app/delete-account	banxxBaseUrl

The Base column names the constant; its URL is on page 1.

1. Login

Base URL: <https://uat.banxx.com:8000/in/home/app/api>

Send OTP

Endpoint	POST /send-otp
Full URL	https://uat.banxx.com:8000/in/home/app/api/send-otp
Called from	Login screen — BanxxAuthService.sendOtp()
Request headers	Content-Type: application/json Accept: application/json
Request body	mobile_phone_number required The 10-digit mobile number. Sent as a JSON number when it parses as one, otherwise as a string. <pre>{ "mobile_phone_number": 9876543210 }</pre>
Response body	<pre>{ "success": "true", "app_id": "a91f4c7e-2b18-4d55-9f30-6c1e8ab47d02", "message": "OTP sent successfully" }</pre> success is read as a string or a boolean. app_id is carried into /verifyotp and /resend-otp.
Errors	The message shown to the user is taken from `message`, or from the first entry of a Laravel `errors` map when `message` is a generic validation sentence. Fallback: "Failed to send OTP. Please try again."
cURL	<pre>curl -X POST 'https://uat.banxx.com:8000/in/home/app/api/send-otp' \ -H 'Content-Type: application/json' \ -H 'Accept: application/json' \ -d '{"mobile_phone_number": 9876543210}'</pre>

2. OTP Verification

Base URL: <https://uat.banxx.com:8000/in/home/app/api>

Verify OTP

Endpoint	POST /verifyotp
Full URL	https://uat.banxx.com:8000/in/home/app/api/verifyotp
Called from	OTP screen — BanxxAuthService.verifyOtp()
Request headers	Content-Type: application/json
Request body	code required The OTP. Sent as a JSON number when it parses as one. credit_prospect_id required The app_id from /send-otp. <pre>{ "code": 123456, "credit_prospect_id": "a91f4c7e-2b18-4d55-9f30-6c1e8ab47d02" }</pre>
Response body	<pre>{ "success": "true", "message": "OTP verified", "token": "eyJ0eXAiOiJKV1QiLCJhbGciOi...", "greeting": "Hi Rahul", "lastLoginDateTime": "2026-08-19T06:42:11Z", "data": { "credituid": "cp_8f21d0b4e7a3", "greeting": "Hi Rahul", "lastLoginDateTime": "2026-08-19T06:42:11Z" } }</pre> token and data.credituid are written to secure storage and key every authenticated call thereafter. greeting and lastLoginDateTime are read from the root or from data, whichever carries them.
cURL	<pre>curl -X POST 'https://uat.banxx.com:8000/in/home/app/api/verifyotp' \ -H 'Content-Type: application/json' \ -d '{"code": 123456, "credit_prospect_id": "a91f4c7e-2b18-4d55-9f30-6c1e8ab47d02"}'</pre>

Resend OTP

Endpoint	POST /resend-otp/{app_id}
Full URL	https://uat.banxx.com:8000/in/home/app/api/resend-otp/{app_id}
Called from	OTP screen, Resend link — BanxxAuthService.resendOtp()
Request headers	Content-Type: application/json
Path parameters	app_id required The login attempt id returned by /send-otp.
Request body	None.
Response body	<pre>{ "success": "true", "app_id": "a91f4c7e-2b18-4d55-9f30-6c1e8ab47d02", "message": "OTP resent successfully" }</pre> A new app_id in this response replaces the one held by the screen.
cURL	<pre>curl -X POST 'https://uat.banxx.com:8000/in/home/app/api/resend-otp/{app_id}' \ -H 'Content-Type: application/json'</pre>

3. Signup — Personal Details Registration

Base URL: <https://uat.banxx.com:8000/in/home/app/api>

Save personal data

Endpoint	POST /v2/personal-data/{credit_uid}
Full URL	https://uat.banxx.com:8000/in/home/app/api/v2/personal-data/{credit_uid}
Called from	Registration screen and Settings > My Account — BanxxAuthService.savePersonalData()
Request headers	Authorization: Bearer <token> Content-Type: application/json
Path parameters	credit_uid required The credit prospect uid.
Request body	first_name required last_name required email required birth_date required ISO date, YYYY-MM-DD <pre>{ "first_name": "Rahul", "last_name": "Sharma", "email": "rahul.sharma@example.com", "birth_date": "1992-04-17" }</pre>
Response body	<pre>{ "success": "true", "message": "Details saved" }</pre> HTTP 200 with success:false is treated as a failure and `message` is shown to the user — that is how "The email has already been taken." reaches the screen. A success drops the app's cached profile so every screen re-reads the new values.
cURL	<pre>curl -X POST \ 'https://uat.banxx.com:8000/in/home/app/api/v2/personal-data/cp_8f21d0b4e7a3' \ -H 'Authorization: Bearer <token>' \ -H 'Content-Type: application/json' \ -d '{ "first_name": "Rahul", "last_name": "Sharma", "email": "rahul.sharma@example.com", "birth_date": "1992-04-17" }'</pre>

4. Session management

Base URL: <https://uat.banxx.com:8000/in/home/app/api>

Refresh token

Endpoint	POST /refresh-token
Full URL	https://uat.banxx.com:8000/in/home/app/api/refresh-token
Called from	Banxx401RefreshInterceptor, via TokenRefreshCoordinator — fired automatically on any 401 from a Banxx endpoint, then the failed request is retried once with the new token.
Request headers	Authorization: Bearer <current token> X-CSRF-TOKEN: (sent as an empty value)
Request body	None.
Response body	<pre>{ "success": true, "token": "eyJ0eXAiOiJKV1QiLCJhbGciOi...", "message": "Token refreshed" }</pre> success is read as a real boolean here. An empty or failed response signs the user out. This call goes out on a bare client with no interceptor, so a 401 on the refresh itself never recurses.
cURL	<pre>curl -X POST 'https://uat.banxx.com:8000/in/home/app/api/refresh-token' \ -H 'Authorization: Bearer <token>' \ -H 'X-CSRF-TOKEN;'</pre>

5. Dashboard

Base URL: <https://uat.banxx.com:8000/in/home/app/api>

Get personal data

Endpoint	GET /v2/personal-data/{credit_uid}
Full URL	https://uat.banxx.com:8000/in/home/app/api/v2/personal-data/{credit_uid}
Called from	Prefetched at launch and after OTP verification, then shared from an in-memory cache by the dashboard, Charles card, nav bar, sidebar, Settings, Activity, Pay Bills, the offers bloc and the lead form — BanxxAuthService.getPersonalData()
Request headers	Authorization: Bearer <token>
Path parameters	credit_uid required The credit prospect uid returned as data.credituid by /verifyotp.
Response body	<pre>{ "data": { "first_name": "Rahul", "last_name": "Sharma", "email": "rahul.sharma@example.com", "birth_date": "1992-04-17", "creditprospect_uuid": "cp_8f21d0b4e7a3", "messageTitle": "3 offers are waiting for you", "messageBody": "Two personal loans and a gold loan matched your profile." } }</pre> The same fields are also accepted at the root. messageTitle and messageBody are the authoritative copy for the Charles card. A user with no profile yet returns HTTP 200 with an empty body — that is "not onboarded", not a failure, and is not cached.
cURL	<pre>curl -X GET \ 'https://uat.banxx.com:8000/in/home/app/api/v2/personal-data/cp_8f21d0b4e7a3' \ -H 'Authorization: Bearer <token>'</pre>

Get user record

Endpoint	GET /user/{credit_uid}
Full URL	https://uat.banxx.com:8000/in/home/app/api/user/{credit_uid}
Called from	Prefetched at launch; read by the offers bloc, the Applications tab and the loan status tracker — BanxxAuthService.getUserDetails() and LoanApplicationService.fetchUserRecord()
Request headers	Authorization: Bearer <token> X-CSRF-TOKEN: (sent as an empty value by the loan-status caller)
Path parameters	credit_uid required The creditprospect_uuid.
Response body	<pre>{ "data": { "creditapp_uuid": "ca_5b7290ffe14c", "creditprospect_uuid": "cp_8f21d0b4e7a3", "pl_completed": 1, "gl_completed": 0, "hl_completed": 1, "tin": "ABCDE1234F", "pincode": "400072", "monthly_income": "85000", "picked_lenders": "Muthoot,IIFL", "knockout_lenders": "HDFC", "updated_at": "2026-08-20T11:04:52Z" } }</pre> Never cached across calls: the *_completed flags change the moment an application is submitted. creditapp_uuid is what keys the two lender hand-off calls, so this request always runs before them. Flags are accepted as 1, "1" or true.
cURL	<pre>curl -X GET 'https://uat.banxx.com:8000/in/home/app/api/user/cp_8f21d0b4e7a3' \ -H 'Authorization: Bearer <token>'</pre>

Product list

Endpoint	GET /product-list
-----------------	-------------------

Full URL	https://uat.banxx.com:8000/in/home/app/api/product-list
Called from	Credit-card offers card, insurance card and the product selection screens — ProductListApiService
Request headers	Authorization: Bearer <token> Accept: application/json
Response body	<pre>{ "data": [{ "category_name": "Insurance", "long_description": "Cover for you and your family", "short_description": "Insurance", "category_img_thumb": "https://cdn.banxx.com/cat/insurance.png", "subcategories": [{ "category_name": "Health Insurance", "long_description": "Cashless cover at 7,000+ hospitals", "category_img_thumb": "https://cdn.banxx.com/ins/health.png", "subcategories": [{ "category_name": "FlashAid", "category_img_thumb": "https://cdn.banxx.com/ins/flashaid.png", "utm_link": "https://partner.example.com/apply?utm_source=banxx" }] }] }] }</pre> A bare top-level array is also accepted. Cached for the session; a failed fetch is remembered so a card can tell an API failure apart from a genuinely empty catalogue. Insurance is located by a category_name containing "insurance"; a subcategory with no children is treated as the offer itself.
cURL	<pre>curl -X GET 'https://uat.banxx.com:8000/in/home/app/api/product-list' \ -H 'Authorization: Bearer <token>' \ -H 'Accept: application/json'</pre>

6. Offers and Credit — partner APIs

Base URL: <https://loanet.in:8000/api>

Dedupe check

Endpoint	POST /partner/dedupe
Full URL	https://loanet.in:8000/api/partner/dedupe
Called from	Offers bloc, before the lead form is offered — OffersRemoteDataSource.checkDedupe()
Request headers	apikey: <partner API key> Content-Type: application/json
Request body	mobileNumber required string { "mobileNumber": "9876543210" }
Response body	{ "success": "true", "leadId": "LN2026081900412" } When no lead exists, success is not "true" and no leadId is returned — the app then shows the lead creation form.
cURL	curl -X POST 'https://loanet.in:8000/api/partner/dedupe' \ -H 'apikey: <partner API key>' \ -H 'Content-Type: application/json' \ -d '{"mobileNumber": "9876543210"}'

Create lead

Endpoint	POST /v2/partner/create-lead
Full URL	https://loanet.in:8000/api/v2/partner/create-lead
Called from	Personal loan lead form — OffersRemoteDataSource.createLead()
Request headers	apikey: <partner API key> Content-Type: application/json
Request body	mobileNumber required string firstName required string lastName required string pan required string dob required YYYY-MM-DD email required string pincode required string monthlyIncome required number employmentStatus required number consumerConsentDate required built by the app consumerConsentIp required built by the app employerName optional sent when employmentStatus is 1 officePincode optional sent when employmentStatus is 1 businessRegistrationType optional number residenceType optional number businessCurrentTurnover optional number businessYears optional number businessAccount optional number { "mobileNumber": "9876543210", "firstName": "Rahul", "lastName": "Sharma", "pan": "ABCDE1234F", "dob": "1992-04-17", "email": "rahul.sharma@example.com", "pincode": "400072", "monthlyIncome": 85000, "employmentStatus": 1, "consumerConsentDate": "2026-08-21 10:15:00", "consumerConsentIp": "49.36.180.14" }
Response body	{ "success": "true", "leadId": "LN2026082100518" } "id" is accepted in place of "leadId". A refusal comes back HTTP 200 with no lead id and the reason in the body; the app raises that reason verbatim rather than reporting a generic failure.

cURL	<pre>curl -X POST 'https://loanet.in:8000/api/v2/partner/create-lead' \ -H 'apikey: <partner API key>' \ -H 'Content-Type: application/json' \ -d '{ "mobileNumber": "9876543210", "firstName": "Rahul", "lastName": "Sharma", "pan": "ABCDE1234F", "dob": "1992-04-17", "email": "rahul.sharma@example.com", "pincode": "400072", "monthlyIncome": 85000, "employmentStatus": 1, "consumerConsentDate": "2026-08-21 10:15:00", "consumerConsentIp": "49.36.180.14" }'</pre>
-------------	--

Update lead

Endpoint	POST /v2/partner/update-lead/{lead_id}																																																						
Full URL	https://loanet.in:8000/api/v2/partner/update-lead/{lead_id}																																																						
Called from	Lead form, when a lead already exists — OffersRemoteDataSource.updateLead()																																																						
Request headers	apikey: <partner API key> Content-Type: application/json																																																						
Path parameters	lead_id required Taken from the event when supplied, otherwise the lead id in local storage. If neither is available the call is not made and the screen shows "No lead ID found. Please create a lead first."																																																						
Request body	The complete lead payload — the same fields as create-lead, not only the changed ones. <table><tr><td>mobileNumber</td><td>required</td><td>string</td></tr><tr><td>firstName</td><td>required</td><td>string</td></tr><tr><td>lastName</td><td>required</td><td>string</td></tr><tr><td>pan</td><td>required</td><td>string</td></tr><tr><td>dob</td><td>required</td><td>YYYY-MM-DD</td></tr><tr><td>email</td><td>required</td><td>string</td></tr><tr><td>pincode</td><td>required</td><td>string</td></tr><tr><td>monthlyIncome</td><td>required</td><td>number</td></tr><tr><td>employmentStatus</td><td>required</td><td>number</td></tr><tr><td>consumerConsentDate</td><td>required</td><td>built by the app</td></tr><tr><td>consumerConsentIp</td><td>required</td><td>built by the app</td></tr><tr><td>employerName</td><td>optional</td><td>sent when employmentStatus is 1</td></tr><tr><td>officePincode</td><td>optional</td><td>sent when employmentStatus is 1</td></tr><tr><td>businessRegistrationType</td><td>optional</td><td>number</td></tr><tr><td>residenceType</td><td>optional</td><td>number</td></tr><tr><td>businessCurrentTurnover</td><td>optional</td><td>number</td></tr><tr><td>businessYears</td><td>optional</td><td>number</td></tr><tr><td>businessAccount</td><td>optional</td><td>number</td></tr></table>	mobileNumber	required	string	firstName	required	string	lastName	required	string	pan	required	string	dob	required	YYYY-MM-DD	email	required	string	pincode	required	string	monthlyIncome	required	number	employmentStatus	required	number	consumerConsentDate	required	built by the app	consumerConsentIp	required	built by the app	employerName	optional	sent when employmentStatus is 1	officePincode	optional	sent when employmentStatus is 1	businessRegistrationType	optional	number	residenceType	optional	number	businessCurrentTurnover	optional	number	businessYears	optional	number	businessAccount	optional	number
mobileNumber	required	string																																																					
firstName	required	string																																																					
lastName	required	string																																																					
pan	required	string																																																					
dob	required	YYYY-MM-DD																																																					
email	required	string																																																					
pincode	required	string																																																					
monthlyIncome	required	number																																																					
employmentStatus	required	number																																																					
consumerConsentDate	required	built by the app																																																					
consumerConsentIp	required	built by the app																																																					
employerName	optional	sent when employmentStatus is 1																																																					
officePincode	optional	sent when employmentStatus is 1																																																					
businessRegistrationType	optional	number																																																					
residenceType	optional	number																																																					
businessCurrentTurnover	optional	number																																																					
businessYears	optional	number																																																					
businessAccount	optional	number																																																					
Response body	Not read by the app.																																																						
CURL	curl -X POST 'https://loanet.in:8000/api/v2/partner/update-lead/LN2026082100518' \ -H 'apikey: <partner API key>' \ -H 'Content-Type: application/json' \ -d '{ ...the full lead payload... }'																																																						

Get offers

Endpoint	GET /partner/get-offers/{lead_id}
Full URL	https://loanet.in:8000/api/partner/get-offers/{lead_id}
Called from	Personal / gold / housing loan offer carousels — OffersRemoteDataSource.getOffers()
Request headers	apikey: <partner API key> Content-Type: application/json
Path parameters	lead_id required The lead id, or the creditprospect_uuid when the personal loan is already completed.

Response body	<pre>{ "success": "true", "leadId": "LN2026082100518", "offers": [{ "lenderId": 12, "lenderName": "Muthoot Finance", "lenderLogo": "https://cdn.loannet.in/lenders/muthoot.png", "lenderLogoDark": "https://cdn.loannet.in/lenders/muthoot-dark.png", "description": "Gold loan at attractive rates", "offerAmountUpTo": "500000", "offerTenure": "12 months", "offerInterestRate": "12.5%", "offerProcessingFees": "1%", "offerLink": "https://partner.example.com/apply/690289", "kfs": "https://cdn.loannet.in/kfs/muthoot-gold.pdf", "status": "Accepted", "lenderLeadId": "MU-88213", "lenderLeadType": "gold" }], "unmatchedOffers": [] }</pre> offers[] renders the cards; unmatchedOffers[] renders the "not matched" list. Field names are accepted in both spellings (lenderLogo / LenderLogo, lenderId / LenderId). The KFS link is read from kfs, Kfs, KFS, kfsUrl, kfsLink or keyFactStatement, and offerLink / applyUrl / redirectionUrl all work as the apply URL.
cURL	<pre>curl -X GET 'https://loannet.in:8000/api/partner/get-offers/LN2026082100518' \ -H 'apikey: <partner API key>' \ -H 'Content-Type: application/json'</pre>

Gold loan application

Endpoint	POST /v2/partner/gold-loans
Full URL	https://loannet.in:8000/api/v2/partner/gold-loans
Called from	Gold loan form — OffersRemoteDataSource.createGoldLoan()
Request headers	apikey: <partner API key> Content-Type: application/json
Request body	<pre>mobileNumber required string firstName required string lastName required string pan required string email required string pincode required string loanAmount required number consumerConsentDate required built by the app consumerConsentIp required built by the app { "mobileNumber": "9876543210", "firstName": "Rahul", "lastName": "Sharma", "pan": "ABCDE1234F", "email": "rahul.sharma@example.com", "pincode": "400072", "loanAmount": 250000, "consumerConsentDate": "2026-08-21 10:15:00", "consumerConsentIp": "49.36.180.14" }</pre>
Response body	The same structure as /partner/get-offers — success, message, offers[], unmatchedOffers[] — carrying the lenders matched for this application and their opening status. Each offer carries its own live `status`, which is why the app reads the per-lender statuses straight off this response and makes no second call for them. A rejection answers HTTP 200 with success:false and no offers; the app raises `message` so the user is told why instead of being shown an empty card.

cURL	<pre>curl -X POST 'https://loannet.in:8000/api/v2/partner/gold-loans' \ -H 'apikey: <partner API key>' \ -H 'Content-Type: application/json' \ -d '{ "mobileNumber": "9876543210", "firstName": "Rahul", "lastName": "Sharma", "pan": "ABCDE1234F", "email": "rahul.sharma@example.com", "pincode": "400072", "loanAmount": 250000, "consumerConsentDate": "2026-08-21 10:15:00", "consumerConsentIp": "49.36.180.14" }'</pre>
-------------	--

Housing loan application

Endpoint	POST /v2/partner/housing-loan
Full URL	https://loannet.in:8000/api/v2/partner/housing-loan
Called from	Housing loan form — OffersRemoteDataSource.createHousingLoan()
Request headers	apikey: <partner API key> Content-Type: application/json
Request body	mobileNumber required string firstName required string lastName required string pan required string dob required YYYY-MM-DD email required string pincode required string monthlyIncome required number housingLoanAmount required number propertyType required string – House Flat Plot consumerConsentDate required built by the app consumerConsentIp required built by the app <pre>{ "mobileNumber": "9876543210", "firstName": "Rahul", "lastName": "Sharma", "pan": "ABCDE1234F", "dob": "1992-04-17", "email": "rahul.sharma@example.com", "pincode": "400072", "monthlyIncome": 85000, "housingLoanAmount": 4500000, "propertyType": "House", "consumerConsentDate": "2026-08-21 10:15:00", "consumerConsentIp": "49.36.180.14" }</pre>
Response body	The same structure as /partner/get-offers and the gold loan call, and the same HTTP 200 rejection handling.
cURL	curl -X POST 'https://loannet.in:8000/api/v2/partner/housing-loan' \ -H 'apikey: <partner API key>' \ -H 'Content-Type: application/json' \ -d '{ ...the housing payload above... }'

7. Loan status and lender hand-off

Base URL: <https://uat.banxx.com:8000/in/home/app/api>

Offer summary

Endpoint	GET /partner/get-summary/{creditprospect_uuid}
Full URL	https://uat.banxx.com:8000/in/home/app/api/partner/get-summary/{creditprospect_uuid}
Called from	Loan status tracker — LoanApplicationService.fetchSummary()
Request headers	apikey: <partner API key> X-CSRF-TOKEN: (sent as an empty value) Note: authenticated by the partner apikey, not the bearer token — the bearer is deliberately not sent on this one.
Path parameters	creditprospect_uuid required From the /user record.
Response body	<pre>{ "summary": { "offersTotal": 3, "maxLoanAmount": "500000", "minMPR": "1.04", "maxMPR": "1.75" }, "redirectionUrl": "https://uat.banxx.com/in/home/app/offers/cp_8f21d0b4e7a3" }</pre>
cURL	<pre>curl -X GET \ 'https://uat.banxx.com:8000/in/home/app/api/partner/get-summary/cp_8f21d0b4e7a3' \ -H 'apikey: <partner API key>' \ -H 'X-CSRF-TOKEN:'</pre>

Gold loan hand-off

Endpoint	POST /gold-loans/{creditapp_uuid}
Full URL	https://uat.banxx.com:8000/in/home/app/api/gold-loans/{creditapp_uuid}
Called from	Loan status tracker and the gold offer card's Continue button — LoanApplicationService.submitGoldLoan()
Request headers	Authorization: Bearer <token> X-CSRF-TOKEN: (sent as an empty value)
Path parameters	creditapp_uuid required creditapp_uuid from GET /user/{credit_uid}. Only that record hands it out, so /user always runs first.
Request body	None.
Response body	<pre>{ "status": "true", "applicationId": 690289, "lenderId": 12, "redirect": "lenders-transaction/muthootgold?applicationId=690289" }</pre> Reads as a write but is idempotent — repeated calls return the same applicationId — so it is safe on every status refresh. `status` is "true" even when nothing was created, so applicationId > 0 is the only usable success signal. `redirect` is a path relative to https://uat.banxx.com/in/home/app (the same host on the default port; these paths 404 on :8000). The lender slug is read out of that path — muthootgold above — rather than from a lenderId map.
cURL	<pre>curl -X POST 'https://uat.banxx.com:8000/in/home/app/api/gold-loans/ca_5b7290ffe14c' \ -H 'Authorization: Bearer <token>' \ -H 'X-CSRF-TOKEN:'</pre>

Housing loan hand-off

Endpoint	POST /housing-loan/{creditapp_uuid}
Full URL	https://uat.banxx.com:8000/in/home/app/api/housing-loan/{creditapp_uuid}
Called from	Loan status tracker and the housing offer card's Continue button — LoanApplicationService.submitHousingLoan()
Request headers	Authorization: Bearer <token> X-CSRF-TOKEN: (sent as an empty value)
Path parameters	creditapp_uuid required creditapp_uuid from GET /user/{credit_uid}.

Request body	None.
Response body	Identical in shape to the gold hand-off — status, applicationId, lenderId, redirect — and read the same way.
cURL	<pre>curl -X POST 'https://uat.banxx.com:8000/in/home/app/api/housing-loan/ca_5b7290ffe14c' \ -H 'Authorization: Bearer <token>' \ -H 'X-CSRF-TOKEN;'</pre>

8. Notifications

Base URL: <https://uat.banxx.com:8000/in/home/app/api>

Notification list

Endpoint	GET /banxx/notifications									
Full URL	https://uat.banxx.com:8000/in/home/app/api/banxx/notifications									
Called from	Notification dropdown and the notifications screen, and once per navbar mount for the unread badge — NotificationsService.fetchPage() / refreshUnreadCount()									
Request headers	Authorization: Bearer <token> Accept: application/json X-CSRF-TOKEN: (sent as an empty value)									
Query parameters	<table><tr><td>page</td><td>optional</td><td>1-based page number. Default 1.</td></tr><tr><td>per_page</td><td>optional</td><td>Rows per page. Default 20; clamped by the app to 1..100 because the server caps at 100 and clamps silently.</td></tr><tr><td>opened</td><td>optional</td><td>true / false to filter by read state. Omitted for the mixed list the dropdown shows; sent as false for the badge-only call.</td></tr></table>	page	optional	1-based page number. Default 1.	per_page	optional	Rows per page. Default 20; clamped by the app to 1..100 because the server caps at 100 and clamps silently.	opened	optional	true / false to filter by read state. Omitted for the mixed list the dropdown shows; sent as false for the badge-only call.
page	optional	1-based page number. Default 1.								
per_page	optional	Rows per page. Default 20; clamped by the app to 1..100 because the server caps at 100 and clamps silently.								
opened	optional	true / false to filter by read state. Omitted for the mixed list the dropdown shows; sent as false for the badge-only call.								
Response body	<pre>{ "data": [{ "notificationId": "ntf_00194823", "category": "loan_alerts", "title": "Gold loan application received", "text": "Your gold loan application has been sent to Muthoot...", "timestamp": "2026-08-20T09:15:40Z", "opened": false }], "pagination": { "current_page": 1, "last_page": 4, "has_more": true }, "total_unopened": 7}</pre> total_unopened is account-wide, not per page — it is what the bell badge shows, which is why the badge refresh asks for per_page=1. `opened` is accepted as a boolean, 0/1 or "true"/"1". has_more is trusted when sent; otherwise current_page < last_page is used. `text` is truncated here and complete in the fetch response. HTTP 400 means the account has no inbox yet — the app reads that as an empty list with a zero badge, not as an error.									
cURL	<pre>curl -X GET \ 'https://uat.banxx.com:8000/in/home/app/api/banxx/notifications?page=1&per_page=20' \ -H 'Authorization: Bearer <token>' \ -H 'Accept: application/json' \ -H 'X-CSRF-TOKEN;' # unread badge only curl -X GET 'https://uat.banxx.com:8000/in/home/app/api/banxx/notifications? page=1&per_page=1&opened=false' \ -H 'Authorization: Bearer <token>' \ -H 'Accept: application/json'</pre>									

Fetch one notification

Endpoint	GET /banxx/notifications/fetch/{notification_id}
Full URL	https://uat.banxx.com:8000/in/home/app/api/banxx/notifications/fetch/{notification_id}
Called from	Notification detail screen — NotificationsService.openNotification()
Request headers	Authorization: Bearer <token> Accept: application/json X-CSRF-TOKEN: (sent as an empty value)
Path parameters	notification_id required notificationId from the list response.

Response body	<pre>{ "data": { "notificationId": "ntf_00194823", "category": "loan_alerts", "title": "Gold loan application received", "text": "Your gold loan application has been sent to Muthoot Finance and is currently under review. You will be notified once a decision is made.", "timestamp": "2026-08-20T09:15:40Z", "opened": true } }</pre> Calling this marks the notification opened server-side — this request is what "reading" a notification means. The app drops its unread badge by one at the same moment rather than waiting for the next list fetch. A failure here costs the full body, not the screen: the detail view is already showing the list's copy of the title and text.
cURL	<pre>curl -X GET \ 'https://uat.banxx.com:8000/in/home/app/api/banxx/notifications/fetch/ntf_00194823' \ -H 'Authorization: Bearer <token>' \ -H 'Accept: application/json' \ -H 'X-CSRF-TOKEN:'</pre>

9. Promotional banners

Base URL: <https://uat.banxx.com:8000/in/home/app/api>

Banner list

Endpoint	GET /banxx/banners
Full URL	https://uat.banxx.com:8000/in/home/app/api/banxx/banners
Called from	Dashboard only, through BannersService.refresh(), at three points: - on mount dashboard_screen.dart initState - on foreground returning to the app; cheap, the TTL usually serves it - connection restored refresh(force: true) — the mount fetch is the one the dead connection ate, so cache is not worth serving Cached for 10 minutes in between. One request fills all four placements, and this is the app's only banner call.
Request headers	Authorization: Bearer <token> Accept: application/json X-CSRF-TOKEN: (sent as an empty value)
Query parameters	active sent as 'true' on every call — only live banners are wanted.
Response body	<pre>{ "success": true, "data": [{ "bannerId": "5a11ple0-0000-4000-8000-000000000001", "image_url": "https://<bucket>/banner/1787827554-zqjT3R34zZ.png", "name": "Popup 330x200", "title": "Popup banner", "text": "Opens over the dashboard on load, once a day", "type": "330x200", "active": true, "timestamp": "2026-08-27T10:05:00+05:30" }], "count": 5 }</pre> success must be a real boolean true; anything else leaves the currently loaded banners in place rather than blanking them. bannerId is also accepted as banner_id or id; image_url as imageUrl. A record with no id is dropped. `active` absent means shown (the list is already filtered). `timestamp` sorts the list newest first, which is what decides the banner a single-banner slot shows.
Placement	The app picks the placement from `type` alone — nothing else routes it. The server enum is 284x167 330x200 354x227 354x100 popup. 354x227 Dashboard top: full-width card under Charles. A second banner at this size is placed lower, below the opening widget. 330x200 Popup over the dashboard, once per banner per day. The alias "popup" lands here too. 284x167 Slides inside the Personal Loan Offers carousel, spread through it (two offers, a banner, two offers, ...). 354x100 Sticky strip above the bottom navigation bar; several page through with arrows. Matching is lenient about how the size is written — "354 X 227", "354x227" and "354x227" all resolve. A type this build has no slot for is parsed and logged but never drawn, and so is a banner with no artwork or one switched off. Banners are cleared on logout.
Live, not sample	Every build calls this endpoint. The bundled sample response (lib/features/banners/data/sample_banners.dart) is opt-in through --dart-define=BANNER_SAMPLE_DATA=true, defaults to false, and is ANDed with kDebugMode — so a release build talks to the real API even with the define set. A slot with no banner at its size renders nothing; that is "no banner published at that size", not a failure. While the sample is on, the load logs a "SAMPLE DATA" line every time and the artwork is stamped SAMPLE BANNER — worth checking first when a banner published in the panel does not appear.
cURL	<pre>curl -X GET 'https://uat.banxx.com:8000/in/home/app/api/banxx/banners?active=true' \ -H 'Authorization: Bearer <token>' \ -H 'Accept: application/json' \ -H 'X-CSRF-TOKEN;'</pre>

10. AI Chat and Voice Command

Base URL: <https://uat.banxx.com:8000/in/home/app/api/chat/>

Start conversation

Endpoint	POST chat/start
Full URL	https://uat.banxx.com:8000/in/home/app/api/chat/start
Called from	AI chat popup and the voice command card, on open — AiChatPopup._initializeChat(), VoiceCommandCard._initializeChat(), SupportScreen._startNewThread()
Request headers	apikey: <partner API key> Authorization: Bearer <token> Content-Type: application/json
Request body	{} An empty JSON object. No phone number is sent.
Response body	{ "conversation_id": "cnv_7d4a19b2" } conversation_id is held for the life of the popup and sent with every message. A thread is titled server-side from its first message, so the conversation list is only worth re-reading after a message lands.
cURL	curl -X POST 'https://uat.banxx.com:8000/in/home/app/api/chat/start' \ -H 'apikey: <partner API key>' \ -H 'Authorization: Bearer <token>' \ -H 'Content-Type: application/json' \ -d '{}'

Send message

Endpoint	POST chat/message
Full URL	https://uat.banxx.com:8000/in/home/app/api/chat/message
Called from	AI chat popup, voice command card and the support chat tab
Request headers	apikey: <partner API key> Authorization: Bearer <token> Content-Type: application/json
Request body	conversation_id required From chat/start. message required Free text – typed or dictated. { "conversation_id": "cnv_7d4a19b2", "message": "How much can I borrow against my gold?" }
Response body	{ "message": "Based on the current rate, you could borrow up to **75%** of your gold's value..." } Rendered as markdown in the chat bubble. A missing `message` shows "No response". The request is cancellable — the Stop button drops it, and the message itself stays in the thread because it did reach the server.
cURL	curl -X POST 'https://uat.banxx.com:8000/in/home/app/api/chat/message' \ -H 'apikey: <partner API key>' \ -H 'Authorization: Bearer <token>' \ -H 'Content-Type: application/json' \ -d '{ "conversation_id": "cnv_7d4a19b2", "message": "How much can I borrow against my gold?" }'

Prepared questions

Endpoint	GET chat/questions
Full URL	https://uat.banxx.com:8000/in/home/app/api/chat/questions
Called from	AI chat popup and the support chat tab, on open
Request headers	apikey: <partner API key> Authorization: Bearer <token> (omitted when no token is held) Content-Type: application/json

Response body	<pre>{ "questions": ["What is my credit score?", "Which loan offers do I qualify for?", "How do I improve my eligibility?"] }</pre> Answers without a bearer token — the list is the same for everyone.
cURL	<pre>curl -X GET 'https://uat.banxx.com:8000/in/home/app/api/chat/questions' \ -H 'apikey: <partner API key>'</pre>

11. AI Advice

Base URL: <https://uat.banxx.com:8000/in/home/app/api/chat/>

AI advice

Endpoint	POST chat/advice
Full URL	https://uat.banxx.com:8000/in/home/app/api/chat/advice
Called from	AI advice card on the dashboard and the advice popup — AiAdviceItem.fetchLatest()
Request headers	apikey: <partner API key> Authorization: Bearer <token> Content-Type: application/json
Request body	<pre>{}</pre> An empty JSON object. No phone number is sent.
Response body	<pre>[{ "short": "Move your EMI date", "long": "Your salary credits on the 1st but your EMI debits on the 28th. Shifting the EMI to the 3rd would avoid the shortfall you hit in June and July.", "category": "Loans", "saving": "1200" }]</pre> The first item is used. A single object at the root is also accepted, as is the list nested under any of advice, advice_list, advices, data, results or items. Field aliases: short / title / advice / message for the headline, long / description / detail / body for the text, category / type, saving / amount.
cURL	<pre>curl -X POST 'https://uat.banxx.com:8000/in/home/app/api/chat/advice' \ -H 'apikey: <partner API key>' \ -H 'Authorization: Bearer <token>' \ -H 'Content-Type: application/json' \ -d '{}'</pre>

12. Deals and Offers (vouchers)

Base URL: <https://uat.banxx.com:8000/in/home/app/api>

Get voucher brands

Endpoint	POST /vouchagram/getbrands
Full URL	https://uat.banxx.com:8000/in/home/app/api/vouchagram/getbrands
Called from	Deals & Offers strip on the dashboard — OffersRemoteDataSource.getDealsAndOffers()
Request headers	Authorization: Bearer <token> Accept: application/json X-CSRF-TOKEN: (sent as an empty value)
Request body	None.
Response body	<pre>{ "success": true, "data": "[{"BrandName\\":\\"Amazon\\", \\\"Brandtype\\":\\"E-Gift Voucher\\", \\\"Category\\":\\"Shopping\\", \\\"BrandImage\\":\\"https://cdn.vouchagram.com/brands/amazon.png\\"}]"</pre> <code>`data`</code> currently arrives as the brand list re-encoded as a JSON string. The app peels the envelope and decodes strings in a loop (up to four levels), so a bare array, or <code>`data`</code> holding the list itself, works too — a backend that stops double-encoding will not blank the strip. Fields read per brand: BrandName (title), Brandtype (tag and fallback description), Category, BrandImage (artwork), Descriptions (fallback). Never throws: an empty or failed response renders the strip's empty state. A 401 is refreshed and retried once like every other Banxx call.
cURL	<pre>curl -X POST 'https://uat.banxx.com:8000/in/home/app/api/vouchagram/getbrands' \ -H 'Authorization: Bearer <token>' \ -H 'Accept: application/json' \ -H 'X-CSRF-TOKEN;'</pre>

13. Support — Help and Chat

Base URL: <https://uat.banxx.com:8000/in/home/app/api/chat/>

FAQ

Endpoint	GET chat/faq
Full URL	https://uat.banxx.com:8000/in/home/app/api/chat/faq
Called from	Support screen, Help tab — <code>SupportScreen.fetchFaq()</code>
Request headers	apikey: <partner API key> Authorization: Bearer <token> (omitted when no token is held) Content-Type: application/json
Response body	<pre>{ "faq": [{ "title": "Loans", "subtitles": [{ "subtitle": "Gold loans", "questions": [{ "question": "How is my gold valued?", "answer": "The lender values your gold at the prevailing rate..." }] }] }] }</pre> Sometimes delivered as a JSON string rather than an object; the app detects and decodes that case. Categories with no questions are dropped. A failure puts a retry on the tab rather than an empty list.
cURL	<pre>curl -X GET 'https://uat.banxx.com:8000/in/home/app/api/chat/faq' \ -H 'apikey: <partner API key>' \ -H 'Content-Type: application/json'</pre>

FAQ feedback

Endpoint	POST chat/faq-feedback
Full URL	https://uat.banxx.com:8000/in/home/app/api/chat/faq-feedback
Called from	Support screen, the thumbs up / down on an FAQ article — SupportScreen.submitFaqFeedback()
Request headers	apikey: <partner API key> Authorization: Bearer <token> Content-Type: application/json
Request body	title required FAQ category title. subtitle required FAQ group subtitle. question required The article question. reaction required The reaction recorded. { "title": "Loans", "subtitle": "Gold loans", "question": "How is my gold valued?", "reaction": "up" }
Response body	{ "message": "Thanks for your feedback!" } `message` is shown to the user when present; otherwise the app shows "Thanks for your feedback!". One reaction per article per session.
cURL	curl -X POST 'https://uat.banxx.com:8000/in/home/app/api/chat/faq-feedback' \ -H 'apikey: <partner API key>' \ -H 'Authorization: Bearer <token>' \ -H 'Content-Type: application/json' \ -d '{ "title": "Loans", "subtitle": "Gold loans", "question": "How is my gold valued?", "reaction": "up" }'

Conversation list

Endpoint	GET chat/conversations
Full URL	https://uat.banxx.com:8000/in/home/app/api/chat/conversations
Called from	Support screen, the chat history drawer — SupportScreen._fetchConversations()
Request headers	apikey: <partner API key> Authorization: Bearer <token> Content-Type: application/json
Response body	<pre>{ "conversations": [{ "id": 4192, "thread_id": "cnv_7d4a19b2", "created_at": "2026-08-18T14:22:03Z", "first_message": "How much can I borrow against my gold?" }] }</pre> first_message titles the drawer row. Re-read after a message is sent, since the server titles a thread from its first message.
cURL	curl -X GET 'https://uat.banxx.com:8000/in/home/app/api/chat/conversations' \ -H 'apikey: <partner API key>' \ -H 'Authorization: Bearer <token>' \ -H 'Content-Type: application/json'

Conversation history

Endpoint	GET chat/history/{conversation_id}
Full URL	https://uat.banxx.com:8000/in/home/app/api/chat/history/{conversation_id}
Called from	Support screen, on opening a past thread — SupportScreen._fetchChatHistory()
Request headers	apikey: <partner API key> Authorization: Bearer <token> Content-Type: application/json
Path parameters	conversation_id required thread_id from the conversation list.
Response body	<pre>{ "messages": [{ "id": 88301, "role": "user", "content": "How much can I borrow against my gold?", "created_at": "2026-08-18T14:22:03Z" }, { "id": 88302, "role": "assistant", "content": "Based on the current rate...", "created_at": "2026-08-18T14:22:07Z" }] }</pre>
cURL	curl -X GET 'https://uat.banxx.com:8000/in/home/app/api/chat/history/cnv_7d4a19b2' \ -H 'apikey: <partner API key>' \ -H 'Authorization: Bearer <token>' \ -H 'Content-Type: application/json'

14. Settings

Base URL: <https://uat.banxx.com:8000/in/home/app/api>

Get notification preferences

Endpoint	GET /banxx/notification-preferences
Full URL	https://uat.banxx.com:8000/in/home/app/api/banxx/notification-preferences
Called from	Settings > Notifications, on open — BanxxAuthService.getNotificationPreferences()
Request headers	Authorization: Bearer <token> X-CSRF-TOKEN: (sent as an empty value)
Response body	<pre>{ "data": { "payments": true, "bills_due_reminders": true, "credit_card_alerts": false, "loan_alerts": true, "critical_alerts": true } }</pre> The same fields are also accepted at the root. Each is read as true only on a real boolean true — a 1 or "true" reads as off.
cURL	<pre>curl -X GET \ 'https://uat.banxx.com:8000/in/home/app/api/banxx/notification-preferences' \ -H 'Authorization: Bearer <token>' \ -H 'X-CSRF-TOKEN;'</pre>

Save notification preferences

Endpoint	POST /banxx/notification-preferences
Full URL	https://uat.banxx.com:8000/in/home/app/api/banxx/notification-preferences
Called from	Settings > Notifications, on each toggle — BanxxAuthService.saveNotificationPreferences()
Request headers	Authorization: Bearer <token> X-CSRF-TOKEN: (sent as an empty value) Content-Type: application/json
Request body	payments required boolean bills_due_reminders required boolean credit_card_alerts required boolean loan_alerts required boolean critical_alerts required boolean { "payments": true, "bills_due_reminders": true, "credit_card_alerts": false, "loan_alerts": true, "critical_alerts": true } All five are always sent, not just the one that changed.
Response body	{ "success": true } A "status" key is accepted in place of "success". If the response carries neither, any 2xx is taken as saved.
cURL	curl -X POST \ 'https://uat.banxx.com:8000/in/home/app/api/banxx/notification-preferences' \ -H 'Authorization: Bearer <token>' \ -H 'X-CSRF-TOKEN;' \ -H 'Content-Type: application/json' \ -d '{ "payments": true, "bills_due_reminders": true, "credit_card_alerts": false, "loan_alerts": true, "critical_alerts": true }'

Delete account

Endpoint	POST /banxx-app/delete-account
Full URL	https://uat.banxx.com:8000/in/home/app/api/banxx-app/delete-account
Called from	Settings > Delete account, after the confirmation dialog — BanxxAuthService.deleteAccount()
Request headers	Authorization: Bearer <token> X-CSRF-TOKEN: (sent as an empty value) Content-Type: application/json
Request body	phone required The signed-in mobile number. { "phone": "9876543210" }
Response body	{ "success": true, "message": "Account deleted" } A "status" key is accepted in place of "success"; accepted values are true and 1, and anything else is treated as a failure. On success the app wipes secure storage and every cache — profile, user record, banners, notification badge — and returns the user to login.
cURL	curl -X POST 'https://uat.banxx.com:8000/in/home/app/api/banxx-app/delete-account' \ -H 'Authorization: Bearer <token>' \ -H 'X-CSRF-TOKEN;' \ -H 'Content-Type: application/json' \ -d '{"phone": "9876543210"}'