



پُلاریس

Wallet as a service

تهیه کننده: استارتاپ پُلاریس

نسخه مستند: ۱.۱

تاریخ مستند: ۱۴۰۲/۰۹/۰۸

تاریخچه نگارش

تاریخ ویرایش	تغییرات ویرایش	ویرایش
۱۴۰۲/۰۹/۰۱	ایجاد سند	۱.۰
۱۴۰۲/۰۹/۰۸	بروزرسانی سرویس‌ها	۱.۱
۱۴۰۳/۰۴/۳۰	اضافه شدن سرویس‌های بومی ترون و TRC۲۰	۱.۲
۱۴۰۳/۰۵/۰۱	اضافه شدن سرویس‌های بومی سولانا و SPL	۱.۳
۱۴۰۳/۰۵/۰۷	اضافه شدن سرویس‌های توکن تون	۱.۴
۱۴۰۳/۰۶/۰۱	اضافه شدن سرویس‌های پولکادات، کاردانو و ریپل	۱.۵



فهرست مطالب

۱- مقدمه	۶
۲- اهداف مستند	۶
۳- موارد کلی	۶
• X-API-Key	۶
• برچسب کلید	۶
• امکان استخراج کلید	۷
• دسترسی به اطلاعات با Ref	۷
• اتصال به KMS	۷
• پردازش آسنکرون تراکنش‌ها	۷
• مقادیر عددی بزرگ	۸
۴- مدیریت کیف پول	۸
• ساخت کیف پول جدید	۸
• افزودن کیف پول با استفاده از کلید خصوصی	۹
• افزودن کلید براساس mnemonics	۱۰
۵- سرویس‌های بومی شبکه‌های اتریومی	۱۱
• دریافت موجودی	۱۱
• انتقال وجه (رمزارز مبنای شبکه)	۱۲
• نتیجه وضعیت تراکنش	۱۳
۶- سرویس‌های مبتنی بر توکن ERC۲۰	۱۴
• دریافت موجودی	۱۴
• دریافت مقدار تمام ذخیره یک توکن	۱۵
• انتقال یک توکن ERC۲۰	۱۶
• انتقال یک توکن ERC۲۰ با مجوز برداشت	۱۷
• مشاهده مجوز برداشت	۱۸
• اجازه برداشت به یک کیف پول	۱۹



- ۷- سرویس‌های مبتنی بر توکن ERC۲۱ ۲۰
- بارگذاری کالکشن ۲۰
 - مینت توکن ۲۱
 - سوزاندن توکن ۲۱
 - انتقال یک توکن ERC۲۱ ۲۲
- ۸- سرویس‌های بومی شبکه ترون ۲۳
- دریافت موجودی ۲۳
 - انتقال وجه (رمزارز مبنای شبکه) ۲۴
 - نتیجه وضعیت تراکنش ۲۵
- ۹- سرویس‌های مبتنی بر توکن TRC۲۰ ۲۶
- دریافت موجودی ۲۶
 - دریافت مقدار تمام ذخیره یک توکن ۲۷
 - انتقال یک توکن TRC20 ۲۸
 - نتیجه وضعیت تراکنش ۲۹
- ۱۰- سرویس‌های بومی شبکه سولانا ۳۰
- دریافت موجودی ۳۰
 - انتقال وجه (رمزارز مبنای شبکه) ۳۰
 - نتیجه وضعیت تراکنش ۳۱
- ۱۱- سرویس‌های مبتنی بر توکن‌های SPL ۳۲
- دریافت موجودی ۳۲
 - انتقال یک توکن SPL ۳۲
 - ساخت اکانت توکن SPL ۳۳
 - ساخت اکانت توکن SPL و انتقال آن ۳۴
- ۱۲- سرویس‌های شبکه تون ۳۵
- دریافت موجودی ۳۵
 - نتیجه وضعیت تراکنش توکن ۳۶
 - دریافت تراکنش‌های توکنی ۳۷



- دریافت بلاک آخر..... ۳۸
- ۱۳- سرویس‌های شبکه پولکادات..... ۳۹
- دریافت موجودی..... ۳۹
- نتیجه وضعیت تراکنش..... ۴۰
- انتقال وجه (رمزارز مبنای شبکه)..... ۴۱
- ۱۴- سرویس‌های شبکه کاردانو..... ۴۲
- دریافت موجودی..... ۴۲
- نتیجه وضعیت تراکنش..... ۴۳
- انتقال وجه (رمزارز مبنای شبکه)..... ۴۴
- ۱۵- سرویس‌های شبکه ریپل..... ۴۵
- دریافت موجودی..... ۴۵
- نتیجه وضعیت تراکنش..... ۴۶
- انتقال وجه (رمزارز مبنای شبکه)..... ۴۷
- فعال‌سازی حساب (رمزارز مبنای شبکه)..... ۴۸



۱- مقدمه

مجموعه سرویس‌های Blockchain as a Service پلاریس تلاش می‌کند تا زمان توسعه محصول را برای مشتریان خود در حوزه بلاکچین را کاهش داده و نیاز و وابستگی آن‌ها را به دانش فنی مرتبط با بلاکچین به حداقل میزان ممکن برساند. بدین ترتیب کسب‌وکارهای نوپا و نیز سازمان یافته می‌توانند در سریع‌ترین زمان ممکن با استفاده از سرویس‌های مبتنی بر وب پلاریس محصولات خود را توسعه و گسترش دهند.

۲- اهداف مستند

این مستند در راستای معرفی سرویس‌های مدیریت کیف پول کریپتو و نیز مدیریت تراکنش‌های کیف پول ارائه شده است. لازم به ذکر است که این سامانه نقش کلیدی و مرکزی در ارائه سرویس‌های پلاریس دارد و در واقع حلقه متصل کننده سرویس‌های مدیریت کلید (Crypto KMS) و نیز شبکه‌ها (Node as a Service) می‌باشد. سرویس WaaS یا Wallet as a Service یک سرویس کیف پول حضانتی می‌باشد که این امکان را به کاربران می‌دهد تراکنش‌های خود را در انواع شبکه‌های اتریومی و بیت کوین (UTXO-based) انجام دهد.

۳- موارد کلی

• X-API-Key

مقدار api-key یک رشته تولید است که جهت احراز هویت و تنظیم دسترسی به سرویس‌ها استفاده می‌شود.

• برچسب کلید

در بعضی از عملیات کاربر می‌تواند یک برچسب (Label) برای خود مشخص کند. در آینده این برچسب جهت تحلیل داده‌ها توسط کاربر در داشبورد می‌تواند مورد استفاده قرار گیرد.



• امکان استخراج کلید

در هنگام ثبت اطلاعات کلید و یا تولید کلید در سامانه، کاربر می‌تواند با مقدار دهی به فیلد `Exportable=true` مشخص کند که در آینده امکان استخراج اطلاعات کلید وجود خواهد داشت. این امکان به صورت مستقیم در سرویس‌های مدیریت کیف پول ارائه نمی‌گردد و در صورت نیاز کاربر می‌بایست به سرویس‌های KMS مراجعه نماید.

• دسترسی به اطلاعات با Ref

در بسیاری از موارد، سرویس‌های ارائه شده توسط WaaS این امکان را به کاربران می‌دهد که اطلاعات را با کلید یکتای `uuid` ارائه شده (مثلاً `Wallet-Id` یا `Token-Id` یا `Blockchain-Id`) استفاده کنند و یا اینکه اطلاعات را با نام یا آدرس شناخته شده آن مورد فراخوانی قرار دهند.

در نتیجه شما می‌توانید برای انتقال وجه به یک آدرس، آدرس مقصد را به صورت `public-address` و یا `wallet-id` ارسال کنید. همچنین می‌توانید برای فراخوانی سرویس‌های یک توکن `ERC20` آن را با `token-address` و یا `token-id` و یا `token-name` فراخوانی کنید و سرویس به صورت خودکار این موضوع را متوجه خواهد شد.

• اتصال به KMS

سامانه WaaS امکان اتصال به چندین KMS متفاوت را به صورت همزمان به کاربران خود می‌دهد. با این روش، کاربران، می‌توانند بسته به امنیت مورد نظر به سرویس مدیریت کلید `cloud` و یا اینکه سرویس مدیریت کلید خودشان متصل گردند. جهت ثبت لینک KMS کاربران می‌بایست از طریق داشبورد کاربران اقدام کنند. در صورت عدم امکان استفاده از داشبورد کاربران، می‌توانید از طریق پشتیبانی پلاریس جهت اتصال به سرویس خود استفاده نمایید. مقدار `KmsRef` در حالت عادی می‌بایست به مقدار `"cloud-kms"` مقدار دهی شود.

• پردازش آسنکرون تراکنش‌ها

جهت جلوگیری از ایجاد وقفه در پردازش تراکنش‌ها، تراکنش‌ها (مخصوصاً تراکنش‌های شبکه‌های اتریومی) به صورت آسنکرون انجام می‌شود و شما می‌توانید نتیجه تراکنش را با کمک سرویس `/api/evm/status` دریافت کنید.



• مقادیر عددی بزرگ

در بسیاری از سرویس‌ها، مقادیر عددی بزرگ به صورت uint۲۵۶ می‌باشد که در استاندارد مقدار عددی json number نمی‌گنجد و لذا به صورت رشته حرفی (string) دریافت و یا ارسال می‌شود.

۴- مدیریت کیف پول

• ساخت کیف پول جدید

این سرویس یک کیف پول جدید با کلید جدید تولید می‌کند.

Creates new wallet by random private key			
<server-url>/api/wallets/create			
Request	Method	POST	
	Header	Content-Type	application/json
		X-API-Key	Authorization Token
	Body	<pre>{ "blockchainRef": "<string>", "networkRef": "<string>", "label": "<string>", "kmsRef": "<string>", "keyExportable": "<boolean>" }</pre>	
Response	WalletId	کد یکتای کیف پول ایجاد شده	
	Example	<uuid>	



• افزودن کیف پول با استفاده از کلید خصوصی

این سرویس یک کیف پول با استفاده از کلید خصوصی کاربر تولید می کند. لازم به ذکر است که مقدار chaincodeHex یک مقدار اختیاری است و جهت اشتقاق کلید^۱ به روش HD-Wallet استفاده می گردد و در صورتی که نیازی به آن ندارید می توانید مقدار آن را null ارسال کنید. جهت دریافت اطلاعات بیشتر لطفاً به مستندات BEP-۰۰۳۲^۲ مراجعه بفرمایید.

Imports a wallet using private key			
<server-url>/api/wallets/import/privatekey			
Request	Method	POST	
	Header	Content-Type	application/json
		X-API-Key	Authorization Token
	Body	<pre>{ "blockchainRef": "<string>", "networkRef": "<string>", "label": "<string>", "kmsRef": "<string>", "privateKeyHex": "<string>", "chainCodeHex": "<string>", "keyExportable": "<boolean>" }</pre>	
Response	Key Id	کد یکتا کلید وارد شده	
	Example	<uuid>	

^۱ Key Derivation

^۲ <https://github.com/bitcoin/bips/blob/master/bip-0032.mediawiki>



• افزودن کلید براساس mnemonics

این سرویس یک کیف پول با کلید تولید شده بر اساس کلمات mnemonic تولید می‌کند.

Imports wallet using mnemonics			
<server-url>/api/wallets/import/mnemonic			
Request	Method	POST	
	Header	Content-Type	application/json
		X-API-Key	Authorization Token
	Body	<pre>{ "blockchainRef": "<string>", "networkRef": "<string>", "label": "<string>", "kmsRef": "<string>", "mnemonics": "<string>", "keyExportable": "<boolean>" }</pre>	
Response	Key Id	کد یکتا کلید وارد شده	
	Example	<uuid>	



۵- سرویس‌های بومی^۳ شبکه‌های اتریومی

در این بخش سرویس‌های بومی شبکه‌های اتریومی مانند موجودی و انتقال وجه ارائه می‌گردد.

• دریافت موجودی

Returns native currency token balance, equal to eth_balance			
<server-url>/api/evm/balance			
Request	Method	GET	
	Header	Content-Type	application/json
		X-API-Key	Authorization Token
	Query Params	BlockchainRef	blockchain name or id
		NetworkRef	blockchain network name or id
		WalletRef	public address or wallet id
Response	Example	<string>	

³ native



• انتقال وجه (رمزارز مبنای شبکه)

از طریق این سرویس کاربر می‌تواند رمزارز مبنای شبکه (مانند Ether برای شبکه Ethereum و یا Matic برای شبکه Polygon) را منتقل کند. لازم است که سرویس به صورت لحظه‌ای نتیجه تراکنش را مشخص نمی‌کند (جهت جلوگیری از ایجاد وقفه طولانی در پاسخ‌دهی سرویس) و شما می‌توانید نتیجه تراکنش را در ادامه از طریق سرویس `/api/evm/status` دریافت کنید.

Transfers native token from source to destination			
<server-url>/api/evm/transfer			
Request	Method	POST	
	Header	Content-Type	application/json
		X-API-Key	Authorization Token
	Body	<pre>{ "blockchainRef": "<string>", "networkRef": "<string>", "sourceWalletId": "<uuid>", "destinationRef": "<string>", "amountInWei": "<string>", "gasFeeInWei": "<string>" }</pre>	
Response	Example	<string>	هش تراکنش



• نتیجه وضعیت تراکنش

این سرویس نتیجه تراکنش را به کاربر اعلام می‌کند.

Returns transaction status for evm based networks			
<server-url>/api/evm/status			
Request	Method	GET	
	Header	Content-Type	application/json
		X-API-Key	Authorization Token
	Query Params	BlockchainRef	blockchain name or id
		NetworkRef	blockchain network name or id
		TransactionHash	public address or wallet id
Response	Example	<pre>{ "transactionHash": "string", "transactionIndex": "number", "blockHash": "string", "blockNumber": "number", "from": "string", "to": "string", "cumulativeGasUsed": "number", "gasUsed": "number", "effectiveGasPrice": "number", "status": "Success" }</pre>	



۶- سرویس‌های مبتنی بر توکن ERC۲۰

• دریافت موجودی

این سرویس مقدار موجودی توکن ERC۲۰ را برای یک آدرس باز می‌گرداند.

Returns erc۲۰ currency token balance			
<server-url>/api/erc۲۰/balance			
Request	Method	GET	
	Header	Content-Type	application/json
		X-API-Key	Authorization Token
	Query Params	BlockchainRef	blockchain name or id
		NetworkRef	blockchain network name or id
		WalletRef	public address or wallet id
		TokenRef	token id or token name or token address
Response	Example	<string>	مقدار موجودی



• دریافت مقدار تمام ذخیره یک توکن

این سرویس مقدار کل ذخیره یک توکن را باز می‌گرداند.

Returns total supply of the token			
<server-url>/api/erc20/total-supply			
Request	Method	GET	
	Header	Content-Type	application/json
		X-API-Key	Authorization Token
	Query Params	BlockchainRef	blockchain name or id
		NetworkRef	blockchain network name or id
		TokenRef	token id or token name or token address
Response	Example	<string>	



• انتقال یک توکن ERC۲۰

این سرویس مقدار مشخصی را از کیف پول مبدأ (sourceWalletId) به یک آدرس ارسال می‌کند. مقدار بازگشتی، transaction hash خواهد بود.

Transfers erc۲۰ token amount from source to destination			
<server-url>/api/erc۲۰/transfer			
Request	Method	POST	
	Header	Content-Type	application/json
		X-API-Key	Authorization Token
	Body	<pre>{ "blockchainRef": "<string>", "networkRef": "<string>", "tokenRef": "<string>", "sourceWalletId": "<uuid>", "destinationRef": "<string>", "amountInWei": "<string>", "gasFeeinWei": "<string>" }</pre>	
Response	Example	<string>	



• انتقال یک توکن ERC۲۰ با مجوز برداشت

این سرویس مقدار مشخصی را از یک آدرس مبدأ (sourcePublicRef) به آدرس مقصد (DestinationPublicRef) ارسال می‌کند.

لازم به ذکر است که عاملیت تراکنش با ownerWalletId می‌باشد و در واقع می‌بایست قبل از این متد، از طریق approve به کیف پول مذکور اجازه برداشت داده شود. مقدار بازگشتی، transaction hash خواهد بود.

transfers from source (other than owner wallet) to destination address			
<server-url>/api/erc۲۰/transfer-from			
Request	Method	POST	
	Header	Content-Type	application/json
		X-API-Key	Authorization Token
	Body	<pre>{ "blockchainRef": "<string>", "networkRef": "<string>", "tokenRef": "<string>", "ownerWalletId": "<uuid>", "sourcePublicRef": "<string>", "destinationPublicRef": "<string>", "amountInWei": "<string>", "gasFeeInWei": "<string>" }</pre>	
Response	Example	<string>	



• مشاهده مجوز برداشت

این سرویس مجوز برداشت از یک آدرس کیف پول به یک آدرس کیف پول یا قرارداد هوشمند را نمایش می‌دهد.

Get how much spender is allowed to spend owner's token			
<server-url>/api/erc20/allowance			
Request	Method	GET	
	Header	Content-Type	application/json
		X-API-Key	Authorization Token
	Query Params	BlockchainRef	blockchain name or id
		NetworkRef	blockchain network name or id
		WalletRef	public address or wallet id
		TokenRef	token id or token name or token address
		SourcePublicRef	Source address or wallet id
		SpenderPublicRef	Spender address or wallet id
Response	Example	<string>	مقدار مجوز برداشت



• اجازه برداشت به یک کیف پول

این سرویس به یک آدرس اجازه می‌دهد که به مقدار مشخصی از حساب کیف پول مبدأ برداشت نماید.

Approves erc۲۰ token amount from owner to spender			
<server-url>/api/erc۲۰/approve			
Request	Method	POST	
	Header	Content-Type	application/json
		X-API-Key	Authorization Token
	Body	<pre>{ "blockchainRef": "<string>", "networkRef": "<string>", "tokenRef": "<string>", "sourceWalletId": "<uuid>", "spenderRef": "<string>", "destinationPublicRef": "<string>", "amount": "<string>", "gasFeeinWei": "<string>" }</pre>	
Response	Example	<string>	



۷- سرویس‌های مبتنی بر توکن ERC۷۲۱

• بارگذاری^۴ کالکشن

این سرویس کالکشن یا قرارداد هوشمند را بارگذاری کرده و transaction hash را باز می‌گرداند. این کالکشن‌ها می‌توانند سه خصوصیت Auto-increment، Burnable و Enumerable را دارا باشند. توضیح این خصوصیات به این شرح می‌باشد:

- Auto-increment: Auto-increment اجازه می‌دهد تا زمانی که یک توکن جدید در کانترکت درج می‌شود، شناسه^۵ آن به صورت خودکار گذاشته شود (شناسه‌ها از ۰ شروع شده و یکی یکی بالا می‌روند).
- Burnable: Burnable توکن را از گردش قرارداد هوشمند خارج می‌کند، در نتیجه عرضه کل را کاهش می‌دهد و تقاضا را افزایش می‌دهد.
- Enumerable: Enumerable قابلیت شمارش همه شناسه‌های توکن به قرارداد هوشمند اضافه می‌کند.

Deploy an ERC-۷۲۱ contract			
<server-url>/api/erc۷۲۱/deploy			
Request	Method	POST	
	Header	Content-Type	application/json
		Authorization	JWT Authorization
	Body	<pre>{ "blockchainRef": "<string>", "networkRef": "<string>", "tokenRef": "<string>", "sourceWalletId": "<uuid>", "name": "<string>", "symbol": "<string>", "isAutoIncrement": "<bool>", "isBurnable": "<bool>", "isEnumerable": "<bool>", "gasFeeinWei": "<string>" }</pre>	
Response	Example	<string>	

deploy⁴

id⁵



• مینت توکن

این سرویس با گرفتن metadata برای nft همچنین آدرس کالکشن، یک توکن بر روی کالکشن مینت نموده و transaction hash را باز می‌گرداند.

Mint a token on ERC-۷۲۱ contract			
<server-url>/api/erc۷۲۱/mint			
Request	Method	POST	
	Header	Content-Type	application/json
		Authorization	JWT Authorization
	Body	<pre>{ "blockchainRef": "<string>", "networkRef": "<string>", "tokenRef": "<string>", "sourceWalletId": "<uuid>", "destinationAddress": "<string>", "collectionAddress": "<string>", "isAutoIncrement": "<bool>", "tokenId": "<BigInteger>", "metaDataUri": "<string>", "gasFeeinWei": "<string>" }</pre>	
Response	Example	<string>	

• سوزاندن توکن

این سرویس با گرفتن شناسه توکن، آن توکن را از قرارداد هوشمند حذف می‌کند و به اصطلاح می‌سوزاند. پس از آن transaction hash را باز می‌گرداند.

برای تست این سرویس در محیط local (اگر از قبل قرار داد هوشمندی ندارید). قبل از همه نیاز است یک کیف پول بسازیم. برای این کار باید سرویس kms بالا باشد و یک کیف پول بسازید (این بخش در "مدیریت کیف پول" توضیح داده شده است). سپس نیاز است که با سرویس "بارگذاری کالکشن" که در بالا توضیح آن داده شده است یک قرارداد هوشمند بسازید. سپس با transaction hash برگردانده شده در محیط blockchain explorer آدرس قرارداد هوشمند را بگیرید. در مرحله بعدی باید از سرویس "مینت توکن" توکنی را مینت کنید. می‌توانید بعد این توکن را در سایت opensea ببینید. پس از آن با سرویس سوزاندن توکن همین توکن را از بین ببرید. پس از آن، توکن نباید در opensea دیده شود.



Burn a token on ERC-۷۲۱ contract			
<server-url>/api/erc۷۲۱/mint			
Request	Method	POST	
	Header	Content-Type	application/json
		Authorization	JWT Authorization
	Body	<pre>{ "blockchainRef": "<string>", "networkRef": "<string>", "tokenRef": "<string>", "sourceWalletId": "<uuid>", "collectionAddress": "<string>", "gasFeeinWei": "<string>" }</pre>	
Response	Example		

• انتقال یک توکن ERC۷۲۱

این سرویس توکن مشخصی را از آدرس مبدأ (sourceAddress) به آدرس مقصد (destinationAddress) ارسال می کند. مقدار بازگشتی، transaction hash خواهد بود.

Transfer a token on ERC-۷۲۱ contract			
<server-url>/api/erc۷۲۱/transfer			
Request	Method	POST	
	Header	Content-Type	application/json
		Authorization	JWT Authorization
	Body	<pre>{ "blockchainRef": "<string>", "networkRef": "<string>", "tokenRef": "<string>", "sourceWalletId": "<uuid>", "sourceAddress": "<string>", "destinationAddress": "<string>", "collectionAddress": "<string>", "tokenId": "<BigInteger>", "gasFeeinWei": "<string>" }</pre>	
Response	Example		



۸- سرویس های بومی شبکه ترون

در این بخش سرویس های بومی شبکه ترون مانند موجودی و انتقال وجه ارائه می گردد.

• دریافت موجودی

Returns native currency token(trx) balance			
<server-url>/api/tron/balance			
Request	Method	GET	
	Header	Content-Type	application/json
		X-API-Key	Authorization Token
	Query Params	BlockchainRef	blockchain name or id
		NetworkRef	blockchain network name or id
		WalletRef	public address or wallet id
Response	Example	<decimal>	



• انتقال وجه (رمزارز مبنای شبکه)

از طریق این سرویس کاربر می‌تواند رمزارز توکن بومی شبکه را منتقل کند. لازم است که سرویس به صورت لحظه‌ای نتیجه تراکنش را مشخص نمی‌کند (جهت جلوگیری از ایجاد وقفه طولانی در پاسخ‌دهی سرویس) و شما می‌توانید نتیجه تراکنش را در ادامه از طریق سرویس `/api/tron/status` دریافت کنید.

Transfers native token from source to destination			
<server-url>/api/tron/transfer			
Request	Method	POST	
	Header	Content-Type	application/json
		X-API-Key	Authorization Token
	Body	<pre>{ "blockchainRef": "<string>", "networkRef": "<string>", "sourceWalletId": "<uuid>", "destinationRef": "<string>", "amount": "<long>" }</pre>	
Response	Example	<string>	هش تراکنش



• نتیجه وضعیت تراکنش

این سرویس نتیجه تراکنش را به کاربر اعلام می‌کند.

Returns transaction status for tron networks			
<server-url>/api/tron/status			
Request	Method	GET	
	Header	Content-Type	application/json
		X-API-Key	Authorization Token
	Query Params	BlockchainRef	blockchain name or id
		NetworkRef	blockchain network name or id
		TransactionHash	Transaction hash
Response	Example	<json>	



۹- سرویس‌های مبتنی بر توکن TRC20

• دریافت موجودی

این سرویس مقدار موجودی توکن TRC۲۰ را برای یک آدرس باز می‌گرداند.

Returns trc۲۰ currency token balance			
<server-url>/api/trc۲۰/balance			
Request	Method	GET	
	Header	Content-Type	application/json
		X-API-Key	Authorization Token
	Query Params	BlockchainRef	blockchain name or id
		NetworkRef	blockchain network name or id
		WalletRef	public address or wallet id
		TokenRef	token id or token name or token address
Response	Example	<decimal>	مقدار موجودی



• دریافت مقدار تمام ذخیره یک توکن

این سرویس مقدار کل ذخیره یک توکن را باز می‌گرداند.

Returns total supply of the token			
<server-url>/api/trc۲۰/total-supply			
Request	Method	GET	
	Header	Content-Type	application/json
		X-API-Key	Authorization Token
	Query Params	BlockchainRef	blockchain name or id
		NetworkRef	blockchain network name or id
		TokenRef	token id or token name or token address
Response	Example	<decimal>	



• انتقال یک توکن TRC20

این سرویس مقدار مشخصی را از کیف پول مبدأ (sourceWalletId) به یک آدرس ارسال می‌کند. مقدار بازگشتی، transaction hash خواهد بود.

Transfers trc۲۰ token amount from source to destination			
<server-url>/api/trc۲۰/transfer			
Request	Method	POST	
	Header	Content-Type	application/json
		X-API-Key	Authorization Token
	Body	<pre>{ "blockchainRef": "<string>", "networkRef": "<string>", "tokenRef": "<string>", "sourceWalletId": "<uuid>", "destinationRef": "<string>", "amount": "<decimal>", "Memo": "<string?>", "feeAmount": "<long?>" }</pre>	
Response	Example	<string>	



• نتیجه وضعیت تراکنش

این سرویس نتیجه تراکنش را به کاربر اعلام می‌کند.

Returns transaction status for tron networks			
<server-url>/api/tc۲۰/status			
Request	Method	GET	
	Header	Content-Type	application/json
		X-API-Key	Authorization Token
	Query Params	BlockchainRef	blockchain name or id
		NetworkRef	blockchain network name or id
		TransactionHash	Transaction hash
Response	Example	<json>	



۱۰- سرویس های بومی شبکه سولانا

در این بخش سرویس های بومی شبکه سولانا مانند موجودی و انتقال وجه ارائه می گردد.

• دریافت موجودی

Returns native currency token(SOL) balance			
<server-url>/api/solana/balance			
Request	Method	GET	
	Header	Content-Type	application/json
		X-API-Key	Authorization Token
	Query Params	BlockchainRef	blockchain name or id
		NetworkRef	blockchain network name or id
		WalletRef	public address or wallet id
Response	Example	<string>	

• انتقال وجه (رمزارز مبنای شبکه)

از طریق این سرویس کاربر می تواند رمزارز توکن بومی شبکه را منتقل کند. لازم به ذکر است که سرویس به صورت لحظه ای نتیجه تراکنش را مشخص نمی کند (جهت جلوگیری از ایجاد وقفه طولانی در پاسخ دهی سرویس) و شما می توانید نتیجه تراکنش را در ادامه از طریق سرویس `/api/solana/status` دریافت کنید.



Transfers native token from source to destination			
<server-url>/api/solana/transfer			
Request	Method	POST	
	Header	Content-Type	application/json
		X-API-Key	Authorization Token
	Body	<pre>{ "blockchainRef": "<string>", "networkRef": "<string>", "sourceWalletId": "<uuid>", "destinationRef": "<string>", "amountInSOL": "<decimal>" }</pre>	
Response	Example	<string>	هش تراکنش

• نتیجه وضعیت تراکنش

این سرویس نتیجه تراکنش را به کاربر اعلام می‌کند.

Returns transaction status for solana networks			
<server-url>/api/solana/status			
Request	Method	GET	
	Header	Content-Type	application/json
		X-API-Key	Authorization Token
	Query Params	BlockchainRef	blockchain name or id
		NetworkRef	blockchain network name or id
		TransactionHash	Transaction hash
Response	Example	Json	



۱۱- سرویس‌های مبتنی بر توکن های SPL

• دریافت موجودی

این سرویس مقدار موجودی توکن SPL را برای یک آدرس باز می‌گرداند.

Returns SPL currency token balance			
<server-url>/api/spl/balance			
Request	Method	GET	
	Header	Content-Type	application/json
		X-API-Key	Authorization Token
	Query Params	BlockchainRef	blockchain name or id
		NetworkRef	blockchain network name or id
		WalletRef	public address or wallet id
		TokenRef	token id or token name or token address
Response	Example	<ulong>	مقدار موجودی

• انتقال یک توکن SPL

این سرویس مقدار مشخصی را از کیف پول مبدأ (sourceWalletId) به یک آدرس ارسال می‌کند. مقدار بازگشتی، transaction hash خواهد بود.



Transfers SPL token amount from source to destination			
<server-url>/api/spl/transfer			
Request	Method	POST	
	Header	Content-Type	application/json
		X-API-Key	Authorization Token
	Body	<pre>{ "blockchainRef": "<string>", "networkRef": "<string>", "tokenRef": "<string>", "sourceWalletId": "<uuid>", "destinationRef": "<string>", "amount": "<ulong>" }</pre>	
Response	Example	<string>	

• ساخت اکانت توکن SPL

این سرویس یک توکن اکانت برای کیف پول مورد نظر (destinationRef) و توکن داده شده (tokenRef) ایجاد میکند. مقدار بازگشتی، transaction hash خواهد بود.

Creates SPL token account for destination			
<server-url>/api/spl/create-token-account			
Request	Method	POST	
	Header	Content-Type	application/json
		X-API-Key	Authorization Token
	Body	<pre>{ "blockchainRef": "<string>", "networkRef": "<string>", "tokenRef": "<string>", "sourceWalletId": "<uuid>", "destinationRef": "<string>" }</pre>	
Response	Example	<string>	



• ساخت اکانت توکن SPL و انتقال آن

این سرویس یک توکن اکانت برای کیف پول مورد نظر (destinationRef) و توکن داده شده (tokenRef) ایجاد میکند و به مقدار (amount) از این توکن از مبدا به مقصد انتقال می‌دهد. مقدار بازگشتی، transaction hash خواهد بود.

Creates SPL token account for destination and transfers amount from source to destination			
<server-url>/api/spl/create-token-account-and-transfer			
Request	Method	POST	
	Header	Content-Type	application/json
		X-API-Key	Authorization Token
	Body	<pre>{ "blockchainRef": "<string>", "networkRef": "<string>", "tokenRef": "<string>", "sourceWalletId": "<uuid>", "destinationRef": "<string>", "amount": "<ulong>" }</pre>	
Response	Example	<string>	



۱۲- سرویس‌های شبکه تون

• دریافت موجودی

این سرویس مقدار موجودی کوین بومی تون را برای یک آدرس باز می‌گرداند.

Returns native currency token(TON) balance			
<server-url>/api/ton/balance			
Request	Method	GET	
	Header	Content-Type	application/json
		X-API-Key	Authorization Token
	Query Params	BlockchainRef	Ton
		NetworkRef	Mainnet
		WalletRef	public address or wallet id
Response	Example	<decimal>	مقدار موجودی



• نتیجه وضعیت تراکنش توکن

این سرویس نتیجه و اطلاعات تراکنش توکنی را به کاربر اعلام می‌کند.

Returns token transaction status for Ton network			
<server-url>/api/ton/status			
Request	Method	GET	
	Header	Content-Type	application/json
		X-API-Key	Authorization Token
	Query Params	BlockchainRef	Ton
		NetworkRef	Mainnet
		TransactionHash	Transaction hash
		TokenMaster	Token smart contract address
Response	Example	<pre>{ "aborted": true, "destroyed": true, "transactionHash": "string", "transactionType": "string", "from": "string", "to": ["string"], "tokenAmount": 0, "amount": 0, "contractAddress": "string", "fee": "string", "comments": ["string"], "timestamp": "2024-08-13T08:26:42.304Z", "blockWorkchain": 0, "blockShard": "string", "blockSeqno": 0 }</pre>	



• دریافت تراکنش های توکنی

این سرویس تراکنش های توکنی یک آدرس را برای توکن مورد نظر باز می گرداند.

Returns token transactions for an address			
<server-url>/api/ton/token-transactions			
Request	Method	GET	
	Header	Content-Type	application/json
		X-API-Key	Authorization Token
	Query Params	BlockchainRef	Ton
		NetworkRef	Mainnet
		TokenMaster	Token smart contract address
		Address	Public address
		Pagination	Transactions pagination (50 transactions per page & sort = desc)
Response	Example	<pre>[{ "aborted": true, "destroyed": true, "transactionHash": "string", "transactionType": "string", "from": "string", "to": ["string"], "tokenAmount": 0, "amount": 0, "contractAddress": "string", "fee": "string", "comments": ["string"], "timestamp": "2024-08-13T08:48:49.331Z", "blockWorkchain": 0, "blockShard": "string", "blockSeqno": 0 }]</pre>	



• دریافت بلاک آخر

این سرویس متادیتا و تراکنش های بلاک آخر را برای شبکه تون مورد نظر باز می گرداند.

Returns last block metadata and transactions for given network			
<server-url>/api/ton/last-block			
Request	Method	GET	
	Header	Content-Type	application/json
		X-API-Key	Authorization Token
	Query Params	BlockchainRef	blockchain name or id
		NetworkRef	blockchain network name or id
		BlockWorkchain	Block workchain
		BlockShard	Block shard
Response	Example	<pre>{ "metaData": { "workchain": 0, "shard": "string", "seqno": 0, "transactionCount": 0, "timestamp": "2024-08-13T08:58:25.179Z", "rootHash": "string", "fileHash": "string" }, "transactions": [{ "aborted": true, "destroyed": true, "transactionHash": "string", "transactionType": "string", "from": "string", "to": ["string"], "tokenAmount": 0, "amount": 0, "contractAddress": "string", "fee": "string", "comments": ["string"], "timestamp": "2024-08-13T08:58:25.179Z", "blockWorkchain": 0, "blockShard": "string", "blockSeqno": 0 }] }</pre>	



۱۳- سرویس‌های شبکه پولکادات

• دریافت موجودی

این سرویس مقدار موجودی کوین بومی پولکادات (DOT) را برای یک آدرس باز می‌گرداند.

Returns native currency(DOT) balance			
<server-url>/api/polkadot/balance			
Request	Method	GET	
	Header	Content-Type	application/json
		X-API-Key	Authorization Token
	Query Params	BlockchainRef	Polkadot
		NetworkRef	Mainnet
		WalletRef	public address or wallet id
Response	Example	<string>	مقدار موجودی



• نتیجه وضعیت تراکنش

این سرویس نتیجه و اطلاعات تراکنش را به کاربر اعلام می‌کند.

Returns token transaction status for Polkadot network			
<server-url>/api/polkadot/status			
Request	Method	GET	
	Header	Content-Type	application/json
		X-API-Key	Authorization Token
	Query Params	BlockchainRef	Polkadot
		NetworkRef	Mainnet
		ExtrinsicHash	Extrinsic hash
		BlockHash	Block hash
Response	Example	<string>	نتیجه تراکنش



• انتقال وجه (رمزارز مبنای شبکه)

از طریق این سرویس کاربر می‌تواند رمزارز توکن بومی شبکه را منتقل کند. لازم به ذکر است که سرویس به صورت لحظه‌ای نتیجه تراکنش را مشخص نمی‌کند (جهت جلوگیری از ایجاد وقفه طولانی در پاسخ‌دهی سرویس) و شما می‌توانید نتیجه تراکنش را در ادامه از طریق سرویس `/api/polkadot/status` دریافت کنید.

Transfers native token from source to destination			
<server-url>/api/polkadot/transfer			
Request	Method	POST	
	Header	Content-Type	application/json
		X-API-Key	Authorization Token
	Body	<pre>{ "blockchainRef": "<string>", "networkRef": "<string>", "sourceWalletId": "<uuid>", "destinationRef": "<string>", "amount": "<decimal>" }</pre>	
Response	Example	<string>	هش تراکنش



۱۴- سرویس‌های شبکه کاردانو

• دریافت موجودی

این سرویس مقدار موجودی کوین بومی کاردانو (ADA) را برای یک آدرس باز می‌گرداند.

Returns native currency(ADA) balance			
<server-url>/api/cardano/balance			
Request	Method	GET	
	Header	Content-Type	application/json
		X-API-Key	Authorization Token
	Query Params	BlockchainRef	Cardano
		NetworkRef	Mainnet
		WalletRef	public address or wallet id
Response	Example	<string>	مقدار موجودی



• نتیجه وضعیت تراکنش

این سرویس نتیجه و اطلاعات تراکنش را به کاربر اعلام می‌کند.

Returns token transaction status for Cardano network			
<server-url>/api/cardano/status			
Request	Method	GET	
	Header	Content-Type	application/json
		X-API-Key	Authorization Token
	Query Params	BlockchainRef	Cardano
		NetworkRef	Mainnet
		TransactionHash	Transaction hash
Response	Example	<string>	نتیجه تراکنش



• انتقال وجه (رمزارز مبنای شبکه)

از طریق این سرویس کاربر می‌تواند رمزارز توکن بومی شبکه را منتقل کند. لازم به ذکر است که سرویس به صورت لحظه‌ای نتیجه تراکنش را مشخص نمی‌کند (جهت جلوگیری از ایجاد وقفه طولانی در پاسخ‌دهی سرویس) و شما می‌توانید نتیجه تراکنش را در ادامه از طریق سرویس `/api/cardano/status` دریافت کنید.

Transfers native token from source to destination			
<server-url>/api/cardano/transfer			
Request	Method	POST	
	Header	Content-Type	application/json
		X-API-Key	Authorization Token
	Body	<pre>{ "blockchainRef": "<string>", "networkRef": "<string>", "sourceWalletId": "<uuid>", "destinationRef": "<string>", "amount": "<decimal>" }</pre>	
Response	Example	<string>	هش تراکنش



۱۵- سرویس‌های شبکه ریپل

• دریافت موجودی

این سرویس مقدار موجودی کوین بومی ریپل (XRP) را برای یک آدرس باز می‌گرداند.

Returns native currency(XRP) balance			
<server-url>/api/ripple/balance			
Request	Method	GET	
	Header	Content-Type	application/json
		X-API-Key	Authorization Token
	Query Params	BlockchainRef	Ripple
		NetworkRef	Mainnet
		WalletRef	public address or wallet id
Response	Example	<string>	مقدار موجودی



• نتیجه وضعیت تراکنش

این سرویس نتیجه و اطلاعات تراکنش را به کاربر اعلام می‌کند.

Returns token transaction status for Ripple network			
<server-url>/api/ripple/status			
Request	Method	GET	
	Header	Content-Type	application/json
		X-API-Key	Authorization Token
	Query Params	BlockchainRef	Ripple
		NetworkRef	Mainnet
		TransactionHash	Transaction hash
Response	Example	<string>	نتیجه تراکنش



• انتقال وجه (رمزارز مبنای شبکه)

از طریق این سرویس کاربر می‌تواند رمزارز توکن بومی شبکه را منتقل کند. لازم به ذکر است که سرویس به صورت لحظه‌ای نتیجه تراکنش را مشخص نمی‌کند (جهت جلوگیری از ایجاد وقفه طولانی در پاسخ‌دهی سرویس) و شما می‌توانید نتیجه تراکنش را در ادامه از طریق سرویس `/api/ripple/status` دریافت کنید.

Transfers native token from source to destination			
<server-url>/api/ripple/transfer			
Request	Method	POST	
	Header	Content-Type	application/json
		X-API-Key	Authorization Token
	Body	<pre>{ "blockchainRef": "<string>", "networkRef": "<string>", "sourceWalletId": "<uuid>", "destinationRef": "<string>", "amount": "<decimal>" }</pre>	
Response	Example	<string>	هش تراکنش



• فعال سازی حساب (رمزارز مبنای شبکه)

از طریق این سرویس کاربر می تواند اکانت خود را فعال سازی کند. در شبکه Ripple، اصطلاح activate account (فعال سازی حساب) به فرآیند فعال کردن یک حساب کاربری جدید در ریپل اشاره دارد. زمانی که یک حساب ریپل (Ripple account) ایجاد می شود، تا زمانی که حداقل ۱۰ XRP به آن واریز نشود، حساب فعال نمی شود. این حداقل مقدار ۱۰ XRP به عنوان reserve (ذخیره) در نظر گرفته می شود. در این فرآیند نیاز است مقدار توکن در بخش amount باید از ۱۰ XRP بیشتر باشد. لازم به ذکر است که سرویس به صورت لحظه ای نتیجه تراکنش را مشخص نمی کند (جهت جلوگیری از ایجاد وقفه طولانی در پاسخ دهی سرویس) و شما می توانید نتیجه تراکنش را در ادامه از طریق سرویس `/api/ripple/status` دریافت کنید.

Activate an account in Ripple network			
<server-url>/api/ripple/activate-account			
Request	Method	POST	
	Header	Content-Type	application/json
		X-API-Key	Authorization Token
	Body	<pre>{ "blockchainRef": "<string>", "networkRef": "<string>", "sponsorWalletRef": "<uuid>", "destinationRef": "<string>", "amount": "<decimal>" }</pre>	
Response	Example	<string>	هش تراکنش