



پُلاریس

سامانه مدیریت کلید

Crypto KMS

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

فهرست مطالب

۱ - مقدمه	۱
۲ - اهداف مستند	۱
۳ - موارد کلی	۱
X-API-KEY	۱
برچسب کلید	۱
امکان استخراج کلید	۲
۴ - سرویس‌های مدیریت کلید	۲
ساخت کلید جدید	۲
افزودن کلید براساس کلید خصوصی	۳
افزودن کلید براساس MNEMONICS	۳
حذف کلید	۴
اشتقاق کلید	۵
استخراج کلید	۵
۵ - سرویس‌های شبکه‌های مبتنی بر اتریوم	۶
امضا توسط کلید	۶
دریافت اطلاعات عمومی کلید	۷
۶ - سرویس‌های شبکه‌های مبتنی بر UTXO	۸
امضا توسط کلید	۸
دریافت اطلاعات عمومی کلید	۹



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 کلید مبتنی بر بلاکچین (Crypto Key Management Service) می‌باشد. بر اساس طراحی سامانه‌های پلاریس، سرویس‌های BaaS پلاریس می‌توانند کلیدهای خود را در بیش از یک محل ذخیره کنند.

در این راستا، سرویس‌های KMS به صورت مستقل از Wallet ارائه شده است و قابلیت این را دارد که به صورت اختصاصی برای مشتریان نصب شده و یا اینکه به صورت مشترک و از سرویس مدیریت کلید cloud پلاریس استفاده نماید.

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.

۴ - سرویس‌های مدیریت کلید

• ساخت کلید جدید

این سرویس یک کلید کاملاً جدید تولید می‌کند و شناسه یکتای آن را باز می‌گرداند.

Generate Key			
<server-url>/api/keys/generate			
Request	Method	POST	
	Header	Content-Type	application/json
		X-API-Key	Authorization Token
	Body	<pre>{ "label": "<string>", "exportable": "<boolean>" }</pre>	
Response	keyId	شناسه یکتای کلید	
	Example	<pre>{ "keyId": "<uuid>" }</pre>	



• افزودن کلید براساس کلید خصوصی

این سرویس یک کلید را بر اساس کلید خصوصی در سیستم ثبت می‌کند. لازم به ذکر است که مقدار chaincodeHex یک مقدار اختیاری است و جهت اشتقاق کلید^۱ به روش HD-Wallet استفاده می‌گردد و در صورتی که نیازی به آن ندارید می‌توانید مقدار آن را null ارسال کنید. جهت دریافت اطلاعات بیشتر لطفاً به مستندات BEP-۰۰۳۲^۲ مراجعه بفرمایید.

Import-Privatekey			
<server-url>/api/keys/import-privatekey			
Request	Method	POST	
	Header	Content-Type	application/json
		X-API-Key	Authorization Token
	Body	<pre>{ "privateKeyHex": "<string>", "chainCodeHex": "<string>", "label": "<string>", "exportable": "<boolean>" }</pre>	
Response	keyId	کلید یکتا	
	Example	<pre>{ "keyId": "<uuid>" }</pre>	

• افزودن کلید براساس Mnemonics

این سرویس یک کلید خصوصی بر اساس mnemonic ارسال شده در سامانه ثبت می‌کند.

^۱ Key Derivation

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



Import-Privatekey			
<server-url>/api/keys/import-mnemonics			
Request	Method	POST	
	Header	Content-Type	application/json
		X-API-Key	Authorization Token
	Body	<pre>{ "mnemonics": "<string>", "label": "<string>", "exportable": "<boolean>" }</pre>	
Response	keyId	کلید یکتا	
	Example	<pre>{ "keyId": "<uuid>" }</pre>	

• حذف کلید

با این سرویس کاربر می‌تواند با ارائه شناسه کلید (تولید شده در سرویس‌های فوق) اقدام به حذف کلید خود نماید. در صورت حذف کلیدی که از آن کلید دیگری مشتق شده باشد، مشکلی به وجود نخواهد آمد و در واقع تنها کاربر است که ارتباط بین کلیدها را می‌داند.

Delete Key			
<server-url>/api/keys/delete			
Request	Method	POST	
	Header	Content-Type	application/json
		X-API-Key	Authorization Token
	Body	<pre>{ "keyId": "<string>", }</pre>	
Response	keyId	کلید یکتا	
	Example	f1a2772-e366-45c2-a06d-ce758a92fdb5702	



• اشتقاق کلید

این سرویس امکان اشتقاق کلید (Key Derivation) بر اساس استاندارد BEP-۰۰۳۲ را به کاربر می‌دهد. لازم به ذکر است که کلید جدید تولید شده نیز امکان اشتقاق را خواهد داشت.

Derive Key			
<server-url>/api/keys/derive			
Request	Method	POST	
	Header	Content-Type	application/json
		X-API-Key	Authorization Token
	Body	<pre>{ "parentId": "<uuid>", "label": "<string>", "exportable": "<boolean>" }</pre>	
Response	keyId	کلید یکتا	
	Example	<pre>{ "keyId": "<uuid>" }</pre>	

• استخراج کلید

با کمک این سرویس می‌توانید مقدار کلید خصوصی، chain code و سایر اطلاعات کلید را دریافت نمایید. در صورتی که در زمان تولید کلید، آن را exportable معرفی نکرده باشید، امکان export آن وجود نخواهد داشت.



Export Key			
<server-url>/api/keys/export-key			
Request	Method	GET	
	Header	Content-Type	application/json
		X-API-Key	Authorization Token
	Query Params	"keyId": "<uuid>"	
Response	privatekeyHex	کلید خصوصی در مبنای شانزده	
	chainCodeHex	کد مخصوص به صورت hex	
	timestamp	زمان دریافت کلید	
	label	برچسب کلید	
	Example	<pre>{ "privatekeyHex": "<string>", "timestamp": "<dateTime>", "chainCodeHex": "<string>", "label": "<string>" }</pre>	

۵ - سرویس‌های شبکه‌های مبتنی بر اتریوم

• امضا توسط کلید

این سرویس hash یک تراکنش را بر اساس کلید مشخص شده امضا می‌کند.



Signs Transaction Hash			
<server-url>/api/evm/sign/hash			
Request	Method	POST	
	Header	Content-Type	application/json
		X-API-Key	Authorization Token
	Body	<pre>{ "keyId": "<uuid>", "hashHex": "<string>", "chainId": "<integer>" }</pre>	
Response	keyId	کلید یکتا	
	hashHex	مقدار هاش به صورت hex	
	chainId	کد یکتای شبکه	
	Example	<pre>{ "r_Hex": "<string>", "s_Hex": "<string>", "v_Hex": "<string>", "signature": "<string>" }</pre>	

• دریافت اطلاعات عمومی کلید

این سرویس اطلاعات کلید و آدرس تولید شده بر اساس استانداردهای شبکه‌های مبتنی بر اتریوم را مشخص می‌کند.



Returns public information

<server-url>/api/evm/get-public-info

Request	Method	GET	
	Header	Content-Type	application/json
		X-API-Key	Authorization Token
Response	keyId	کد یکتای کلید	
	publicKeyHex	کلید عمومی	
	publicAddress	آدرس اتریمی	
	Example	<pre>{ "keyId": "<uuid>", "publicKeyHex": "<string>", "publicAddress": "<string>" }</pre>	

۶ - سرویس‌های شبکه‌های مبتنی بر UTXO

در این بخش سرویس‌های شبکه‌های مبتنی بر ساختار UTXO^۳ مانند بیت کوین توضیح داده خواهند شد.

• امضا توسط کلید

این سرویس یک تراکنش را رمز می‌کند. لازم به ذکر است که مقدار hexadecimal transactionDataHex در واقع داده json تراکنش است که به صورت کد شده است. این سرویس تلاش می‌کند که ورودی‌های تراکنش را توسط کلیدهای ارائه شده امضا نماید. بدیهی است که در صورت عدم ارائه تمامی کلیدها، تراکنش تولید شده در خروجی، به طور کامل رمز نشده باشد.

این موضوع به کاربر امکان ارائه سرویس رمزنگاری یک تراکنش توسط چندین نفر را می‌دهد به این صورت که هر یک از کاربران اطلاعات تراکنش را به صورت جداگانه رمز می‌کند و نتیجه را در اختیار نفر بعد قرار می‌دهد.

^۳ Unspent Transaction Output



Signs a UTXO (like bitcoin) transaction

<server-url> /api/utxo/sign-tx

Request	Method	POST	
	Header	Content-Type	application/json
		X-API-Key	Authorization Token
	Body	<pre>{ "keyIds": ["<uuid>", "<uuid>"], "transactionDataHex": "<string>", "network": "MainNet" }</pre>	
Response	Hash string	داده تراکنش UTXO که به صورت hex کد شده است.	
	Example	<string>	

• دریافت اطلاعات عمومی کلید

این سرویس اطلاعات کلید و آدرس تولید شده بر اساس استانداردهای شبکه‌های مبتنی بر UTXO مانند بیتکوین را مشخص می‌کند. با توجه به اینکه آدرس تولید شده ممکن است که بر اساس نوع شبکه (mainnet و یا testnet و ...) متفاوت باشد، کاربر می‌بایست نوع شبکه خود را مشخص نماید.



Returns public information (public key and address based on utxo networks)

<server-url> /api/utxo/get-public-info

Request	Method	GET	
	Header	Content-Type	application/json
		X-API-Key	Authorization Token
	Query	KeyId	کد یکتای کلید
		ChainType	نوع شبکه
Response	keyId	کد یکتای کلید	
	publicKeyHex	کلید عمومی	
	publicAddress	آدرس در شبکه	
	Example	<pre>{ "keyId": "<uuid>", "publicKeyHex": "<string>", "publicAddress": "<string>" }</pre>	