



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

## فهرست مطالب

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

۱۷	دریافت موجودی.....
۱۸	دریافت مقدار تمام ذخیره یک توکن.....
۱۹	انتقال یک توکن ERC۲۰.....
۲۰	انتقال یک توکن ERC۲۰ با مجوز برداشت.....
۲۱	مشاهده مجوز برداشت.....
۲۲	اجازه برداشت به یک کیف پول.....



## 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 متصل‌کننده 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 `ERC۲۰` 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.

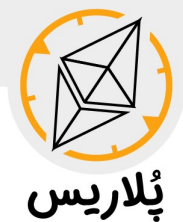
## ۴- مدیریت شبکه‌های بلاکچین

سامانه WaaS این امکان را به کاربر خود می‌دهد که به انواع شبکه‌های بلاکچین متصل گردد و در این رابطه جهت اطلاع کاربران لیست شبکه‌های مورد پشتیبانی به شرح زیر ارائه می‌گردد.

#### • ارائه لیست شبکه‌های بلاکچین

این سرویس لیست انواع شبکه‌های مورد پشتیبانی (مانند اتریوم، پلیگان، بیت کوین و ...) را ارائه می‌کند.

Supported blockchains			
<server-url>/api/blockchains			
Request	Method	GET	
	Header	Content-Type	application/json
		X-API-Key	Authorization Token
Response	Example	<pre>[   {     "id": "&lt;uuid&gt;",     "name": "&lt;string&gt;",     "type": "Evm"   },   {     "id": "&lt;uuid&gt;",     "name": "&lt;string&gt;",     "type": "Utxo"   } ]</pre>	

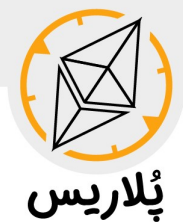


### • لیست شبکه‌های یک بلاکچین

هر بلاکچین ممکن است که شبکه‌های متفاوتی را جهت تست پشتیبانی کند. مثلاً شبکه اتریوم ممکن است که شبکه Mainnet برای انجام تراکنش‌های اصلی و kavan جهت انجام تراکنش‌های تست را پشتیبانی کند.

Supported blockchains			
<server-url>/api/blockchains/:blockchainRef/networks			
Request	Method	GET	
	Header	Content-Type	application/json
		X-API-Key	Authorization Token
	Path Variable	blockchainRef	blockchain name or id
Response	Example	<pre>[   {     "id": "&lt;uuid&gt;",     "name": "&lt;string&gt;",     "blockchain": "&lt;string&gt;",     "blockchainType": "Evm",     "bitcoinChainType":       "MainNet"   },   {     "id": "&lt;uuid&gt;",     "name": "&lt;string&gt;",     "blockchain": "&lt;string&gt;",     "blockchainType": "Evm",     "bitcoinChainType":       "MainNet"   } ]</pre>	





## • دریافت اطلاعات یک شبکه مشخص

با این سرویس کاربر می‌تواند اطلاعات یک شبکه مشخص را دریافت کند.

Supported blockchains			
<server-url>/api/blockchains/:blockchainRef/:networkRef			
Request	Method	GET	
	Header	Content-Type	application/json
		X-API-Key	Authorization Token
	Path Variable	blockchainRef	blockchain name or id
networkRef		blockchain network name or id	
Response	Example	<pre>[   {     "id": "&lt;uuid&gt;",     "name": "&lt;string&gt;",     "blockchain": "&lt;string&gt;",     "blockchainType": "Evm",     "bitcoinChainType": "MainNet"   },   {     "id": "&lt;uuid&gt;",     "name": "&lt;string&gt;",     "blockchain": "&lt;string&gt;",     "blockchainType": "Evm",     "bitcoinChainType": "MainNet"   } ]</pre>	



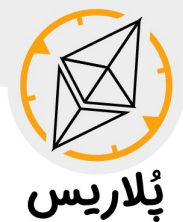
### • لیست تمام شبکه‌های پشتیبانی شده

این سرویس لیست تمامی شبکه‌های تمامی بلاکچین‌های مورد پشتیبانی را ارائه می‌کند.

List all supported networks			
<server-url>/api/networks			
Request	Method	GET	
	Header	Content-Type	application/json
		X-API-Key	Authorization Token
Response	Example	<pre>[   {     "id": "&lt;uuid&gt;",     "name": "&lt;string&gt;",     "blockchain": "&lt;string&gt;",     "blockchainType": "Evm",     "bitcoinChainType": "MainNet"   },   {     "id": "&lt;uuid&gt;",     "name": "&lt;string&gt;",     "blockchain": "&lt;string&gt;",     "blockchainType": "Evm",     "bitcoinChainType": "MainNet"   } ]</pre>	

### • لیست مشخصات توکن‌های پشتیبانی شده

این سرویس لیست توکن‌های پشتیبانی شده در سیستم را ارائه می‌کند. در حال حاضر لیست مذکور تنها توکن‌های ERC۲۰ را ارائه می‌کند.



List all supported Tokens			
<server-uri>/api/tokens			
Request	Method	GET	
	Header	Content-Type	application/json
		X-API-Key	Authorization Token
	Example	<pre>[   {     "id": "&lt;uuid&gt;",     "name": "&lt;string&gt;",     "address": "&lt;string&gt;",     "networkName": "&lt;string&gt;",     "blockchainName": "&lt;string&gt;"   },   {     "id": "&lt;uuid&gt;",     "name": "&lt;string&gt;",     "address": "&lt;string&gt;",     "networkName": "&lt;string&gt;",     "blockchainName": "&lt;string&gt;"   } ]</pre>	

## ۵- مدیریت کیف پول

### • ساخت کیف پول جدید

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



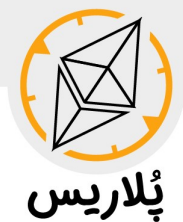
Creates new wallet by random private key			
<server-url>/api/wallets/create			
Request	Method	POST	
	Header	Content-Type	application/json
		X-API-Key	Authorization Token
	Body	<pre>{   "blockchainRef": "&lt;string&gt;",   "networkRef": "&lt;string&gt;",   "label": "&lt;string&gt;",   "kmsRef": "&lt;string&gt;",   "keyExportable": "&lt;boolean&gt;" }</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<sup>۱</sup> 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<sup>۲</sup> مراجعه بفرمایید.

<sup>۱</sup> Key Derivation

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



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": "&lt;string&gt;",   "networkRef": "&lt;string&gt;",   "label": "&lt;string&gt;",   "kmsRef": "&lt;string&gt;",   "privateKeyHex": "&lt;string&gt;",   "chainCodeHex": "&lt;string&gt;",   "keyExportable": "&lt;boolean&gt;" }</pre>	
Response	Key Id	کد یکتا کلید وارد شده	
	Example	<uuid>	

#### • 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": "&lt;string&gt;",   "networkRef": "&lt;string&gt;",   "label": "&lt;string&gt;",   "kmsRef": "&lt;string&gt;",   "mnemonics": "&lt;string&gt;",   "keyExportable": "&lt;boolean&gt;" }</pre>	
Response	Key Id	کد یکتا کلید وارد شده	
	Example	<uuid>	



## ۶- مدیریت لینک‌های KMS

همانطور که در بخش ۳ توضیح داده شد، سامانه WaaS امکان اتصال به سامانه‌های مدیریت کلید متفاوت به صورت همزمان را داراست.

### • لیست تمام لینک‌های KMS

این سرویس لیست تمام لینک‌های KMS متصل را ارائه می‌کند.

Lists all kms links configured inside the system			
<server-url>/api/networks			
Request	Method	GET	
	Header	Content-Type	application/json
		X-API-Key	Authorization Token
Response	Example	<pre>[   {     "id": "&lt;uuid&gt;",     "name": "&lt;string&gt;"   },   {     "id": "&lt;uuid&gt;",     "name": "&lt;string&gt;"   } ]</pre>	

## ۷- سرویس‌های بومی<sup>۳</sup> شبکه‌های اتریومی

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

<sup>3</sup> native

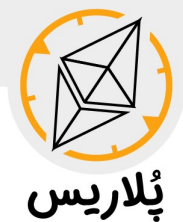


## • دریافت موجودی

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>	

## • 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.

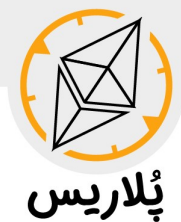


Transfers native token from source to destination			
<server-url>/api/evm/transfer			
Request	Method	POST	
	Header	Content-Type	application/json
		X-API-Key	Authorization Token
	Body	<pre>{   "blockchainRef": "&lt;string&gt;",   "networkRef": "&lt;string&gt;",   "sourceWalletId": "&lt;uuid&gt;",   "destinationRef":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

#### • نتیجه وضعیت تراکنش

این سرویس نتیجه تراکنش را به کاربر اعلام می‌کند.



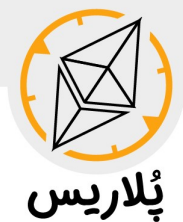


Returns transaction status for evm based networks			
<server-url>/api/evm/status			
Request	Method	GET	
	Header	Content-Type	application/json
		X-API-Key	Authorization Token
	Query Params	BlockchainRef	blockchain name or id
		NetworkRef	blockchain network name or id
		TransactionHash	public address or 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>	

## ۸- سرویس‌های مبتنی بر توکن ERC۲۰

### • دریافت موجودی

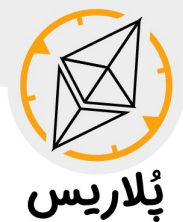
این سرویس مقدار موجودی توکن ERC۲۰ را برای یک آدرس باز می‌گرداند.



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 ERC20

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

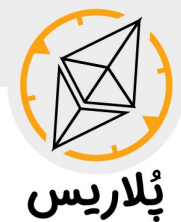


Transfers erc۲۰ token amount from source to destination			
<server-url>/api/erc۲۰/transfer			
Request	Method	POST	
	Header	Content-Type	application/json
		X-API-Key	Authorization Token
	Body	<pre>{   "blockchainRef": "&lt;string&gt;",   "networkRef": "&lt;string&gt;",   "tokenRef": "&lt;string&gt;",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

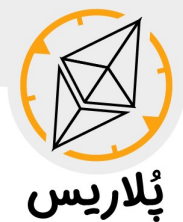
لازم به ذکر است که عاملیت تراکنش با ownerWalletId می‌باشد و در واقع می‌بایست قبل از این متد، از طریق approve به کیف پول مذکور اجازه برداشت داده شود. مقدار بازگشتی، transaction hash خواهد بود.



transfers from source (other than owner wallet) to destination address			
<server-url>/api/erc20/transfer-from			
Request	Method	POST	
	Header	Content-Type	application/json
		X-API-Key	Authorization Token
	Body	<pre>{   "blockchainRef": "&lt;string&gt;",   "networkRef": "&lt;string&gt;",   "tokenRef": "&lt;string&gt;",   "ownerWalletId": "&lt;uuid&gt;",   "sourcePublicRef": "&lt;string&gt;",   "destinationPublicRef": "&lt;string&gt;",   "amountInWei": "&lt;string&gt;",   "gasFeeinWei": "&lt;string&gt;" }</pre>	
Response	Example	<string>	

#### • مشاهده مجوز برداشت

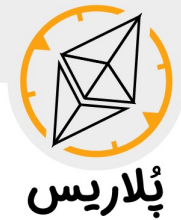
این سرویس مجوز برداشت از یک آدرس کیف پول به یک آدرس کیف پول یا قرارداد هوشمند را نمایش می‌دهد.



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": "&lt;string&gt;",   "networkRef": "&lt;string&gt;",   "tokenRef": "&lt;string&gt;",   "sourceWalletId": "&lt;uuid&gt;",   "spenderRef": "&lt;string&gt;",   "destinationPublicRef": "&lt;string&gt;",   "amount": "&lt;string&gt;",   "gasFeeinWei": "&lt;string&gt;" }</pre>	
Response	Example		

## ۹- سرویس‌های مبتنی بر توکن ERC۲۱

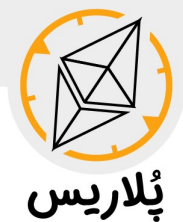
### • بارگذاری<sup>۴</sup> کالکشن

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<sup>۵</sup> 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 4

id 5

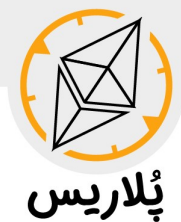


Deploy an ERC-۷۲۱ contract			
<server-url>/api/erc۷۲۱/deploy			
Request	Method	POST	
	Header	Content-Type	application/json
		Authorization	JWT Authorization
	Body	<pre>{   "blockchainRef": "&lt;string&gt;",   "networkRef": "&lt;string&gt;",   "tokenRef": "&lt;string&gt;",   "sourceWalletId": "&lt;uuid&gt;",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

#### • 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.

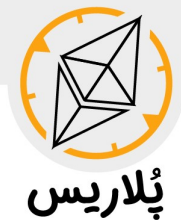




Mint a token on ERC-۷۲۱ contract			
<server-url>/api/erc۷۲۱/mint			
Request	Method	POST	
	Header	Content-Type	application/json
		Authorization	JWT Authorization
	Body	<pre>{   "blockchainRef": "&lt;string&gt;",   "networkRef": "&lt;string&gt;",   "tokenRef": "&lt;string&gt;",   "sourceWalletId": "&lt;uuid&gt;",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

### • 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 کیف‌پول بسازیم. برای این کار باید سرویس kms بالا باشد و یک کیف‌پول بسازید (این بخش در "مدیریت کیف‌پول" 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.

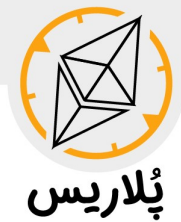


Burn a token on ERC-۷۲۱ contract			
<server-url>/api/erc۷۲۱/mint			
Request	Method	POST	
	Header	Content-Type	application/json
		Authorization	JWT Authorization
	Body	<pre>{   "blockchainRef": "&lt;string&gt;",   "networkRef": "&lt;string&gt;",   "tokenRef": "&lt;string&gt;",   "sourceWalletId": "&lt;uuid&gt;",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

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. می توانید بعد این توکن را در سایت opensea ببینید. پس از آن با سرویس انتقال توکن همین توکن را به اکانت دیگری انتقال دهید. پس از آن، توکن نباید در opensea اکانت سازنده دیده شود و باید در اکانتی که به آن فرستادید باشد.



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": "&lt;string&gt;",   "networkRef": "&lt;string&gt;",   "tokenRef": "&lt;string&gt;",   "sourceWalletId": "&lt;uuid&gt;",   "sourceAddress": "&lt;string&gt;",   "destinationAddress": "&lt;string&gt;",   "collectionAddress": "&lt;string&gt;",   "tokenId": "&lt;BigInteger&gt;",   "gasFeeinWei": "&lt;string&gt;" }</pre>	
Response	Example		