



KATUN API SERVICES

Versión 0.0.1

Descripción de contenido

Este manual es una guía de uso para la capa de servicios web Katun. Se muestra la interacción cliente-servidor mediante un servicio JSON/RPC.

Notas de lanzamiento

Versión	Fecha	Descripción
0.0.1	24, junio 2019	Versión inicial

Contenido

Katun Endpoint.....	3
Registro de usuario.....	4
Autenticación de usuario.....	4
Creación de transacción.....	5
Obtención de transacción por hash.....	6
Obtención de transacción por receiver	6
Obtención de transacción por sender	7
Obtención de todas transacciones.....	8
Creación de un Smart Contract	9
Obtención de contrato por nombre	9
Obtención de contrato por hash.....	10
Obtención de todos contratos.....	11
Ejecución de contrato por hash	11
Ejecución de contrato por nombre.....	12

Katun Endpoint

La capa de servicios de Katun te permite interactuar de una manera fácil al Blockchain, este manual describe el uso detallado de este servicio.

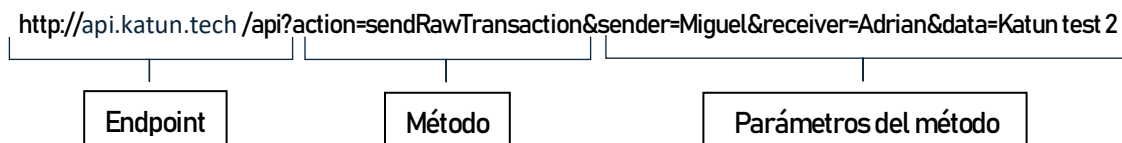
A continuación, se enlista los usos del servicio:

- Registro de nuevo usuario
- Autenticación de un usuario
- Creación de una transacción
- Obtención de una transacción
- Creación de un Smart Contract
- Obtención de un Smart Contract
- Ejecución de un Smart Contract

Endpoint	Ambiente
http://api.katun.tech/	Testing

Para todos los casos de uso, la comunicación mediante la URL se define de la siguiente manera:

Ejemplo URL:



Ejemplo de respuesta **[SUCCESS]**:

```
{
  "jsonrpc": "2.0",
  "method": "sendRawTransaction",
  "result": "cc3274a068828b7c39bc1c503e2ac72911514384da548d01c4cb1ccbf2ef4dac",
  "id": "166"
}
```

Ejemplo de respuesta **[FAIL]**:

```
{
  "jsonrpc": "2.0",
```

```

"method": "sendRawTransaction",
"error": "descripción de error",
"id": "19"
}

```

Registro de usuario

Método	Path	Método HTTP
S/N	user	POST

Parámetros	Tipo	Requerido
user	String	Si
email	String	Si
identifier	String	Si
password	String	Si

Ejemplo de body (json body)

```

{
  "user": "xellCoders",
  "email": "xell.dart@xellcoders.com",
  "identifier": "summit",
  "password": "WBqpDkgBkDdl"
}

```

Ejemplo de respuesta [SUCCESS]:

```

{
  "status": true,
  "publicKey": "Tmh7i7QsUVaUpEuse89qxH",
  "secretKey": "2a7BDUh7voswd3qVQNo2r1XEvw6Yrq1wuu1nuaN43PIF"
}

```

Autenticación de usuario

Método	Path	Método HTTP
S/M	user/auth	POST

Ejemplo de petición

```
{
  "public": "FT3fLBgM37zSFb7GTWAYh",
  "secret": "2pKhb9VQVfLEBemAiiYebhxcs5eGurK9e1HfRhgW8Hg"
}
```

Ejemplo de respuesta [SUCCESS]:

```
{
  "accessToken":
    "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJBdXRoZW50aWNhdGlvbmlzmlzcyl6ImthdHVuLnRlY2giLCJfaWQiOiI1ZDBlYTl2ZWZmZWQ0MTIzZTdlMzNkN2liLCJleHAiOiJlNjEyNzQyNzZ9.Kh7sLP8H39vqOepTwtFht0W02ysmQnlrOADVgiSoYdZg23CF_Y99qZv0GL-iXrjNOrBVuNTqRSyUbrt-ZPJA",
  "expireIn": "1 hour"
}
```

Ejemplo de salida [FAIL]:

Código HTTP 401 – No autorizado

El token generado debe enviarse en los siguientes métodos como manera de autenticación en las cabeceras de la petición, el token tiene una vida de una 1 hora:

Ejemplo de cabeceras:

Authorization: Bearer

eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJBdXRoZW50aWNhdGlvbmlzmlzcyl6ImthdHVuLnRlY2giLCJfaWQiOiI1ZDBlZTA5ZDgyMMwMDRjMWE3NDg1NjYiLCJleHAiOiJlNjEyNjM2NjV9.Tmw5uHqFXyRdv80QML4j2li4vZsTcttYBRtUXNCbWHfyeFYRG2jDcRyuimVWP-13-dwTRNmZdDSixplv3u4Hbg

Creación de transacción

Método	Path	Método HTTP
sendRawTransaction	api	GET

Parámetros	Tipo	Requerido
sender	String	Si
receiver	String	Si
data	String	Si

Ejemplo de respuesta [SUCCESS]:

```
{
  "jsonrpc": "2.0",
```

```

"method": "sendRawTransaction",
"result": "cc3274a068828b7c39bc1c503e2ac72911514384da548d01c4cb1ccbf2ef4dac",
"id": "166"
}

```

Obtención de transacción por hash

Método	Path	Método HTTP
getTransactionByHash	api	GET

Parámetros	Tipo	Requerido
txHash	String	Si

Ejemplo de respuesta [SUCCESS]:

```

{
  "jsonrpc": "2.0",
  "method": "getTransactionByHash",
  "result": {
    "txHash": "577cb92186450b7a6ba7b67efd4094b441a3e4e69cac23ca5ea982bfa092c63b",
    "from": "Miguel",
    "to": "Adrián",
    "data": "Katun test 2",
    "version": "1561396228622",
    "creation": {
      "num": 1561396228622
    }
  },
  "id": "362"
}

```

Obtención de transacción por receiver

Método	Path	Método HTTP
getTransactionByReceiver	api	GET

Parámetros	Tipo	Requerido
receiver	String	Si
page	String	No
offset	String	No

- Parámetros para paginado: page = número de página, offset = resultados por página

Ejemplo de respuesta [SUCCESS]:

```
{
  "jsonrpc": "2.0",
  "method": "getTransactionByHash",
  "result": {
    "txHash": "577cb92186450b7a6ba7b67efd4094b441a3e4e69cac23ca5ea982bfa092c63b",
    "from": "Miguel",
    "to": "Adrián",
    "data": "Katun test 2",
    "version": "1561396228622",
    "creation": {
      "num": 1561396228622
    }
  },
  "id": "362"
}
```

Obtención de transacción por sender

Método	Path	Método HTTP
getTransactionBySender	api	GET

Parámetros	Tipo	Requerido
sender	String	Si
page	String	No
offset	String	No

- Parámetros para paginado: page = número de página, offset = resultados por página

Ejemplo de respuesta [SUCCESS]:

```
{
  "jsonrpc": "2.0",
  "method": "getTransactionByHash",
  "result": {
    "txHash": "577cb92186450b7a6ba7b67efd4094b441a3e4e69cac23ca5ea982bfa092c63b",
```

```

"from": "Miguel",
"to": "Adrián",
"data": "Katun test 2",
"version": "1561396228622",
"creation": {
  "num": 1561396228622
}
},
"id": "362"
}

```

Obtención de todas transacciones

Método	Path	Método HTTP
getAllTransactions	api	GET

Parámetros	Tipo	Requerido
page	String	No
offset	String	No

- Parámetros para paginado: page = número de página, offset = resultados por página

Ejemplo de respuesta [SUCCESS]:

```

[
{
  "jsonrpc": "2.0",
  "method": "getTransactionByHash",
  "result": {
    "txHash": "577cb92186450b7a6ba7b67efd4094b441a3e4e69cac23ca5ea982bfa092c63b",
    "from": "Miguel",
    "to": "Adrián",
    "data": "Katun test 2",
    "version": "1561396228622",
    "creation": {
      "num": 1561396228622
    }
  },
  "id": "362"
}

```



```
}
```

Creación de un Smart Contract

Método	Path	Método HTTP
createSmartContract	api	POST

Parámetros	Tipo	Requerido
File .java	Multipart	Si

Ejemplo de respuesta **[SUCCESS]**:

```
{
  "jsonrpc": "2.0",
  "method": "createSmartContract",
  "result": "cc3274a068828b7c39bc1c503e2ac72911514384da548d01c4cb1ccbf2ef4dac",
  "id": "166"
}
```

Errores

Error	Descripción
Class not public	Necesitas declarar tu clase principal con public
Contract name already exist	Ya hay un contrato con asociado a tu cuenta
Extension java not detected	El archivo debe terminar en .java
Sintaxis error	Descripción de errores de sintaxis en el archivo
File name not match with class	El archivo y la clase deben llamarse igual

Obtención de contrato por nombre

Método	Path	Método HTTP
getContractByName	api	GET

Parámetros	Tipo	Requerido
contractName	String	Si

Ejemplo de respuesta **[SUCCESS]**:

```
{
  "jsonrpc": "2.0",
  "method": "getContractByName",
  "result": {
    "txContract": "f41954a699dfa2b3b4a02eb846dcfa7343ff05ad905a67c0efd672cf5063b189",
    "contractName": "Test",
    "compilationMessages": "Must declare java class as public.",
    "content": "class Test {\r\n  public static void main(String args[]) {\r\n    System.out.println(\"Hello Java\");\r\n  }\r\n\r\n  public int getSum(int a, int b) {\r\n    return a + b;\r\n  }\r\n}",
    "version": "1561264253401"
  },
  "id": "307"
}
```

Obtención de contrato por hash

Método	Path	Método HTTP
getContractByHash	api	GET

Parámetros	Tipo	Requerido
txContract	String	Si

Ejemplo de respuesta [SUCCESS]:

```
{
  "jsonrpc": "2.0",
  "method": "getContractByName",
  "result": {
    "txContract": "f41954a699dfa2b3b4a02eb846dcfa7343ff05ad905a67c0efd672cf5063b189",
    "contractName": "Test",
    "compilationMessages": "Must declare java class as public.",
    "content": "class Test {\r\n  public static void main(String args[]) {\r\n    System.out.println(\"Hello Java\");\r\n  }\r\n\r\n  public int getSum(int a, int b) {\r\n    return a + b;\r\n  }\r\n}",
    "version": "1561264253401"
  },
  "id": "307"
}
```

Obtención de todos contratos

Método	Path	Método HTTP
getAllContracts	api	GET

Ejemplo de respuesta [SUCCESS]:

```
{
  "jsonrpc": "2.0",
  "method": "getContractByName",
  "result": {
    "txContract": "f41954a699dfa2b3b4a02eb846dcfa7343ff05ad905a67c0efd672cf5063b189",
    "contractName": "Test",
    "compilationMessages": "Must declare java class as public.",
    "content": "class Test {\r\n  public static void main(String args[]) {\r\n    System.out.println(\"Hello Java\");\r\n  }\r\n\r\n  public int getSum(int a, int b) {\r\n    return a + b;\r\n  }\r\n}",
    "version": "1561264253401"
  },
  "id": "307"
}
```

Ejecución de contrato por hash

Método	Path	Método HTTP
executeContractByHash	api	POST

Parámetros	Tipo	Requerido
TxContract ()	String	Si
execute	String	Si

Ejemplo de body (json body)

```
{
```

```

    "execute": "function:getSum:arg:integer:{10}:arg:integer:{20}:function:getMessage:"
  }

```

Ejemplo de respuesta [SUCCESS]:

```

{
  "jsonrpc": "2.0",
  "method": "executeContractByHash",
  "result": {
    "txContract": "9ac859353cbf15b1eb3c72c27c0a68d0113880c8ecb9bcf194472579667ed200",
    "result": {
      "getSum": {
        "value": "30"
      },
      "getMessage": {
        "value": "Message: Example..."
      }
    }
  },
  "id": "280"
}

```

Ejecución de contrato por nombre

Método	Path	Método HTTP
executeContractByName	api	POST

Parámetros	Tipo	Requerido
contractName (query)	String	Si
execute	String	Si

Ejemplo de respuesta [SUCCESS]:

```

{
  "jsonrpc": "2.0",
  "method": "executeContractByName",
  "result": {
    "txContract": "9ac859353cbf15b1eb3c72c27c0a68d0113880c8ecb9bcf194472579667ed200",

```

```
"result": {  
  "getSum": {  
    "value": "30"  
  },  
  "getMessage": {  
    "value": "Message: Example..."  
  }  
},  
"id": "280"  
}
```