🚀 QuantPay Chain API Documentation

Overview

The QuantPay Chain API provides endpoints for Web3 document signing, blockchain integration, and user management. This RESTful API enables developers to integrate decentralized document workflows into their applications.

Base URL: `https://www.quantpaychain.com/api`
Authentication: JWT tokens or Web3 signatures
Rate Limit: 100 requests per minute

🔐 Authentication

1. Email/Password Authentication

```
POST /api/auth/signin
Content-Type: application/json

{
  "email": "user@example.com",
  "password": "securepassword123"
}
```

Response:

```
{
  "user": {
    "id": "user_123",
    "email": "user@example.com",
    "name": "John Doe",
    "plan": "starter"
  },
  "token": "jwt_token_here"
}
```

2. Web3 Wallet Authentication (SIWE)

```
POST /api/auth/siwe
Content-Type: application/json

{
  "message": "Sign in to QuantPay Chain: 0x742d35...",
  "signature": "0x1234567890abcdef..."
}
```

👤 User Management

Create User Account

```
POST /api/signup
Content-Type: application/json

{
  "email": "newuser@example.com",
  "password": "securepassword123",
  "firstName": "Jane",
  "lastName": "Smith",
  "walletAddress": "0x742d35Cc6634C0532925a3b8D29B5A33B3f7B0A5"
}
```

Get User Profile

```
GET /api/user/profile
Authorization: Bearer <jwt_token>
```

📄 Document Management

1. Upload Document

Uploads a document to IPFS and registers it on the blockchain.

```
POST /api/documents/upload
Authorization: Bearer <jwt_token>
Content-Type: application/json

{
  "title": "Contract Agreement",
  "description": "Service agreement contract",
  "ipfsHash": "QmYour43CharacterIPFSHashHere123456789",
  "blockchainId": "0xblockchain_transaction_hash",
  "fileName": "contract.pdf",
  "fileSize": 245760,
  "fileType": "application/pdf",
  "signers": ["0x742d35Cc...", "0x8f9e10Ab..."],
  "requiresMultiSig": false
}
```

Response:

```
{
  "id": "doc_123456",
  "title": "Contract Agreement",
  "status": "PENDING",
  "ipfsHash": "QmYour43CharacterIPFSHashHere123456789",
  "blockchainId": "0xblockchain_transaction_hash",
  "createdAt": "2024-09-24T10:30:00Z",
  "signatures": [
    {
      "id": "sig_123",
      "status": "PENDING",
      "signerAddress": "0x742d35Cc..."
    }
  ]
}
```

2. Get Document Details

```
GET /api/documents/{id}
Authorization: Bearer <jwt_token>
```

Response:

```
{
  "id": "doc_123456",
  "title": "Contract Agreement",
  "description": "Service agreement contract",
  "status": "PENDING",
  "ipfsHash": "QmYour43CharacterIPFSHashHere123456789",
  "blockchainId": "0xblockchain_transaction_hash",
  "fileName": "contract.pdf",
  "fileSize": 245760,
  "fileType": "application/pdf",
  "createdAt": "2024-09-24T10:30:00Z",
  "updatedAt": "2024-09-24T10:30:00Z",
  "creator": {
    "id": "user_123",
    "name": "John Doe",
    "email": "john@example.com"
  },
  "signatures": [
    {
      "id": "sig_123",
      "status": "PENDING",
      "signerAddress": "0x742d35Cc...",
      "createdAt": "2024-09-24T10:30:00Z"
    }
  ]
}
```

3. Download Document

```
GET /api/documents/{id}/download
Authorization: Bearer <jwt_token>
```

Returns the original document file as binary data.

4. List User Documents

```
GET /api/documents?page=1&limit=10&status=PENDING
Authorization: Bearer <jwt_token>
```

Query Parameters:
- `page` (optional): Page number (default: 1)
- `limit` (optional): Items per page (default: 10, max: 100)
- `status` (optional): Filter by status (`PENDING` , `COMPLETED` , `REJECTED`)
- `search` (optional): Search in document titles

Response:

```json
{
  "documents": [
    {
      "id": "doc_123456",
      "title": "Contract Agreement",
      "status": "PENDING",
      "createdAt": "2024-09-24T10:30:00Z",
      "signaturesCount": 1,
      "completedSignatures": 0
    }
  ],
  "pagination": {
    "page": 1,
    "limit": 10,
    "total": 25,
    "totalPages": 3
  }
}
```

✍️ Signature Management

1. Sign Document

Signs a document using blockchain verification.

```
POST /api/documents/{id}/sign
Authorization: Bearer <jwt_token>
Content-Type: application/json

{
  "signedIpfsHash": "QmSignedDocumentHashHere123456789",
  "transactionHash": "0xblockchain_signature_transaction",
  "signature": "0xweb3_signature_data"
}
```

Response:

```
{
  "id": "sig_123",
  "status": "SIGNED",
  "signedAt": "2024-09-24T11:15:00Z",
  "transactionHash": "0xblockchain_signature_transaction",
  "document": {
    "id": "doc_123456",
    "status": "COMPLETED"
  }
}
```

2. Get Signature Details

```
GET /api/signatures/{id}
Authorization: Bearer <jwt_token>
```

📊 Usage & Analytics

Get Usage Statistics

```
GET /api/usage/stats
Authorization: Bearer <jwt_token>
```

Response:

```
{
  "currentMonth": {
    "documentsUploaded": 8,
    "documentsSigned": 6,
    "documentsCompleted": 5
  },
  "plan": {
    "name": "starter",
    "documentsLimit": 50,
    "documentsRemaining": 42
  },
  "allTime": {
    "totalDocuments": 23,
    "totalSignatures": 31,
    "totalStorageUsed": "15.7MB"
  }
}
```

Reset Monthly Usage (Admin)

```
POST /api/usage/reset
Authorization: Bearer <admin_jwt_token>
```

🔍 Blockchain Integration

Verify Document on Blockchain

```
GET /api/blockchain/verify/{blockchainId}
Authorization: Bearer <jwt_token>
```

Response:

```json
{
  "blockchainId": "0xblockchain_transaction_hash",
  "verified": true,
  "blockNumber": 18524567,
  "timestamp": "2024-09-24T10:30:00Z",
  "gasUsed": 156742,
  "documentHash": "0xdocument_content_hash",
  "signers": [
    {
      "address": "0x742d35Cc...",
      "signed": true,
      "timestamp": "2024-09-24T11:15:00Z"
    }
  ]
}
```

🚨 Error Handling

All API endpoints return appropriate HTTP status codes and error messages.

Common Status Codes

- `200 OK` - Request successful
- `201 Created` - Resource created successfully
- `400 Bad Request` - Invalid request data
- `401 Unauthorized` - Authentication required
- `403 Forbidden` - Insufficient permissions
- `404 Not Found` - Resource not found
- `429 Too Many Requests` - Rate limit exceeded
- `500 Internal Server Error` - Server error

Error Response Format

```json
{
  "error": {
    "code": "DOCUMENT_NOT_FOUND",
    "message": "The requested document could not be found",
    "details": {
      "documentId": "doc_123456"
    }
  }
}
```

Common Error Codes

- `AUTHENTICATION_REQUIRED` - Missing or invalid authentication
- `INVALID_SIGNATURE` - Web3 signature verification failed
- `DOCUMENT_NOT_FOUND` - Document doesn’t exist or access denied
- `PLAN_LIMIT_EXCEEDED` - Monthly document limit reached
- `INVALID_IPFS_HASH` - Malformed IPFS hash
- `BLOCKCHAIN_ERROR` - Smart contract interaction failed
- `FILE_TOO_LARGE` - Document exceeds size limit (10MB)

📚 SDK Examples

JavaScript/TypeScript

```
import { QuantPayChainAPI } from '@quantpaychain/sdk';

const api = new QuantPayChainAPI({
  baseURL: 'https://www.quantpaychain.com/api',
  apiKey: 'your_api_key_here'
});

// Upload document
const document = await api.documents.upload({
  title: 'Contract Agreement',
  file: fileBuffer,
  signers: ['0x742d35Cc...']
});

// Sign document
await api.documents.sign(document.id, {
  signature: await wallet.signMessage(message)
});
```

Python

```
from quantpaychain import QuantPayChainClient

client = QuantPayChainClient(
    base_url="https://www.quantpaychain.com/api",
    api_key="your_api_key_here"
)

# Upload document
document = client.documents.upload(
    title="Contract Agreement",
    file=file_data,
    signers=["0x742d35Cc..."]
)

# Get document status
status = client.documents.get(document.id)
```

🔄 Webhooks

QuantPay Chain supports webhooks for real-time event notifications.

Supported Events

- `document.uploaded` - New document uploaded
- `document.signed` - Document signature completed
- `document.completed` - All required signatures collected
- `document.expired` - Document signature deadline passed

Webhook Payload Example

```
{
  "event": "document.signed",
  "timestamp": "2024-09-24T11:15:00Z",
  "data": {
    "documentId": "doc_123456",
    "signatureId": "sig_123",
    "signer": "0x742d35Cc...",
    "transactionHash": "0xblockchain_signature_transaction"
  }
}
```

Webhook Configuration

```
POST /api/webhooks
Authorization: Bearer <jwt_token>
Content-Type: application/json

{
  "url": "https://your-app.com/webhooks/quantpaychain",
  "events": ["document.signed", "document.completed"],
  "secret": "webhook_secret_for_verification"
}
```

🔒 Security Best Practices

API Key Management

- Store API keys securely (environment variables)
- Rotate keys regularly
- Use different keys for different environments

Rate Limiting

- Implement client-side rate limiting
- Handle 429 responses gracefully with exponential backoff

Data Validation

- Always validate IPFS hashes format
- Verify blockchain transaction hashes
- Sanitize user inputs

Web3 Integration

- Verify wallet signatures on both client and server
- Use secure random nonces for SIWE messages
- Implement proper session management

📞 Support

Technical Support

- **Email**: api-support@quantpaychain.com
- **Documentation**: docs.quantpaychain.com (https://docs.quantpaychain.com)
- **Discord**: Join Developer Community (https://discord.gg/quantpay-dev)

API Status

- **Status Page**: status.quantpaychain.com (https://status.quantpaychain.com)
- **Uptime**: 99.9% SLA
- **Response Time**: <500ms average

Last Updated: September 24, 2024
API Version: v1.0