📚 Portal NewLoc - Documentação da API

🔐 Autenticação

Todas as rotas protegidas requerem autenticação via cookie de sessão. O middleware injeta automaticamente os headers `x-user-type` e `x-user-client` nas requisições autenticadas.

🛣️ Rotas Disponíveis

1. Health Check

Endpoint: `/api/health`
Método: `GET`
Autenticação: Não requerida
Descrição: Verifica se a API está funcionando corretamente.

Exemplo de Resposta:

```
{
  "status": "ok",
  "timestamp": "2024-11-14T19:00:00.000Z",
  "service": "Portal NewLoc"
}
```

Status Codes:
- `200` - Serviço funcionando

---

2. Login

Endpoint: `/api/auth/login`
Método: `POST`
Autenticação: Não requerida
Descrição: Autentica um usuário e cria uma sessão.

Body da Requisição:

```
{
  "email": "admin@newloc.com",
  "password": "Admin@123"
}
```

Exemplo de Resposta (Sucesso):

```
{
  "user": {
    "id": "123e4567-e89b-12d3-a456-426614174000",
    "email": "admin@newloc.com",
    "tipo": "admin",
    "cliente": null
  },
  "token": "abc123def456..."
}
```

Headers de Resposta:

- `Set-Cookie: session_token=...` (HttpOnly, SameSite=Strict)

Status Codes:

- `200` - Login bem-sucedido
- `400` - Campos obrigatórios faltando
- `401` - Credenciais inválidas
- `500` - Erro interno

Exemplo cURL:

```
curl -X POST https://app.newloc.com/api/auth/login \
  -H "Content-Type: application/json" \
  -d '{"email":"admin@newloc.com","password":"Admin@123"}'
```

---

3. Logout

Endpoint: `/api/auth/logout`
Método: `POST`
Autenticação: Requerida
Descrição: Encerra a sessão do usuário.

Exemplo de Resposta:

```
{
  "message": "Logout realizado com sucesso"
}
```

Status Codes:

- `200` - Logout bem-sucedido
- `401` - Não autenticado

Exemplo cURL:

```
curl -X POST https://app.newloc.com/api/auth/logout \
  -H "Cookie: session_token=YOUR_TOKEN"
```

---

4. Listar Documentos

Endpoint: `/api/documentos`
Método: `GET`
Autenticação: Requerida
Descrição: Lista documentos de acordo com as permissões do usuário.

Regras de Permissão:
- **Admin:** Vê todos os documentos
- **Cliente:** Vê apenas documentos do próprio cliente

Exemplo de Resposta:

```
[
  {
    "id": "123e4567-e89b-12d3-a456-426614174000",
    "date": "2024-11-14T00:00:00.000Z",
    "cliente": "Construtora Silva",
    "dataDocumento": "2024-11-14T00:00:00.000Z",
    "remessa": "REM-2024-001",
    "contrato": "CTR-2024-001",
    "operacao": "entrega",
    "patrimonios": ["PAT-001", "PAT-002", "PAT-003"],
    "documentacaoImagem": "...",
    "status": "ativo",
    "createdAt": "2024-11-14T10:30:00.000Z",
    "updatedAt": "2024-11-14T10:30:00.000Z"
  }
]
```

Status Codes:
- `200` - Sucesso
- `401` - Não autenticado
- `500` - Erro interno

Exemplo cURL:

```
curl -X GET https://app.newloc.com/api/documentos \
  -H "Cookie: session_token=YOUR_TOKEN"
```

---

5. Obter Documento por ID

Endpoint: `/api/documento/{id}`
Método: `GET`
Autenticação: Requerida
Descrição: Retorna os detalhes completos de um documento específico.

Parâmetros de URL:
- `id` (UUID) - ID do documento

Regras de Permissão:
- **Admin:** Pode acessar qualquer documento
- **Cliente:** Só pode acessar documentos do próprio cliente

Exemplo de Resposta:

```
{
  "id": "123e4567-e89b-12d3-a456-426614174000",
  "date": "2024-11-14T00:00:00.000Z",
  "cliente": "Construtora Silva",
  "dataDocumento": "2024-11-14T00:00:00.000Z",
  "remessa": "REM-2024-001",
  "contrato": "CTR-2024-001",
  "operacao": "entrega",
  "patrimonios": ["PAT-001", "PAT-002", "PAT-003"],
  "documentacaoImagem": "...",
  "status": "ativo",
  "createdAt": "2024-11-14T10:30:00.000Z",
  "updatedAt": "2024-11-14T10:30:00.000Z"
}
```

Status Codes:
- `200` - Sucesso
- `401` - Não autenticado
- `403` - Acesso negado (cliente tentando acessar documento de outro cliente)
- `404` - Documento não encontrado
- `500` - Erro interno

Exemplo cURL:

```
curl -X GET https://app.newloc.com/api/documento/123e4567-e89b-12d3-a456-426614174000
\
  -H "Cookie: session_token=YOUR_TOKEN"
```

---

6. Listar Usuários (Admin Only)

Endpoint: `/api/usuarios`
Método: `GET`
Autenticação: Requerida (Admin)
Descrição: Lista todos os usuários do sistema (apenas para administradores).

Exemplo de Resposta:

```
[
  {
    "id": "123e4567-e89b-12d3-a456-426614174000",
    "email": "admin@newloc.com",
    "tipo": "admin",
    "cliente": null,
    "ativo": true,
    "criadoEm": "2024-11-14T10:00:00.000Z"
  },
  {
    "id": "987e6543-e21b-12d3-a456-426614174000",
    "email": "cliente@empresa.com",
    "tipo": "cliente",
    "cliente": "Construtora Silva",
    "ativo": true,
    "criadoEm": "2024-11-14T11:00:00.000Z"
  }
]
```

Status Codes:
- `200` - Sucesso
- `401` - Não autenticado
- `403` - Acesso negado (não é admin)
- `500` - Erro interno

---

7. Criar Usuário (Admin Only)

Endpoint: `/api/usuarios`
Método: `POST`
Autenticação: Requerida (Admin)
Descrição: Cria um novo usuário do tipo Cliente (apenas para administradores).

Body da Requisição:

```
{
  "email": "novo@cliente.com",
  "password": "senha123",
  "cliente": "Nome da Empresa"
}
```

Validações:
- Email deve ser único
- Senha deve ter no mínimo 6 caracteres
- Campo `cliente` é obrigatório
- Tipo sempre será “cliente”

Exemplo de Resposta (Sucesso):

```
{
  "id": "456e7890-e12b-34d5-a678-901234567890",
  "email": "novo@cliente.com",
  "tipo": "cliente",
  "cliente": "Nome da Empresa",
  "ativo": true,
  "criadoEm": "2024-11-14T15:00:00.000Z"
}
```

Status Codes:
- `201` - Usuário criado com sucesso
- `400` - Dados inválidos ou email já existe
- `401` - Não autenticado
- `403` - Acesso negado (não é admin)
- `500` - Erro interno

Exemplo cURL:

```
curl -X POST https://app.newloc.com/api/usuarios \
  -H "Content-Type: application/json" \
  -H "Cookie: session_token=YOUR_TOKEN" \
  -d '{
    "email": "novo@cliente.com",
    "password": "senha123",
    "cliente": "Nome da Empresa"
  }'
```

🔒 Segurança

Cookies de Sessão

A autenticação é gerenciada através de cookies HttpOnly com as seguintes características:
- **Nome:** `session_token`
- **HttpOnly:** `true` (não acessível via JavaScript)
- **SameSite:** `Strict`
- **Secure:** `true` (apenas HTTPS em produção)
- **Expiração:** 8 horas

Headers Injetados pelo Middleware

Para rotas autenticadas, o middleware injeta automaticamente:
- `x-user-type`: Tipo do usuário (`admin` ou `cliente`)
- `x-user-client`: Nome do cliente (apenas para tipo `cliente`)
- `x-user-id`: ID do usuário

📊 Modelos de Dados

UsuariosPortal

```
{
  id: string;         // UUID
  email: string;      // Único
  senhaHash: string;  // Bcrypt hash
  cliente: string | null; // Nome do cliente (apenas para tipo 'cliente')
  tipo: 'admin' | 'cliente';
  ativo: boolean;
  criadoEm: Date;
}
```

DocumentosOperacoes

```
{
  id: string;                 // UUID
  date: Date;
  cliente: string;
  dataDocumento: Date;
  remessa: string;
  contrato: string;
  operacao: string;           // entrega, retirada, devolução, etc.
  patrimonios: any[];         // Array JSON de patrimônios
  documentacaoImagem: string; // Base64 da imagem
  status: string;
  createdAt: Date;
  updatedAt: Date;
}
```

SessionsPortal

```
{
  id: string;       // UUID
  token: string;    // Token único da sessão
  userId: string;   // UUID do usuário
  criadoEm: Date;
  expiracao: Date;  // 8 horas após criação
}
```

⚠️ Códigos de Status HTTP

Código	Significado
200	Sucesso
201	Recurso criado com sucesso
400	Requisição inválida
401	Não autenticado
403	Acesso negado
404	Recurso não encontrado
500	Erro interno do servidor

🔄 Fluxo de Autenticação Completo

1. **Login:**
   `POST /api/auth/login`
   `→ Retorna cookie session_token`

2. **Requisições Autenticadas:**
   `GET /api/documentos`
   `Cookie: session_token=...`
   `→ Middleware valida sessão`
   `→ Injeta headers x-user-*`
   `→ API processa com permissões`

3. **Logout:**
   `POST /api/auth/logout`
   `→ Deleta sessão do banco`
   `→ Remove cookie`

🧪 Testando a API

Usando cURL

```
# 1. Login
LOGIN_RESPONSE=$(curl -s -c cookies.txt -X POST https://app.newloc.com/api/auth/login
\
  -H "Content-Type: application/json" \
  -d '{"email":"admin@newloc.com","password":"Admin@123"}')

echo $LOGIN_RESPONSE

# 2. Listar documentos
curl -b cookies.txt https://app.newloc.com/api/documentos

# 3. Obter documento específico
curl -b cookies.txt https://app.newloc.com/api/documento/ID_DO_DOCUMENTO

# 4. Criar usuário (admin apenas)
curl -b cookies.txt -X POST https://app.newloc.com/api/usuarios \
  -H "Content-Type: application/json" \
  -d '{
    "email": "novo@cliente.com",
    "password": "senha123",
    "cliente": "Empresa Teste"
  }'

# 5. Logout
curl -b cookies.txt -X POST https://app.newloc.com/api/auth/logout
```

Usando Postman

1. **Configurar Environment:**
   - `base_url`: `https://app.newloc.com`
   - `token`: (será preenchido automaticamente)

2. **Login Collection:**
   - Criar requisição POST para `{{base_url}}/api/auth/login`
   - Em “Tests”, adicionar:
   `javascript`
   `pm.environment.set("token", pm.cookies.get("session_token"));`

3. **Outras Requisições:**
   - Automaticamente utilizarão o cookie armazenado

📝 Notas Importantes

1. **Imagens em Base64:** O campo `documentacaoImagem` contém a imagem completa em base64. Para otimização, considere implementar thumbnails em uma versão futura.

2. **CORS:** Em produção, configure o CORS adequadamente no nginx.conf ou no Next.js para permitir requisições do n8n.

3. **Rate Limiting:** O nginx.conf inclui rate limiting. Ajuste conforme necessário:
   - API: 10 requisições/segundo
   - Login: 5 tentativas/minuto

4. **Timeout:** Requisições grandes (com imagens) podem demorar mais. Configure timeouts adequados no nginx e no cliente.

5. **Validação de Sessão:** As sessões expiram após 8 horas. Implemente refresh automático no cliente se necessário.