

18/02/2020

DOCUMENTAÇÃO UPQUERY BACKEND (API)

1. CONTEXTO

A. Oque é?

- i. Um conjunto de varias APIs para o envio de e-mails, sms e relatórios a clientes da UpQuery.

B. Aonde está disponível?

- i. No servidor 177.11.48.195/root/pass@#2019
- ii. Na url **backend.upquery.com/api/v1**

C. Possui banco de dados?

- i. Sim, Postgres 12.1, database upquery (backend.upquery.com:5432/upquery)
- ii. Usuário admin: postgres/upQu3ryR00tS3c43t
- iii. Usuário da aplicação: backend/b1ck3ndS3cr3t

D. Quais tabelas existem?

- i. **CLIENT_CONFIG**: liga a chave de API a um código de cliente da UpQuery, possui as configurações de smtp e sms.
- ii. **MAIL**: Registra os e-mails enviados (endereços, assunto, tamanhos, url do relatório e o tipo (SIMPLE para e-mails comuns, REPORT para relatórios).
- iii. **SMS**: Registra os smss enviados (telefone, mensagem, status).

E. Quais endpoints existem?

- i. **/client**: Para o cadastro/update/remoção de clientes.
- ii. **/mail**: Para o envio de e-mails.
- iii. **/report**: Para a geração e envio de relatórios (páginas web em PDF).
- iv. **/sms**: Para o envio de smss.

F. Detalhes adicionais

- i. Enviar em toda request o HEADER api-key.
- ii. Para os endpoints terminados em /secure (/client/secure por exemplo) é utilizada a criptografia 3DES para todo o conteúdo (headers não) e o content-type "application/octet-stream"
- iii. Chave de API cadastrada para a UpQuery (código cliente **08080002**): **9193c7b5a2636edb8637927c787f4975**, Secret 3DES: porcocaneco
- iv. A chave da upquery é a única que consegue cadastrar e remover clientes.

2. ENDPOINT: CLIENTES

A. POST /client, /client/secure

- i. Cria um novo registro de Cliente e chave de API (apenas chave da UpQuery, api-key, identificador único).
- ii. Campos:
 1. *@NotNull String apiKey;*
 2. *@NotNull String codigoCliente;*
 3. *@NotNull String secretMd5;*
 4. *@NotNull String smtpHost;*
 5. *@NotNull @Min(80) Integer smtpPort;*
 6. *@NotNull String smtpUser;*
 7. *@NotNull String smtpPassword;*
 8. *@NotNull String smtpFrom;*
 9. *@NotNull Boolean smtpSsl;*
 10. *String smsFrom;*
 11. *String smsSecret;*

B. PUT /client, /client/secure

- i. Atualiza um registro de Cliente. (apenas chave da UpQuery ou do próprio cliente).
- ii. Campos:
 1. *@NotNull String apiKey;*
 2. *String secretMd5;*
 3. *String smtpHost;*
 4. *@Min(80) Integer smtpPort;*
 5. *String smtpUser;*
 6. *String smtpPassword;*
 7. *String smtpFrom;*
 8. *Boolean smtpSsl;*
 9. *String smsFrom;*
 10. *String smsSecret;*

C. DELETE /client/\$api-key

- i. Remove um registro de Cliente (apenas chave da UpQuery, a própria chave da UpQuery não pode ser removida).
- ii. Parâmetros:
 1. *api-key: Passado na URL.*

3. ENDPOINT: E-MAILS

A. POST /mail, /mail/secure

- i. Envia um e-mail.
- ii. Campos:
 - 1. *@NotNull String[] to;*
 - 2. *String[] toCc;*
 - 3. *String[] toBcc;*
 - 4. *String mailType;*
 - 5. *String metadata;*
 - 6. *String fromAddress;*
 - 7. *String fromName;*
 - 8. *@NotNull String subject;*
 - 9. *@NotNull String message;*
 - 10. *byte[] attachment;*
 - 11. *String attachmentName; (obrigatório se attachment informado)*
 - 12. *String attachmentType; (obrigatório se attachment informado)*

4. ENDPOINT: RELATÓRIOS

A. POST /report, /report/secure

- i. Envia uma URL como anexo PDF por e-mail.
- ii. Campos:
 - 1. *@NotNull String[] to;*
 - 2. *String[] toCc;*
 - 3. *String[] toBcc;*
 - 4. *String fromAddress;*
 - 5. *String fromName;*
 - 6. *@NotNull String subject;*
 - 7. *@NotNull String message;*
 - 8. *@NotNull String url;*
 - 9. *@Min(100) Integer width;*
 - 10. *@Min(100) Integer height;*
 - 11. *String urlUser;*
 - 12. *String urlPassword;*
 - 13. *String attachmentName;*

5. ENDPOINT: SMS

A. POST /sms, /sms/secure

- i. Envia um sms.
- ii. Campos:
 - 1. *@NotNull @Min(100) Long to;*
 - 2. *@NotNull String message;*
 - 3. *String from;*
 - 4. *String id; (ID externo de controle)*
 - 5. *String aggregateId; (ID de agregação da Zenvia (centro de custo))*
 - 6. *@Formato("yyyyMMddHHmmss") Date schedule;*

6. DETALHES DOS CAMPOS

- A. @NotNull obrigatórios
- B. @Min(VALOR): valor mínimo, se informado.
- C. Arrays devem ser declarados com [] no json (conforme exemplo com múltiplos endereços.
- D. toCc = Com cópia, toBcc = Com cópia oculta
- E. byte[] = String com a representação base64 do anexo.
- F. fromName/fromAddress opcionais, se não informados pega da configuração.
- G. mailType opcional, se não informado é "SIMPLE"
- H. attachmentName opcional, se não informado é "anexo.pdf"
- I. Resolução padrão do relatório: 1920x1080 (fullHD)