CHECKLIST: INICIO DO DIA DE TRABALHO COM FIREBASE CLOUD FUNCTIONS (LOCAL)

Este guia ajuda a configurar e testar suas Cloud Functions (index.js) no seu computador (localhost) usando os Emuladores do Firebase.

PREREQUISITOS (JA CONFIGURADOS):

* Node.js instalado.
* Firebase CLI (firebase-tools) instalado globalmente.
* Sua pasta de projeto local do Firebase (ex: C:\Users\Ana Carla\Desktop\meu-crm-firebase).
* Login no Firebase CLI (firebase login) feito alguma vez.

PASSO A PASSO PARA COMECAR A TRABALHAR:

PASSO 1: ABRIR O TERMINAL Abra o Terminal, Prompt de Comando ou PowerShell.

PASSO 2: NAVEGAR ATE A PASTA DO SEU PROJETO Use o comando 'cd' para ir para a pasta raiz do seu projeto. Comando: cd C:\Users\Ana Carla\Desktop\meu-crm-firebase

PASSO 3: INICIAR OS EMULADORES DO FIREBASE Este passo "liga" o ambiente de teste local. Deixe este terminal ABERTO. Comando: firebase emulators:start

O que esperar:

* Mensagens sobre quais emuladores estao iniciando.
* Mensagem: "✔ All emulators ready!".
* IMPORTANTES: Anote a URL local do Functions (ex: [http://127.0.0.1:5001](http://127.0.0.1:5001/) ) e a URL da UI do Emulador (ex: [http://127.0.0.1:4000](http://127.0.0.1:4000/) ).
* Este terminal mostrara os logs das suas funcoes.

PASSO 4: ABRIR O EDITOR DE CODIGO (OPCIONAL, RECOMENDADO) Abra seu editor e o arquivo functions/index.js para trabalhar no codigo.

PASSO 5: PREPARAR AMBIENTE PARA TESTAR (Segundo Terminal ou Postman) Voce precisa de outro local para chamar as funcoes rodando no emulador.

* Opcao 1 (Segundo Terminal): Abra outro Terminal. Navegue de novo para a pasta do projeto (cd ...).
* Opcao 2 (Postman): Abra o Postman (Desktop ou Web com Agent configurado para localhost).

PASSO 6: TESTAR SUAS FUNCOES LOCALMENTE! Use o segundo terminal (com curl) ou Postman para enviar requisicoes para as URLs locais ( <http://127.0.0.1:5001/> ...) que o emulador te deu no Passo 3.

* Use o Metodo HTTP correto (GET, POST, PUT, DELETE).
* Use a URL local correta para cada funcao (ex: .../crm-com-api/us-central1/nomeDaFuncao).
* Envie os dados necessarios no corpo (para POST/PUT, usando -d "{"campo":"valor"}" no curl para Windows) ou na URL (para GET/PUT/DELETE, usando ?param=valor).

Exemplos de Comandos curl (no Segundo Terminal):

Adicionar Cliente (adicionarPrimeiroCliente - POST): curl -X POST -H "Content-Type: application/json" -d "{"nomeCompleto": "Novo Cliente Teste", "cpf": "111.222.333-44", "telefone": "+5531987654321", "telefoneWhatsapp": "+5531987654321", "enderecoCompleto": "Rua Nova, 789", "beneficioMatricula": "GHI789KLM", "bancoEmprestimo": "Banco Y", "valorEmprestimo": 15000, "valorParcela": 400, "prazoContratado": 48, "dataContratacao": "2024-05-20"}" <http://127.0.0.1:5001/crm-com-api/us-central1/adicionarPrimeiroCliente>

Listar Clientes (listarClientes - GET): curl <http://127.0.0.1:5001/crm-com-api/us-central1/listarClientes>

Obter Cliente por ID (obterCliente - GET): curl <http://127.0.0.1:5001/crm-com-api/us-central1/obterCliente?id=COLE_O_ID_DO_CLIENTE_AQUI>

Mover Cliente no Funil (moverClienteNoFunil - POST): curl -X POST -H "Content-Type: application/json" -d "{"clienteId": "COLE\_O\_ID\_DO\_CLIENTE\_AQUI", "novaEtapa": "em\_negociacao"}" <http://127.0.0.1:5001/crm-com-api/us-central1/moverClienteNoFunil>

Obter Metricas (obterMetricasDashboard - GET): curl <http://127.0.0.1:5001/crm-com-api/us-central1/obterMetricasDashboard>

(Adapte para as outras funcoes: atualizarCliente (PUT), excluirCliente (DELETE), criarAgendamento (POST), listarAgendamentosCliente (GET), adicionarDadosEmprestimo (POST), listarClientesComAlertaRecorrencia (GET), listarClientesEmProspeccao (GET), importarClientes (POST - espera array JSON)).

PASSO 7: MONITORAR LOGS E RESPOSTAS

* No PRIMEIRO Terminal (emuladores): Veja os logs das funcoes (linhas com "functions:").
* No SEGUNDO Terminal (curl/Postman): Veja a resposta HTTP da funcao (JSON ou mensagem).

PASSO 8: DESENVOLVER E CONTINUAR TESTANDO

* Faca alteracoes no functions/index.js (Passo 4).
* Salve o arquivo. O emulador recarrega automaticamente.
* Rode os testes relevantes novamente no Segundo Terminal/Postman (Passo 6).
* Observe logs/respostas (Passo 7). Repita.

AO FINAL DO DIA (PARAR EMULADORES):

* No PRIMEIRO Terminal (emuladores rodando), pressione Ctrl + C (ou Cmd + C no macOS) para parar os emuladores. Confirme se perguntar.

**Como usar este texto:**

1. **Copie**todo o texto acima, desde "CHECKLIST:" até o final.
2. **Abra um editor de texto simples**no seu computador (Pode ser o Bloco de Notas no Windows, TextEdit no macOS, ou qualquer outro editor simples).
3. **Cole**o texto copiado no editor.
4. **Salve**o arquivo.
   * Ao salvar, escolha um nome, por exemplo: checklist\_firebase\_functions.txt .
   * Verifique se o editor está salvando como "Texto Puro" ou com a extensão .txt . Evite salvar com extensões de documentos (como .doc ou .rtf ) que adicionam formatação.
5. Agora você tem um arquivo de texto com o checklist para consultar rapidamente sempre que for trabalhar nas suas Cloud Functions localmente!

Espero que este checklist simplificado te ajude no seu dia a dia! Você está fazendo um progresso excelente! 💪