Arquivo de Configuração MinhasContas

Essa aplicação MinhasContas tem objeto de disponibilizar as seguintes informações:

exibir o log de movimentações de forma ordenada;

* informar o total de gastos por categoria;
* informar qual categoria cliente gastou mais;
* informar qual foi o mês que cliente mais gastou;
* quanto de dinheiro o cliente gastou;
* quanto de dinheiro o cliente recebeu;
* saldo total de movimentações do cliente.

Para a execução com sucesso da aplicação, é necessario seguir os passos abaixo:

1 – Criar uma pasta no diretório C, criar a pasta com o nome workspaceigor

2 – Abir o GitBash na pasta criada, fazer o clone do diretorio ..... na pasta criada.

3- A aplicação foi desenvolvida em Java 8, é necessário verificar as variaveis de ambiente para obter sucesso nos proximos passos;

4 – Abrir o prompt do MSDOS, digitar cd C:\workspaceigor\myMovements

5 – Digitar no prompt compile.bat, executar o arquivo. O projeto java será compilado, gerado classes e o arquivo executavel .jar

6 – Digiratar no prompt e executar start.bar.

Segue configurações Importantes:

ENDPOINT\_RECEBIMENTO = "https://my-json-server.typicode.com/cairano/backend-test/recebimentos"; Constante do Endpoint de recebimentos

ENDPOINT\_PAGAMENTO = "https://my-json-server.typicode.com/cairano/backend-test/pagamentos"; Constante do Endpoint de pagamentoss

ARQUIVO = "C://workspaceaccounttest//myMovements//dados.log"; Constante com o path com o arquivo log.

* Como delimitador foi utilizado o caractere “;”

Aplicação MinhasContas segue a arquitetura:

View Swing, Console

É possível utilizar o programa atrás de uma interfaca gráfica, ou por linha de comando no MSDOS.

Para interface gráfica foi utilizado swing awt.

Controller

Camada controller desaclopado da View para poder utilizar outros view.

No caso da Aplicação MinhasContas, foi reutilizado os metods da Controller na interface gráfica e no console.

A controller invoca os metodos da Facade, que contem toda a lógica.

Facade

Camada Facade para chamar WsRest e classe que le arquivos log.  
Foi desenvolvida uma classe Facade, no qual essa classe invoca as classes Service, e a classe LancamentoFile, que é responsavel pela leitura do arquivo .log.

A classe Facade se utiliza de duas listas para gerar as informações de saida.

Uma lista do tipo RecebimentoData e PagamentoData, essas listas são alimentadas pelas classes Service e File, ou seja, informações da ApiRest e do Arquivo Log.

Segue os metodos mais relevantes da Classe Facade

carregarDadosArquivo()

Carregar as listas do arquivo log, e faz divisão e inseri os dados de acordo com Pagamento e recebimento.

mergeLancamentos()

Mescla informações; convertendo a lista de objetos Pagamento e Recebimento para MovimentacaoDTO, ordenando por Data.

Metodo chamado no metodo load, para preencher List<MovimentacaoDTO> movimentacao

todosLancamentos()

Exibir o log de movimentações de forma ordenada;

Informações obtida na lista mergeLancamentos() que foi preenchida no metodo load

maxCategoria()

Retorna um HashMap, com a categoria que mais teve gastos.

gastosPorCategoria()

Retorna um HashMap<String,Double>, com as categorias e o total de gastos das categorias.

maxMes()

Retorna HashMap com o nome do Mês e gasto do mês, o mês que teve mais gasto.

gastosPorMes()

Gastos por Mês, lista consolidade os meses e seus gastos.

Utilizado para informar qual foi o mês que cliente mais gastou;

totalGasto()

Retorna um Double com o total de gasto do cliente.

totalRecebido()

Retorna um Double com o total recebido.

saldoTotal()

Retorna um double, que é o resultado do TotalRecebido – TotalGasto.

configList()

Retorna um lista de configurações.

Service

Classes Client do Ws rest.

Classes que invocam a API Rest com informações de Recebimento e Pagamentos.

File

Classe que le arquivo .log

Data

Objeto que representa o response

Dto

Objeto para transferir dados da Facade até a View passando pela controller

Util

Classes para conversoes e padronizar dados

Config

Classe que armazena os dados de configurações:

ENDPOINT\_RECEBIMENTO = "https://my-json-server.typicode.com/cairano/backend-test/recebimentos";

Constante do Endpoint de recebimentos

ENDPOINT\_PAGAMENTO = "https://my-json-server.typicode.com/cairano/backend-test/pagamentos";

Constante do Endpoint de pagamentos

ARQUIVO = "C://workspaceaccounttest//myMovements//dados.log";

Constante com o path com o arquivo log.

* Como delimitador foi utilizado o caractere “;”