

Documentação da Biblioteca simplesDB

Eduardo Garcia Mendonça
Hector Fernando Tarasiuk
Jair Eugenio Ferreira
João Pedro Beltrão Cortes

Índice

1 mas afinal, o que o simplesDB vai fazer?	3
2 Das funcionalidades de simplesDB	3
2.1 definir um diretório para o armazenamento de currículos e fotos	3
2.2 criar um currículo a partir dos dados fornecidos	3
2.3 salvar um currículo no banco e dados como um JSON	4
2.4 carrega um currículo do banco de dados armazenado como JSON	4
2.5 Verifica se um currículo existe no banco de dados (por CPF ou por ID)	4
2.6 Editar os dados de um currículo já existente	4
2.7 Deletar um currículo já existente	4
2.8 Verifica se o CPF e senha fornecidos são válidos	5
3 Documentação de simplesDB	5
4 Compilando simplesDB	6
5 Documentação da classe sdb	6
5.1 Atributos	7
5.1.1 private static String DIR_DADOS	7
5.1.2 private static String DIR_FOTOS	7
5.2 Métodos	7
5.2.1 public static String getDIR_DADOS()	7
5.2.2 public static boolean setDados(String dir)	7
5.2.3 public static boolean setFotos(String dir)	7
5.2.4 public static boolean noDiscoCPF(String cpf)	7

5.2.5	public static boolean noDiscoID(String id)	7
5.2.6	public static Curriculo carregaCurriculoId(String id) . .	7
5.2.7	public static Curriculo carregaCurriculoCpf(String cpf)	8
5.2.8	public static Curriculo carregaCurriculoJson(String json)	8
5.2.9	public static Enviando curriculoPraEnviar(Curriculo cur)	8
5.2.10	public static Curriculo recebendoPraCurriculo(Recebendo en)	8
5.2.11	public static boolean novoUsuario(Recebendo dados) . .	8
5.2.12	public static boolean salvaCurriculo(Curriculo cur) . .	8
5.2.13	public static boolean autenticaUsuario(String cpf, String senha)	8
5.2.14	public static boolean delUsuario(String cpf, String senha)	9
5.2.15	public static boolean editaUsuario(Recebendo novos-dados)	9

6 Documentação da classe aux

6.1	Métodos	9
6.1.1	protected static String calculaIdUsuario(String cpf) . .	9
6.1.2	protected static int calculaHashSenha(String senha) . .	9
6.1.3	protected static String nomeArquivo(String ID)	9
6.1.4	protected static String caminhoArquivo(String nomeArquivo)	9
6.1.5	protected static String curriculoPraJson(Curriculo entraida)	9
6.1.6	protected static Curriculo jsonPraCurriculo(String json)	10
6.1.7	protected static Curriculo arquivoPraCurriculo(FileReader arquivo)	10

7 Documentação da classe Curriculo

7.1	Atributos	10
7.1.1	private final String cpf	10
7.1.2	private final String id_usuario	10
7.1.3	private final String nome_arquivo	10
7.1.4	private final int hash_senha	10
7.1.5	private String nome	10
7.1.6	private String telefone	11
7.1.7	private String email	11
7.1.8	private String endereco	11
7.1.9	private String habilidades	11

7.1.10	private String formacao	11
7.1.11	private String experiencia	11
7.1.12	private String descricao	11
7.2	Métodos	11
8	Documentação do record Enviando	11
9	Documentação do record Recebendo	12

1 mas afinal, o que o simplesDB vai fazer?

- definir um diretório para o armazenamento de currículos e fotos.
- criar um currículo a partir dos dados fornecidos
- salvar um currículo no banco de dados como um JSON
- carrega um currículo do banco de dados armazenado como JSON
- Verifica se um currículo existe no banco de dados (por CPF ou por ID)
- Editar os dados de um currículo já existente
- Deletar um currículo já existente
- Verifica se o CPF e senha fornecidos são válidos

2 Das funcionalidades de simplesDB

2.1 definir um diretório para o armazenamento de currículos e fotos

as variáveis DIR_DADOS e DIR_FOTOS armazenam a localização dos diretórios de dados e fotos respectivamente.

Métodos: setFotos(String dir), setDados(String dir);

2.2 criar um currículo a partir dos dados fornecidos

um método que chama o construtor de classe da classe currículo

método: novoUsuario(<parâmetros>)

2.3 salvar um currículo no banco e dados como um JSON

simplesDB usa a biblioteca GSON do google pra serializar um objeto do tipo currículo como um JSON. GSON foi escolhido devido a ser mais fácil de utilizar do que a utiliza de JSON nativa do java.

Método: salvaCurriculo(Curriculo cur)

2.4 carrega um currículo do banco de dados armazenado como JSON

Isso pode ser realizado por meio do CPF ou pelo ID de usuário. Caso o currículo exista no banco de dados, a biblioteca GSON é utilizada para ler o currículo e inicializar um objeto currículo, que será retornado por uma função.

Métodos: carregaCurriculoCpf(String CPF), carregaCurriculoId(String id)

2.5 Verifica se um currículo existe no banco de dados (por CPF ou por ID)

as funções noDiscoCPF e noDiscoID são utilizadas para tal verificação, elas verificam isto por meio da classe file do java. Retornam true se existe, false caso contrário

Da pra usar as próprias funções carregaCurriculo. Se o currículo não existe elas retornam um null.

2.6 Editar os dados de um currículo já existente

Caso um currículo exista, pode-se chamar uma função para editar seus atributos, o CPF e a senha não são editáveis no entanto.

Método: editaUsuario (<parâmetros>)

2.7 Deletar um currículo já existente

Uma função que recebe o CPF e a senha, verifica se existe currículo associado ao CPF no banco de dados e então deleta ele caso a senha fornecida esteja correta.

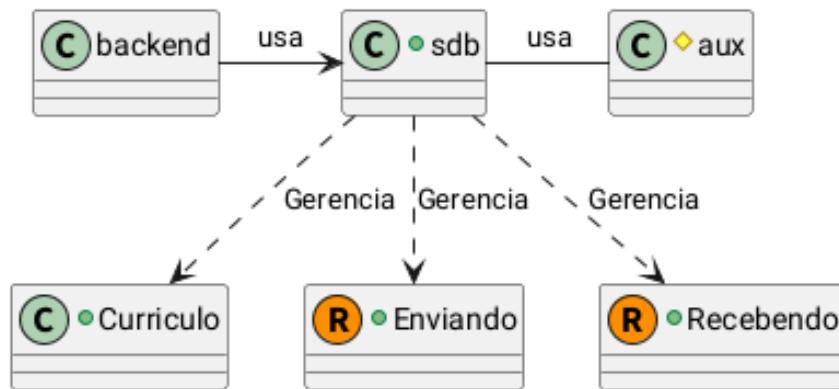
Método: delUsuario(String cpf, String senha)

2.8 Verifica se o CPF e senha fornecidos são válidos

Uma função que recebe CPF e Senha e retorna verdadeiro caso o currículo exista, falso caso o contrário

Método: autenticaUsuario(String cpf, String senha)

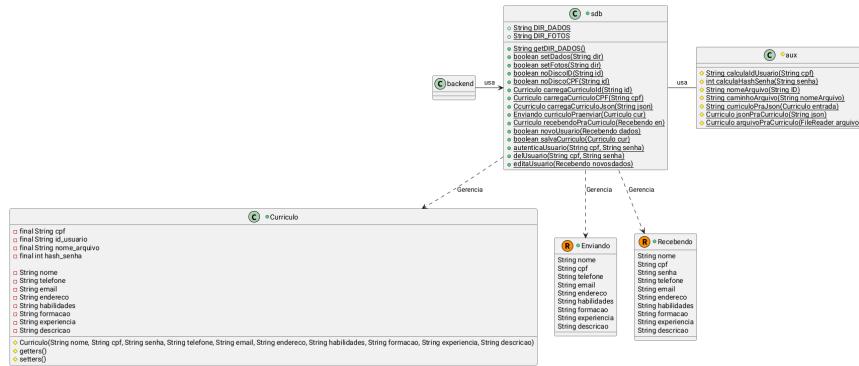
3 Documentação de simplesDB



simplesDB é uma biblioteca criada para gerenciar um banco de dados de currículos. A interface de interação com os currículos é a classe sdb, que provê uma série de métodos para gerenciar currículos no disco. A biblioteca segue o padrão de projetos facade: Uma interface simples (sdb) abstrai todos os detalhes de escrita e leitura de arquivos no disco, verificação de erros e da criação e conversão de objetos do usuário final de forma que este pode realizar todas as tarefas desejadas no banco de dados por meio de algumas poucas funções simples, sem conhecimento aprofundado dos detalhes de implementação do banco de dados.

Os currículos são armazenados no disco como arquivos JSON com os dados do usuário em seus devidos campos. A senha é armazenada como um hash por motivos de segurança, embora o tipo de hash utilizado não seja forte o suficiente para propósitos de segurança reais, sendo apenas uma demonstração do conceito. A busca por um arquivo no banco de dados é simplificada pelo uso de um hash calculado a partir do CPF, o arquivo é nomeado com este hash e então salvo no disco. Buscar por um currículo no banco de dados consiste em simplesmente calcular o hash a partir de um CPF fornecido e carregar o arquivo com o nome correspondente no disco. Tal arquitetura de banco de dados foi escolhida devido sua simplicidade e facilidade de implementação.

Abaixo o diagrama de classe com todos os atributos:



simplesDB é uma biblioteca complexa e é responsável pela lógica de funcionamento do backend.

4 Compilando simplesDB

simplesDB pode ser compilado com o seguinte comando em seu diretório:

```
mvn package
```

Que produzirá um .jar na pasta targets. Preferivelmente no entanto, se pode utilizar o script compilaemove.sh, que não só compila simplesDB como também move ele para seu devido lugar na estrutura do backend, onde poderá ser então instalado com o script instala_biblioteca.sh. O script executa o comando:

```
#!/bin/bash
mvn package && \
mv ./target/simpleDB-1.0-SNAPSHOT.jar ./target/simpleDB.jar && \
mv -f ./target/simpleDB.jar ../backend/src/main/resources/
```

5 Documentação da classe sdb

A classe `sdb` é a interface que possibilita a interação entre o programa e o banco de dados.

5.1 Atributos

5.1.1 `private static String DIR_DADOS`

O caminho para o diretório onde os arquivos JSON são armazenados.

5.1.2 `private static String DIR_FOTOS`

O caminho para o diretório onde armazenariam-se as fotos do usuário.

5.2 Métodos

5.2.1 `public static String getDIR_DADOS()`

Retorna o caminho para o diretório de dados.

5.2.2 `public static boolean setDados(String dir)`

Define o diretório de dados como sendo o provido no argumento. Retorna true se a operação foi um sucesso e false caso contrário.

5.2.3 `public static boolean setFotos(String dir)`

Define o diretório de fotos como sendo o provido no argumento. Retorna true se a operação foi um sucesso e false caso contrário.

5.2.4 `public static boolean noDiscoCPF(String cpf)`

Verifica se existe no disco um currículo associado ao CPF fornecido. Retorna true se sim, false se não.

5.2.5 `public static boolean noDiscoID(String id)`

Verifica se existe no disco um currículo associado ao ID de usuário fornecido. O ID de usuário é um hash calculado a partir do CPF. Retorna true se sim, false se não.

5.2.6 `public static Curriculo carregaCurriculoId(String id)`

Carrega um currículo do banco de dados com base no ID de usuário fornecido. Retorna um objeto Curriculo caso o currículo tenha sido encontrado, null caso contrário.

5.2.7 public static Curriculo carregaCurriculoCpf(String cpf)

Carrega um currículo do banco de dados com base o CPF fornecido. Faz uso de carregaCurriculoId no seu funcionamento interno. Retorna um objeto Curriculo caso o currículo tenha sido encontrado no disco, retorna null caso contrário.

5.2.8 public static Curriculo carregaCurriculoJson(String json)

Transforma uma String contendo um Json de currículo em um currículo diretamente. Retorna um objeto Curriculo se teve sucesso, null caso contrário.

5.2.9 public static Enviando curriculoPraEnviar(Curriculo cur)

Cria um record Enviando a partir de um objeto Curriculo. Este record pode ser então retornado diretamente como um objeto JSON pelo back end.

5.2.10 public static Curriculo recebendoPraCurriculo(Recebendo en)

Cria um novo objeto currículo a partir dos dados de um record Recebendo fornecidos.

5.2.11 public static boolean novoUsuario(Recebendo dados)

Cria um novo objeto currículo a partir do record Recebendo dados e então grava o objeto Curriculo no disco. retorna true se a operação foi um sucesso, false caso contrário.

5.2.12 public static boolean salvaCurriculo(Curriculo cur)

Grava um currículo no disco. Retorna true se a operação foi um sucesso, false caso contrário.

5.2.13 public static boolean autenticaUsuario(String cpf, String senha)

Verifica se existe um currículo no banco de dados associado ao CPF fornecido e caso exista, se a senha fornecida está correta. Retorna true caso ambas as cláusulas sejam verdadeiras, false caso contrário.

5.2.14 public static boolean delUsuario(String cpf, String senha)

Deleta um currículo do disco caso o CPF e senha fornecidos sejam válidos. Retorna true se a operação foi um sucesso, false caso contrário.

5.2.15 public static boolean editaUsuario(Recebendo novosdados)

Muda os dados de um currículo com base nos dados no record Recebendo novosdados. Só realiza a mudança caso o record de entrada passe no processo de autenticação. Retorna true se a operação foi um sucesso, false caso contrário.

6 Documentação da classe aux

A classe aux define uma série de funções que são utilizadas pelas classes sdb e Curriculo.

6.1 Métodos

6.1.1 protected static String calculaIdUsuario(String cpf)

Calcula o ID do usuário a partir de seu CPF.

6.1.2 protected static int calculaHashSenha(String senha)

Calcula o hash da senha do usuário a partir do argumento fornecido

6.1.3 protected static String nomeArquivo(String ID)

Recebe um ID de usuário e adiciona ".json" no final para nomear o arquivo.

6.1.4 protected static String caminhoArquivo(String nomeArquivo)

Adiciona o nome do arquivo no caminho do diretório de dados.

6.1.5 protected static String curriculoPraJson(Curriculo entrada)

Usa a biblioteca Gson do google para transformar um objeto currículo em um JSON.

6.1.6 protected static Curriculo jsonPraCurriculo(String json)

Usa a biblioteca Gson do google para transformar um objeto JSON em um currículo.

6.1.7 protected static Curriculo arquivoPraCurriculo(FileReader arquivo)

Usa a biblioteca Gson do google para transformar os dados em um arquivo do disco diretamente em um currículo.

7 Documentação da classe Curriculo

A classe Curriculo define as variáveis e métodos de um currículo. Objetos Curriculo são salvos no banco de dados como arquivos JSON. os atributos referentes ao cpf, id_usuario, nome_arquivo e hash_senha são imutáveis.

7.1 Atributos

7.1.1 private final String cpf

O CPF de um usuário.

7.1.2 private final String id_usuario

O ID de um usuário, calculado como um hash a partir do CPF.

7.1.3 private final String nome_arquivo

O nome do arquivo salvo no disco, consiste no id de usuário com a terminação ".json".

7.1.4 private final int hash_senha

A senha é armazenada como um hash da String fornecida pro construtor de objeto.

7.1.5 private String nome

O nome do usuário.

7.1.6 private String telefone

O telefone do usuário.

7.1.7 private String email

O email do usuário.

7.1.8 private String endereco

O endereço do usuário

7.1.9 private String habilidades

As habilidades do usuário.

7.1.10 private String formacao

A formação acadêmica do usuário.

7.1.11 private String experiencia

A experiência profissional do usuário.

7.1.12 private String descricao

Uma descrição fornecida pelo usuário.

7.2 Métodos

O construtor e os getters e setters fazem exatamente o que você espera deles.

8 Documentação do record Enviando

Define as variáveis com os dados relevantes de um currículo que o backend enviaria para o frontend. A única diferença entre este record e o record Recebendo é a omissão da variável String senha.

9 Documentação do record Recebendo

Define as variáveis com os dados relevantes de um currículo que o frontend enviaria para o backend. A única diferença entre este record e o record Enviando é a adição da variável String senha.