

2023/2024	Projet JAVA – Etape 04
BTS SIO	Auteur : Loric Worms / Stefen Rialland
1SIOB - SLAM	Date de rédaction : 16/04/2024

Compte-rendu étape 04

Teste de la classe « TestDaoService »

```

Output - JavaApplication_1Slam_Projet2_Sirh-Infoware (run)

run:

Test 1 : DaoService.getOneById
getConnexion : jdbc:mysql://localhost:3307/SIRH
Service d'id 2 trouv   :
Service{code=2, designation=Adminstration, email=administration@infoware.com, telephone=0169983210}

Test 2 : DaoService.getAll
Service{code=1, designation=Informatique, email=informatique@infoware.com, telephone=0169983212}
Service{code=2, designation=Adminstration, email=administration@infoware.com, telephone=0169983210}
Service{code=3, designation=Commercial, email=commercial@infoware.com, telephone=0169983215}
Service{code=4, designation=Comptable, email=comptable@infoware.com, telephone=0169983218}
4 services trouv  s

Connexion    la BDD ferm  e
BUILD SUCCESSFUL (total time: 0 seconds)

```

Cr  ation de la classe « DaoCategorie »

```

/**
 * Rechercher un enregistrement dans la table CATEGORIE d'apr  s son code (String)
 * et en faire un objet de type Service
 *
 * @param id code de la cat  gorie recherch  
 * @return objet de type Categorie si trouv   dans la BDD, null sinon
 * @throws SQLException
 */
public static Categorie getOneById(String id) throws SQLException, IOException {
    Categorie serviceTrouve = null;
    Connection cnx = ConnexionBDD.getConnexion();
    PreparedStatement pstmt = cnx.prepareStatement(sql:"SELECT * FROM Categorie WHERE Code = ?");
    pstmt.setString(parameterIndex: 1, x: id);
    ResultSet rs = pstmt.executeQuery();
    if (rs.next()) {
        serviceTrouve = new Categorie(
            code: rs.getString(columnLabel: "Code"),
            libelle: rs.getString(columnLabel: "Libelle"),
            salaireMini: rs.getDouble(columnLabel: "salaireMini"),
            caisseRetraite: rs.getString(columnLabel: "CaissedeRetraite"),
            prime: rs.getInt(columnLabel: "PrimeResultat")
        );
    }
    return serviceTrouve;
}

```

2023/2024	Projet JAVA – Etape 04
BTS SIO	Auteur : Loric Worms / Stefen Rialland
1SIOB - SLAM	Date de rédaction : 16/04/2024

```

/**
 * Extraire l'ensemble des enregistrements de la table CATEGORIE
 * @return liste d'objets de type Categorie
 * @throws SQLException
 */
public static ArrayList<Categorie> getAll() throws SQLException, IOException {
    ArrayList<Categorie> lesCatTrouves = new ArrayList<>();
    Connection cnx = ConnexionBDD.getConnexion();
    PreparedStatement pstmt = cnx.prepareStatement(sql: "SELECT * FROM Categorie");
    ResultSet rs = pstmt.executeQuery();
    while (rs.next()) {
        Categorie uneCategorie;
        uneCategorie = new Categorie(
            code: rs.getString(columnLabel: "Code"),
            libelle: rs.getString(columnLabel: "Libelle"),
            salaireMini: rs.getDouble(columnLabel: "salaireMini"),
            caisseRetraite: rs.getString(columnLabel: "CaissedeRetraite"),
            prime: rs.getInt(columnLabel: "PrimeResultat")
        );
        lesCatTrouves.add(e: uneCategorie);
    }
    return lesCatTrouves;
}

```

Création de la classe testDaoCategorie »

```

/**
 * @param args the command line arguments
 */
public static void main(String[] args) {
    // Test 1 getOneById
    System.out.println(x: "\n Test 1 : DaoCategorie.getOneById");
    try {
        String idRecherche = "C2";
        Categorie cat = DaoCategorie.getOneById(id: idRecherche);
        if(cat != null){
            System.out.println("Catégorie d'id "+idRecherche+" trouvé : \n"+cat.toString());
        }else{
            System.out.println("Catégorie d'id "+idRecherche+" non trouvé : \n");
        }
    } catch (SQLException ex) {
        System.out.println("TestDaoCategorie - échec getOneById : " + ex.getMessage());
    } catch (IOException ex) {
        System.out.println("TestDaoCategorie - échec getOneById : " + ex.getMessage());
    }
}

```

← 1^{er} test

```
// Test 2 getAll
System.out.println(x: "\n Test 2 : DaoService.getAll");
try {
    ArrayList<Categorie> lesCategories = DaoCategorie.getAll();
    for (Categorie cat : lesCategories) {
        System.out.println(x: cat.toString());
    }
    System.out.println(lesCategories.size()+" services trouvés");
} catch (SQLException ex) {
    System.out.println("TestDaoCategorie - échec getAll : " + ex.getMessage());
} catch (IOException ex) {
    System.out.println("TestDaoCategorie - échec getAll : " + ex.getMessage());
}
```

← 2ème test

fermeture de la
connexion →

```
// Fermeture de la connexion
try {
    ConnexionBDD.getConnexion().close();
    System.out.println(x: "\nConnexion à la BDD fermée");
} catch (SQLException ex) {
    System.out.println("TestDaoService - échec de la fermeture de la connexion : " + ex.getMessage());
} catch (IOException ex) {
    System.out.println("TestDaoService - échec de la fermeture de la connexion : " + ex.getMessage());
}
}
```

Teste de la classe TestDaoCategorie

```
run:

Test 1 : DaoCategorie.getOneById
getConnexion : jdbc:mysql://localhost:3307/SIRH
Catégorie d'id C2 trouvée :
Catégorie{code=C2, libelle=Cadre supérieur, salaire minimum=3500.0, caisse retraite=AGIRC, prime=1}

Test 2 : DaoService.getAll
Catégorie{code=C1, libelle=Cadre moyen, salaire minimum=2500.0, caisse retraite=AGIRC, prime=1}
Catégorie{code=C2, libelle=Cadre supérieur, salaire minimum=3500.0, caisse retraite=AGIRC, prime=1}
Catégorie{code=E1, libelle=Employé niveau 1, salaire minimum=1600.0, caisse retraite=ARRCO, prime=0}
Catégorie{code=E2, libelle=Employé niveau 2, salaire minimum=2100.0, caisse retraite=ARRCO, prime=0}
4 services trouvés

Connexion la BDD fermée
```

2023/2024	Projet JAVA – Etape 04
BTS SIO	Auteur : Loric Worms / Stefen Rialland
1SIOB - SLAM	Date de rédaction : 16/04/2024

Modification de la classe DaoSalarie

```
private static Salarie creerSalarie(ResultSet rs) throws SQLException, IOException {
    Salarie unSalarie = null;
    // Récupération du service du salarié
    Service unService = DaoService.getOneById(id: rs.getInt(columnLabel: "CodeServ"));
    Categorie uneCategorie = DaoCategorie.getOneById(id: rs.getString(columnLabel: "NumCat"));
    unSalarie = new Salarie(
        code: rs.getString(columnLabel: "Code"),
        nom: rs.getString(columnLabel: "Nom"),
        prenom: rs.getString(columnLabel: "Prenom"),
        (java.util.Date) rs.getDate(columnLabel: "DateNaiss"),
        (java.util.Date) rs.getDate(columnLabel: "DateEmbauche"),
        fonction: rs.getString(columnLabel: "Fonction"),
        tauxHoraire: rs.getDouble(columnLabel: "TauxHoraire"),
        situationFamilliale: rs.getString(columnLabel: "situationFamilliale"),
        nbrEnfants: rs.getInt(columnLabel: "NbrEnfants"),
        serv: unService,
        categorie: uneCategorie
    );
    return unSalarie;
}
```

← Prise en compte de la catégorie du salarié.

Création de la classe testDaoSalarie

```
// Test 1 getOneById
System.out.println(x: "\n Test 1 : DaoCategorie.getOneById");
try {
    String idRecherche = "S02";
    Salarie sal = DaoSalarie.getOneById(id: idRecherche);
    if(sal != null){
        System.out.println("Salarié d'id "+idRecherche+" trouvé : \n"+sal.toString());
    }else{
        System.out.println("Salarié d'id "+idRecherche+" non trouvé : \n");
    }
} catch (SQLException ex) {
    System.out.println("TestDaoCategorie - échec getOneById : " + ex.getMessage());
} catch (IOException ex) {
    System.out.println("TestDaoCategorie - échec getOneById : " + ex.getMessage());
}
```

← 1^{er} test

```
// Test 2 getAll
System.out.println(x: "\n Test 2 : DaoService.getAll");
try {
    ArrayList<Salarie> lesSalaries = DaoSalarie.getAll();
    for (Salarie sal : lesSalaries) {
        System.out.println(x: sal.toString());
    }
    System.out.println(lesSalaries.size()+" services trouvés");
} catch (SQLException ex) {
    System.out.println("TestDaoSalarie - échec getAll : " + ex.getMessage());
} catch (IOException ex) {
    System.out.println("TestDaosalarie - échec getAll : " + ex.getMessage());
}
```

← 2ème test

