

## TModeler ORM - Documentation Technique

TModeler est une bibliothèque ORM (Object-Relational Mapping) C++ orientée objet, compatible Linux et Windows. Elle permet la gestion complète des entités métiers avec des opérations CRUD, des filtres complexes, une gestion des relations (via `ModelField`, `ListField`), l'observation d'événements, et la définition de structures personnalisées.

### 1. Structures Simples

Les modèles simples sont des classes dérivées de `TModel` et contiennent des champs classiques.

```
struct Person : public TModel {
    ModelField<int> id;
    ModelField<std::string> name;
    ModelField<std::string> dob;
    ModelField<JsonField> meta;
    ModelField<float> ratio;
    ModelField<bool> empty;
};
```

Les opérations CRUD sont supportées :

```
Person person;
person.id = 100;
person.name = "Olivier TAMBO";
person.save();      // Insert or update
person.del();       // Delete
```

### 2. Structures Avancées : Relations

Les relations entre modèles sont exprimées avec `ModelField` (relation 1-1) et `ListField` (relation 1-N).

```
struct Client : public TModel {
    ModelField<std::string> name;
    ListField<Client*> friends;
};

struct Product : public TModel {
    ModelField<std::string> name;
};

struct Cmd : public TModel {
```

```

    ModelField<Client> client;
    ModelField<Product> product;
};

```

Exemple d'insertion avec relations :

```

Client c11, c12;
c11.name = "Olivier";
c12.name = "Michel";
c11.friends = { &c12 };
c11.save();

```

### 3. Modèles Personnalisés

On peut hériter d'un modèle existant ou ajouter des champs personnalisés.

```

struct CustomPerson : public Person {
    ModelField<std::string> role;
};

struct ExtendedPerson : public CustomPerson {
    ModelField<std::string> status;
};

```

### 4. Observateurs Multithread

Les gestionnaires de modèles ('Tms<T>') permettent d'écouter les modifications sur les objets, même dans un contexte multithread.

```

Tms<Person> tms;
tms.onCreate([]() {
    Log::d("Création détectée");
});
tms.onUpdate([]() {
    Log::d("Mise à jour détectée");
});

```

### 5. Filtres

Trois catégories :

- **Simple** (équivalent WHERE)
- **Jointure**
- **Agrégation**

Simple (WHERE) :

```
auto f1 = Person::tms.with(p).filter((p.dob >= "2000-11-04") && (p.name != "Michel TAMBO"));
```

Traduction : WHERE dob >= '2000-11-04' AND name != 'Michel TAMBO'

Jointure :

```
auto join = Client::tms.with<Cmd>(cl, cm).filter(cm.client == cl);
```

Traduction : FROM Client INNER JOIN Cmd ON Cmd.client\_id = Client\_id

Agrégation :

```
auto g = Client::tms.with(cl).group(cl.name).filter(cl._id.count() == 2);
```

Traduction : GROUP BY name HAVING COUNT(\_id) = 2

## 6. Installation

À compléter : prérequis, dépendances, compilation sous Linux/Windows.