AppGene Help

# What is AppGene?

AppGene is an application that helps developers or users to build their applications quickly.

# Getting Start

AppGene can build a database application; the minimum inputs are database entities.

AppGene is a data model driving system

## Prerequisites

## First Entity

## Run It!

# For Beginners

## Entities

## Connection String

## Basic Entity Attributes

# For Middle Level Users

## Change Product Basic Information

## Entity Model/Business Entity

## Conventions

## Entity Configurations

## Key Interfaces

## Key Attributes

# For Advance Users

## User Interface Styles

# Table Categories in AppGene

* Simple Table
  + 1 pk and 0..\* index
  + pk has 1 columns
  + Id + Code + Name
  + Id + Name
  + Id + \* (PK, editable)
* Simple Table with reference columns
  + reference key s only include 1 column
* Table: PK has multiple columns
  + some PK columns are reference another table.
* Tree Table
  + PK has multiple columns
* One record table
* PK is editable table
* History Table

Same as the original table, but has an additional column "HistoryId"

* CheckOut Table

Same as the original table, but has some checkout information, checkout user, checkout time.

We focus on 1 primary key + 1 unique key (optional) only.

* Has Primary Key
* Does Display Id
* Is Tree Table
* Special columns
  + timestamp
  + Is Deleted
    - Case 1: The column's data type is same as PK, the value is 0 for non-deleted records, and is same as Id for deleted records.
    - Case 2: The code column is indexed but not unique
    - Case 3: No code columns.
  + Create User
  + Create Time
  + Modify User
  + Modify Time
  + Modify Times
  + Disabled

The column indicates if the data can be referenced now.

* + Order

# Design Patterns in AppGene

# Terminology

* AppGene

AppGene is this product.

* Entity

Entity is mainly a data model concept, and an entity probably be shared in all layers as data transition.

* Data Entity

Data Entity is entity.

* Entity Model / Business Model / Business Entity

Model is a concept on the presentation layer, which can help the presentation objects to understand the function better

There are 2 situations your entity and model are same one.

Situation 1: Your business service uses instances of model.

Situation2: Your business service uses instances of entity, and the entity class provides enough information to get the presentation layer to use it.

* Model
* MVVM
* View
* View Model
* View Controller
* View Constructor

# Licenses