dasz.at
Benutzbare Technologie

Kistl Guide

The Hitchhiker's guide to the Kistl galaxy

Contents

Chapter 1

Introduction

Kistl is a programming framework to provide the complete process from defining data structures, designing data access and transfer objects, designing servers and GUIs and the necessary parts to make everything work together.

Chapter 2

Programming

This chapter describes the various ways and pieces the Kistl system is programmed and customized.

2.1 Objects

2.2 Modules

2.3 Enhancing Kistl's inner workings

2.3.1 Database Providers

2.4 Graphical User Interface

Like other subsystems, the GUI core is designed to be platform independent. Therefore only the "outermost" shell contains toolkit specific code.

2.4.1 Architecture

The GUI is modeled after the Model-View-ViewModel architecture. The *Model* represents the underlying data structures and business logic. It is provided by the generated classes from the actual datamodel. *View Models* provide display specific functionality like formatting, transient state holding and implementing the user's possible actions. They always inherit from `Kistl.Client.Presentables.PresentableModel`. Common implementations reside in the `Kistl.Client.Presentables` namespace. Finally, *Views* (editors and displays) are the actual components taking care of showing content to the user and converting the users keypresses and clicks into calls on the view models interface. Views are toolkit[1] specific and reside in the toolkit's respective assembly.

[1]Toolkits are GUI libraries like GTK# or Windows Forms.

List of Figures