Table of Contents

The JVx Enterprise Application Framework is a powerful tool for the creation of database applications. However, attention is always paid to simplicity and modularity. Therefore, JVx consists of several packages, which can be used separately or in combination.

The following packages are components of JVx:

- User Interface (UI)
  - Generic User Interface (Generic UI)
  - Swing
  - Other UI Platforms
- Generic Model
- Remote
- Server
- Persist

User Interface (UI) (jvx.rad.ui)

The JVx UI defines which possibilities the user interface must offer in order to build an enterprise application. These are based on minimal requirements so as not to create any extra ballast. Basically, the UI consists of simple interfaces with which the necessary attributes are defined in a technology-independent manner. The goal was to use the one application with various GUI technologies without even having to change a line of code.

Generic User Interface (Generic UI) (jvx.rad.genui)

The generic UI is a technology-independent implementation of the UI. It consists of concrete classes which act as wrappers for the preferred technology. Thanks to the generic UI, it is possible to reach a high level of comfort when developing because it is easily possible to derive one's own classes and integrate them into applications. Only with interfaces is this not possible.

Swing (com.sibvisions.ui.swing)

The Swing package contains the technology-independent UI implementation for Swing. On the one hand, standard Swing components were used, on the other, extended Swing components were implemented. The extended components offer extra functionality and can also be used separately with other Swing-based applications.

Dynamic Model (jvx.rad.model, com.sibvisions.rad.model)

The dynamic model defines an active model for usage in the UI. This is, in principle, several interfaces with which the attributes of the model are described. All UI controls use these interfaces to display and manipulate data. The model has already been implemented for the use of data in the main memory and for the access to databases. The actual database access takes place thanks to the Remote package.

Remote (jvx.rad.remote, com.sibvisions.rad.remote)

In the remote package, the protocol-independent communication between client and server is, on the one hand, defined, and on the other hand, its use with Master- and SubConnection is implemented. Each communication between client and server is executed with this package. The communication via HTTP(S) has already been fully implemented.

Server (jvx.rad.server, com.sibvisions.rad.server)

The Server package defines the attributes of a server that is fully supported by Jvx and the communication package (Remote). This definition covers the communication possibilities, a Master-/SubSession administration, and the lifecycle management. The communication via HTTP(S) has already been fully implemented.

Persistence (jvx.rad.persist, com.sibvisions.rad.persist)

Interfaces define how data from various sources (databases, XML, etc.) are to be accessed. The definition is based on a record-oriented data storage. An implementation already exists for databases.

From:
http://doc.sibvisions.com/ - **Documentation**

Permanent link:
http://doc.sibvisions.com/jvx/package_overview

Last update: **2024/11/18 10:24**