

# Table of Contents

Das **JVx** Framework ist ein Full Stack Applikations Framework für die Erstellung von Datenbank Applikationen. Bei der Entwicklung wird jedoch stets auf Einfachheit und Modularität geachtet. Aus diesem Grund besteht JVx aus mehreren Paketen die voneinander getrennt oder kombiniert, verwendet werden können.

Folgende Pakete sind Bestandteil von JVx:

- **User Interface (UI)**
  - **Generic User Interface (Generic UI)**
  - **Swing**
  - andere UI Plattformen
- **Generic Model**
- **Remote**
- **Server**
- **Persist**

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

Durch das JVx UI wird definiert welche Möglichkeiten ein User Interface bieten muss um Enterprise Applikationen zu erstellen. Dabei wurde von den minimalen Anforderungen ausgegangen, um bereits von Beginn an keinen Ballast entstehen zu lassen. Das UI besteht im Grunde aus simplen Interfaces mit denen technologieunabhängig die notwendigen Eigenschaften definiert werden. Das Ziel dabei war, ein und dieselbe Applikation, mit unterschiedlichen GUI Technologien zu verwenden ohne auch nur eine Codezeile verändern zu müssen.

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

Das generische UI ist eine Technologie unabhängige Implementierung des UI. Es handelt sich um konkrete Klassen, die als Wrapper für die gewünschte Technologie fungieren. Durch das generische UI wird es ermöglicht den bestmöglichen Komfort beim Entwickeln zu erreichen, weil es ohne Probleme möglich ist eigene Klassen abzuleiten und in Applikationen zu integrieren. Mit reinen Interfaces wäre das nicht möglich.

### **Swing (com.sibvisions.ui.swing)**

Das Swing Paket enthält die Technologie abhängige UI Implementierung für Swing. Es wurden einerseits Standard Swing Komponenten verwendet und andererseits erweiterte Swing Komponenten implementiert. Die erweiterten Komponenten bieten zusätzliche Funktionalität und können auch separat mit anderen Swing basierenden Applikationen verwendet werden.

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

Das generische Model definiert ein aktives Model für die Verwendung im UI. Im Prinzip handelt es sich hierbei um Interfaces mit denen die Eigenschaften des Models beschrieben werden. Sämtliche UI

Controls verwenden diese Interfaces um Daten darzustellen und zu manipulieren. Das Model wurde bereits für die Verwendung von Daten im Hauptspeicher und für den Zugriff auf Datenbanken implementiert. Der Datenbankzugriff selbst wird über das Remote Paket abgewickelt.

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

Im Remote Paket wird einerseits die protokollunabhängige Kommunikation zwischen Client und Server definiert und andererseits die Verwendung mit Master- und SubConnection implementiert. Jegliche Kommunikation zwischen Client und Server wird mit diesem Paket durchgeführt. Die Kommunikation über HTTP(S) wurde bereits vollständig implementiert.

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

Das Server Paket definiert die Eigenschaften eines Servers, der von JVx und dem Kommunikations Paket (Remote) vollständig unterstützt wird. Diese Definition umfasst die Kommunikationsmöglichkeiten, eine Master-/SubSession Verwaltung und das Life-Cycle Management. Die Kommunikation über HTTP(S) wurde bereits vollständig implementiert.

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

Durch Interfaces wird definiert wie auf Daten von unterschiedlichsten Quellen (Datenbanken, XML, usw.) zugegriffen werden kann. Die Definition geht von einer Tabellen orientierten Datenhaltung aus. Für Datenbanken ist bereits eine Implementierung vorhanden.

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

Permanent link:  
[http://doc.sibvisions.com/de/jvx/package\\_overview](http://doc.sibvisions.com/de/jvx/package_overview)



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