Volle Kontrolle über den gesamten Entwicklungsprozess auch für Webanwendungen

Holger Klemt, November 2019

Was ist aus unserer Sicht das größte Problem bei Webanwendungen, die von typischen Delphi-Entwicklern erstellt wurden?

Viele Delphi-Entwickler konzentrieren sich auf ihre aktuelle Plattform und kennen die Details der VCL im Allgemeinen und die Komponenten, die sie seit Jahren häufig verwenden. Natürlich gehört auch die SQL-Sprache der Datenbank zu ihren Kenntnissen.

Alle Entwickler, mit denen wir gesprochen haben, haben erkannt, dass sie bestimmte Lösungen benötigen, die auf einem mobilen Gerät verwendet werden können. Aber ihre tägliche Arbeit ermöglicht es ihnen nicht, so viel Zeit auf einer völlig neuen Plattform zu investieren.

Einige von ihnen begannen mit Lösungen, die auf FireMonkey-iOS- oder Android-Technologien von Delphi basieren, aber selbst einfache Anwendungen, die auf diese Weise erstellt wurden, erfordern viel zusätzliches Know-how, insbesondere für den iOS-Markt, bevor die erste echte Anwendung von Kunden benutzt werden kann.

Einer der größten Nachteile einer solchen nativen Android- oder iOS-App sehen wir auch in der sehr eingeschränkten Möglichkeit, unmittelbar ein Update für die App bereitzustellen, nachdem ein Fehler erkannt und behoben wurde.
Es kann zwei Wochen oder länger dauern, bis die Bugfix-Version von Ihrer IDE hochgeladen wurde und dem Kunden auf dem Gerät wieder zur Verfügung steht.

Benutzer müssen möglicherweise immer noch mit älteren Versionen arbeiten, obwohl im App Store bereits ein Update verfügbar ist.

Wenn die alte Version noch funktioniert, haben die Anwender Glück, aber wenn ein Fehler Daten zerstört oder die Nutzung der App blockiert, sind Sie in großen Schwierigkeiten und müssen schnell Entschuldigungen für verärgerte Kunden finden, die zu recht die bereits bezahlte App nutzen möchten.

Ganz zu schweigen von den Problemen, die durch neue Einschränkungen von Google oder Apple verursacht werden, beispielsweise die aktuelle Anforderung für 64-Bit-Android oder das von Apple blockierte Electron-Framework.

Obwohl die 64-Bit-Android-Anforderung bereits seit langer Zeit bekannt ist, verfügt Delphi über keine entsprechende Arbeitsumgebung, auch wenn sie so freundlich sind, Ihnen Zugriff auf eine Beta-Version zu bieten.

Der gesamte von Ihnen erstellte Quellcode ist völlig wertlos, wenn eines der erforderlichen Middleware-Systeme zwischen Ihrem Quellcode und der laufenden App auf dem Mobilgerät Ihres Kunden nicht verfügbar ist.

Eine der Lösungen, die derzeit als Add-On für reine JavaScript-basierte Anwendungen direkt aus Ihrem Pascal-Quellcode auf dem Markt sind, ist pas2js, das auch von TMS WEB Core verwendet wird, wenn es in Delphi oder in die Lazarus-IDE integriert ist.

Im Vergleich zu Lazarus mit pas2js, das zu 100% Open Source ist, bietet TMS WEB Core einige Teile jedoch nur als Closed Source-Module an, die jeder Entwickler lizenzieren muss.

Andere Lösungen wie uniGUI sind mit pas2js nicht vergleichbar, da diese ohne aktive Internet-verbindung nicht auf einem mobilen Gerät ausgeführt werden können.

Jede pas2js-Anwendung wird vollständig in natives JavaScript kompiliert. Sie können Daten auf einfache Weise in einem sogenannten lokalen Speichersystem speichern und lesen. Dieses einfache System ist nicht mit einem voll funktionsfähigen SQL-Server vergleichbar.

Jedoch lassen sich mit dieser Methode einfache Aufgaben, wie beispielsweise Arbeitszeit-dokumentationen auf mobilen Geräten, mobile Zugriffe auf Produktdaten, Kundendaten oder alles anders, was Sie auf dem System speichern möchten, auch wenn es keinen Online-Zugriff hat, einfach umsetzen.

Aber wie können Sie die Daten von Ihrem Büromanagement-System auf die mobilen Geräte übertragen und auf den mobilen Geräten vorgenommene Eingaben an das Büromanagement-System zurücksenden?

Wenn Sie sich auf dem Markt die verschiedenen Funktionsweisen ansehen, werden Sie Dutzende verschiedener Möglichkeiten finden, bei denen REST-API die Lösung für fast alles zu sein scheint.

Aber was braucht eine typische REST-Server-basierte Anwendung?

a) Eine allgemeine Client-Anwendung, mit der Daten angezeigt und bearbeitet werden können. Diese Anwendung sollte weiterhin die Möglichkeit bieten, Teile der Daten im lokalen Speicher zu speichern und zu lesen.

b) Ein Teil Ihrer Client-App, der weiß, welche APIs auf dem REST-Server implementiert sind und wie sie aufgerufen werden müssen, welche Daten für welche Anforderung kommen und wo sie in der in Teil a) genannten GUI-App verwendet werden müssen.

c) Ein REST-Server, der genau das widerspiegelt, was Sie in Teil b) verwenden und benötigen.

d) Auf jeden Fall eine Datenbank, in der die Daten für den REST-Server gespeichert sind.

e) Wenn Ihre Backoffice-Software viel komplizierter ist als Ihre mobilen Apps, benötigen Sie auch einige Import-/Exportverfahren, die möglicherweise bereits auf dem REST-Server oder direkt in der Datenbank implementiert sind.

Die Implementierung von Grundfunktionen direkt in der Datenbank ist beispielsweise dann sinnvoll, wenn Sie allen verbundenen Kunden Ihre Produktdaten mit individuellen Preisen anzeigen lassen möchten.
Sie werden dies definitiv nicht auf allen 5 Ebenen in separaten Quellcodes implementieren wollen.
Zu viele Ebenen in Ihrer Anwendung machen jede Anwendung sehr viel komplexer und die Entwicklungszeiten explodieren bei jeder allgemeinen Änderung, da der Schwerpunkt in der obigen Liste nicht auf der Büromanagement-Software und dem Reporting-System Ihres FAT Clients liegt.

Eine sehr einfache Anforderung, den korrekten Produktpreis für einen Kunden anzuzeigen, der Ersatzteile für eine bestimmte Maschine bestellen möchte, die er anhand der Seriennummer auf dem mobilen Gerät identifiziert hat, ist in einer Multilayer-App in der Tat eine schreckliche Idee, wenn es keine ordnungsgemäße Architektur mit nur so vielen Schichten verwendet, wie wirklich erforderlich sind.

Warum nicht direkte SQL-Anweisungen aus dem Quellcode Ihrer Webanwendung verwenden? Gute Frage! Und aus unserer Sicht ist die Antwort sehr einfach: Wir machen es genauso!

Ein Memo-Steuerelement wird verwendet, um einen gültigen SQL-Code in die verbundene Datenbank einzugeben. Das Button Click Event sendet es von der Anwendung über eine HTTPS-Verbindung und an einen Webserver, auf dem Apache und PHP laufen.
Die PHP-Engine verbindet sich in einem sehr einfachen 25-zeiligen PHP-Skript mit der Firebird-Datenbank und sendet den Befehl direkt an eine gespeicherte Prozedur in der Datenbank, in der die SQL-Anweisung verarbeitet wird.
Das Ergebnis für Selects wird zurückgegeben, und, wenn Objektrechte vergeben wurden, werden andere Anweisungen, wie einfügen-/aktualisieren-/löschen, auf die gleiche Weise ausgeführt.

Ein großer Vorteil:

Haben Sie beim Versuch, eine Datenbankverbindung herzustellen, jemals Probleme mit einer falschen Clientbibliothek oder fehlenden offenen Ports gehabt?

Dies wird hier nicht geschehen, da wir einfach das sichere HTTPS-Protokoll verwenden, um Befehle und Ergebnisse zu übertragen.

Sie können dieselbe Technologie nicht nur von pas2js aus verwenden, sondern auch von jeder anderen kompilierten oder interpretierten Anwendung, die HTTPS-UTL-Aufrufe unterstützt, auf diese zugreifen und die Ergebnisse verarbeiten kann.

Was muss ich dafür lernen?

Für die Lazarus-Konferenz vom 29. bis 30. November 2019 in Eindhoven, Niederlande, haben wir einen vollständigen Satz an Quellcodes vorbereitet, um zu zeigen, wie alles funktioniert.

Der vollständige Quellcode besteht aus ungefähr 600 Codezeilen, einschließlich aller in der Firebird-Datenbank verwendeten Metadatenobjekte, des vollständigen Lazarus-Quellcodes, des einfachen PHP-Quellcodes und allem, was Sie sonst noch benötigen.

Und wir geben Ihnen eine Garantie! - Am Ende des ersten Konferenztages werden Sie zu 100% in der Lage sein, eine Anwendung auf der Grundlage dieser Technologie zu erstellen.

Am ersten Tag der Konferenz beginnen wir mit diesem Projekt und zeigen Ihnen detailliert jedes verwendete Modul.

Wir werden mit einer leeren virtuellen Maschine beginnen, Firebird installieren, Apache installieren, PHP installieren, die erforderlichen pas2js installieren und damit beginnen, die Anwendung Zeile für Zeile zu implementieren.

Der Einstieg in die Web Development-Branche ist heute verhältnismäßig einfach. Aber ein Allrounder zu sein ist, wie auch in jeder anderen Branche, äußerst schwer.
Wir helfen Ihnen, wie Sie Ihrem Ziel als Full-Stack-Developer ein bisschen näher kommen! Los geht's!

Was muss ich dafür kaufen? Was ist kostenlos und Open Source?

Firebird:	Ja	www.firebirdsql.org
Apache:	Ja	www.apache.org
PHP:	Ja	www.php.org
Lazarus:	Ja	lazarus.freepascal.org
DemoDB Firebird	Ja	Quellcode: Auf dem USB-Laufwerk der Konferenzteilnehmer
DemoAPP Lazarus pas2js	Ja	Quellcode: Auf dem USB-Laufwerk der Konferenzteilnehmer

Für den Zugriff auf die Datenbank-Internals wird eine IBExpert-Vollversion verwendet, die für 259 € zzgl. MwSt. erhältlich ist.
Aber alles, was in der DemoDB implementiert ist, kann auch mit der kostenlosen IBExpert Personal Edition abgerufen werden.

Wenn Sie also darauf vorbereitet sein möchten, alles auf Ihrem eigenen Laptop zu erledigen, laden und installieren Sie bitte die kostenlose IBExpert Personal Edition oder die IBExpert Developer Studio Edition Vollversion herunter.

Alle anderen Setups und Dateien werden auf dem USB-Laufwerk der Konferenzteilnehmer gespeichert, das vom Blaise Pascal Magazine überreicht wird.

Sie möchten an der Konferenz teilnehmen?

Weitere Informationen finden Sie unter https://www.lazpro.net. Die Gebühr für einen Tag beträgt 55 € und für beide Tage 100 €.

Sie haben keine Zeit für Ihre Teilnahme oder die Entfernung ist zu weit?

Die Lazarus-Factory wird auch eine virtuelle Konferenz abhalten, wenn eine ausreichend hohe Anzahl an Teilnehmern erreicht wird.

Die virtuelle Konferenz wird als GoToMeeting-Session in englischer Sprache angeboten, an der Sie von zu Hause oder ggfs. von Ihrem Arbeitsplatz aus mit Ihrem eigenen Computer teilnehmen können. Sehen Sie Holger auf der Kamera und gleichzeitig die Präsentation auf dem Bildschirm. Der Inhalt ist entsprechend des Konferenzthemas wie oben aufgeführt.

Sie können auch das GoToMeeting-Chat-System und auf Wunsch Ihr Headset und Ihre eigene Webcam verwenden, um mit uns über Details zu chatten.

Der Preis für diese virtuelle Veranstaltung beträgt 50 € pro Person und wird per Vorauskasse berechnet. Die virtuelle Veranstaltung dauert ca. 4 Stunden.

Unsere virtuelle Veranstaltung beginnt zu verschiedenen Uhrzeiten standortabhängig für folgende Länder:

07:00 Uhr MEZ Asien, Japan, Australien usw.
11:00 MEZ Europa, Afrika
15:00 MEZ USA, Canada, Südamerika

Für Ihre Teilnahme senden Sie uns bitte eine E-Mail an sales@ibexpert.biz. Sie erhalten dann eine Liste mit allen verfügbaren Terminen für Ihr Land. Falls Sie eine abweichende Uhrzeit für Ihr Land bevorzugen, teilen Sie uns das bitte unbedingt rechtzeitig in Ihrer E-Mail mit.