



DataFlex Entwickler Tag 2019

Historienverwaltung

Vincent Oorsprong

Agenda

- › Geschichte und tiefe Verbindung
- › Was hat sich in 19.1 geändert?



[Direkter Link zum Dokument](#)

Wie hat alles angefangen?

- › Benutzer navigieren zu einem Dokument
 - › Verfügbar im Netzwerk (Internet) über eine URL
 - › z.B. <http://dataaccess.com/ourproducts.html>
- › Dokument kann auf ein anderes Dokument verweisen
 - › Das ist es, was ein Hyperlink wirklich bedeutet
 - › z.B. `DataFlex Studio`

Wie hat alles angefangen?

- › Der Browser führt eine Liste von URLs, die verwendet werden können:
 - › Gehe zurück und vorwärts
 - › Direkter Link zum Dokument (Deep Link)



Die ersten DataFlex-Anwendungen

Die ersten DataFlex-Anwendungen

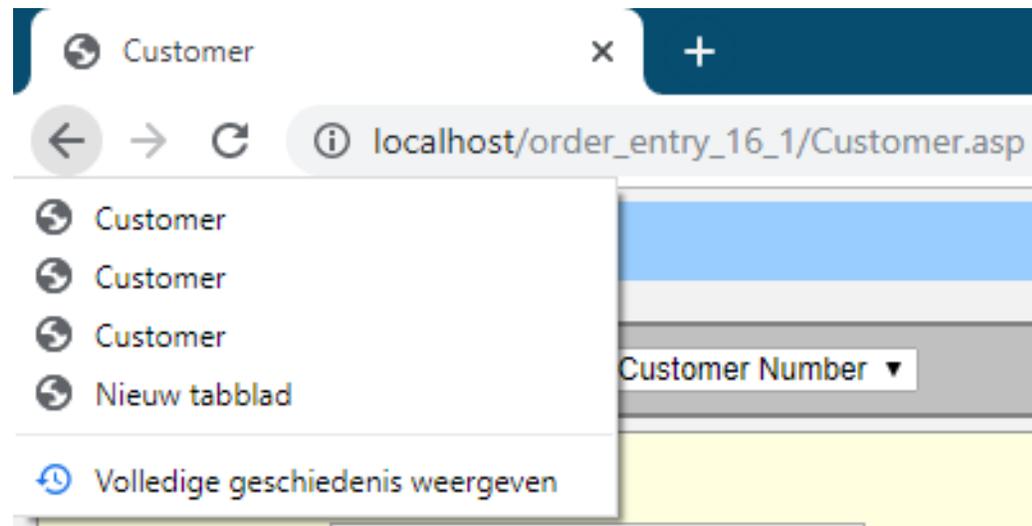
- › Basierend auf ASP (Active Server Pages)
- › Jede Dateneingabe oder Berichtsseite in einer eigenen ASP-Datei
- › Unterstützte Vorwärts- und Rückwärtsoperationen im Browser
- › Unterstützte URL kopieren und einfügen (Deep Linking)

Deep linking Beispiele

- › http://localhost/order_entry_16_1/Customer.asp?rowid=0b000000
 - › Zeigt Kunde mit Kundennummer 11
- › http://localhost/order_entry_16_1/CustomerList.asp?RunReport=1&Index=1
 - › Zeigt eine Liste der ersten 10 Zeilen aus der Kundentabelle an
- › http://localhost/order_entry_16_1/CustomerList.asp?RunReport=1&Start=15000000&Index=1
 - › Zeigt eine weitere Seite in der Kundenliste an

Schaltflächen "Zurück" und "Vor"

- › Da jede URL eine neue Seite generiert, erstellt der Browser einen Verlauf der angeforderten Seiten, und die Schaltflächen für die Vor- und Zurücknavigation können verwendet werden



An aerial view of Oslo, Norway, showing the Akerselva river in the foreground, the Holmenkollen Ski Stadium in the middle ground, and various city buildings in the background. The sky is overcast.

DataFlex AJAX-basierte Anwendungen

DataFlex AJAX-basierte Anwendungen

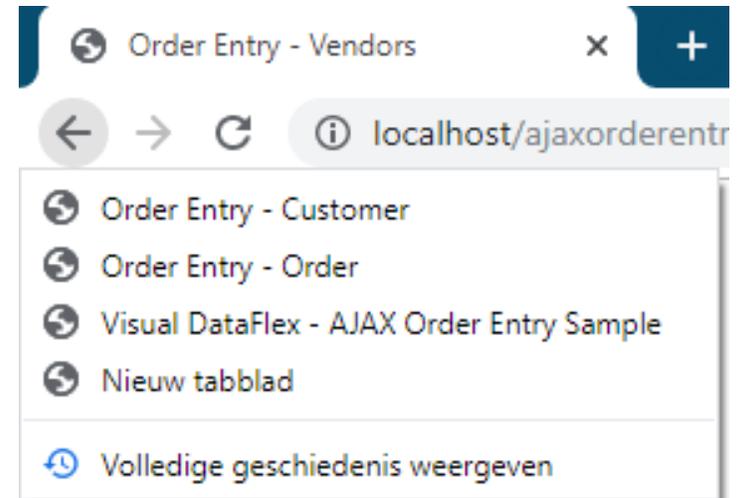
- › Anstatt eine neue URL an den Server zu senden, wurde die Seite einmal geladen und Datenanforderungen an den Server gesendet
 - › Der Blink / Page Reload Effekt wurde entfernt
- › Die Seiten waren eine Mischung aus ASP-, HTML-, CSS- und Javascript-Code
- › Deep Linking ist möglich und funktioniert wie bisher mit reinem ASP
 - › Standardmäßig nicht für einzelne Zeilen in der Tabelle

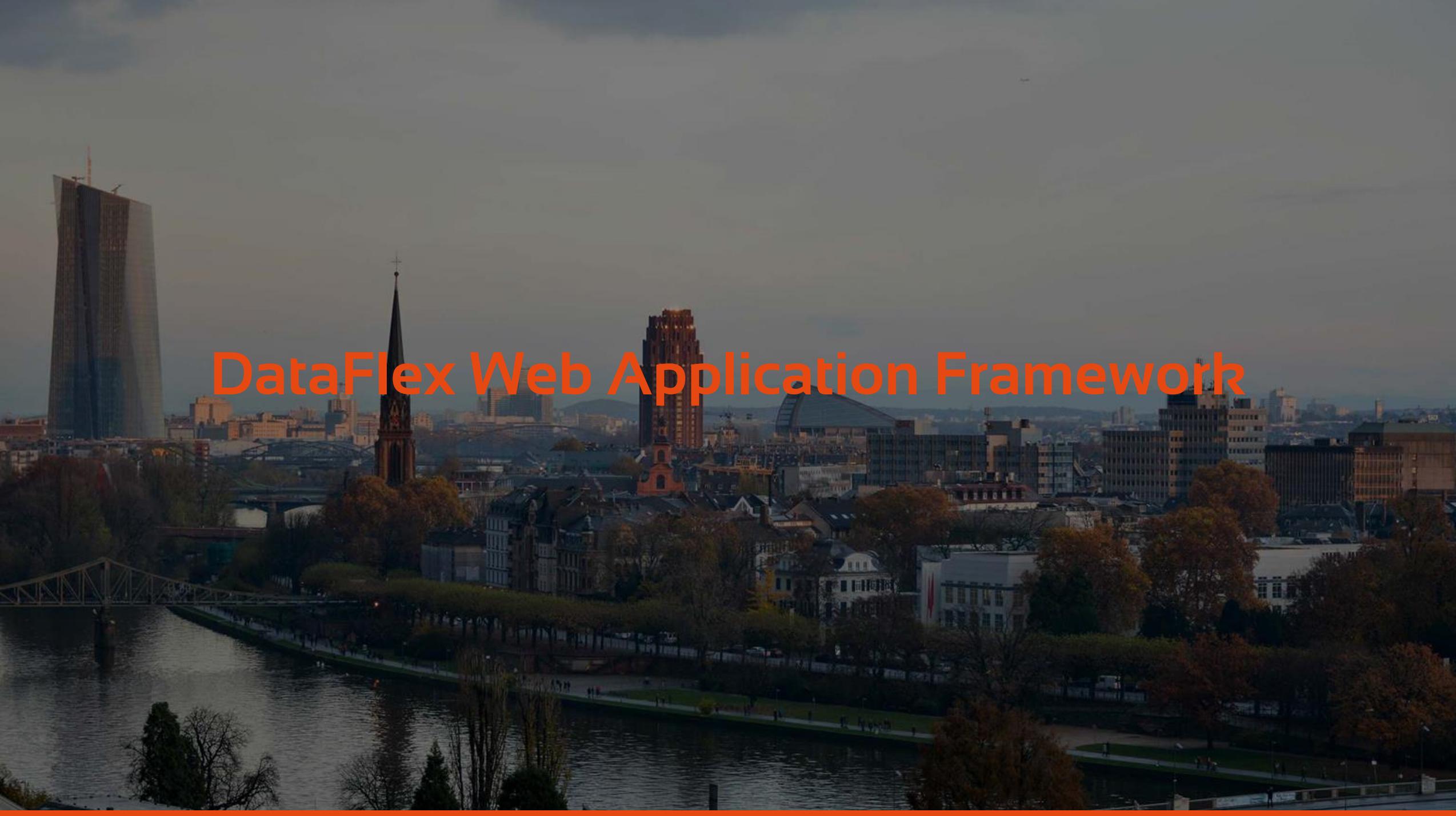
DataFlex AJAX-basierte Anwendungen

- › Anstatt eine neue URL an den Server zu senden, wurde die Seite einmal geladen und Datenanforderungen an den Server gesendet
- › Die Seiten waren eine Mischung aus ASP- und Javascript-Code
- › Blinkereffekt (Seiten-Reload) ist nicht mehr vorhanden

DataFlex AJAX-basierte Anwendungen

- › Deep Linking ist möglich und funktioniert wie bisher mit reinem ASP
- › Die Vor- und Zurück-Schaltflächen funktionieren für geladene ASP-Dateien und nicht für einzelne Zeilen in der Tabelle





DataFlex Web Application Framework

DataFlex WAF-Anwendungen (17.1 – 19.0)

- › Anwendung auf einer Seite
- › Nur eine HTML-Datei
 - › Lädt das Javascript
 - › Erstellt HTML-Knoten basierend auf dem Webobjekt, zu dem der Benutzer navigiert hat
- › ASP wurde nur zum Hoch- und Herunterladen von Dateien benötigt

DataFlex WAF-Anwendungen (17.1 – 19.0)

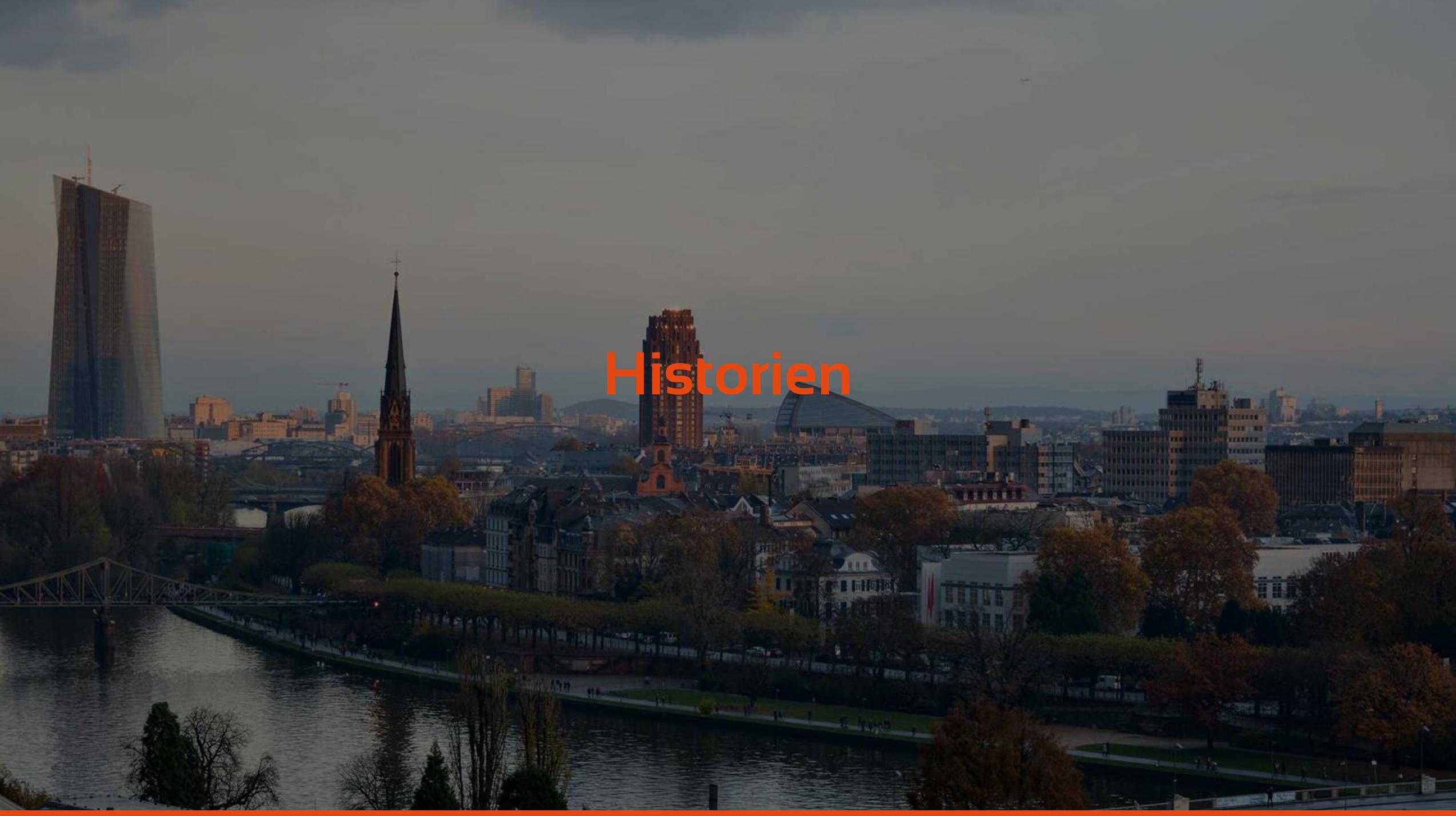
- › Die Schaltflächen "Zurück" und "Vorwärts" funktionieren, jedoch nicht beim Hin- und Hergehen durch die Anwendung
 - › Stattdessen haben sie die DataFlex-Anwendung verlassen
 - › Navigation zu zuvor besuchten Seiten / Websites
- › Deep Linking ist möglich
 - › z.B. <https://hde.dataaccess.eu/?action=ticket&id=6547>
 - › Code muss geschrieben werden



Was hat sich in 19.1 geändert?

Historienverwaltung

- › Browser Hersteller erstellen eine API zum manipulieren des Historie stack
 - › Das DataFlex Web Application Framework benutzt das
 - › Macht “vor” und “zuruck” buttons funktionsfaehig
- › Sie können sich dafür entscheiden, die Anwendung so zu belassen, wie sie war, und ohne Historie und Deep Linking
- › Erleichtert das Upgrade einer vorhandenen DataFlex-Anwendung ab 17.1



Historien

Codeänderungen

- › cWebApp object
 - › peApplicationStateMode
 - › Set to asmHistoryOnly oder asmHistoryAndUrls

| Constant | Meaning |
|-------------------|--|
| asmOff | Turns off history management. |
| asmHistoryOnly | URLs are not updated with the state hash, but history items are created. |
| asmHistoryAndUrls | URLs are updated and history items are created. |

Codeänderungen

› WebRegisterPath

- › Alle möglichen Navigationspfade müssen registriert werden
- › Am einfachsten ist es in dem Objekt, in dem der Zielpfad verwendet wird
 - › Sie müssen die aufrufende Objekt-ID nicht übergeben

› Verwendung von NavigatePath

- › Wenn es nur einen Navigationspfad gibt
 - › Eine IF / CASE-Struktur kann zu unterschiedlichen Aktionen führen
 - › Behalten Sie dann einfach den NavigateForward usw. bei, aber verwenden Sie WebRegisterPath

Codeänderungen

› WebRegisterPath beispiele

› Von:

```
Procedure OnClick String sButton String sRowId  
    Send NavigateForward to oZoomCustomer Self  
End_Procedure
```

› Zu:

```
WebRegisterPath ntNavigateForward oZoomCustomer  
  
Procedure OnClick String sButton String sRowId  
    Send NavigatePath  
End_Procedure
```

Optionale Codeänderungen

- › psApplicationTitle

- › Legt den Basisteil des Titels fest, der auf der Registerkarte angezeigt wird



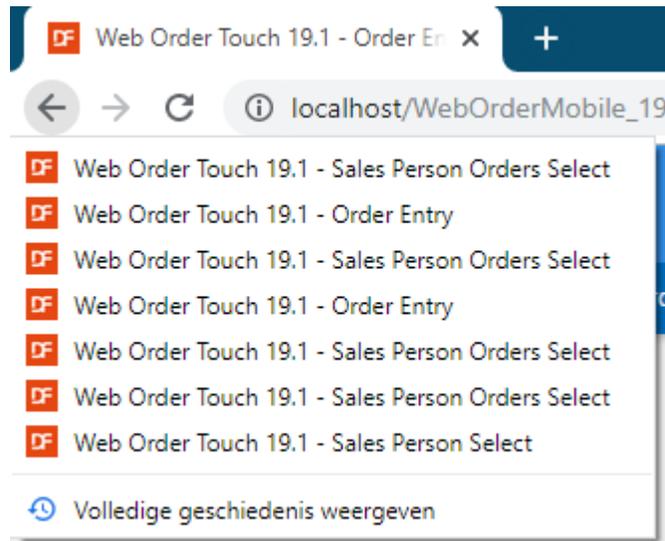
- › psCaption wird für den 2. Teil des Titels verwendet

- › In diesem Fall "Dashboard"
 - › Verwenden Sie zur Dynamisierung die SetHeaderCaption-Nachricht für Drilldown-Stilanwendungen

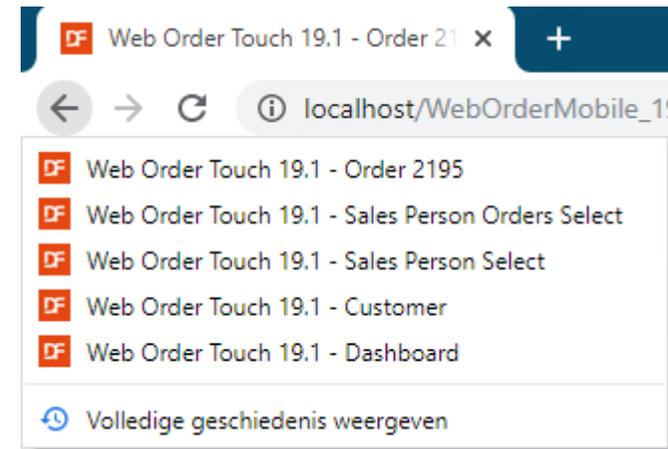
Titel der Verlaufs-URL

Machen Sie die URL kontextsensitiv

Standard



Geändert basierend auf der aktuellen Zeile





URL Änderungen

Deep linking – URL changes

- › Von: http://localhost/WebOrderMobile_19_1/Index.html
- › Zu:
http://localhost/WebOrderMobile_19_1/Index.html#SalesPersons
- › Struktur :
 - › Base URL / #StateKey – rowID – customData

URL – Statekey

- › Name der Webansicht
 - › z.B. oZoomOrder
- › Oder psStateViewName property value
 - › z.B. Order
- › Oder parameter of WebRegisterPath

```
WebRegisterPath eNavigationMode hoNavigateToView [ hoInvokingObject [ sStateKey ] ]
```

- › z.B. NewOrder

```
WebRegisterPath ntNavigateForward oZoomOrder Self "NewOrder"
```

- › http://localhost/WebOrderMobile_19_1/Index.html#Orders/NewOrder

URL – RowId

- › Der Wert der Spalten, die die Datensatzkennung für die Zeilen-ID der Haupt-DD der Ansicht bilden
 - › Beispiele:
 - › 5
 - › Numeric (Integer) ID (or Recnum)
 - › AB
 - › Alphanumerische Spalte
 - › 8A1901E4~5f~671B~5f~431A~5f~871C~5f~EADB2942E9EE
 - › Dies war die Rowguid in einer SQL-Tabelle

URL - RowId

- › Wenn Sie einen anderen Zeilenbezeichner möchten, implementieren (überschreiben) Sie die folgenden zwei Funktionen
 - › GenerateStateHashRecordId
 - › HashRecordIdToRowId

```
Function GenerateStateHashRecordId Integer iFile RowID rRowId Returns String  
    Function_Return (SerializeRowID (rRowId))  
End_Function
```

```
Function HashRecordIdToRowId Integer iFile String sRecordId RowID ByRef rRowId Returns Boolean  
    Boolean bFound
```

```
    Move (DeserializeRowID (sRecordId)) to rRowId  
    Move (FindByRowId (iFile, rRowId)) to bFound  
    Function_Return bFound  
End_Function
```

Custom data

- › Die RowId im Deep Link bietet genügend Informationen, um die Ansicht im selben Status wiederherzustellen, wenn es sich um eine Dateneingabeansicht handelt
- › Werte von Nicht-DEO-Objekten werden nicht gespeichert und wiederhergestellt, wenn der Deep Link ausgeführt wird
 - › Filteroptionen (Typ, von und bis Wert) für eine eingeschränkte Liste
 - › http://localhost/WebOrder_19_1/#oDemoConstrainedGrid-1
 - › Auswahlkriterien für DataFlex Reports-Berichte

Custom data

- › Implementieren Sie das OnDefineCustomStateHash-Ereignis, um den CustomData-Teil des Ansichtshashes zu erstellen

```
Procedure OnDefineCustomStateHash String ByRef sStateHash
    String sValue

    Forward Send OnDefineCustomStateHash (&sStateHash)

    WebGet psValue of oFromCustomerCustomerNumber to sValue
    If (sValue <> "") Begin
        Move (sStateHash + "FromCustNum=" + sValue) to sStateHash
    End

    WebGet psValue of oToCustomerCustomerNumber to sValue
    If (sValue <> "") Begin
        If (sStateHash <> "") Begin
            Move (sStateHash + "|") to sStateHash
        End
        Move (sStateHash + "ToCustNum=" + sValue) to sStateHash
    End
End
```

Benutzerdefinierte Datenverwendung

- › Dekodieren Sie die benutzerdefinierten Daten in einem Ereignis wie OnNavigateForward

```
Get CustomStateHash to sStateHash
Move (StrSplitToArray (sStateHash, "|")) to sValuePairs
Move (SizeOfArray (sValuePairs) - 1) to iElements
For iElement from 0 to iElements
  Move (Pos ("=", sValuePairs[iElement])) to iSeparatorPos
  If (iSeparatorPos > 0) Begin
    Move (Left (sValuePairs[iElement], iSeparatorPos - 1)) to sName
    Case Begin
      Case (sName = "FromCustNum")
        WebSet psValue of oFromCustomerCustomerNumber to (Right (sValuePairs[iElement]
        Case Break
      Case (sName = "ToCustNum")
        WebSet psValue of oToCustomerCustomerNumber to (Right (sValuePairs[iElement]
        Case Break
      Case (sName = "FromCustName")
        WebSet psValue of oFromCustomerCustomerNumber to (Right (sValuePairs[iElement]
```

Benutzerdefinierte Datenverwendung

- › Ich möchte die benutzerdefinierten Daten nicht als :
 - › `SortField%3D%7BCustomer.Customer_Number%7D%7CSortDirection%3D0`
- › Und beschlossen, eine Base64-Zeichenfolge daraus zu machen, was zu führte :
 - › `RnJvbUN1c3R0dW09MTB8VG9DdXN0TnVtPTE1fFNvcnRGaWVsZD17Q3VzdG9tZXIuTmFtZX18U29ydERpcmVjdGlvbj0x`
- › Ja, es ist länger, aber weniger einfach zu ändern und zu manipulieren

Deaktivieren Sie die Deep-Link-Funktion

- › Einige Ansichten müssen möglicherweise keine Tiefenverknüpfung unterstützen
 - › z.B. Ergebnisansicht melden
 - › Auswahlkriterien und -optionen können in benutzerdefinierten Daten auf der Auswahlseite gespeichert werden
 - › Control via peStateMode property
 - › z.B. set to vsmOff

Deep Link & Sicherheit

- › Umgeht das Anmeldesystem nicht 😊
- › Demo

Deep Link & Sicherheit

- › Achten Sie darauf, wenn Sie zurück navigieren dass zu viele Daten angezeigt werden
 - › http://localhost/WebOrderMobile_19_1/Index.html#Orders/Order-2227/OrderLine-2227_1
 - › Zeigt die erste Bestellposition der Bestellung 2227 an
 - › Wenn Sie über den Breadcrumb zurückgehen, werden alle Bestellungen angezeigt



Deep Link & Sicherheit

› Lösung

- › Erstellen Sie ein Login-Konto, mit dem nur bestimmte Bestellungen angezeigt werden
 - › Beschränken Sie den Zugriff auf Teile der Anwendung, die der Benutzer nicht sehen sollte



Zusammenfassung

- › Großer Fortschritt in der Entwicklung von Webanwendungen
- › Benutzer können die Schaltfläche "Zurück" erneut verwenden
- › Ein URL-Deep-Link kann bereitgestellt werden



DataFlex Entwickler Tag 2019

Vielen Dank für Ihre Aufmerksamkeit!

Haben Sie Fragen?