



DataFlex to New Heights

# Custom Web Controls

Henri Reterink

# New stuff

- > PDF Viewer
- > Gantt Chart
- > Online payments in DataFlex



# PDF Viewer

# PDF Viewer

- > Display PDFs in a uniform way
  - > Any browser
  - > Any device
- > Very relevant for mobile devices
- > Uses a WebIframe with ViewerJS library



**Demo**

# PDF Viewer - Usage

- > Add ViewerJs folder to your AppHtml folder
- > Create a WebIframe
- > (Web)Set psUrl to “viewerJS/#../” followed by PDF location
- > PDF location can also be obtained using Get DownloadUrl
- ...

# PDF Viewer - Usage

Object oWebButton1 is a cWebButton

Set piColumnSpan to 3

Set psCaption to "Load pdf 1"

Procedure OnClick

Send ClearIframe of oWebIFrame1

WebSet pbRender of oWebImage1 to False

WebSet pbRender of oWebIFrame1 to True

WebSet psUrl of oWebIFrame1 to "ViewerJS/#../Uploads/Scheduler-information.pdf"

End\_Procedure

End\_Object

# PDF Viewer - Usage

## > Using Get DownloadURL

```
Object oWebButton2 is a cWebButton
  Set piColumnSpan to 3
  Set piColumnIndex to 3
  Set psCaption to "Load pdf 2"

  Procedure OnClick
    String sUrl sFilePath

    Send ClearIframe of oWebIFrame1
    WebSet pbRender of oWebImage1 to False
    WebSet pbRender of oWebIFrame1 to True

    Get GetDownloadFolderPath of ghoWebApp to sFilePath
    Move (sFilePath + "Scheduler changelog 2016-06-03.pdf") to sFilePath

    Get DownloadURL of ghoWebResourceManager sFilePath to sUrl

    Move ("ViewerJS/#../" + sUrl) to sUrl
    WebSet psUrl of oWebIFrame1 to sUrl
  End_Procedure
End_Object
```

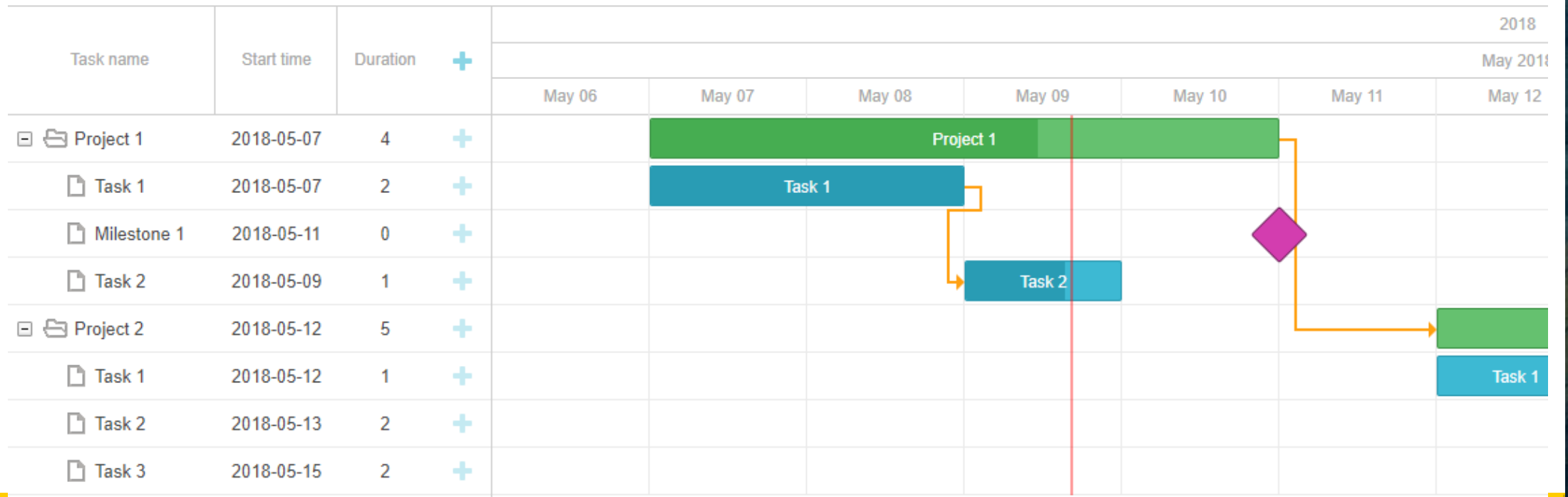


# Gantt Chart

A photograph of a large, rugged mountain peak with a flat top, covered in green vegetation. The sky is a clear, pale blue. The text "Gantt Chart" is overlaid in yellow on the left side of the mountain.

# Gantt Chart

- > Used to plan and keep track of progress
- > Main idea: Tasks & Links → Dependencies
- > Based on dHTMLX Gantt Chart library



# Main features

- > 3 different types of items (projects, tasks, milestones)
- > Links between items
- > Interaction between items
- > Critical path
- > Multiple scales



**Demo**

# Usage

- > Visual interaction is handled by dHtmlx library
- > 1 procedure to augment to load data
  - > OnLoadGanttData
- > Augment key events for data management after interaction
  - > On..Create/Delete/Update..Task
  - > On..Create/Delete/Update..Link

# Loading data

```
Procedure OnLoadGanttData tWebDhxGanttData ByRef aGanttData
    Forward Send OnLoadGanttData (&aGanttData)

//     Gets the Data from the DB and returns it in the right format
    Get TestDataFromTables to aGanttData
End_Procedure
```

# Editing an item

```
Procedure OnTaskDb1Click tWebDhxTask tTask
    Forward Send OnTaskDb1Click tTask

    // Popup edit dialog
    Register_Object oEditTask
    Send Popup of oEditTask (Self)
    Send InitialiseDetails of oEditTask tTask False
End_Procedure
```

```
Procedure OnUpdateTask tWebDhxTask[] tUpdatedTasks
  Integer iSize iIndex

  Move (SizeOfArray(tUpdatedTasks)) to iSize
  While (iIndex < iSize)
    Send Clear of oTask_DD
    Move tUpdatedTasks[iIndex].biId to Task.taskId
    Send Find of oTask_DD EQ 2

    If (Found) Begin
      Set Field_Changed_Value of oTask_DD Field Task.text to tUpdatedTasks[iIndex].sText
      Set Field_Changed_Value of oTask_DD Field Task.startdate to tUpdatedTasks[iIndex].dtStartDate
      // etc
      Send Request_Save of oTask_DD
    End

    Increment iIndex
  Loop
End_Procedure
```





# Online Payments in DataFlex

# Payment Module – the idea

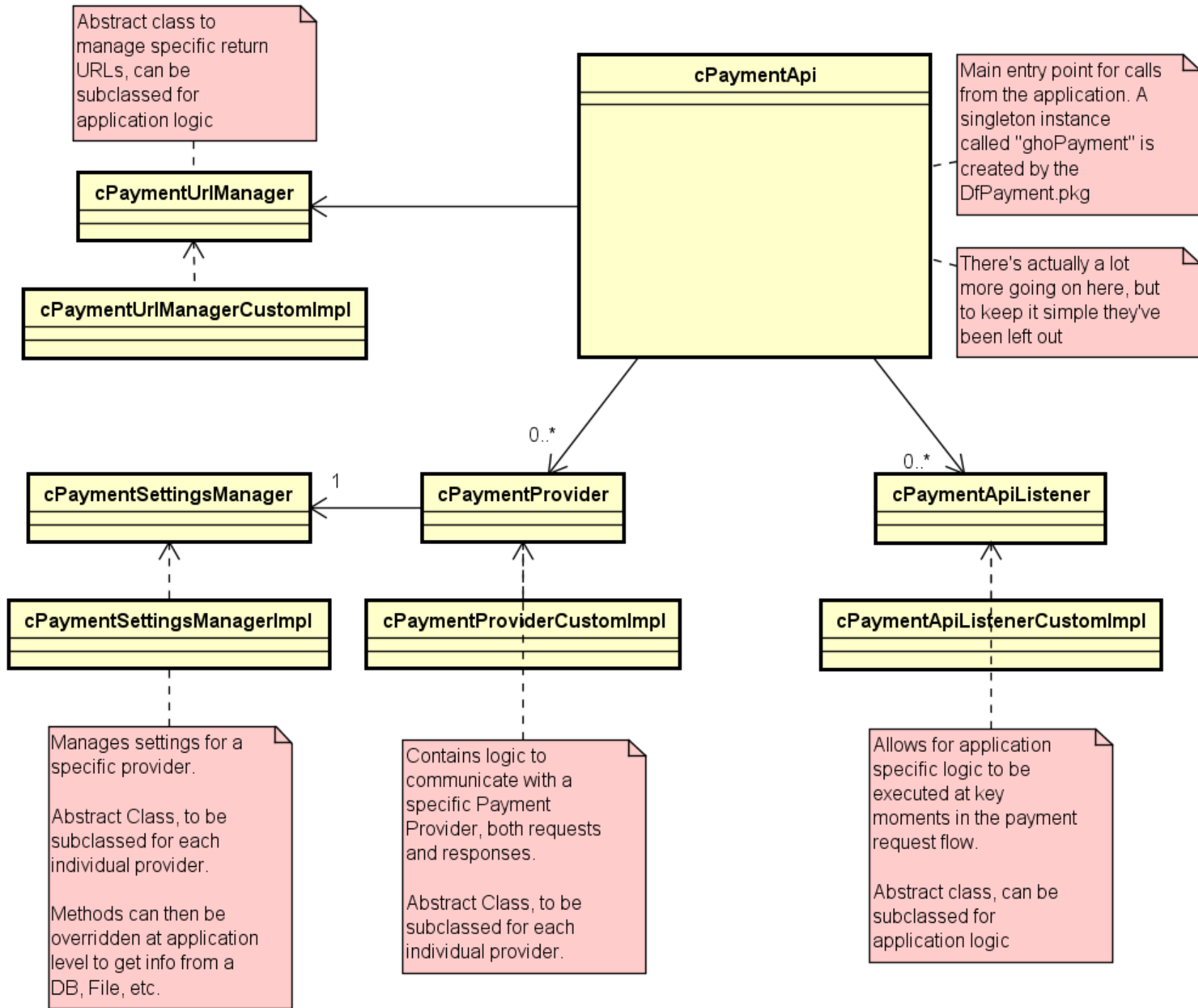
- > Single library that serves as a framework for multiple online payment solutions
- > Same DF interface for every payment provider
- > As easy to integrate as possible
- > Modular, pick what you need
- > Expandable, add your own providers
- > No dependancy on specific table structure



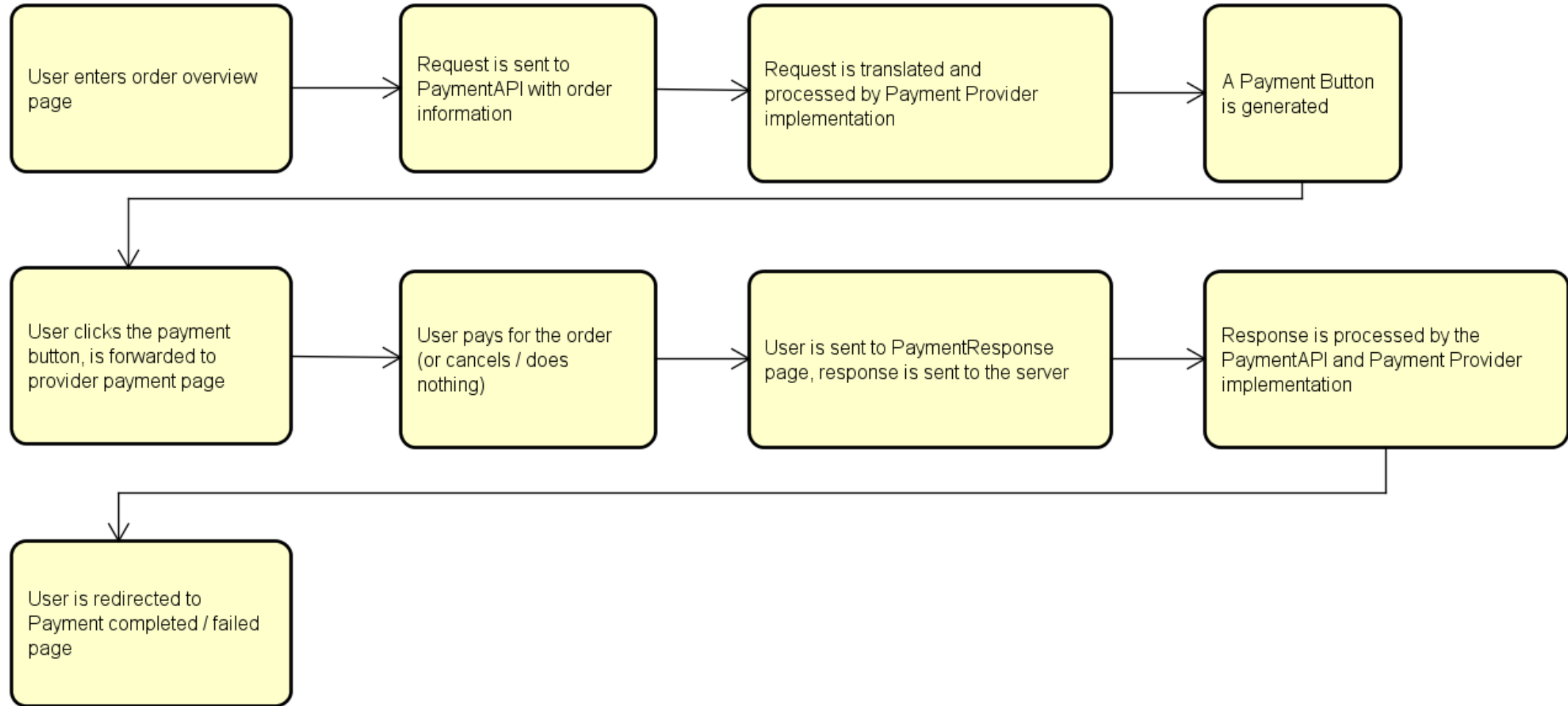
Demo.. Sort of 😊



**How it works – super simplified version**



# How it works – super simplified version



# Using the module – minimalist version

- > Add the module library
- > Copy some JS & ASP files
- > “Use YourProviderNameHere.pkg”
- > Create a WebPaymentButton in your view
- > Call “CreatePayment”, pass the button as a handle
- > In the future: a wizard for this 😊

# Inserting application logic

- > Create custom Listeners
  - > Listeners perform application logic at key moments:
    - > Before a payment request is sent
    - > When a Payment response is received
    - > When errors occur
- > Create custom Settings Managers for your provider
  - > Responsible for getting up-to-date provider specific settings from your DB, file, etc.
- > Create a custom cPaymentUrlManager
  - > Define URLs for various response types



# Expanding the module

- > Create your own cPaymentProvider implementation
- > Mandatory methods
- > Will give a program error when not overridden
- > Create a base cPaymentSettingsManager subclass for your provider if needed

# How do i get this?!

- > PdfViewer
  - > Free! Download link to be provided
- > Gantt Chart
  - > DF Wrapper: €750
  - > dHtmlx license
- > Payment Module
  - > Work in progress, pricing tbd



DataFlex to New Heights

**Thank you!**  
**Are there any questions?**

