GMC and Kony

Integration

Version 9.4

CLASSIFICATION: PUBLIC

GMC Software AG

http://www.gmc.net/documentation

GMC and Kony

Integration

Product version 9.4

Document version 9.4.0.1

Release date: September 2014

GMC Software AG
www.gmc.net

See also all GMC documentation online

Should you have any queries, suggestions or comments concerning these materials, please do not hesitate to contact us at documentation@gmc.net.

This document contains information classified as Public.

The GMC Software Group of companies are ISO 9001:2008 certified.

Copyright

Information contained within this document may contain technical inaccuracies or typographical errors. Changes will be added periodically and modifications will be made thereto without prior notification. GMC Software AG does not enter into any obligations or responsibilities regarding the content of this document and does not assume any legal liability – neither expressed or implied – for its accuracy, completeness and/or usefulness.

Trademarks

Table of Contents

1 Introduction

The GMC Showcase application is a mobile application developed in Kony Studio. It is a working, sample application that demonstrates the functionality of the GMC Mobile API used on the Kony platform.

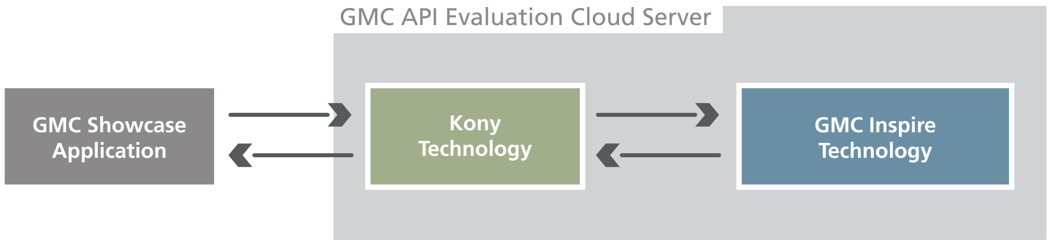
The application supports:

- Integration of variable data into documents.
- Creation of output documents in various formats.
- Displaying the output documents on mobile devices.
- Creation of output documents divided into several pages.
- Listing properties of files contained in a directory.
- Integration with e-signature provider.

The application is designed for mobile devices, with the graphics optimized for smart phones. However, the functionality is usable on both smart phones and tablets.

Supported platforms are:

- Android – for smart phone and tablet
- iOS – for iPhone and iPad



GMC Showcase application Ready-made mobile application installed on a mobile device.

GMC API Evaluation Cloud Server GMC Cloud server with demo installations of necessary components.

The source code of the GMC Showcase application serves programmers as a base material for developing their own mobile applications, designed to their specific needs. The goal is for programmers to use their own environment supporting their own functionalities and document design.

1.1 Readers of this Document

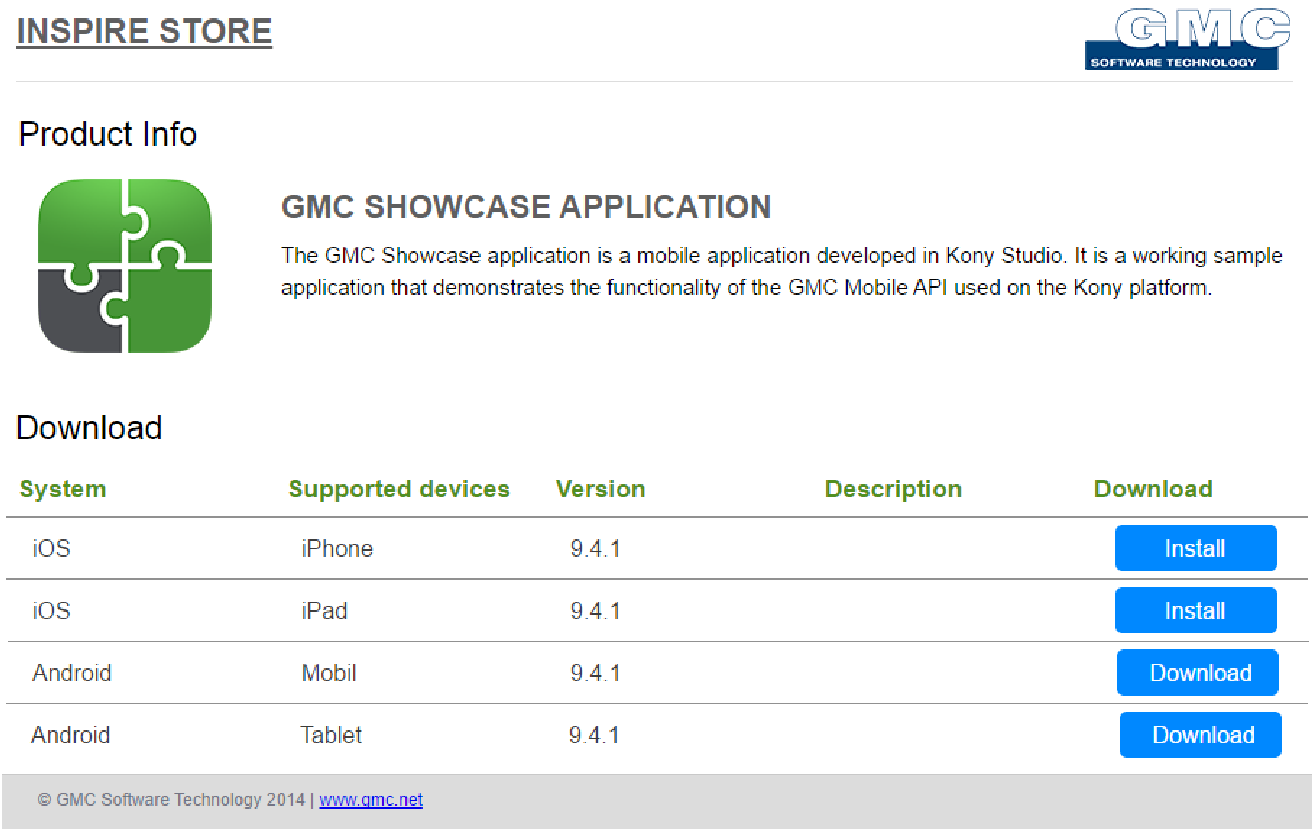
This document is intended for business people and programmers.

Business People

Business people can download the ready-made GMC Showcase application to a mobile device and test it.

The application can be downloaded for free from Inspire Store: http://inspirestore.gmc.net ↗. To obtain the login credentials, contact the GMC Product Management group.

The application only needs to be downloaded (i.e. installed on a mobile device), no additional settings are needed.



INSPIRE STORE

Product Info

GMC SHOWCASE APPLICATION

The GMC Showcase application is a mobile application developed in Kony Studio. It is a working sample application that demonstrates the functionality of the GMC Mobile API used on the Kony platform.

Download

System	Supported devices	Version	Description	Download
iOS	iPhone	9.4.1		Install
iOS	iPad	9.4.1		Install
Android	Mobil	9.4.1		Download
Android	Tablet	9.4.1		Download

© GMC Software Technology 2014 | www.gmc.net

Sections (within this document) providing information for business people are:

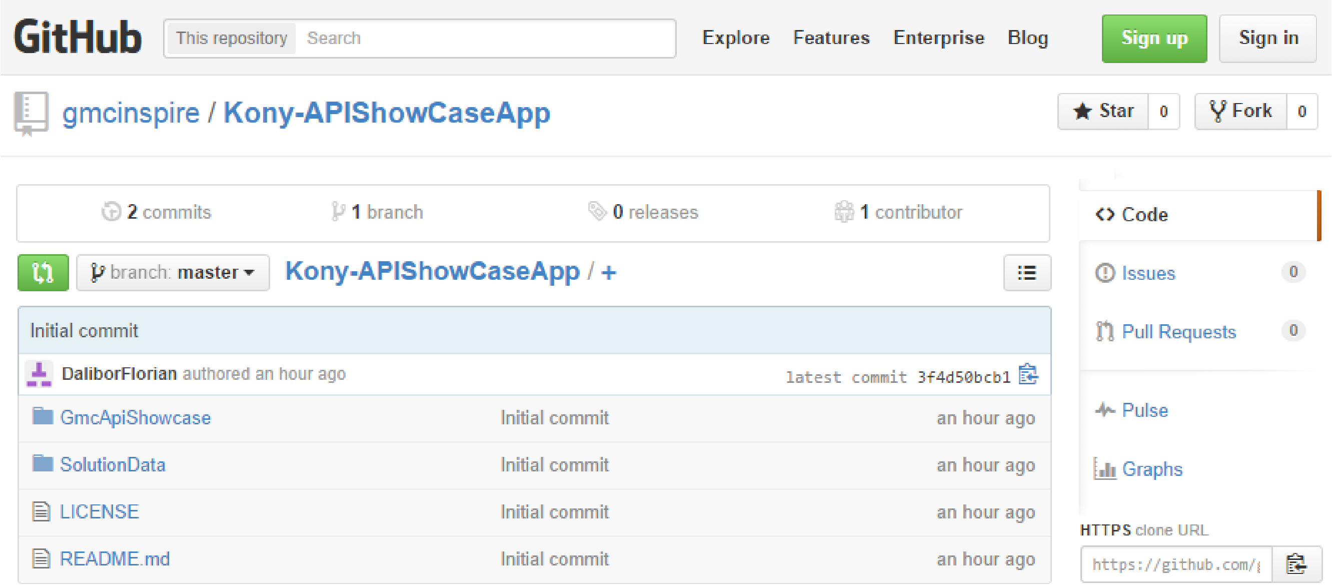
- Showcase Application GUI
- Architecture

Programmers

Besides testing the ready-made GMC Showcase application, programmers can download the whole project of the application, open the project in Kony Studio, examine and customize it and compile their own application.

The project can be downloaded from GitHub: https://github.com/gmcinspire/Kony-APIShowCaseApp ↗.

The project should be set up and customized in three stages described in this document.



Initial commit

DaliborFlorian authored an hour ago latest commit 3f4d50bcb1

GmcApiShowcase	Initial commit	an hour ago
SolutionData	Initial commit	an hour ago
LICENSE	Initial commit	an hour ago
README.md	Initial commit	an hour ago

Sections (within this document) providing information exclusively for programmers are:

- Setting Up
- GMC Mobile API

Tip You can learn more about Kony Studio on www.kony.com ↗.

To examine the GMC Showcase application project in Kony Studio, it is sufficient to have a trial version. Here you can register for a free 90 day trial: http://www.kony.com/products/development/trial ↗.

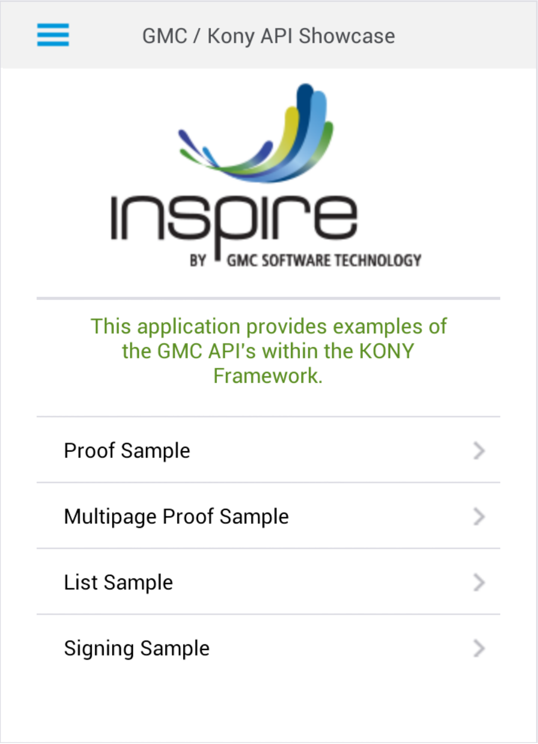
2 Showcase Application GUI

Launch the application by touching its icon on the mobile device.

The initial page displays.

On the initial page (or after touching the **List** ≡ icon), you get the application's menu with the following sections:

- Proof Sample
- Multipage Proof Sample
- List Sample
- Signing Sample



≡ GMC / Kony API Showcase

inspire BY GMC SOFTWARE TECHNOLOGY

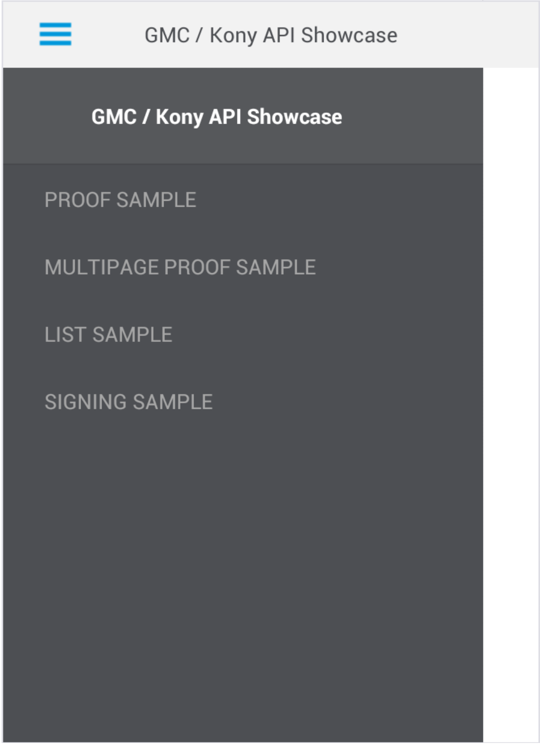
This application provides examples of the GMC API's within the KONY Framework.

Proof Sample >

Multipage Proof Sample >

List Sample >

Signing Sample >



≡ GMC / Kony API Showcase

GMC / Kony API Showcase

PROOF SAMPLE

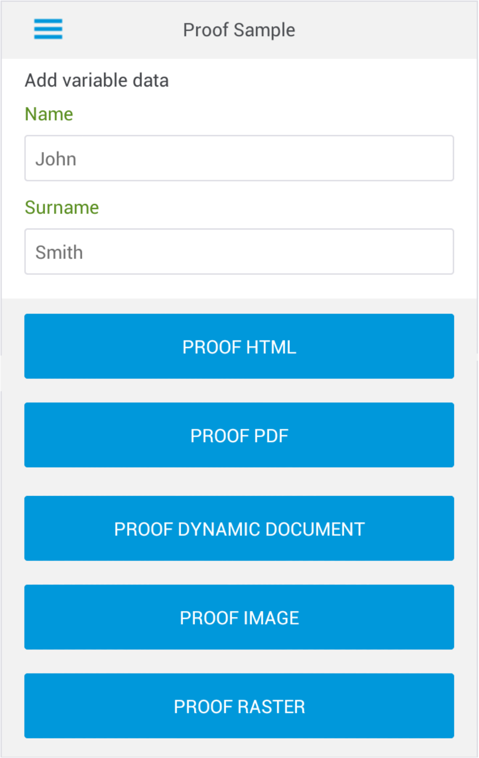
MULTIPAGE PROOF SAMPLE

LIST SAMPLE

SIGNING SAMPLE

2.1 Proof Sample

If you go to Proof Sample, you get the following screen.



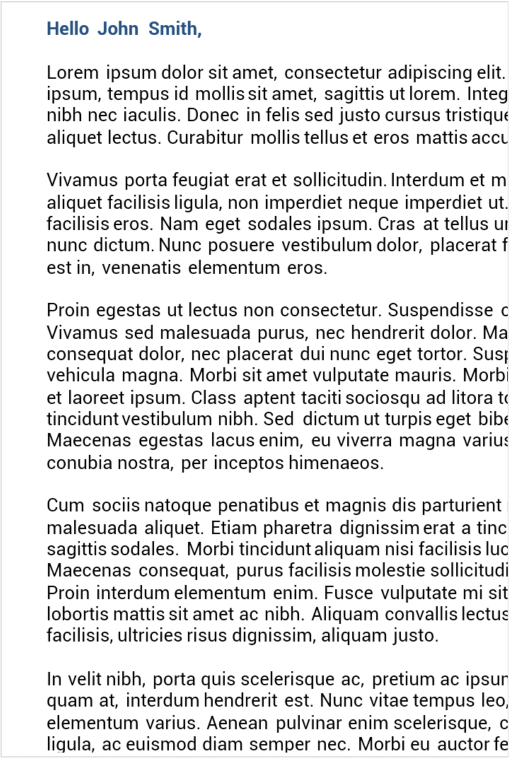
Name: John

Surname: Smith

You can edit the **Name** and **Surname** text fields. Provided values will be included in the output document.

Proof HTML

After touching the **Proof HTML** button, you get the output document in the HTML format.



Hello John Smith,

Lorem ipsum dolor sit amet, consectetur adipiscing elit. ipsum, tempus id mollis sit amet, sagittis ut lorem. Integ nibh nec iaculis. Donec in felis sed justo cursus tristique aliquet lectus. Curabitur mollis tellus et eros mattis accu

Vivamus porta feugiat erat et sollicitudin. Interdum et m aliquet facilisis ligula, non imperdiet neque imperdiet ut. facilisis eros. Nam eget sodales ipsum. Cras at tellus ur nunc dictum. Nunc posuere vestibulum dolor, placerat f est in, venenatis elementum eros.

Proin egestas ut lectus non consectetur. Suspendisse c Vivamus sed malesuada purus, nec hendrerit dolor. Ma consequat dolor, nec placerat dui nunc eget tortor. Susp vehicula magna. Morbi sit amet vulputate mauris. Morbi et laoreet ipsum. Class aptent taciti sociosqu ad litora to tincidunt vestibulum nibh. Sed dictum ut turpis eget bibe Maecenas egestas lacus enim, eu viverra magna varius conubia nostra, per inceptos himenaeos.

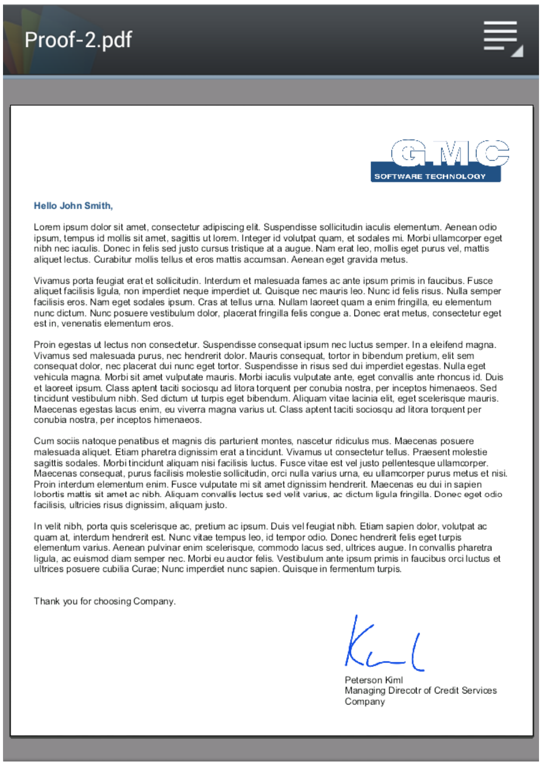
Cum sociis natoque penatibus et magnis dis parturient malesuada aliquet. Etiam pharetra dignissim erat a tinc sagittis sodales. Morbi tincidunt aliquam nisi facilisis luc Maecenas consequat, purus facilisis molestie sollicitudi Proin interdum elementum enim. Fusce vulputate mi sit lobortis mattis sit amet ac nibh. Aliquam convallis lectus facilisis, ultricies risus dignissim, aliquam justo.

In velit nibh, porta quis scelerisque ac, pretium ac ipsur quam at, interdum hendrerit est. Nunc vitae tempus leo, elementum varius. Aenean pulvinar enim scelerisque, c ligula, ac euismod diam semper nec. Morbi eu auctor fe

Values provided in the **Name** and **Surname** text fields are included.

Proof PDF

After touching the **Proof PDF** button, the output document in the PDF format is downloaded to your mobile device.



Hello John Smith,

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse sollicitudin iaculis elementum. Aenean odio ipsum, tempus id mollis sit amet, sagittis ut lorem. Integer id volutpat quam, et sodales mi. Morbi ullamcorper eget nibh nec iaculis. Donec in felis sed justo cursus tristique at a augue. Nam erat leo, mollis eget purus vel, mattis aliquet lectus. Curabitur mollis tellus et eros mattis accumsan. Aenean eget gravida metus.

Vivamus porta feugiat erat et sollicitudin. Interdum et malesuada fames ac ante ipsum primis in faucibus. Fusce aliquet facilisis ligula, non imperdiet neque imperdiet ut. Quisque nec mauris leo. Nunc id felis risus. Nulla semper facilisis eros. Nam eget sodales ipsum. Cras at tellus urna. Nullam laoreet quam a enim fringilla, eu elementum nunc dictum. Nunc posuere vestibulum dolor, placerat fringilla felis congue a. Donec erat metus, consectetur eget est in, venenatis elementum eros.

Proin egestas ut lectus non consectetur. Suspendisse consequat ipsum nec luctus semper. In a eleifend magna. Vivamus sed malesuada purus, nec hendrerit dolor. Mauris consequat, tortor in bibendum pretium, elit sem consequat dolor, nec placerat dui nunc eget tortor. Suspendisse in risus sed dui imperdiet egestas. Nulla eget vehicula magna. Morbi sit amet vulputate mauris. Morbi iaculis vulputate ante, eget convallis ante rhoncus id. Duis et laoreet ipsum. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Sed tincidunt vestibulum nibh. Sed dictum ut turpis eget bibendum. Aliquam vitae lacinia elit, eget scelerisque mauris. Maecenas egestas lacus enim, eu viverra magna varius ut. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos.

Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Maecenas posuere malesuada aliquet. Etiam pharetra dignissim erat a tincidunt. Vivamus ut consectetur tellus. Praesent molestie sagittis sodales. Morbi tincidunt aliquam nisi facilisis luctus. Fusce vitae est vel justo pellentesque ullamcorper. Maecenas consequat, purus facilisis molestie sollicitudin, orci nulla varius urna, eu ullamcorper purus metus et nisi. Proin interdum elementum enim. Fusce vulputate mi sit amet dignissim hendrerit. Maecenas eu dui in sapien lobortis mattis sit amet ac nibh. Aliquam convallis lectus sed velit varius, ac dictum ligula fringilla. Donec eget odio facilisis, ultricies risus dignissim, aliquam justo.

In velit nibh, porta quis scelerisque ac, pretium ac ipsum. Duis vel feugiat nibh. Etiam sapien dolor, volutpat ac quam at, interdum hendrerit est. Nunc vitae tempus leo, id tempor odio. Donec hendrerit felis eget turpis elementum varius. Aenean pulvinar enim scelerisque, commodo lacus sed, ultrices augue. In convallis pharetra ligula, ac euismod diam semper nec. Morbi eu auctor felis. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Nunc imperdiet nunc sapien. Quisque in fermentum turpis.

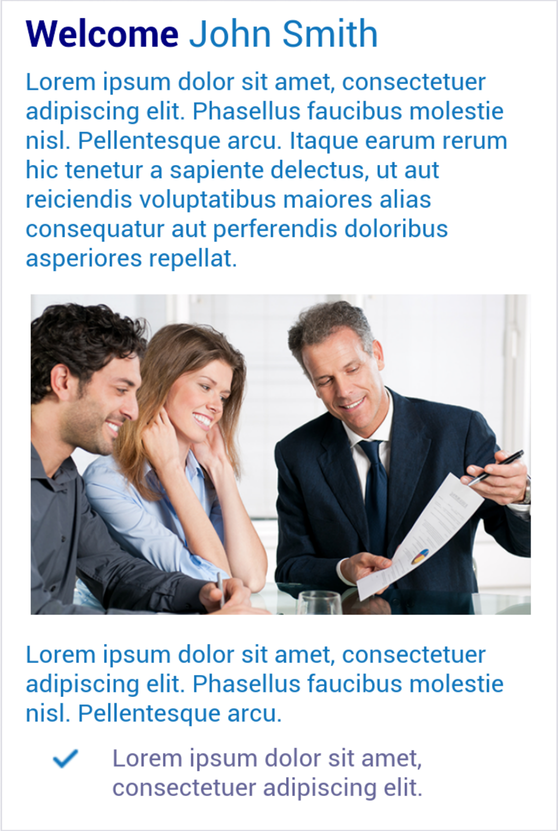
Thank you for choosing Company.

Peterson Kiml
Managing Direcotr of Credit Services
Company

Values provided in the **Name** and **Surname** text fields are included.

Proof Dynamic Document

After touching the **Proof Dynamic Document** button, you get the dynamic document output.



Welcome John Smith

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Phasellus faucibus molestie nisl. Pellentesque arcu. Itaque earum rerum hic tenetur a sapiente delectus, ut aut reiciendis voluptatibus maiores alias consequatur aut perferendis doloribus asperiores repellat.

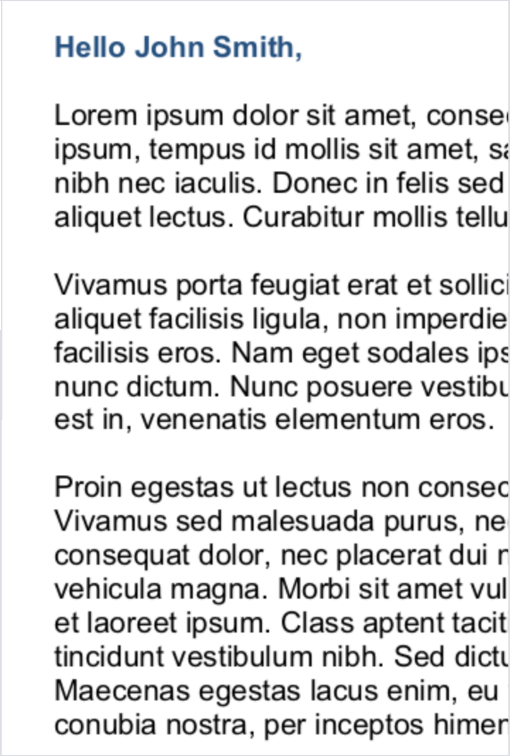
Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Phasellus faucibus molestie nisl. Pellentesque arcu.

- ✓ Lorem ipsum dolor sit amet, consectetuer adipiscing elit.

Values provided in the **Name** and **Surname** text fields are included.

Proof Image

After touching the **Proof Image** button, you get the output document in the Image format.



Hello John Smith,

Lorem ipsum dolor sit amet, conse
ipsum, tempus id mollis sit amet, sa
nibh nec iaculis. Donec in felis sed
aliquet lectus. Curabitur mollis tellu

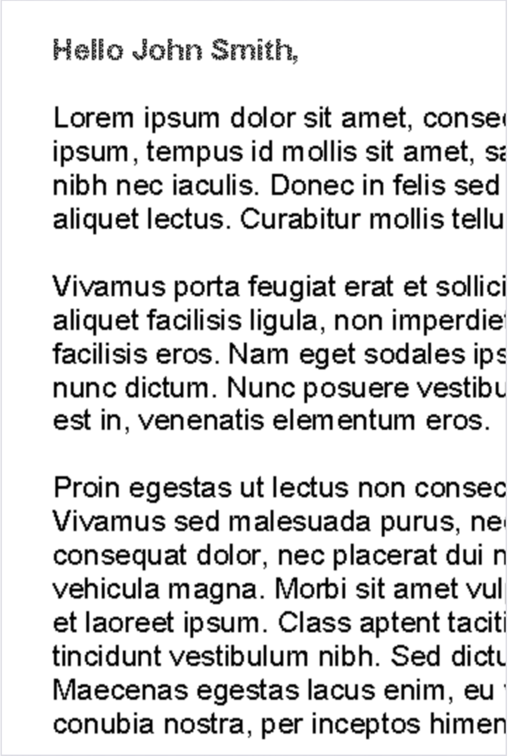
Vivamus porta feugiat erat et sollici
aliquet facilisis ligula, non imperdie
facilisis eros. Nam eget sodales ips
nunc dictum. Nunc posuere vestibu
est in, venenatis elementum eros.

Proin egestas ut lectus non consec
Vivamus sed malesuada purus, ne
consequat dolor, nec placerat dui n
vehicula magna. Morbi sit amet vul
et laoreet ipsum. Class aptent tacit
tincidunt vestibulum nibh. Sed dictu
Maecenas egestas lacus enim, eu
conubia nostra, per inceptos himen

Values provided in the **Name** and **Surname** text fields are included.

Proof Raster

After touching the **Proof Raster** button, you get the output document in the Raster Image format.



Hello John Smith,

Lorem ipsum dolor sit amet, conse
ipsum, tempus id mollis sit amet, sa
nibh nec iaculis. Donec in felis sed
aliquet lectus. Curabitur mollis tellu

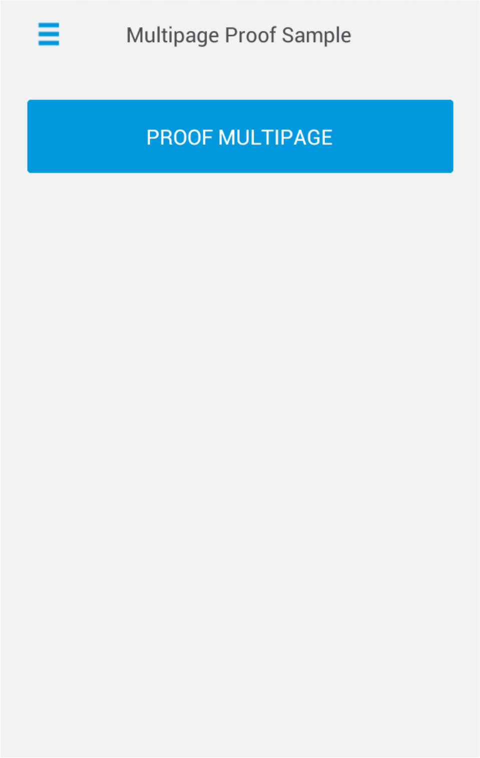
Vivamus porta feugiat erat et sollici
aliquet facilisis ligula, non imperdie
facilisis eros. Nam eget sodales ips
nunc dictum. Nunc posuere vestibu
est in, venenatis elementum eros.

Proin egestas ut lectus non consec
Vivamus sed malesuada purus, ne
consequat dolor, nec placerat dui n
vehicula magna. Morbi sit amet vul
et laoreet ipsum. Class aptent taciti
tincidunt vestibulum nibh. Sed dictu
Maecenas egestas lacus enim, eu
conubia nostra, per inceptos himen

Values provided in the **Name** and **Surname** text fields are included.

2.2 Multipage Proof Sample

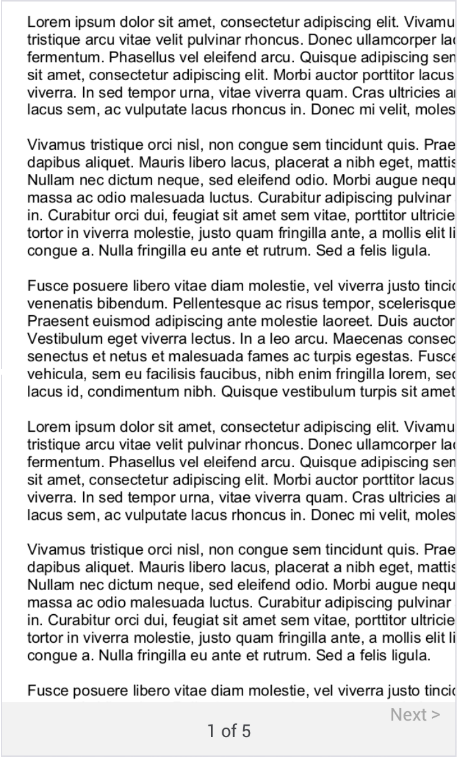
If you go to Multipage Proof Sample, you get the following screen.



No text fields for variable data are currently available.

Proof Multipage

After touching the **Proof Multipage** button, you get the output document in the Image format. The document is divided into multiple pages.



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamu
tristique arcu vitae velit pulvinar rhoncus. Donec ullamcorper lac
fermentum. Phasellus vel eleifend arcu. Quisque adipiscing sen
sit amet, consectetur adipiscing elit. Morbi auctor porttitor lacus
viverra. In sed tempor urna, vitae viverra quam. Cras ultricies ai
lacus sem, ac vulputate lacus rhoncus in. Donec mi velit, moles

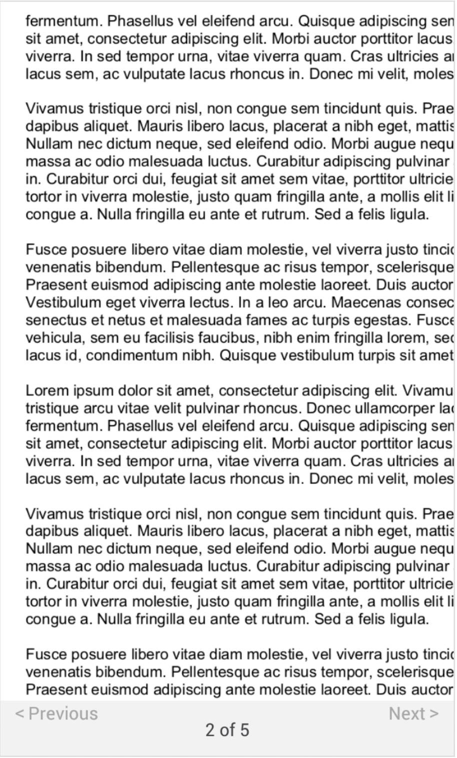
Vivamus tristique orci nisl, non congue sem tincidunt quis. Prae
dapibus aliquet. Mauris libero lacus, placerat a nibh eget, mattis
Nullam nec dictum neque, sed eleifend odio. Morbi augue nequ
massa ac odio malesuada luctus. Curabitur adipiscing pulvinar
in. Curabitur orci dui, feugiat sit amet sem vitae, porttitor ultricie
tortor in viverra molestie, justo quam fringilla ante, a mollis elit li
congue a. Nulla fringilla eu ante et rutrum. Sed a felis ligula.

Fusce posuere libero vitae diam molestie, vel viverra justo tincic
venenatis bibendum. Pellentesque ac risus tempor, scelerisque
Praesent euismod adipiscing ante molestie laoreet. Duis auctor
Vestibulum eget viverra lectus. In a leo arcu. Maecenas consec
senectus et netus et malesuada fames ac turpis egestas. Fusce
vehicula, sem eu facilisis faucibus, nibh enim fringilla lorem, sec
lacus id, condimentum nibh. Quisque vestibulum turpis sit amet

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamu
tristique arcu vitae velit pulvinar rhoncus. Donec ullamcorper lac
fermentum. Phasellus vel eleifend arcu. Quisque adipiscing sen
sit amet, consectetur adipiscing elit. Morbi auctor porttitor lacus
viverra. In sed tempor urna, vitae viverra quam. Cras ultricies ai
lacus sem, ac vulputate lacus rhoncus in. Donec mi velit, moles

Vivamus tristique orci nisl, non congue sem tincidunt quis. Prae
dapibus aliquet. Mauris libero lacus, placerat a nibh eget, mattis
Nullam nec dictum neque, sed eleifend odio. Morbi augue nequ
massa ac odio malesuada luctus. Curabitur adipiscing pulvinar
in. Curabitur orci dui, feugiat sit amet sem vitae, porttitor ultricie
tortor in viverra molestie, justo quam fringilla ante, a mollis elit li
congue a. Nulla fringilla eu ante et rutrum. Sed a felis ligula.

Fusce posuere libero vitae diam molestie, vel viverra justo tincic



fermentum. Phasellus vel eleifend arcu. Quisque adipiscing sen
sit amet, consectetur adipiscing elit. Morbi auctor porttitor lacus
viverra. In sed tempor urna, vitae viverra quam. Cras ultricies ai
lacus sem, ac vulputate lacus rhoncus in. Donec mi velit, moles

Vivamus tristique orci nisl, non congue sem tincidunt quis. Prae
dapibus aliquet. Mauris libero lacus, placerat a nibh eget, mattis
Nullam nec dictum neque, sed eleifend odio. Morbi augue nequ
massa ac odio malesuada luctus. Curabitur adipiscing pulvinar
in. Curabitur orci dui, feugiat sit amet sem vitae, porttitor ultricie
tortor in viverra molestie, justo quam fringilla ante, a mollis elit li
congue a. Nulla fringilla eu ante et rutrum. Sed a felis ligula.

Fusce posuere libero vitae diam molestie, vel viverra justo tincic
venenatis bibendum. Pellentesque ac risus tempor, scelerisque
Praesent euismod adipiscing ante molestie laoreet. Duis auctor
Vestibulum eget viverra lectus. In a leo arcu. Maecenas consec
senectus et netus et malesuada fames ac turpis egestas. Fusce
vehicula, sem eu facilisis faucibus, nibh enim fringilla lorem, sec
lacus id, condimentum nibh. Quisque vestibulum turpis sit amet

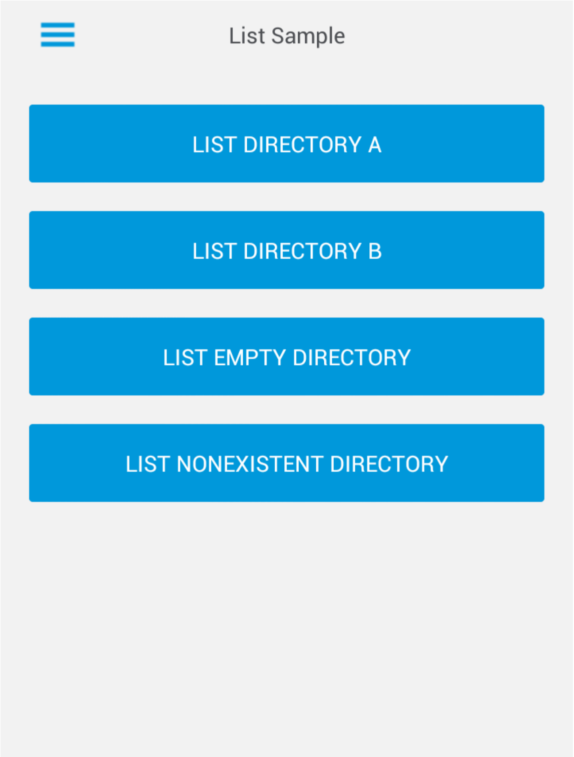
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamu
tristique arcu vitae velit pulvinar rhoncus. Donec ullamcorper lac
fermentum. Phasellus vel eleifend arcu. Quisque adipiscing sen
sit amet, consectetur adipiscing elit. Morbi auctor porttitor lacus
viverra. In sed tempor urna, vitae viverra quam. Cras ultricies ai
lacus sem, ac vulputate lacus rhoncus in. Donec mi velit, moles

Vivamus tristique orci nisl, non congue sem tincidunt quis. Prae
dapibus aliquet. Mauris libero lacus, placerat a nibh eget, mattis
Nullam nec dictum neque, sed eleifend odio. Morbi augue nequ
massa ac odio malesuada luctus. Curabitur adipiscing pulvinar
in. Curabitur orci dui, feugiat sit amet sem vitae, porttitor ultricie
tortor in viverra molestie, justo quam fringilla ante, a mollis elit li
congue a. Nulla fringilla eu ante et rutrum. Sed a felis ligula.

Fusce posuere libero vitae diam molestie, vel viverra justo tincic
venenatis bibendum. Pellentesque ac risus tempor, scelerisque
Praesent euismod adipiscing ante molestie laoreet. Duis auctor

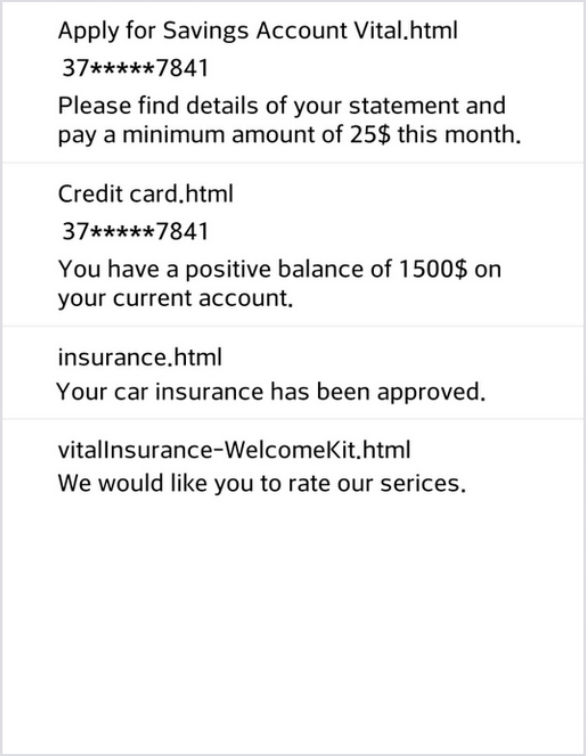
2.3 List Sample

If you go to List Sample, you get the following screen.



List Directory A

After touching the **List Directory A** button, you get properties (metadata) of all files contained in directory A (in the GMC repository).



Apply for Savings Account Vital.html
37*****7841
Please find details of your statement and pay a minimum amount of 25$ this month.

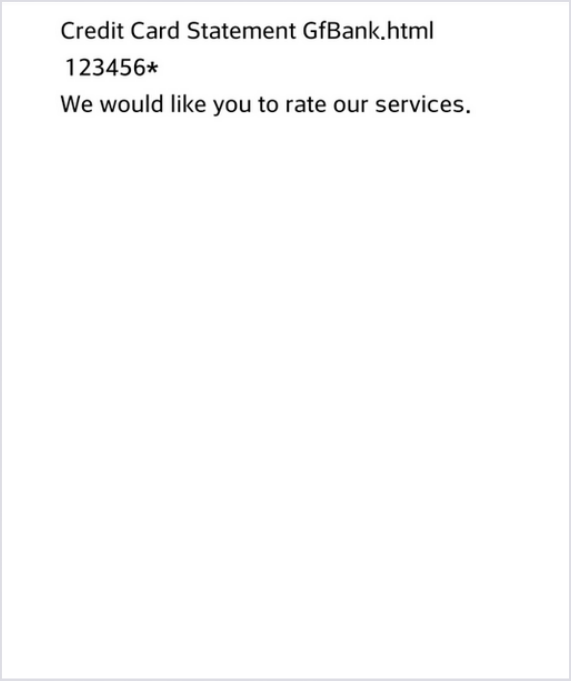
Credit card.html
37*****7841
You have a positive balance of 1500$ on your current account.

insurance.html
Your car insurance has been approved.

vitalInsurance-WelcomeKit.html
We would like you to rate our serices.

List Directory B

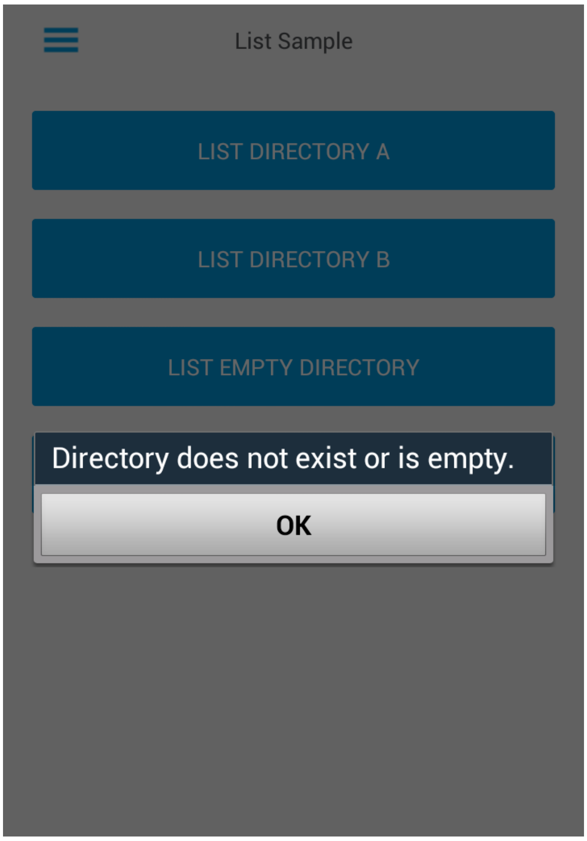
After touching the **List Directory B** button, you get properties (metadata) of the files contained in directory B (in the GMC repository).



Credit Card Statement GfBank.html
123456*
We would like you to rate our services.

List Empty Directory / List Nonexistent Directory

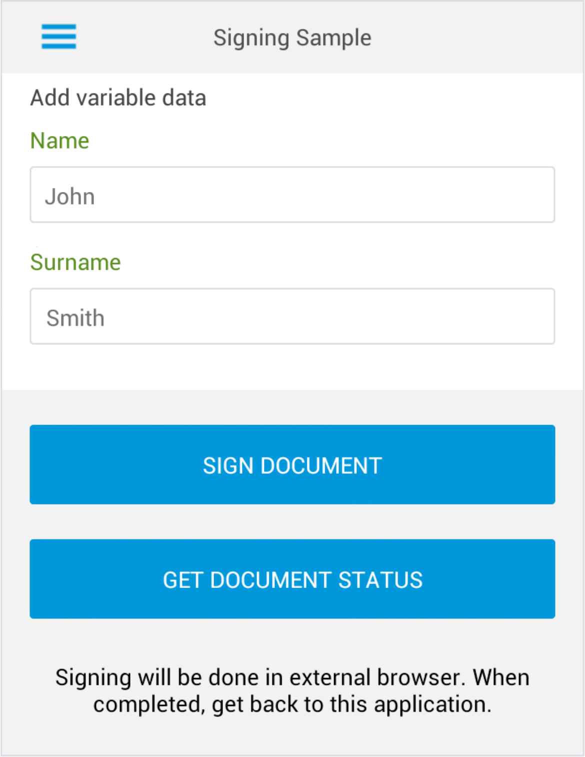
After touching the **List Empty Directory** or **List Nonexistent Directory** button, you get the following message on the screen.



Directory does not exist or is empty.

2.4 Signing Sample

If you go to Signing Sample, you get the following screen.



Signing Sample

Add variable data

Name

John

Surname

Smith

SIGN DOCUMENT

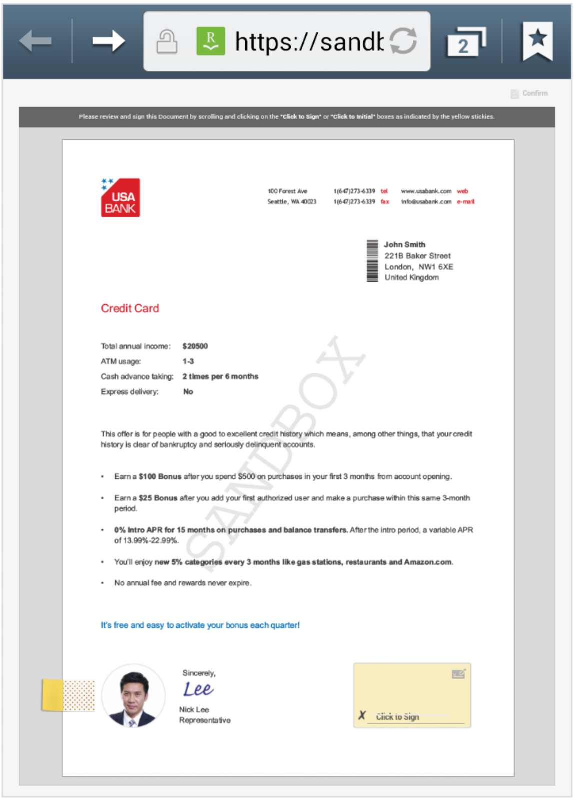
GET DOCUMENT STATUS

Signing will be done in external browser. When completed, get back to this application.

You can edit the **Name** and **Surname** text fields. Provided values will be included in the output document.

Sign Document

After touching the **Sign Document** button, you get the output document ready to be electronically signed.



Please review and sign this Document by scrolling and clicking on the **"Click to Sign"** or **"Click to Initial"** boxes as indicated by the yellow stickies.

USA BANK

100 Forest Ave
Seattle, WA 40023

1(647)273-6339 tel
1(647)273-6339 fax

www.usabank.com web
info@usabank.com e-mail

John Smith
221B Baker Street
London, NW1 6XE
United Kingdom

Credit Card

Total annual income:	**$20500**
ATM usage:	**1-3**
Cash advance taking:	**2 times per 6 months**
Express delivery:	**No**

This offer is for people with a good to excellent credit history which means, among other things, that your credit history is clear of bankruptcy and seriously delinquent accounts.

- Earn a **$100 Bonus** after you spend $500 on purchases in your first 3 months from account opening.
- Earn a **$25 Bonus** after you add your first authorized user and make a purchase within this same 3-month period.
- **0% Intro APR for 15 months on purchases and balance transfers.** After the intro period, a variable APR of 13.99%-22.99%.
- You'll enjoy **new 5% categories every 3 months like gas stations, restaurants and Amazon.com.**
- No annual fee and rewards never expire.

It's free and easy to activate your bonus each quarter!

Sincerely,
Lee
Nick Lee
Representative

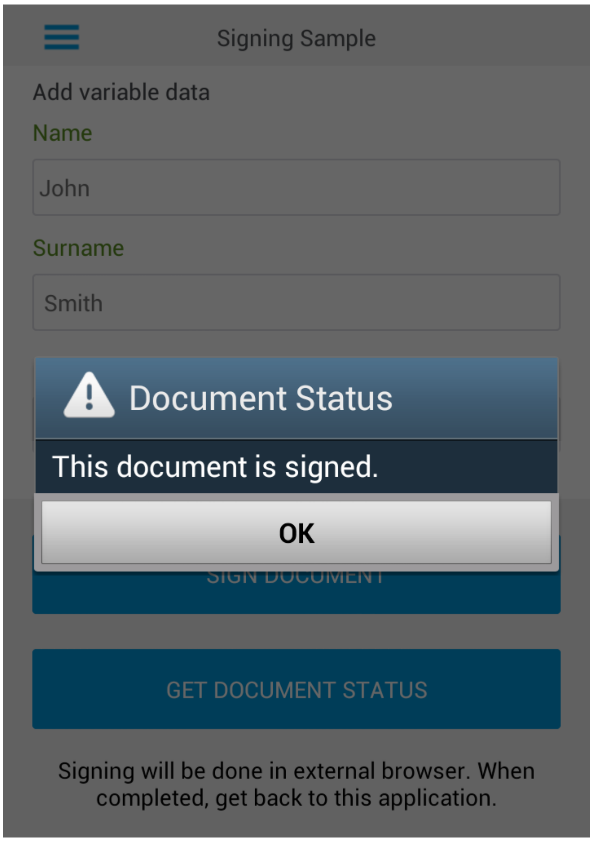
X Click to Sign

SANDBOX

The document is displayed in an external browser.

Get Document Status

After touching the **Get Document Status** button, you are informed whether the document has already been signed.



Signing Sample

Add variable data

Name: John

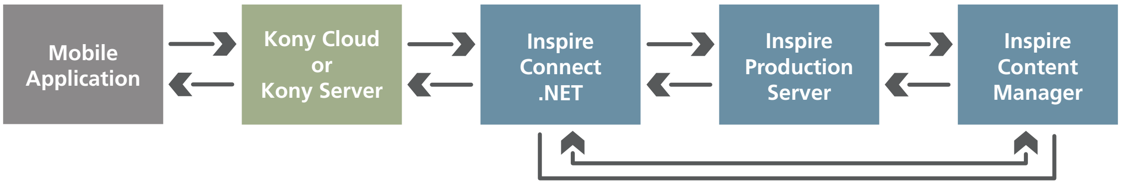
Surname: Smith

Document Status

This document is signed.

Signing will be done in external browser. When completed, get back to this application.

3 Architecture



These components are integrated:

- Mobile Application – Mobile application developed in Kony Studio and installed on a mobile device.
- Kony Cloud / Kony Server – Kony Cloud or a Kony Server installation.
- Inspire Connect .NET – Web application that integrates with Inspire Production Server and Inspire Content Manager services. You can learn more about it in the appendix: Inspire Connect .NET.
- Inspire Production Server – Production console of Inspire Designer, responsible for the document composition. You can learn more about it in the appendix: Inspire Production Server.
- Inspire Content Manager – GMC storage containing WFD files and sample directories. You can learn more about it in the appendix: Inspire Content Manager.

Communication is based on requests and responses:

1. A request is initiated from the mobile application where you can enter input data, e.g. client information.
2. The request is sent to Kony Server (or Kony Cloud) and further passed to Inspire Connect .NET and Inspire Production Server.
3. Design of the output documents is predefined by a WFD file saved in Inspire Content Manager.
4. Output document in the required format (PDF, HTML, image, etc.) is generated and its URL is passed back in a response (from Inspire Production Server via Inspire Connect .NET and Kony Server/Cloud) to the GMC Showcase application where the document is displayed (on the same mobile device).

These sections describe the stages of the setup that programmers should follow:

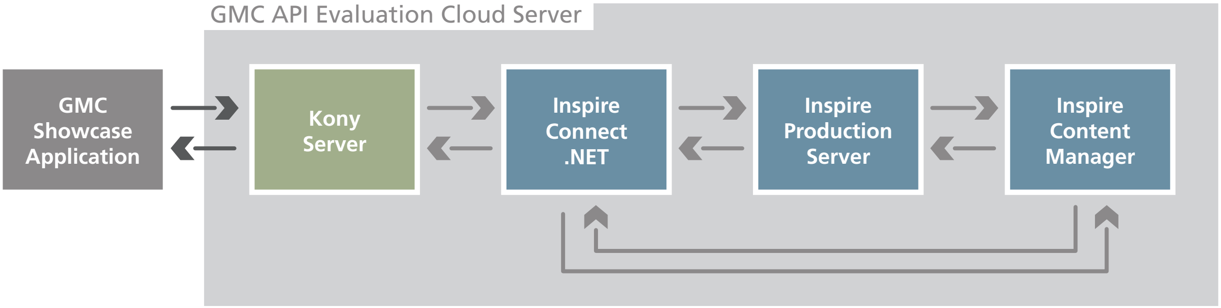
1. Default Demo Installations in GMC Cloud
2. Your Own Kony Cloud
3. Your Own Environment

3.1 Default Demo Installations in GMC Cloud

At this stage, programmers:

- Use demo installations of Kony Server and of the GMC applications (provided on a GMC Cloud server).
- Verify communication between the GMC Showcase application and Kony Server.

Step-by-step instructions for setting up this stage are in the Default Demo Installations in GMC Cloud section.



This architecture is ready to be used after you download the GMC Showcase application project from GitHub.

Programmers need to have Kony Studio (for the purpose of examining the project, a trial version is sufficient).

The GMC API Evaluation Cloud Server runs on the following address: *http://api.gmc.net/mobileapi* and contains demo installations and configurations of all necessary components.

- Kony Server uses services (defined in the project imported to Kony Studio) to call Inspire Connect .NET.
- Inspire Connect .NET uses its methods to call Inspire Production Server.
- Inspire Production Server uses demo WFD files that are stored in Inspire Content Manager. The files define the design of the output documents.

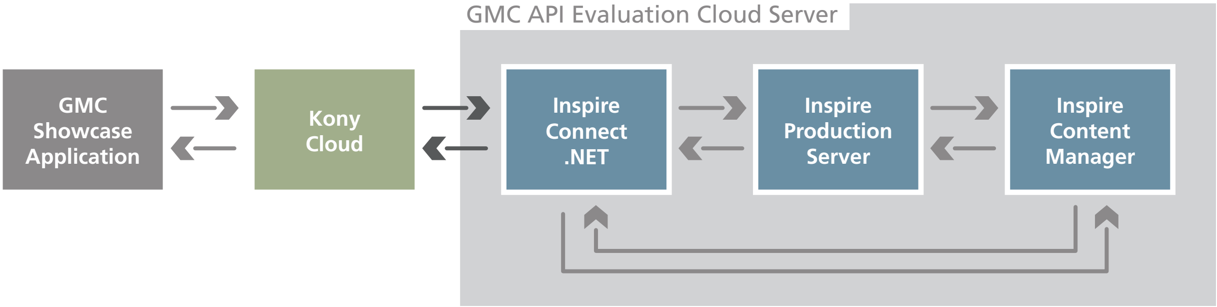
No customization is allowed at this stage.

3.2 Your Own Kony Cloud

At this stage, programmers:

- Use their own Kony Cloud and demo installations of the GMC applications (provided on a GMC Cloud server).
- Verify communication between Kony Cloud and the GMC Inspire Connect .NET.

Step-by-step instructions for setting up this stage are in the Your Own Kony Cloud section.



Programmers need to have Kony Studio and an account on Kony Cloud (for the testing purposes, a trial version of both is sufficient).

> **Note** Depending on your specific needs, you can also use an internal Kony Server (installed on your own server) instead of Kony Cloud, e.g. if your mobile application is intended only for internal users within an intranet or if you need to use web services specific for internal Kony servers.

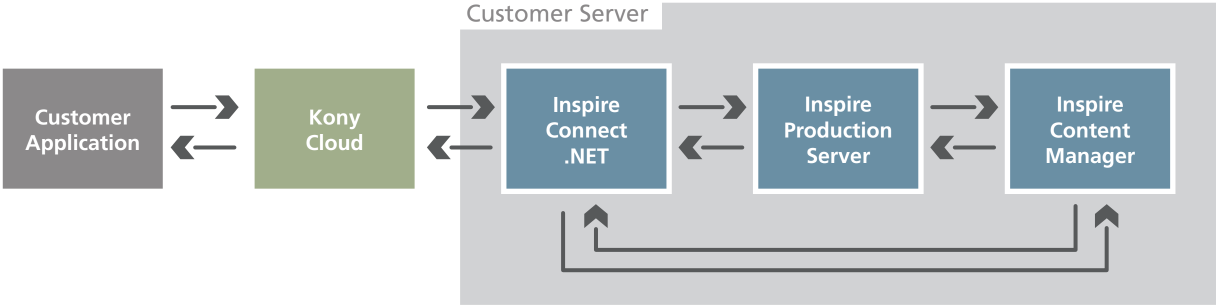
At this stage, you can customize parameters of the services offered by the GMC mobile API. For example, you can change the output format to Docx or use a different output module than the one defined by default.

3.3 Your Own Environment

At this stage, programmers:

- Use their own Kony Cloud (or Kony Server), as well as your own installations of the GMC applications. As all components are in your environment, you can modify them according to your needs.
- Verify communication among all customized components.

Step by step instructions for setting up this stage are in the Your Own Environment section.



Programmers need to have Kony Studio and an account on Kony Cloud (for the testing purposes, a trial version of both is sufficient) and their own installations of the GMC Inspire applications (for the testing purposes, demo versions are sufficient).

> **Note** Like in the previous stage, you can use an internal Kony Server (installed on your own server) instead of Kony Cloud.

At this stage, you can create your own design of documents or implement your own processing logic. You need to refer to documentation of the GMC Inspire applications.

4 Setting Up

4.1 Default Demo Installations in GMC Cloud

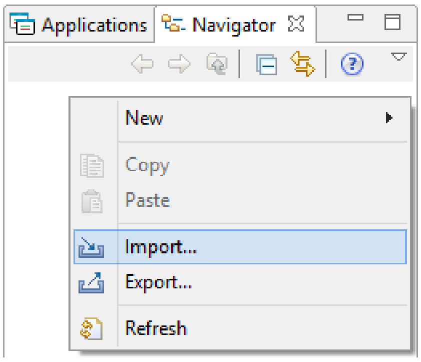
These are instructions for setting up the stage described in the Default Demo Installations in GMC Cloud section.

Prerequisites:

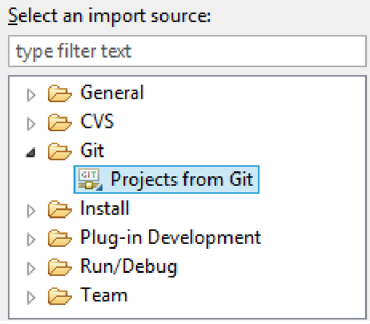
- Get Kony Studio. For the purpose of examining the project, a trial version is sufficient.

Follow these steps:

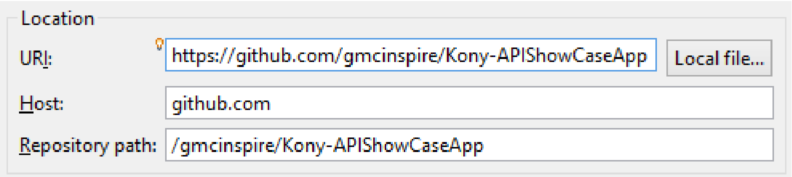
1. Launch Kony Studio.
2. Import the whole project of the GMC Showcase application from GitHub (in the GmcApiShowcase folder) to Kony Studio. You can either create a new empty workspace or import the project to an existing workspace.
    a. Right-click in the Navigator tab and select **Import**.



    b. Select *Projects from Git* as the import source.



    c. Click on the **Clone** button and then specify the project location on the GitHub.



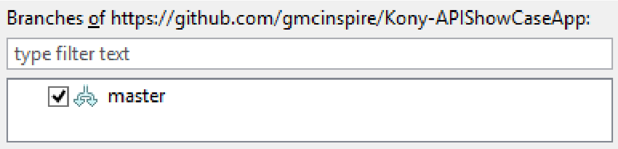
Location

URI: https://github.com/gmcinspire/Kony-APIShowCaseApp

Host: github.com

Repository path: /gmcinspire/Kony-APIShowCaseApp

d. Select the master branch.

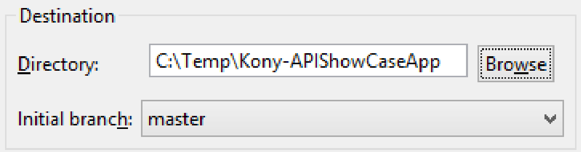


Branches of https://github.com/gmcinspire/Kony-APIShowCaseApp:

type filter text

☑ master

e. Specify a local storage location for the project.

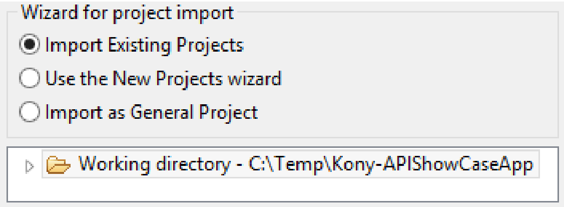


Destination

Directory: C:\Temp\Kony-APIShowCaseApp

Initial branch: master

f. Select *Import Existing Projects*.



Wizard for project import

● Import Existing Projects

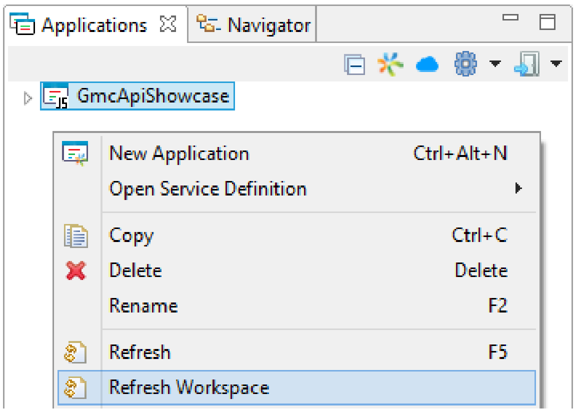
○ Use the New Projects wizard

○ Import as General Project

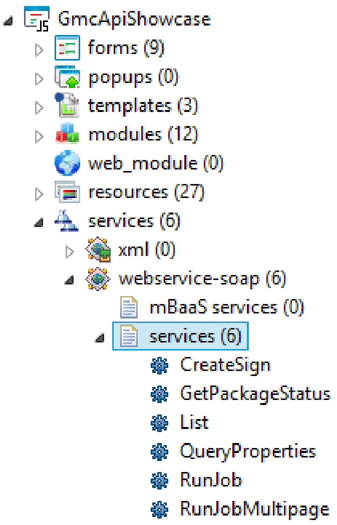
Working directory - C:\Temp\Kony-APIShowCaseApp

3. Refresh the workspace and check the imported services.

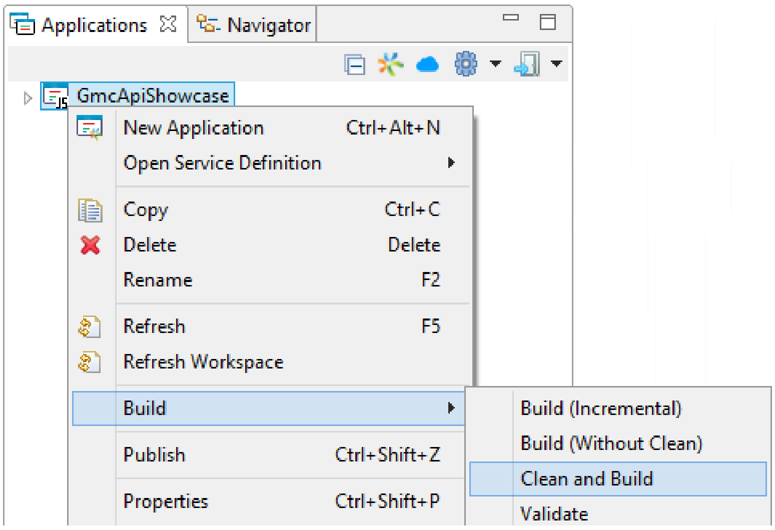
a. Right-click in the Applications tab and select **Refresh Workspace**.



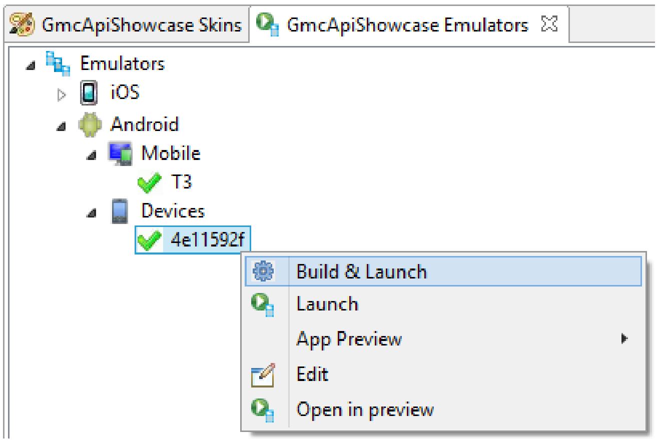
b. Check the services pre-defined in the GmcApiShowcase project.



4. Build the project: Right-click on the project and select **Build | Clean and build**.



5. You can build the GMC Showcase application to your mobile device: Connect the mobile device to your computer and select the device in the list of emulators. Right-click on it and select **Build & Launch**.



4.2 Your Own Kony Cloud

These are instructions for setting up the stage described in the Your Own Kony Cloud section.

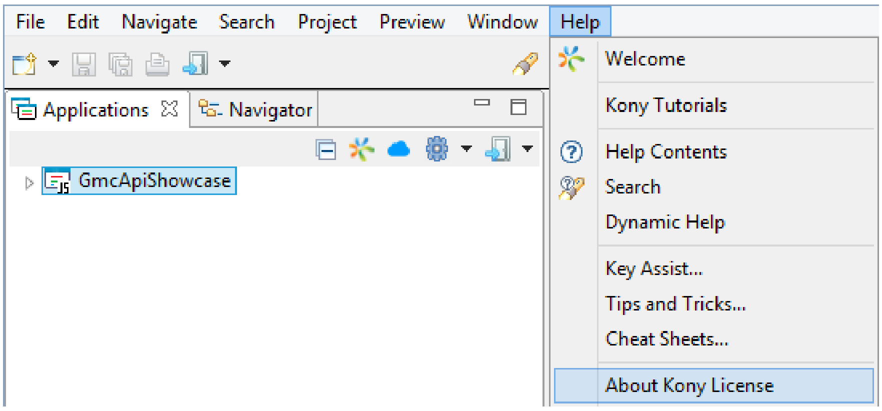
Prerequisites:

- Get Kony Studio and an account on Kony Cloud. For the testing purposes, a trial version of both is sufficient.
- If you use an internal Kony Server instead of Kony Cloud, make sure the server is running.
- Go through steps in the previous stage.

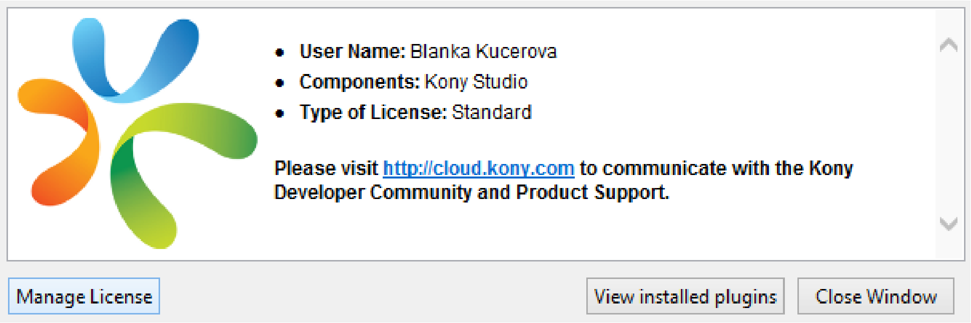
Follow these steps:

1. Enter your license information.

   a. Select **Help | About Kony License**.



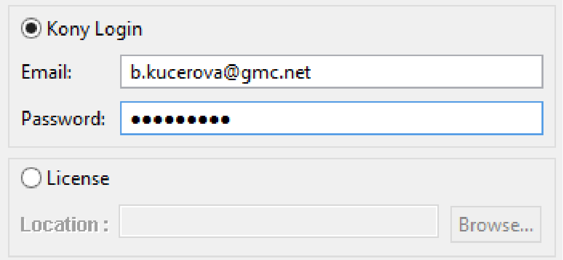
   b. Click on the **Manage License** button.



- **User Name:** Blanka Kucerova
- **Components:** Kony Studio
- **Type of License:** Standard

Please visit http://cloud.kony.com to communicate with the Kony Developer Community and Product Support.

   c. Enter your license information.



● Kony Login

Email: b.kucerova@gmc.net

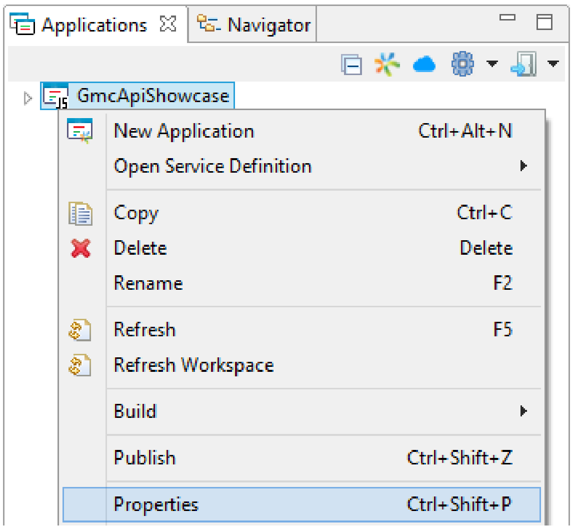
Password: •••••••••

○ License

Location :

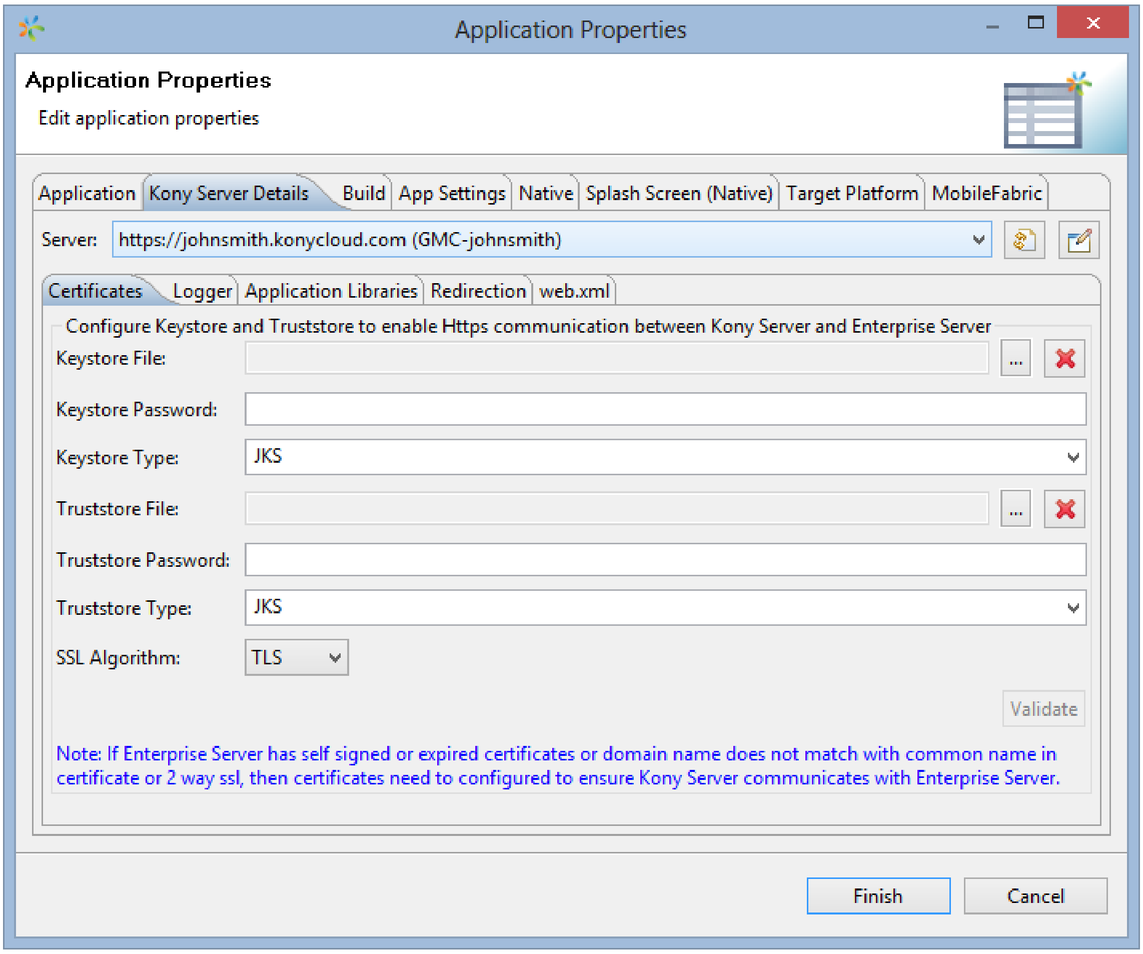
2. Set the URL of Kony Cloud.

   a. Right-click on the GmcApiShowcase project and select **Properties**.



b. In the Kony Server Details tab, click on the **Refresh** icon. In the **Server** combo-box, select your URL on Kony Cloud.

> **Note** If you use an internal Kony Server, click on the **Edit** icon and add the server. Then select it in the combo-box.



Server: https://johnsmith.konycloud.com (GMC-johnsmith)

Configure Keystore and Truststore to enable Https communication between Kony Server and Enterprise Server

Keystore File:

Keystore Password:

Keystore Type: JKS

Truststore File:

Truststore Password:

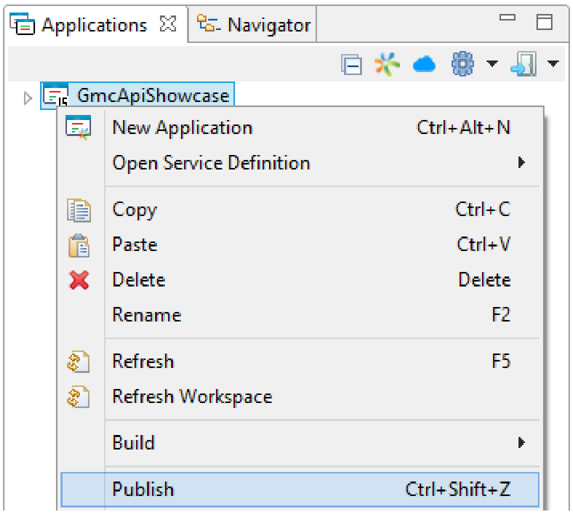
Truststore Type: JKS

SSL Algorithm: TLS

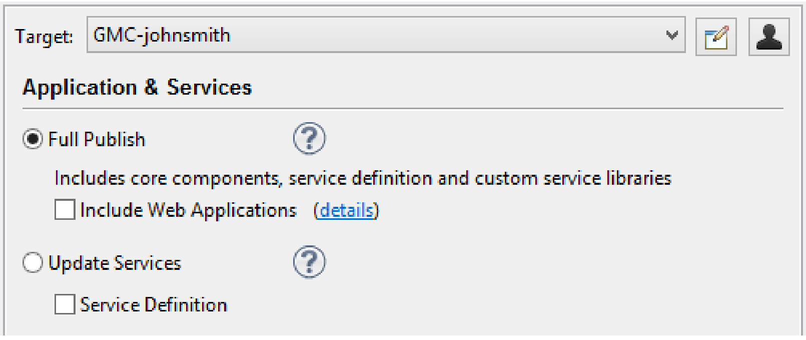
Note: If Enterprise Server has self signed or expired certificates or domain name does not match with common name in certificate or 2 way ssl, then certificates need to configured to ensure Kony Server communicates with Enterprise Server.

3. Publish the services to Kony Cloud.

   a. Right-click on the GmcApiShowcase project and select **Publish**.



b. In the **Target** field, select your account name on Kony Cloud. Below, select either *Full Publish* (e.g. if you publish the project for the first time) or *Update Services* (if you only want to replace existing services).



Target: GMC-johnsmith

Application & Services

● Full Publish

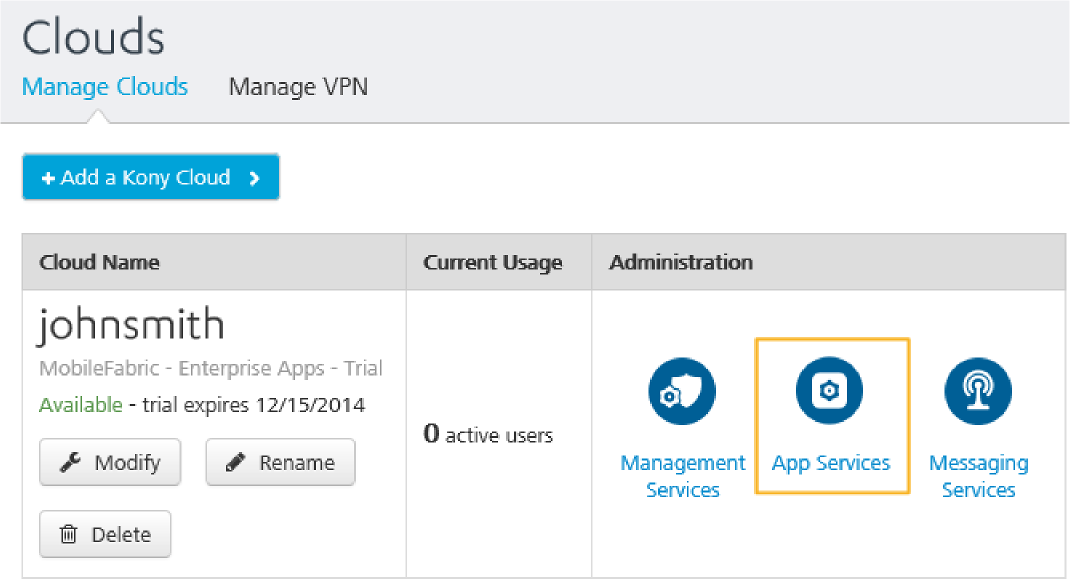
Includes core components, service definition and custom service libraries

□ Include Web Applications (details)

○ Update Services

□ Service Definition

4. Check the project in Kony Cloud: https://manage.kony.com/#/clouds/manage-clouds



Clouds

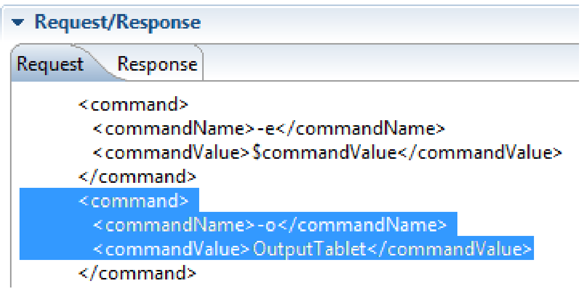
Manage Clouds Manage VPN

+ Add a Kony Cloud

Cloud Name	Current Usage	Administration
johnsmith MobileFabric - Enterprise Apps - Trial Available - trial expires 12/15/2014 Modify Rename Delete	0 active users	Management Services App Services Messaging Services

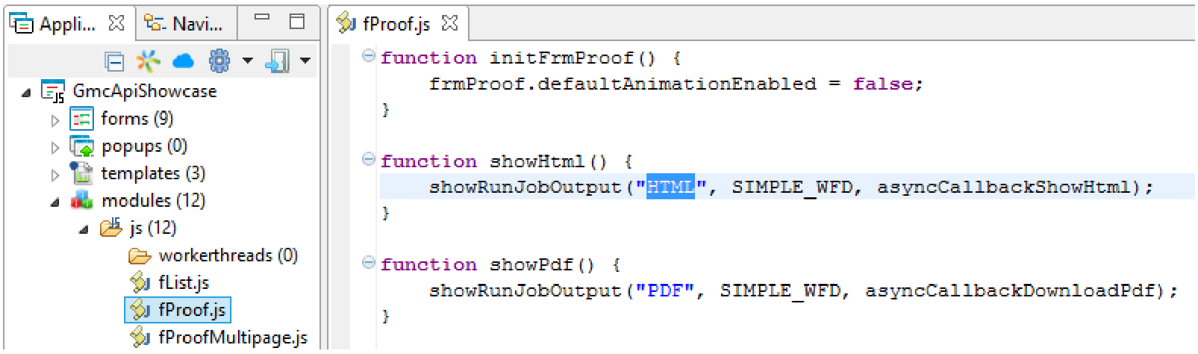
5. When you have tested the pre-defined services, you can try to modify them (i.e. change their parameters). The modified services need to be published to Kony Cloud again.

   - This example shows how you can use a different output module than the one defined by default: In the RunJob service request, change the value of the `-o` command from *OutputTablet* to *OutputMobile*. The output will display different colors in texts and in the heading.



```
<command>
  <commandName>-e</commandName>
  <commandValue>$commandValue</commandValue>
</command>
<command>
  <commandName>-o</commandName>
  <commandValue>OutputTablet</commandValue>
</command>
```

   - This example shows how you can change the output format to Docx: Go to **modules | js | fProof.js** and change the *HTML* value to *Docx*. The **Proof HTML** button will provide output in the Docx format.



```
function initFrmProof() {
    frmProof.defaultAnimationEnabled = false;
}

function showHtml() {
    showRunJobOutput("HTML", SIMPLE_WFD, asyncCallbackShowHtml);
}

function showPdf() {
    showRunJobOutput("PDF", SIMPLE_WFD, asyncCallbackDownloadPdf);
}
```

6. Like in the previous stage, you can build the GMC Showcase application to your mobile device.

4.3 Your Own Environment

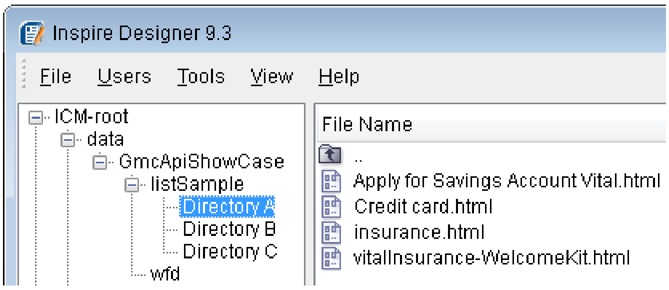
These are instructions for setting up the stage described in the Your Own Environment section.

Prerequisites:

- Get Kony Studio and an account on Kony Cloud. For the testing purposes, a trial version of both is sufficient.
- If you use an internal Kony Server instead of Kony Cloud, make sure the server is running.
- Get your own installations of the GMC Inspire applications: Inspire Connect .NET, Inspire Production Server and Inspire Content Manager. For the testing purposes, demo versions are sufficient.
- Go through steps in the two previous stages.

Follow these steps:

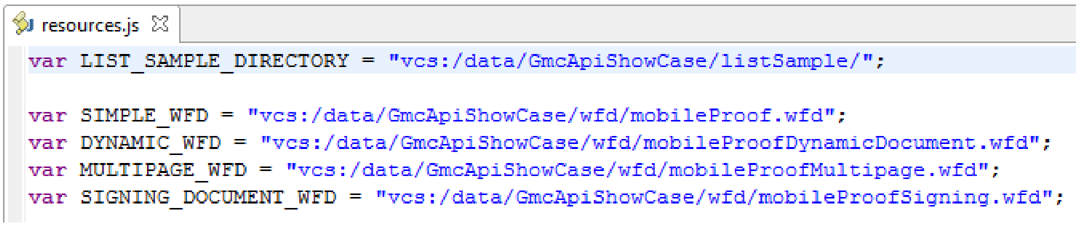
1. Download the `listSampleData.pkg` file from GitHub (in the `SolutionData` folder) and import it to your Inspire Content Manager. The package contains directories used by the List Sample, as well as the sample WFD files.



```
ICM-root
  data
    GmcApiShowCase
      listSample
        Directory A
        Directory B
        Directory C
      wfd
```

..
Apply for Savings Account Vital.html
Credit card.html
insurance.html
vitalInsurance-WelcomeKit.html

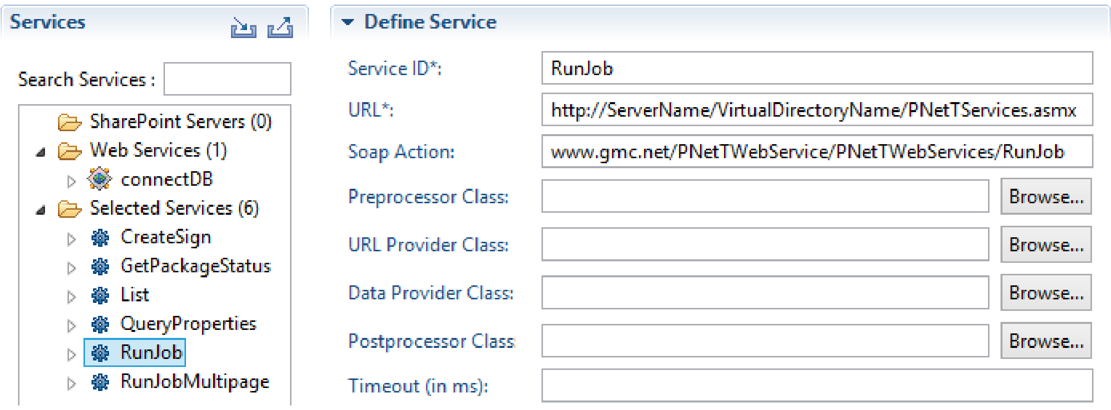
2. In Kony Studio, go to **modules | js | resources.js** and set the ICM paths to:
   - The folder with directories used in the List Sample.
   - The WFD files.



```
var LIST_SAMPLE_DIRECTORY = "vcs:/data/GmcApiShowCase/listSample/";

var SIMPLE_WFD = "vcs:/data/GmcApiShowCase/wfd/mobileProof.wfd";
var DYNAMIC_WFD = "vcs:/data/GmcApiShowCase/wfd/mobileProofDynamicDocument.wfd";
var MULTIPAGE_WFD = "vcs:/data/GmcApiShowCase/wfd/mobileProofMultipage.wfd";
var SIGNING_DOCUMENT_WFD = "vcs:/data/GmcApiShowCase/wfd/mobileProofSigning.wfd";
```

3. Double-click on each of the pre-defined services. In the Define Service section, set the URL of the web service in your Inspire Connect .NET.



Search Services :

- SharePoint Servers (0)
- Web Services (1)
  - connectDB
- Selected Services (6)
  - CreateSign
  - GetPackageStatus
  - List
  - QueryProperties
  - RunJob
  - RunJobMultipage

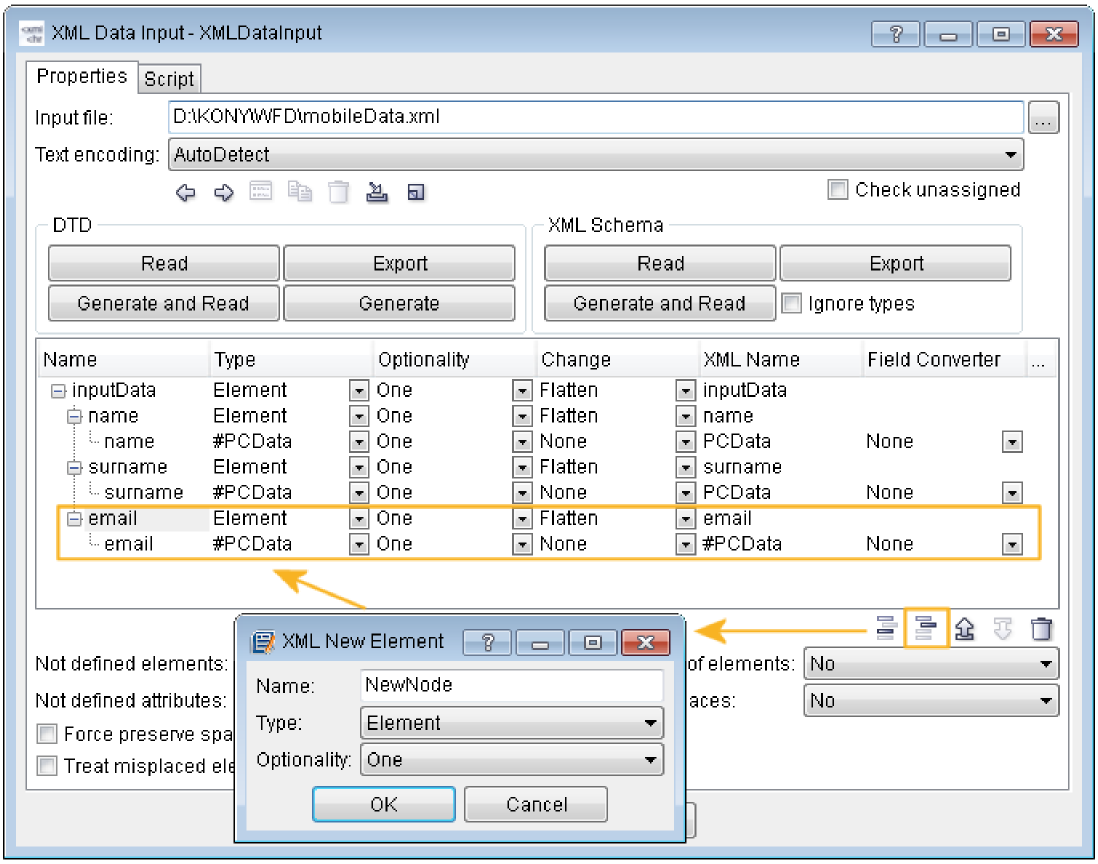
Service ID*: RunJob
URL*: http://ServerName/VirtualDirectoryName/PNetTServices.asmx
Soap Action: www.gmc.net/PNetTWebService/PNetTWebServices/RunJob
Preprocessor Class:
URL Provider Class:
Data Provider Class:
Postprocessor Class
Timeout (in ms):

4. When you have tested the pre-defined services, you can modify them (i.e. change their parameters) or define your own services. Study the Inspire Connect .NET Programming Guide ↗ to learn about all the possibilities.

5. When you have tested the sample WFD files, you can modify them. Study the Inspire Designer User Manual ↗ to learn about all the possibilities.

   This example shows how you can modify the `mobileProof.wfd` file and the GmcApiShowcase project to add a new text field for email address:

   a. In the WFD file, open the XML Data Input module. For the InputData element, create the email sub-element with the #PCData type.



Input file: D:\KONY\WFD\mobileData.xml

Text encoding: AutoDetect

□ Check unassigned

□ Ignore types

Name	Type	Optionality	Change	XML Name	Field Converter
⊟ inputData	Element	One	Flatten	inputData	
⊟ name	Element	One	Flatten	name	
name	#PCData	One	None	PCData	None
⊟ surname	Element	One	Flatten	surname	
surname	#PCData	One	None	PCData	None
⊟ email	Element	One	Flatten	email	
email	#PCData	One	None	#PCData	None

Not defined elements:

Not defined attributes:

□ Force preserve spa

□ Treat misplaced ele

of elements: No

aces: No

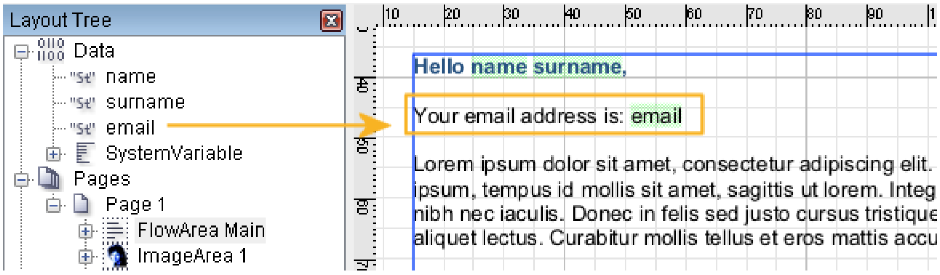
XML New Element

Name: NewNode

Type: Element

Optionality: One

   b. Open the Layout module and add the email variable (and a text) to the main flow area.



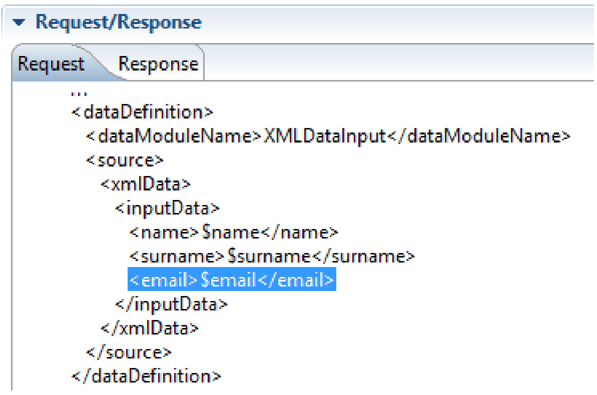
Hello name surname,

Your email address is: email

Lorem ipsum dolor sit amet, consectetur adipiscing elit. ipsum, tempus id mollis sit amet, sagittis ut lorem. Integ nibh nec iaculis. Donec in felis sed justo cursus tristique aliquet lectus. Curabitur mollis tellus et eros mattis accu

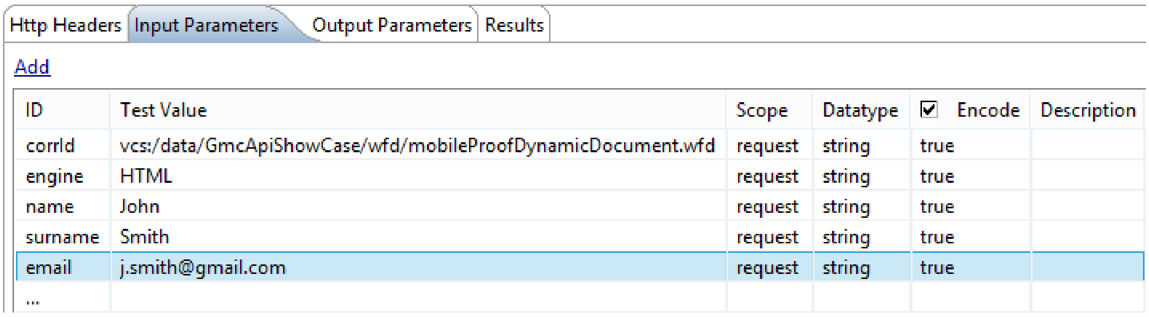
   Save the modified WFD file.

   c. In Kony Studio, open the RunJob service. Add a new line to the request (an element with a variable): *<email>$email</email>*.



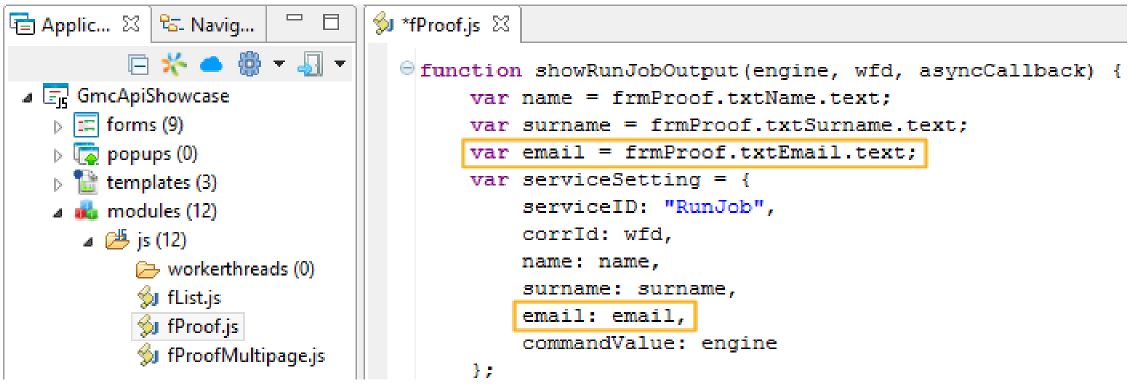
```
...
<dataDefinition>
  <dataModuleName>XMLDataInput</dataModuleName>
  <source>
    <xmlData>
      <inputData>
        <name>$name</name>
        <surname>$surname</surname>
        <email>$email</email>
      </inputData>
    </xmlData>
  </source>
</dataDefinition>
```

d. Add the email input parameter for the RunJob service.



ID	Test Value	Scope	Datatype	☑ Encode	Description
corrId	vcs:/data/GmcApiShowCase/wfd/mobileProofDynamicDocument.wfd	request	string	true	
engine	HTML	request	string	true	
name	John	request	string	true	
surname	Smith	request	string	true	
email	j.smith@gmail.com	request	string	true	
...					

e. In the source code which calls the RunJob service (i.e. in modules | js | `fProof.js`), add the email variable and parameter.



```
function showRunJobOutput(engine, wfd, asyncCallback) {
    var name = frmProof.txtName.text;
    var surname = frmProof.txtSurname.text;
    var email = frmProof.txtEmail.text;
    var serviceSetting = {
        serviceID: "RunJob",
        corrId: wfd,
        name: name,
        surname: surname,
        email: email,
        commandValue: engine
    };
```

6. Publish the modified services to Kony Cloud.

7. Build your own mobile application to your mobile device.

5 GMC Mobile API

The GMC Mobile API provides these functionalities on mobile devices:

- **Document formatting** – see the Proof Sample and the Multipage Proof Sample.

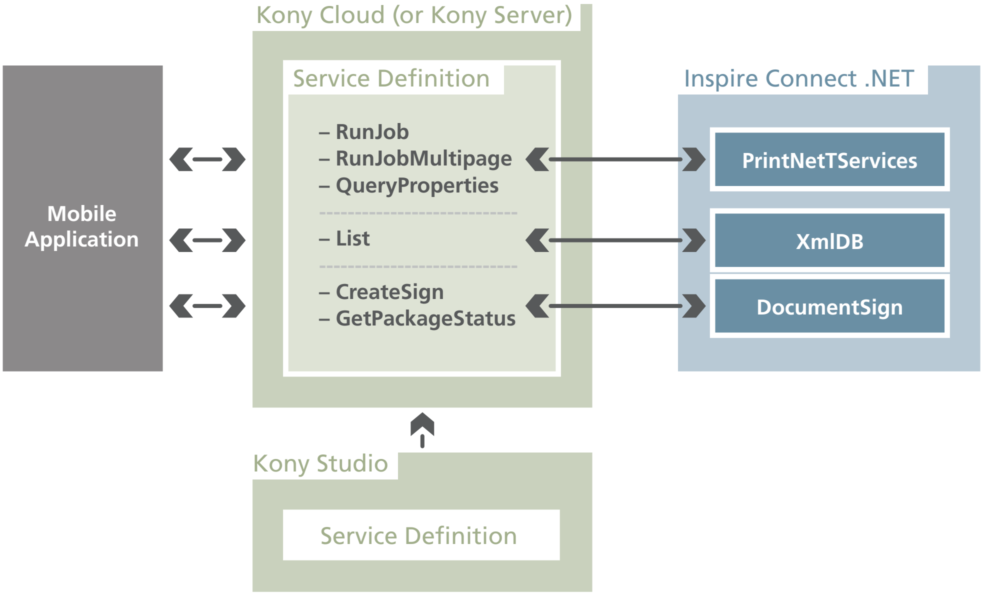
  Production of documents is provided by the RunJob, RunJobMultipage and QueryProperties services defined in Kony and is performed by the PrintNetTServices web service of Inspire Connect .NET.

- **Repository handling** – see the List Sample.

  Handling of the GMC repository (i.e. Inspire Content Manager) is provided by the List service defined in Kony and is performed by the XmlDB web service of Inspire Connect .NET.

- **Document signing** – see the Signing Sample.

  Production and handling of documents intended for electronic signing is provided by the CreateSign and GetPackageStatus services defined in Kony and is performed by the DocumentSign web service of Inspire Connect .NET.



The GMC Mobile API determines how services of a mobile project should be defined in Kony so that they are compatible with the Inspire Connect .NET web services.

The services are defined in Kony Studio and the service definition is deployed (published) to Kony Cloud (or Kony Server).

The Inspire Connect .NET services use the SOAP format of messages.

5.1 Document Formatting

5.1.1 Calling the Service Definition

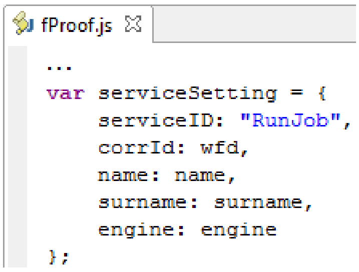
All services defined in Kony are called from the mobile device by this function:

```
kony.net.invokeServiceAsync(url, serviceSetting, callBack, info);
```

- `url` – URL of Kony Cloud or IP address of the internal Kony server.
- `serviceSetting` – String (in the JSON format) containing a list of input parameters, specific for each service.

> **Note** This parameter is called `serviceSetting` for the Proof Sample, `inputParam` for the List Sample and `serviceDefinition` for the Signing Sample.

The Proof Sample uses these input parameters:



```
...
var serviceSetting = {
    serviceID: "RunJob",
    corrId: wfd,
    name: name,
    surname: surname,
    engine: engine
};
```

- `serviceID` – Identifier (a unique name) of the service (defined in Kony) that is called.
- `corrId` – Variable specifying the path to the WFD file that is processed.
- `name` – Variable specifying the name of a variable defined in the input module of the WFD file.
- `surname` – Variable specifying the name of a variable defined in the input module of the WFD file.
- `engine` – Variable specifying the format of the output file.

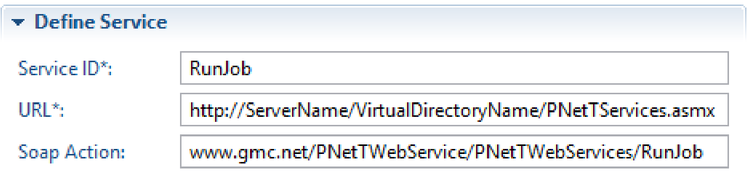
5.1.2 Service Definition in Kony

The definition of all services in Kony consists of:

- Defining which Connect .NET web service is called.
- Defining the request (with input parameters) sent to Inspire Connect .NET.
- Defining which part of the response (i.e. output parameters) is used and returned to the mobile device.

The RunJob service is defined this way:

- The RunJob service calls the PNetTServices web service of Inspire Connect .NET.

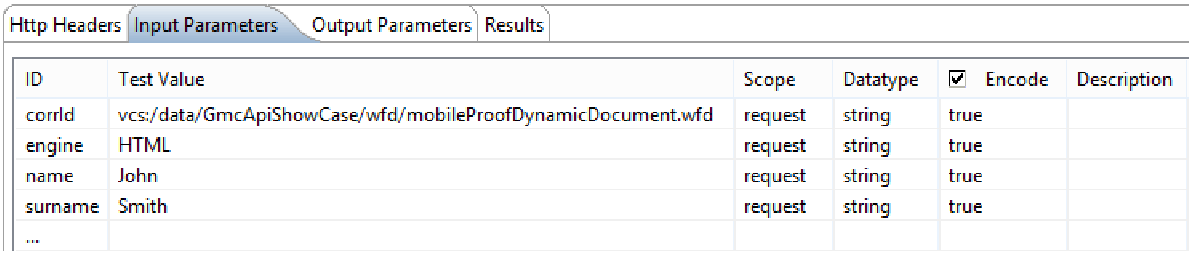


Service ID*: RunJob

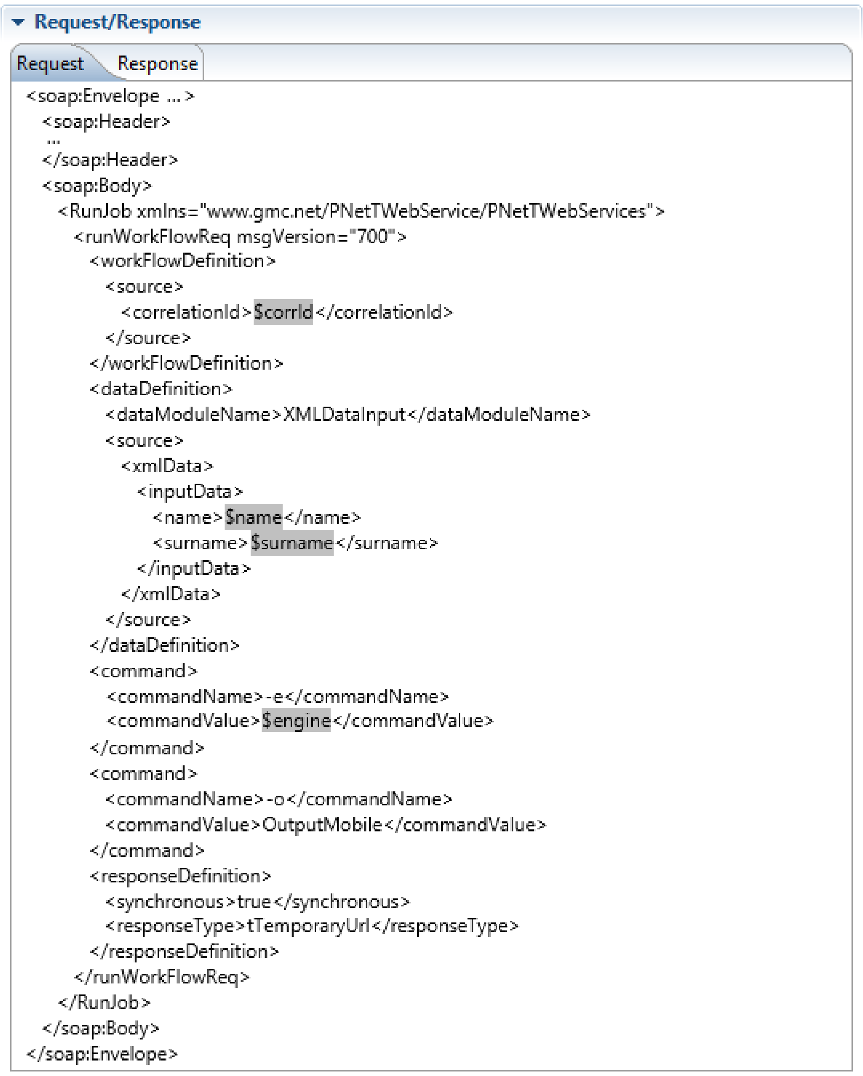
URL*: http://ServerName/VirtualDirectoryName/PNetTServices.asmx

Soap Action: www.gmc.net/PNetTWebService/PNetTWebServices/RunJob

- The RunJob service uses the following request and input parameters.

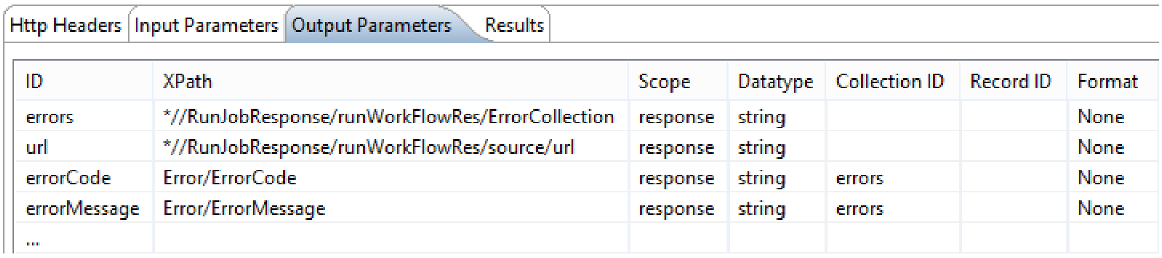


ID	Test Value	Scope	Datatype	☑ Encode	Description
corrId	vcs:/data/GmcApiShowCase/wfd/mobileProofDynamicDocument.wfd	request	string	true	
engine	HTML	request	string	true	
name	John	request	string	true	
surname	Smith	request	string	true	
...					

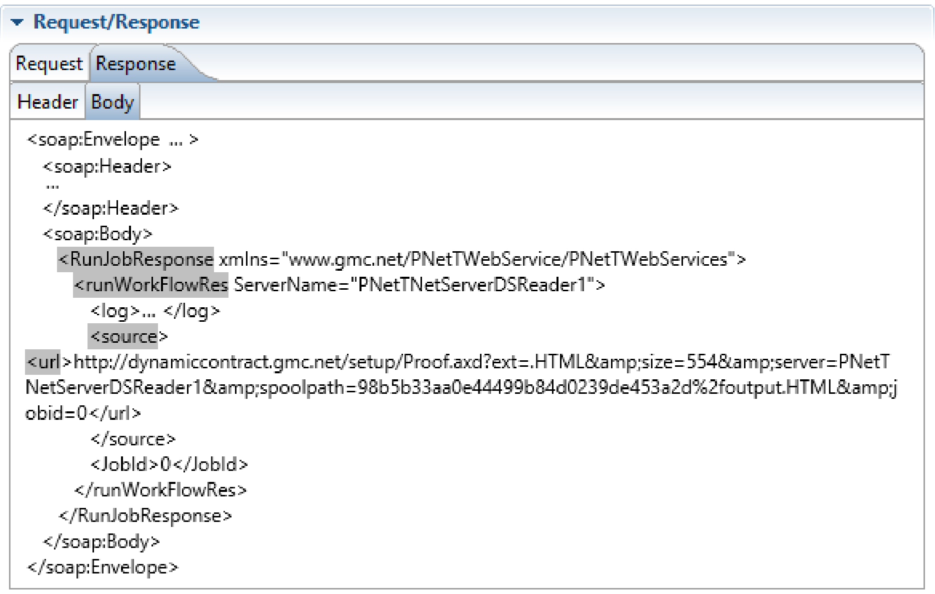


```
<soap:Envelope ...>
  <soap:Header>
    ...
  </soap:Header>
  <soap:Body>
    <RunJob xmlns="www.gmc.net/PNetTWebService/PNetTWebServices">
      <runWorkFlowReq msgVersion="700">
        <workFlowDefinition>
          <source>
            <correlationId>$corrId</correlationId>
          </source>
        </workFlowDefinition>
        <dataDefinition>
          <dataModuleName>XMLDataInput</dataModuleName>
          <source>
            <xmlData>
              <inputData>
                <name>$name</name>
                <surname>$surname</surname>
              </inputData>
            </xmlData>
          </source>
        </dataDefinition>
        <command>
          <commandName>-e</commandName>
          <commandValue>$engine</commandValue>
        </command>
        <command>
          <commandName>-o</commandName>
          <commandValue>OutputMobile</commandValue>
        </command>
        <responseDefinition>
          <synchronous>true</synchronous>
          <responseType>tTemporaryUrl</responseType>
        </responseDefinition>
      </runWorkFlowReq>
    </RunJob>
  </soap:Body>
</soap:Envelope>
```

- The RunJob service processes the following response and output parameters.



ID	XPath	Scope	Datatype	Collection ID	Record ID	Format
errors	*/RunJobResponse/runWorkFlowRes/ErrorCollection	response	string			None
url	*/RunJobResponse/runWorkFlowRes/source/url	response	string			None
errorCode	Error/ErrorCode	response	string	errors		None
errorMessage	Error/ErrorMessage	response	string	errors		None
...						



```
<soap:Envelope  ... >
  <soap:Header>
   ...
  </soap:Header>
  <soap:Body>
    <RunJobResponse xmlns="www.gmc.net/PNetTWebService/PNetTWebServices">
      <runWorkFlowRes ServerName="PNetTNetServerDSReader1">
        <log>... </log>
        <source>
<url>http://dynamiccontract.gmc.net/setup/Proof.axd?ext=.HTML&amp;size=554&amp;server=PNetT
NetServerDSReader1&amp;spoolpath=98b5b33aa0e44499b84d0239de453a2d%2foutput.HTML&amp;j
obid=0</url>
        </source>
        <JobId>0</JobId>
      </runWorkFlowRes>
    </RunJobResponse>
  </soap:Body>
</soap:Envelope>
```

5.1.3 Calling the Inspire Connect .NET Method

Production of documents is ensured by the `RunJob` method of the PrintNetTServices web service in Inspire Connect .NET.

The method can return:

- URL of the output document.
- Binary data of the output document.
- Path to the output document in Inspire Content Manager.

The document design is defined in a WFD file. You can include variables in the document.

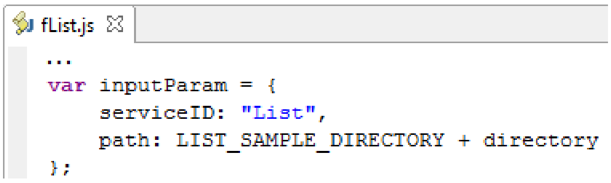
You can find all details about this method in the Inspire Connect .NET Programming Guide ↗ (RunJob Method section).

5.2 Repository Handling

5.2.1 Calling the Service Definition

All services defined in Kony are called from the mobile device by the same function. The second parameter for the List Sample is called `inputParam`.

The List Sample uses these input parameters:



fList.js

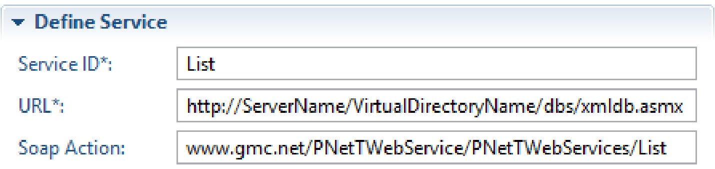
```
...
var inputParam = {
    serviceID: "List",
    path: LIST_SAMPLE_DIRECTORY + directory
};
```

- `serviceID` – Identifier (a unique name) of the service (defined in Kony) that is called.
- `path` – Variable specifying the ICM path to the folder with the sample directories (its value is defined in the `resources.js` file) + variable specifying the directory name.

5.2.2 Service Definition in Kony

The List service is defined this way:

- The List service calls the XmlDB web service of Inspire Connect .NET.



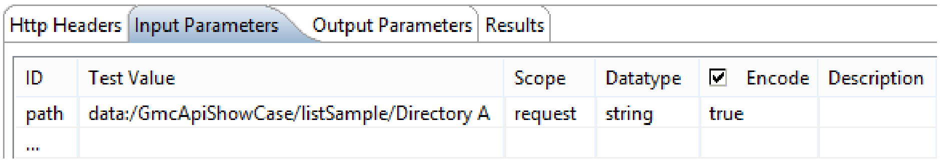
Define Service

Service ID*: List

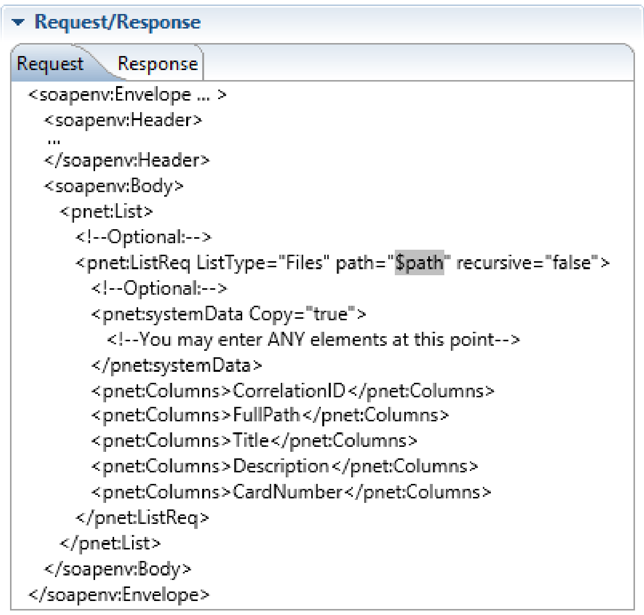
URL*: http://ServerName/VirtualDirectoryName/dbs/xmldb.asmx

Soap Action: www.gmc.net/PNetTWebService/PNetTWebServices/List

- The List service uses the following request and input parameters.



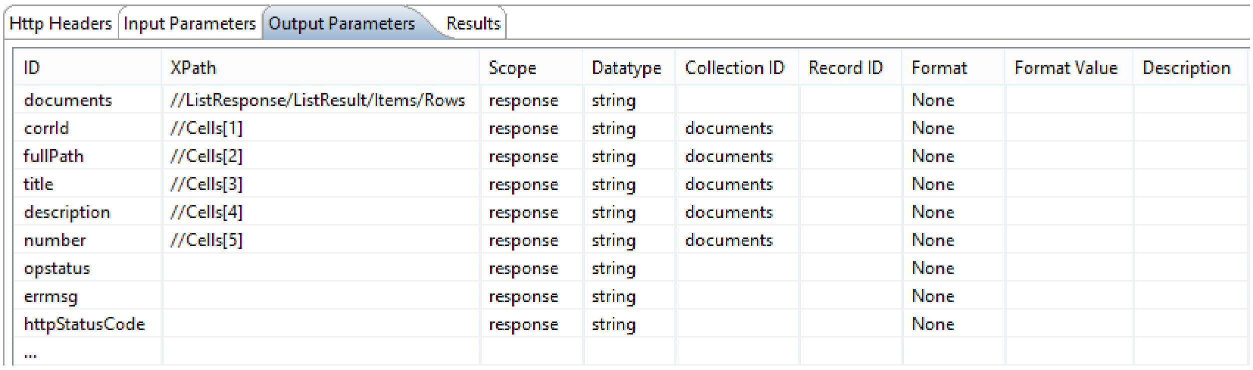
ID	Test Value	Scope	Datatype	☑ Encode	Description
path	data:/GmcApiShowCase/listSample/Directory A	request	string	true	
...					



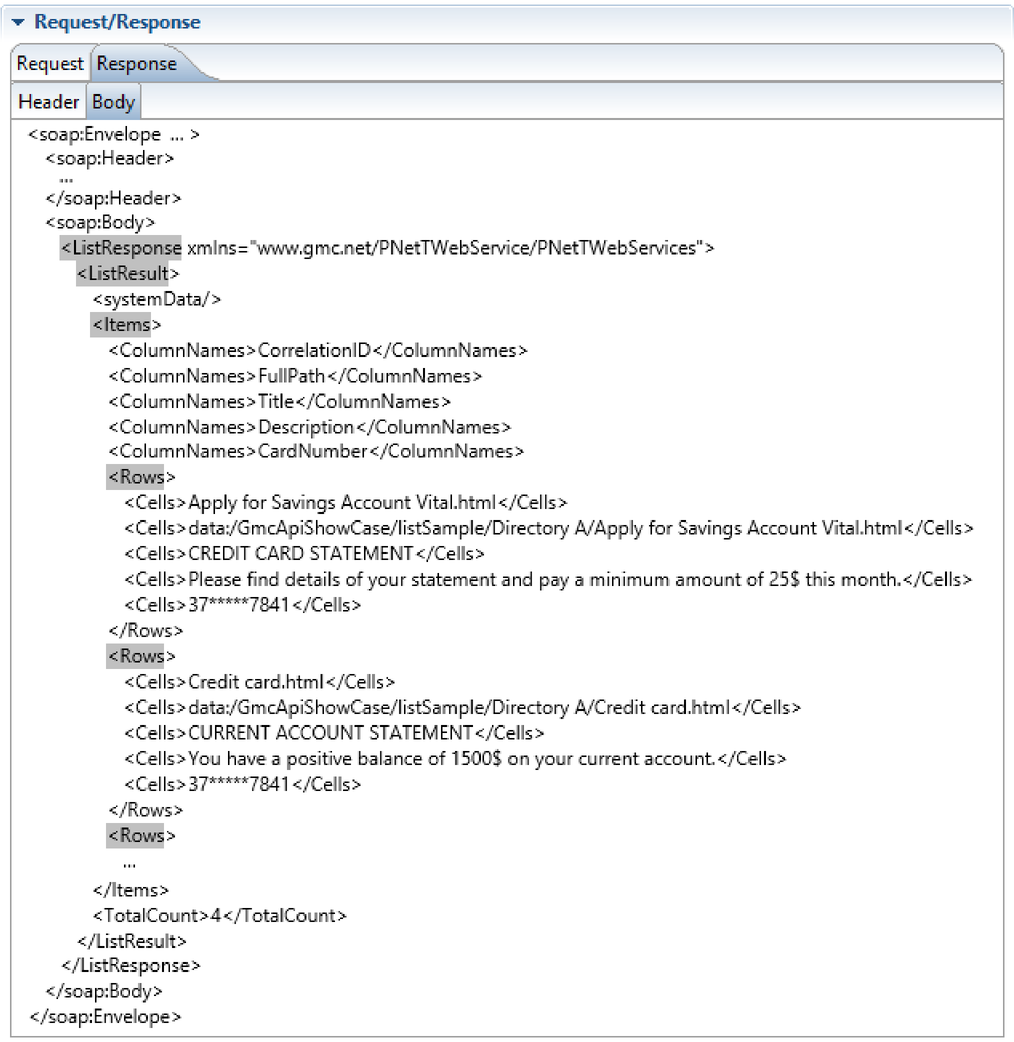
Request/Response

```
<soapenv:Envelope ... >
  <soapenv:Header>
  ...
  </soapenv:Header>
  <soapenv:Body>
    <pnet:List>
      <!--Optional:-->
      <pnet:ListReq ListType="Files" path="$path" recursive="false">
        <!--Optional:-->
        <pnet:systemData Copy="true">
          <!--You may enter ANY elements at this point-->
        </pnet:systemData>
        <pnet:Columns>CorrelationID</pnet:Columns>
        <pnet:Columns>FullPath</pnet:Columns>
        <pnet:Columns>Title</pnet:Columns>
        <pnet:Columns>Description</pnet:Columns>
        <pnet:Columns>CardNumber</pnet:Columns>
      </pnet:ListReq>
    </pnet:List>
  </soapenv:Body>
</soapenv:Envelope>
```

- The List service processes the following response and output parameters.



ID	XPath	Scope	Datatype	Collection ID	Record ID	Format	Format Value	Description
documents	//ListResponse/ListResult/Items/Rows	response	string			None		
corrId	//Cells[1]	response	string	documents		None		
fullPath	//Cells[2]	response	string	documents		None		
title	//Cells[3]	response	string	documents		None		
description	//Cells[4]	response	string	documents		None		
number	//Cells[5]	response	string	documents		None		
opstatus		response	string			None		
errmsg		response	string			None		
httpStatusCode		response	string			None		
...								



```
<soap:Envelope ... >
  <soap:Header>
    ...
  </soap:Header>
  <soap:Body>
    <ListResponse xmlns="www.gmc.net/PNetTWebService/PNetTWebServices">
      <ListResult>
        <systemData/>
        <Items>
          <ColumnNames>CorrelationID</ColumnNames>
          <ColumnNames>FullPath</ColumnNames>
          <ColumnNames>Title</ColumnNames>
          <ColumnNames>Description</ColumnNames>
          <ColumnNames>CardNumber</ColumnNames>
          <Rows>
            <Cells>Apply for Savings Account Vital.html</Cells>
            <Cells>data:/GmcApiShowCase/listSample/Directory A/Apply for Savings Account Vital.html</Cells>
            <Cells>CREDIT CARD STATEMENT</Cells>
            <Cells>Please find details of your statement and pay a minimum amount of 25$ this month.</Cells>
            <Cells>37*****7841</Cells>
          </Rows>
          <Rows>
            <Cells>Credit card.html</Cells>
            <Cells>data:/GmcApiShowCase/listSample/Directory A/Credit card.html</Cells>
            <Cells>CURRENT ACCOUNT STATEMENT</Cells>
            <Cells>You have a positive balance of 1500$ on your current account.</Cells>
            <Cells>37*****7841</Cells>
          </Rows>
          <Rows>
            ...
        </Items>
        <TotalCount>4</TotalCount>
      </ListResult>
    </ListResponse>
  </soap:Body>
</soap:Envelope>
```

5.2.3 Calling the Inspire Connect .NET Method

Handling of the GMC repository (ICM) is ensured by the `List` method of the XmlDB web service in Inspire Connect .NET.

The method can return a list of files stored in ICM and their properties, i.e. ICM metadata.

Using other methods of the XmlDB web service, you can e.g. download files from ICM, upload files, change their location, delete them or set their properties.

You can find all details about these methods in the Inspire Connect .NET Programming Guide (Supported XmlDB Web Service Methods section).

5.3 Document Signing

The Signing Sample includes these steps:

1. A package (containing a PDF document to be signed) is created and sent to the e-signature provider server. The document is displayed to the user (signer).

   The GMC Showcase application displays the document in an external browser, but it is also possible to display it within the mobile application.

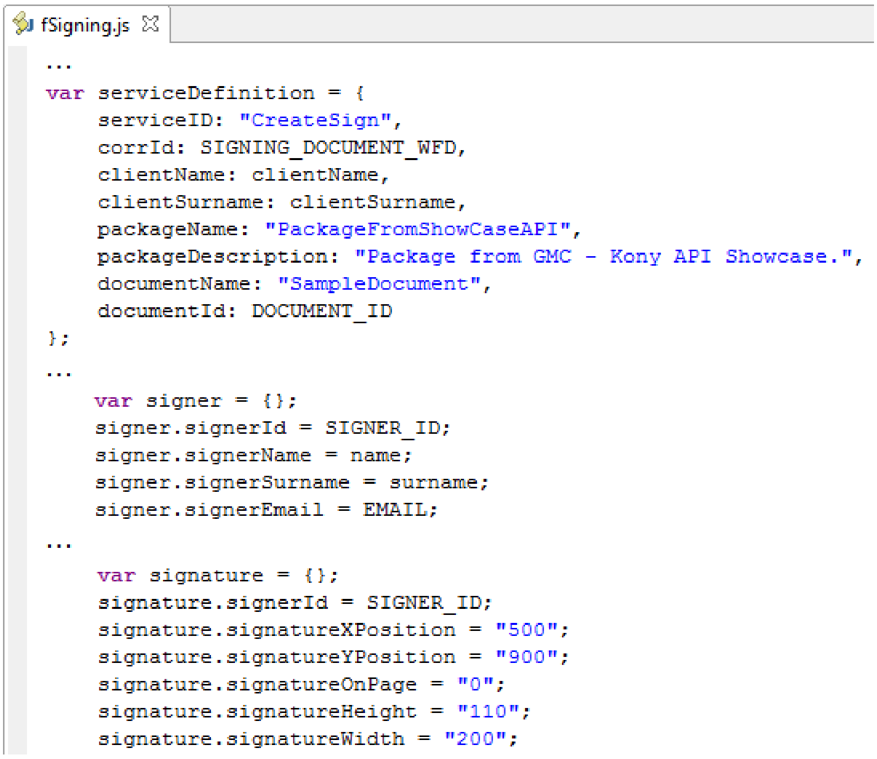
2. The document can be signed.

3. The package status can be queried, i.e. the user can inquire whether the document has been signed already.

5.3.1 Calling the Service Definition

All services defined in Kony are called from the mobile device by the same function. The second parameter for the Signing Sample is called `serviceDefinition`.

The Signing Sample uses these input parameters for package creation:

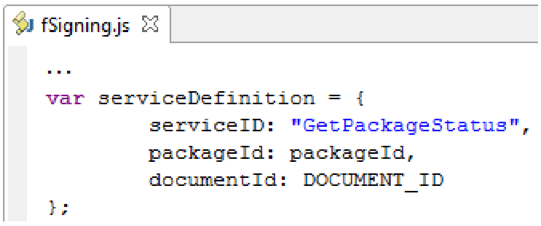


```
...
var serviceDefinition = {
    serviceID: "CreateSign",
    corrId: SIGNING_DOCUMENT_WFD,
    clientName: clientName,
    clientSurname: clientSurname,
    packageName: "PackageFromShowCaseAPI",
    packageDescription: "Package from GMC - Kony API Showcase.",
    documentName: "SampleDocument",
    documentId: DOCUMENT_ID
};
...
    var signer = {};
    signer.signerId = SIGNER_ID;
    signer.signerName = name;
    signer.signerSurname = surname;
    signer.signerEmail = EMAIL;
...
    var signature = {};
    signature.signerId = SIGNER_ID;
    signature.signatureXPosition = "500";
    signature.signatureYPosition = "900";
    signature.signatureOnPage = "0";
    signature.signatureHeight = "110";
    signature.signatureWidth = "200";
```

- `serviceID` – Identifier (a unique name) of the service (defined in Kony) that is called.
- `corrId` – Variable specifying the path to the WFD file that is processed (its value is defined in the `resources.js` file).
- `clientName` – Variable specifying the name of a variable defined in the input module of the WFD file.
- `clientSurname` – Variable specifying the name of a variable defined in the input module of the WFD file.
- `packageName` – String specifying the name of the signing package sent to the e-signature provider server.
- `packageDescription` – String providing the package description.
- `documentName` – String specifying the name of the document included in the package.

- `documentId` – Variable specifying the document identifier within the package. In the GmcApiShowcase project, it is always *1*.
- `signer` – Variable specifying the signer who has to provide the electronic signature: the signer's ID (used for linking signers and signatures), name, surname and email address).
- `signature` – Variable specifying the signature: ID of the signer who has to provide the signature, index of the page on which the signature is placed and exact position of the signature within the page (on the X and Y axes).

For getting the package status, these parameters are used:



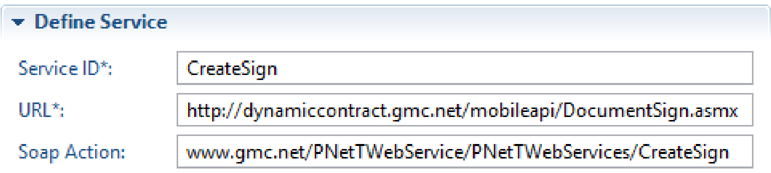
```
...
var serviceDefinition = {
        serviceID: "GetPackageStatus",
        packageId: packageId,
        documentId: DOCUMENT_ID
};
```

- `serviceID` – Identifier (a unique name) of the service (defined in Kony) that is called.
- `packageId` – Variable specifying the identifier of the signing package posted on the e-signature provider server.
- `documentId` – Variable specifying the document identifier within the package. In the GmcApiShowcase project, it is always *1*.

5.3.2 Service Definition in Kony

The CreateSign service is defined this way:

- The CreateSign service calls the DocumentSign web service of Inspire Connect .NET.

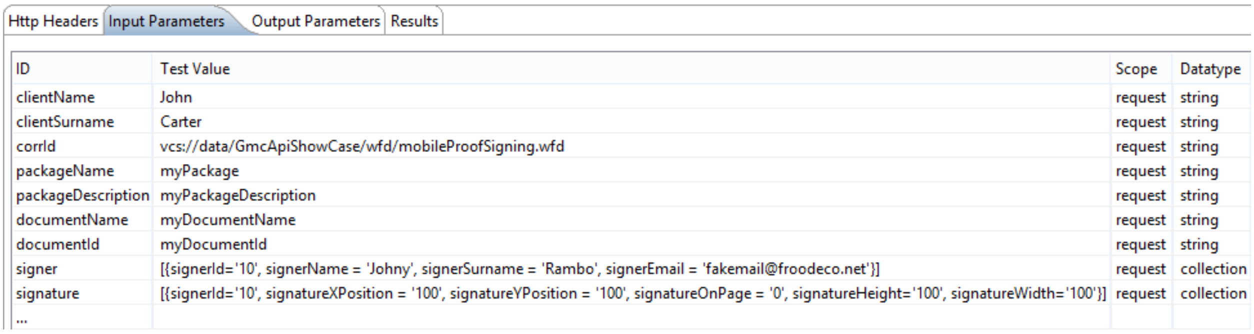


Service ID*: CreateSign

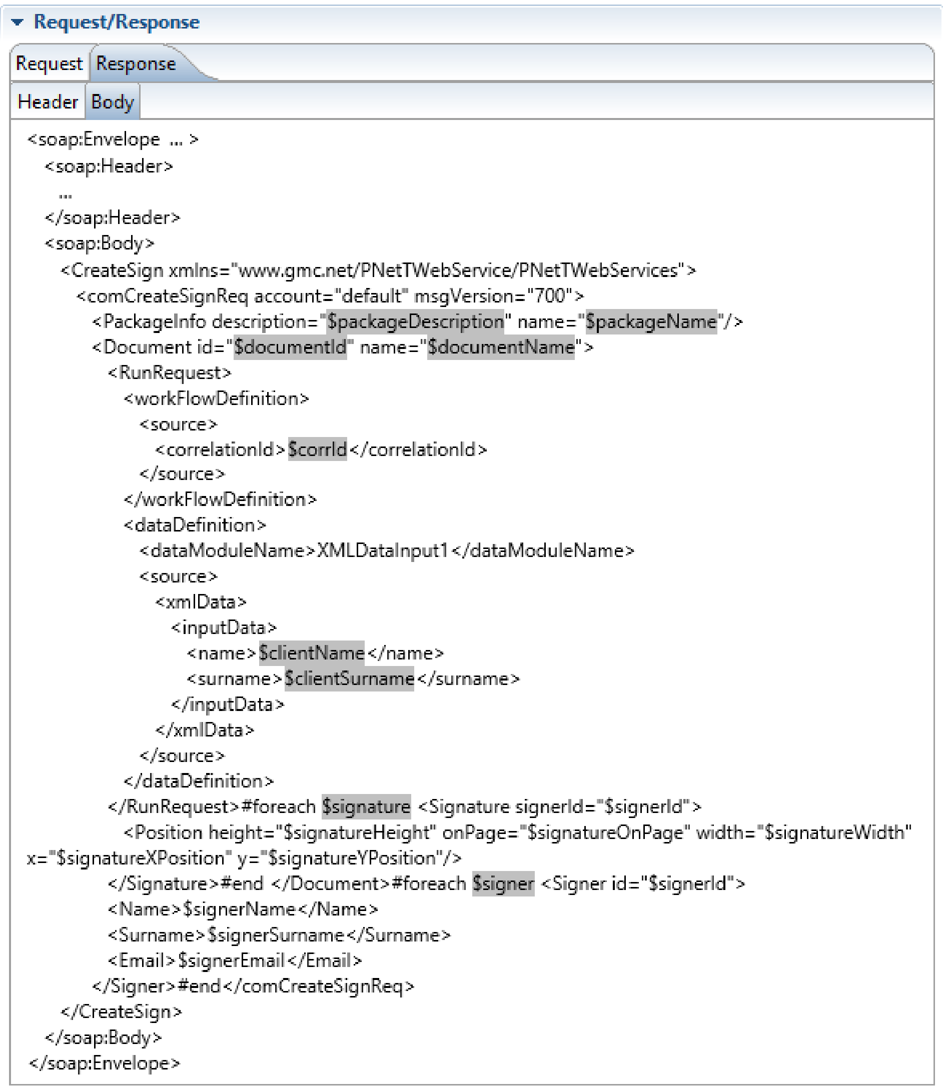
URL*: http://dynamiccontract.gmc.net/mobileapi/DocumentSign.asmx

Soap Action: www.gmc.net/PNetTWebService/PNetTWebServices/CreateSign

- The CreateSign service uses the following request and input parameters.

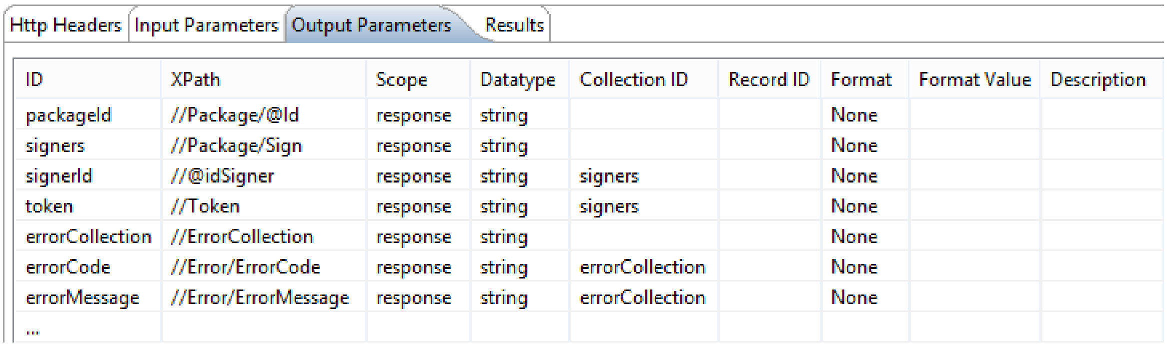


ID	Test Value	Scope	Datatype
clientName	John	request	string
clientSurname	Carter	request	string
corrId	vcs://data/GmcApiShowCase/wfd/mobileProofSigning.wfd	request	string
packageName	myPackage	request	string
packageDescription	myPackageDescription	request	string
documentName	myDocumentName	request	string
documentId	myDocumentId	request	string
signer	[{signerId='10', signerName = 'Johny', signerSurname = 'Rambo', signerEmail = 'fakemail@froodeco.net'}]	request	collection
signature	[{signerId='10', signatureXPosition = '100', signatureYPosition = '100', signatureOnPage = '0', signatureHeight='100', signatureWidth='100'}]	request	collection
...			

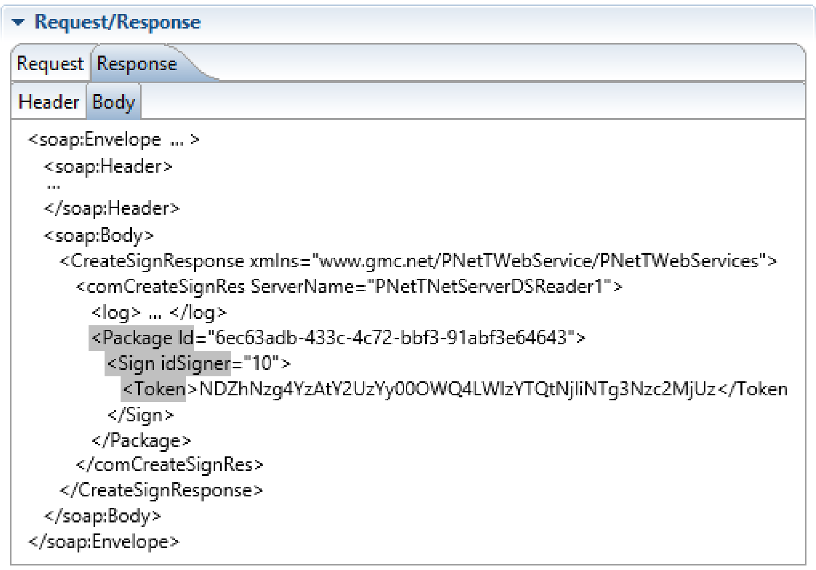


```
<soap:Envelope ... >
  <soap:Header>
    ...
  </soap:Header>
  <soap:Body>
    <CreateSign xmlns="www.gmc.net/PNetTWebService/PNetTWebServices">
      <comCreateSignReq account="default" msgVersion="700">
        <PackageInfo description="$packageDescription" name="$packageName"/>
        <Document id="$documentId" name="$documentName">
          <RunRequest>
            <workFlowDefinition>
              <source>
                <correlationId>$corrId</correlationId>
              </source>
            </workFlowDefinition>
            <dataDefinition>
              <dataModuleName>XMLDataInput1</dataModuleName>
              <source>
                <xmlData>
                  <inputData>
                    <name>$clientName</name>
                    <surname>$clientSurname</surname>
                  </inputData>
                </xmlData>
              </source>
            </dataDefinition>
          </RunRequest>#foreach $signature <Signature signerId="$signerId">
            <Position height="$signatureHeight" onPage="$signatureOnPage" width="$signatureWidth"
x="$signatureXPosition" y="$signatureYPosition"/>
          </Signature>#end </Document>#foreach $signer <Signer id="$signerId">
          <Name>$signerName</Name>
          <Surname>$signerSurname</Surname>
          <Email>$signerEmail</Email>
        </Signer>#end</comCreateSignReq>
    </CreateSign>
  </soap:Body>
</soap:Envelope>
```

- The CreateSign service processes the following response and output parameters.



ID	XPath	Scope	Datatype	Collection ID	Record ID	Format	Format Value	Description
packageId	//Package/@Id	response	string			None		
signers	//Package/Sign	response	string			None		
signerId	//@idSigner	response	string	signers		None		
token	//Token	response	string	signers		None		
errorCollection	//ErrorCollection	response	string			None		
errorCode	//Error/ErrorCode	response	string	errorCollection		None		
errorMessage	//Error/ErrorMessage	response	string	errorCollection		None		
...								



```
<soap:Envelope ... >
  <soap:Header>
  ...
  </soap:Header>
  <soap:Body>
    <CreateSignResponse xmlns="www.gmc.net/PNetTWebService/PNetTWebServices">
      <comCreateSignRes ServerName="PNetTNetServerDSReader1">
        <log> ... </log>
        <Package Id="6ec63adb-433c-4c72-bbf3-91abf3e64643">
          <Sign idSigner="10">
            <Token>NDZhNzg4YzAtY2UzYy00OWQ4LWIzYTQtNjliNTg3Nzc2MjUz</Token
          </Sign>
        </Package>
      </comCreateSignRes>
    </CreateSignResponse>
  </soap:Body>
</soap:Envelope>
```

5.3.3 Calling the Inspire Connect .NET Method

Document Signing is ensured by the`CreateSign` and `GetPackageStatus` methods of the DocumentSign web service in Inspire Connect .NET.

These methods are used for integration with the e-signature provider. The first one creates a package with a PDF document to be signed and sends the package to the e-signature provider server. Once the package is there, the second method can get the current status of the package, i.e. find out whether the document has been signed already.

The methods are internal and are not described in the Inspire Connect .NET documentation.

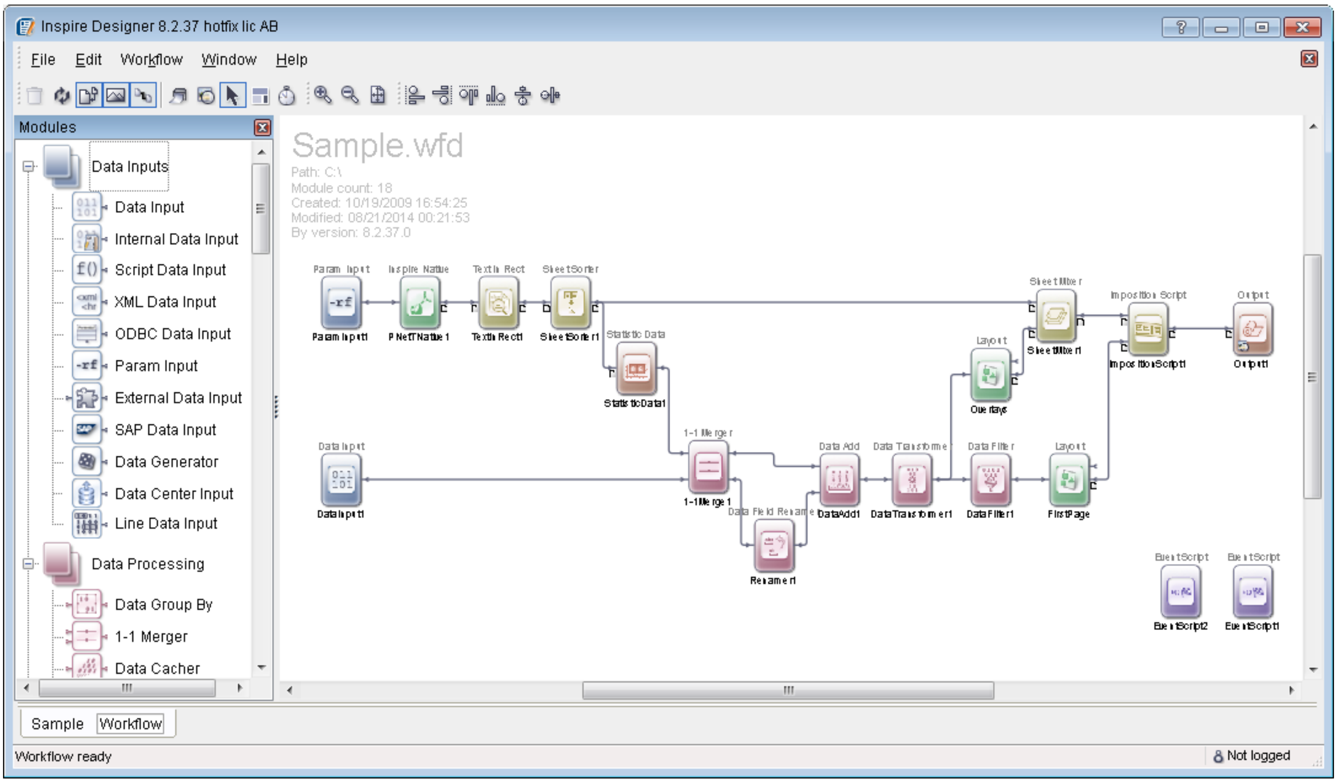
A GMC Inspire Applications

A.1 Inspire Designer

Inspire Designer is a variable-data composition software which formats documents from a range of data and design inputs. It supports various data channels (e.g. data files, databases, web services, social media) as well as all major document output formats, such as PDF, AFP, PCL, HTML, DOC. The whole document setup is stored in a workflow file (.WFD). This file contains the entire processing algorithm of a document, including its graphic layout and configuration of data inputs, data processing, impositioning and outputs.

Inspire Designer GUI

Inspire Designer features an installed desktop application used by document designers to create a WFD file. The visual environment provides the user with building blocks and tools required for designing, previewing and testing of a document. Once the document design is complete, production can be run manually from a production environment. However, this production can be neither automated nor integrated with other products and services.



Inspire Designer 8.2.37 hotfix lic AB

Sample.wfd

Path: C:\
Module count: 18
Created: 10/19/2009 16:54:25
Modified: 08/21/2014 00:21:53
By version: 8.2.37.0

Workflow ready

Inspire Production Server

Inspire Production Server (IPS) is Inspire Designer's production capabilities running as a service. Since the document production consumes significant amounts of processing power, this service is usually installed in several instances, each on a separate server. Other software (e.g. Inspire Connect) can communicate with these services to run the production of a WFD file using specific data and parameters.

Inspire Content Manager

Inspire Content Manager (ICM) is another component of Inspire Designer. It is installed as a service on a separate server. It provides storage and management of data for all Inspire products (only). Inspire products connect to this service as clients to retrieve data.

A.2 Inspire Connect .NET

Inspire Connect (IC) is a web application that integrates with Inspire Production Server and Inspire Content Manager services. It provides a SOAP API accessible via WSDL and also via IBM WebSphere messaging system. This API allows other Inspire and third-party applications to easily access the ICM storage and to run production of WFD files using IPS. Moreover, IC automatically manages and distributes the load among multiple IPS instances.

Inspire Connect is used for automatic processing of client application requests. It features two publicly supported web services:

- XmlDB Web Service

  `http://ServerName/VirtualDirectoryName/dbs/XmlDB.asmx`

  This service provides basic file and directory operations over the ICM storage. The service has dedicated methods for each type of operation, e.g. upload, download, copy, move, rename, create dir, set permissions.

- PrintNetTServices Web Service

  `http://ServerName/VirtualDirectoryName/PNetTServices.asmx`

  This is production web service that invokes IPS operations. The service has dedicated methods for each type of operation, e.g. run production, generate a preview, get WFD properties)

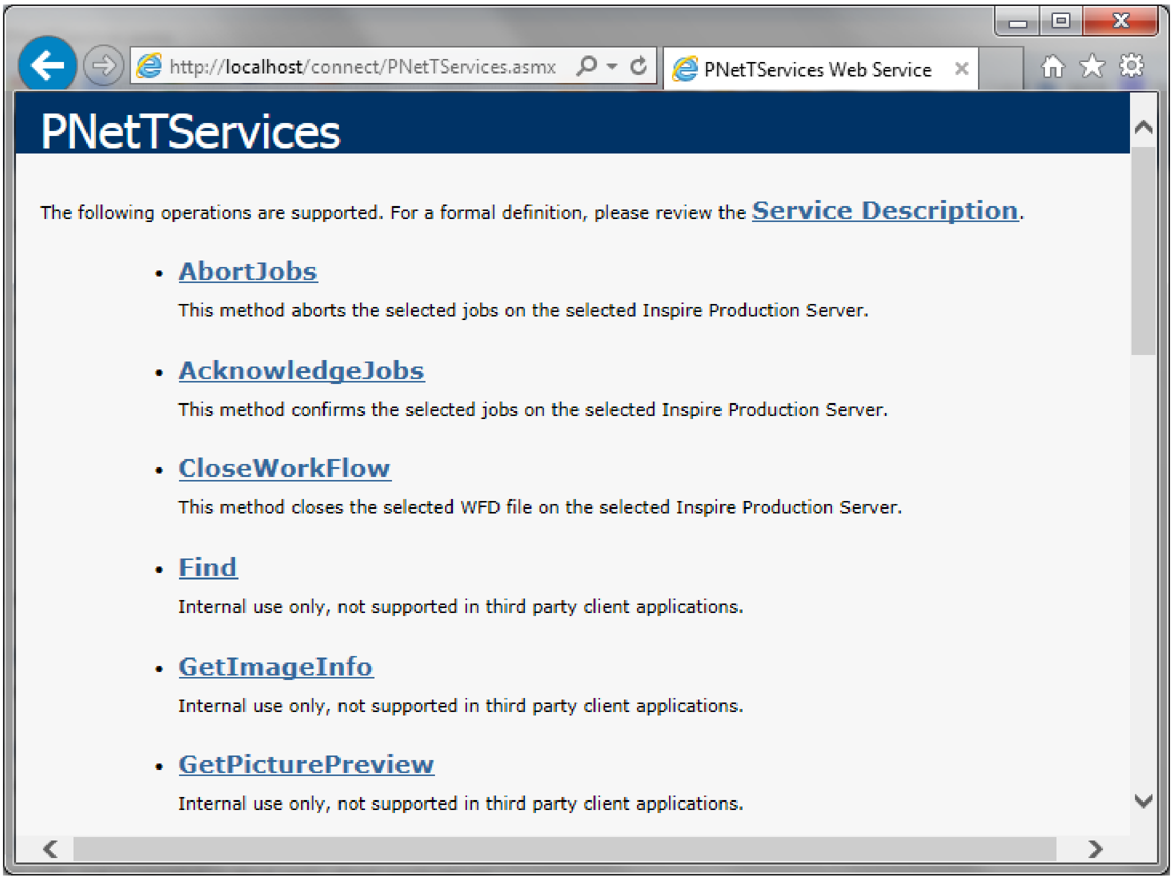
- DocumentSign Web Service

  `http://ServerName/VirtualDirectoryName/DocumentSign.asmx`

  This is a special service for production of signed PDF documents. Produced PDFs are automatically submitted to the e-signature provider server and the service returns URL of the uploaded document. Subsequently, this URL can be used to electronically sign the document or to check the document status.

The communication between these services and a client application is performed using XML request/response messages. The client application sends a web request to the service and calls one of its available methods. The request contains XML data, commands and parameters required for requested operation. Once the request has been processed by the web service, a response is sent back to the client application. The XML data in the response contain information about the result of the requested operation, e.g. log messages, requested values, paths to output files or even Base64-encoded output documents.

The syntax of XML data used for communication with these services is available in XSD and WSDL format in the `VirtualDirectoryName\Doc\` directory. Also, after entering the web service URL into a web browser, the service returns a list of available methods with sample requests and responses.



http://localhost/connect/PNetTServices.asmx
PNetTServices Web Service

PNetTServices

The following operations are supported. For a formal definition, please review the Service Description.

- AbortJobs

  This method aborts the selected jobs on the selected Inspire Production Server.

- AcknowledgeJobs

  This method confirms the selected jobs on the selected Inspire Production Server.

- CloseWorkFlow

  This method closes the selected WFD file on the selected Inspire Production Server.

- Find

  Internal use only, not supported in third party client applications.

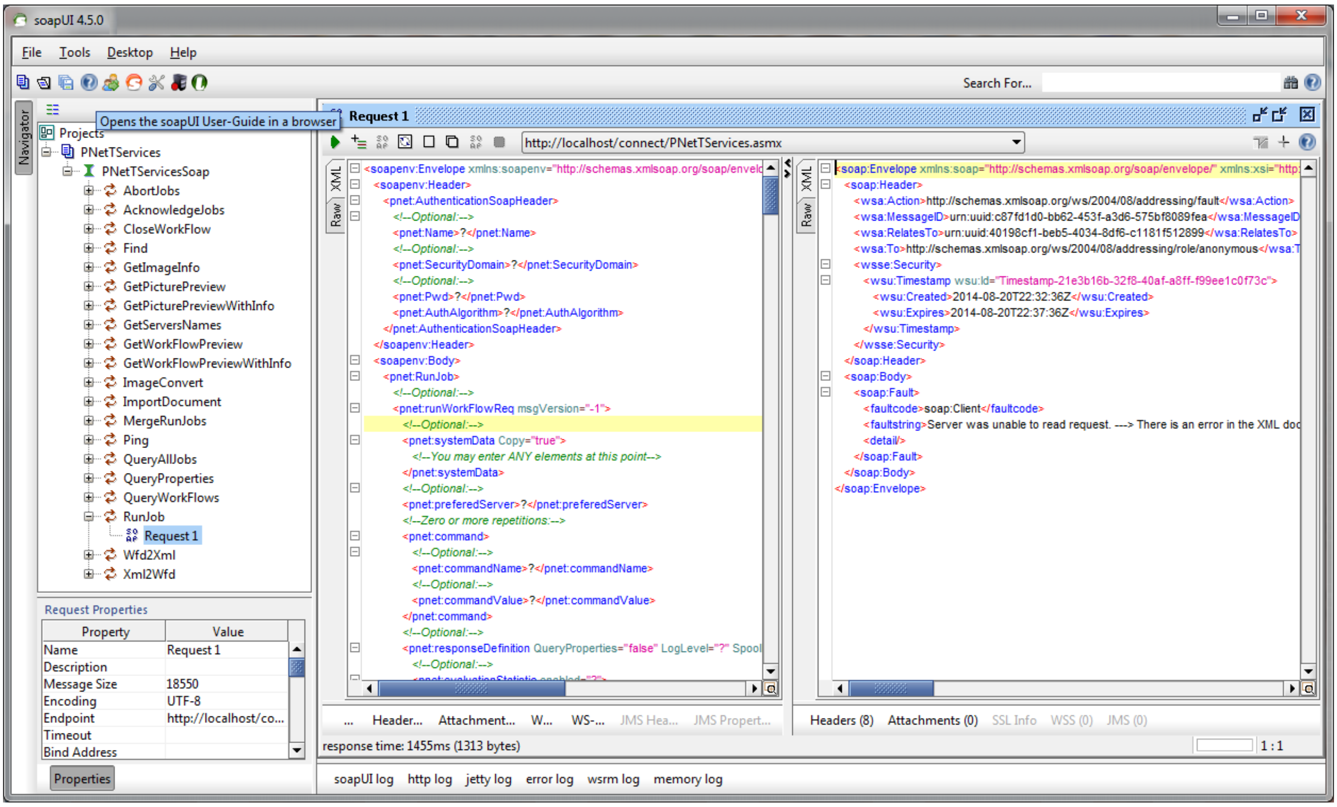
- GetImageInfo

  Internal use only, not supported in third party client applications.

- GetPicturePreview

  Internal use only, not supported in third party client applications.

For parsing by client applications, the WSDLs are directly obtainable using the web service URL appended with `?WSDL` parameter (e.g. `http://ServerName/VirtualDirectoryName/dbs/XmlDB.asmx?WSDL`).



soapUI 4.5.0
File Tools Desktop Help
Search For...
Navigator
Opens the soapUI User-Guide in a browser
Request 1
http://localhost/connect/PNetTServices.asmx
XML
Raw

```
Projects
  PNetTServices
    PNetTServicesSoap
      AbortJobs
      AcknowledgeJobs
      CloseWorkFlow
      Find
      GetImageInfo
      GetPicturePreview
      GetPicturePreviewWithInfo
      GetServersNames
      GetWorkFlowPreview
      GetWorkFlowPreviewWithInfo
      ImageConvert
      ImportDocument
      MergeRunJobs
      Ping
      QueryAllJobs
      QueryProperties
      QueryWorkFlows
      RunJob
        Request 1
      Wfd2Xml
      Xml2Wfd
```

Request Properties

Property	Value
Name	Request 1
Description	
Message Size	18550
Encoding	UTF-8
Endpoint	http://localhost/co...
Timeout	
Bind Address	

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelo
  <soapenv:Header>
    <pnet:AuthenticationSoapHeader>
      <!--Optional:-->
      <pnet:Name>?</pnet:Name>
      <!--Optional:-->
      <pnet:SecurityDomain>?</pnet:SecurityDomain>
      <!--Optional:-->
      <pnet:Pwd>?</pnet:Pwd>
      <pnet:AuthAlgorithm>?</pnet:AuthAlgorithm>
    </pnet:AuthenticationSoapHeader>
  </soapenv:Header>
  <soapenv:Body>
    <pnet:RunJob>
      <!--Optional:-->
      <pnet:runWorkFlowReq msgVersion="-1">
        <!--Optional:-->
        <pnet:systemData Copy="true">
          <!--You may enter ANY elements at this point-->
        </pnet:systemData>
        <!--Optional:-->
        <pnet:preferedServer>?</pnet:preferedServer>
        <!--Zero or more repetitions:-->
        <pnet:command>
          <!--Optional:-->
          <pnet:commandName>?</pnet:commandName>
          <!--Optional:-->
          <pnet:commandValue>?</pnet:commandValue>
        </pnet:command>
        <!--Optional:-->
        <pnet:responseDefinition QueryProperties="false" LogLevel="?" Spool
          <!--Optional:-->
```

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http:
  <soap:Header>
    <wsa:Action>http://schemas.xmlsoap.org/ws/2004/08/addressing/fault</wsa:Action>
    <wsa:MessageID>urn:uuid:c87fd1d0-bb62-453f-a3d6-575bf8089fea</wsa:MessageID
    <wsa:RelatesTo>urn:uuid:40198cf1-beb5-4034-8df6-c1181f512899</wsa:RelatesTo>
    <wsa:To>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:T
    <wsse:Security>
      <wsu:Timestamp wsu:Id="Timestamp-21e3b16b-32f8-40af-a8ff-f99ee1c0f73c">
        <wsu:Created>2014-08-20T22:32:36Z</wsu:Created>
        <wsu:Expires>2014-08-20T22:37:36Z</wsu:Expires>
      </wsu:Timestamp>
    </wsse:Security>
  </soap:Header>
  <soap:Body>
    <soap:Fault>
      <faultcode>soap:Client</faultcode>
      <faultstring>Server was unable to read request. ---> There is an error in the XML doc
      <detail/>
    </soap:Fault>
  </soap:Body>
</soap:Envelope>
```

... Header... Attachment... W... WS-... JMS Hea... JMS Propert...
Headers (8) Attachments (0) SSL Info WSS (0) JMS (0)
1 : 1
soapUI log http log jetty log error log wsrm log memory log

response time: 1455ms (1313 bytes)