



A Guide to the Pool Doctors App and Video

The Pool Doctors app illustrates a simple “real world” app created with Genero Mobile.

This document was last updated on 11 July 2014

Getting Started

If you plan to experiment hands-on while watching the videos, you will want to:

1. Download, install and license Genero Mobile.

You can download Genero Mobile [here](#).

If you select the free personal plan, your license details will be sent to the email address provided. To license Genero Mobile, start the application and go to **Admin >> Genero BDL Licenser** and complete the form.

For more information, refer to the [licensing instructions](#) in the Genero Mobile Developer Guide.

2. Configure Genero Mobile to display to your simulator or device.

[Full instructions](#) for configuring Genero Mobile for iOS or Android are provided in our documentation. The last steps of the instructions have you execute the **OrdersApp** app from the **OfficeStoreMobile** project.

If you can execute and view the **OrdersApp** app, then you are ready to move to the next step.

3. [Download](#) the source for the Pool Doctors application (**pool_doctors.zip**) and extract the files.

By default, the files extract into a **pool_doctors** directory. The Pool Doctors project file is **pool_doctors.4pw**.

4. From Genero Mobile, select **File >> Open**, select **pool_doctors.4pw**, and click Open.

The project opens.

5. To run the app, select **Debug >> Execute**.

The app runs on your device (as specified by the selected configuration).



Design Brief

The technician or courier uses the app at a remote site. He uses a device (iOS or Android) to track his performance and/or the delivery of goods.

The application is generic. It can be used by:

- A technician recording work performed
- A courier driver recording proof of delivery

The application uses many device features:

- Camera
- Barcode Reader
- GPS and Map
- Phone/SMS/Email
- Signature Capture
- Database Synchronization

Why Pool Doctors?

The Pool Doctors app is an example of a technician/ courier driver role created to meet the design brief criteria. After running the app, we hope you can visualize many similarly-named apps: Refrigerator Doctor, TV Doctor, Car Doctor, and so on. All that is needed is new data for the customer and product tables.



Business Process

These are the defined business processes:

- The customer makes contact with the head office or call center and requests work to be done.
- The job is entered into the system and assigned to a technician or rep. The field cm_rep in the customer table indicates which technician is responsible for a customer.
- From within the app, the technician uses the sync option to retrieve the jobs assigned to him and to load them onto his device.
- The technician uses the device to contact the customer.
- The technician uses the device to navigate to the customer site.
- At the site, the technician uses the device to record:
 - i. Parts used.
 - ii. Photos of the work done.
 - iii. Notes about the work done.
 - iv. Timesheet entries for the work done.
- When the job is complete, the technician records the customer's signature as acknowledgement of the work done.
- The sync process uploads the job details back to the head office database.
- Head office can then do the appropriate invoicing and generate appropriate accounting, inventory, and payroll transactions.



How to Use the App (First-Time Users)

Create a random job (for testing) from the main splash screen:

1. Tap the settings icon.
2. Select Create Random Job.
This makes a web service call to create a random job on the parent database server.
3. Note the job number displayed in the dialog box.
4. Tap OK.

Synchronize the job data on your device from the main splash screen:

1. Tap the sync icon.
The job downloads onto your device

Access your new job from the main splash screen:

1. Tap the splash screen (Tap to continue).
The Job List displays, listing the jobs on your device.
2. Find your new job. You will likely have to scroll to the bottom of the list of jobs.
3. Tap the job.
This selects the job and displays the job details in the Job window.
4. Tap the icon in the Customer field.
The Customer Details window displays. You can:
 - View details about the customer.
 - View the customer's location on a map.
 - Call the customer.
 - Send a SMS text message to the customer.
 - Send an email to the customer.
5. Tap Cancel to return to the Job window.

Edit the job:

1. Tap the Start button.
The status of the job changes to in-progress.
You can now edit the parts used (lines), photos, notes, and timesheet for the job.
2. For each of the job areas (parts used, photos, notes and timesheet):
 - Accessing the area provides a list for that job.
 - Tap the append icon (the plus sign) to add a new entry.
 - Tap an entry in the list to edit the entry.
 - Swipe right-to-left to delete the entry.

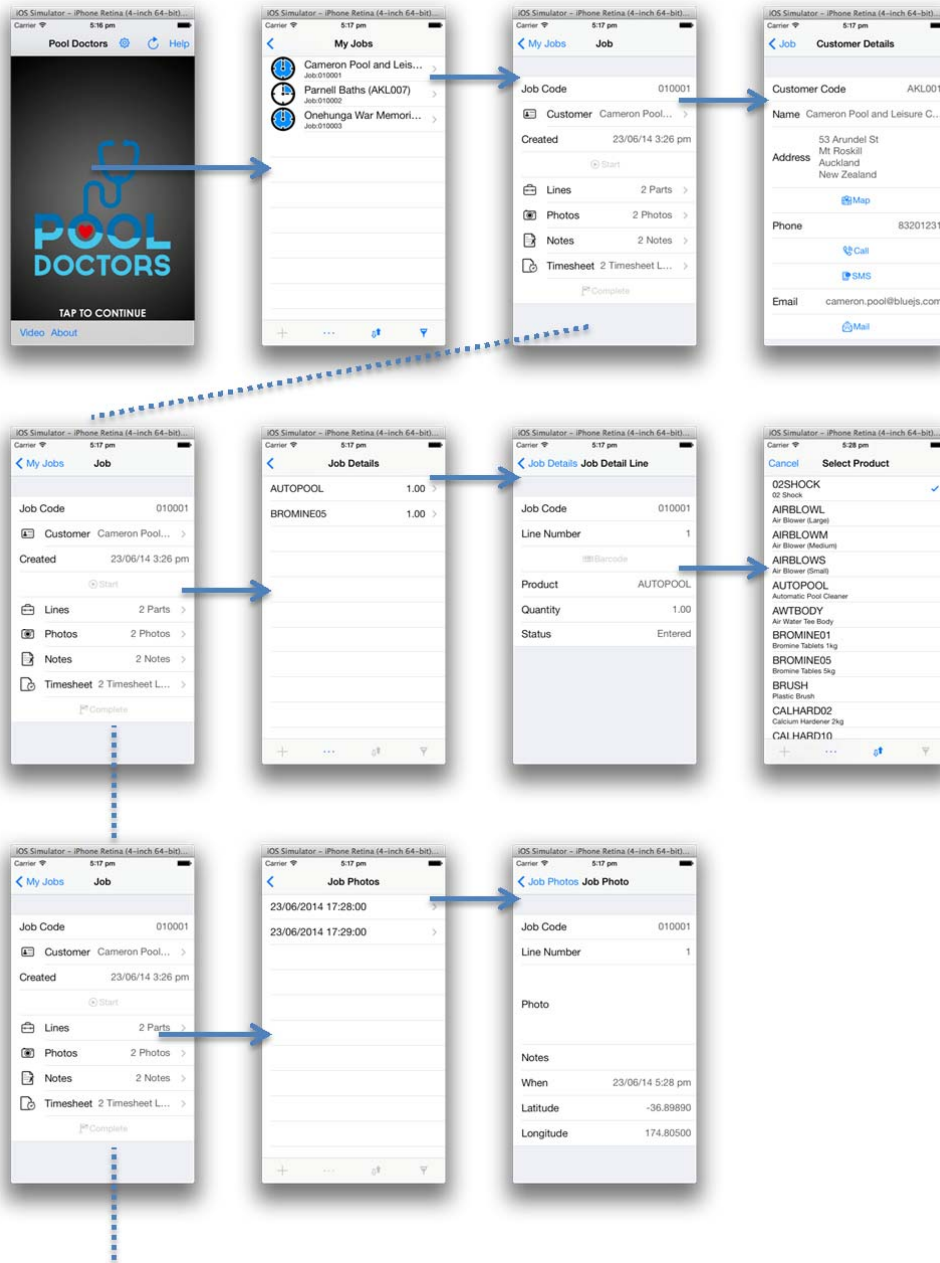


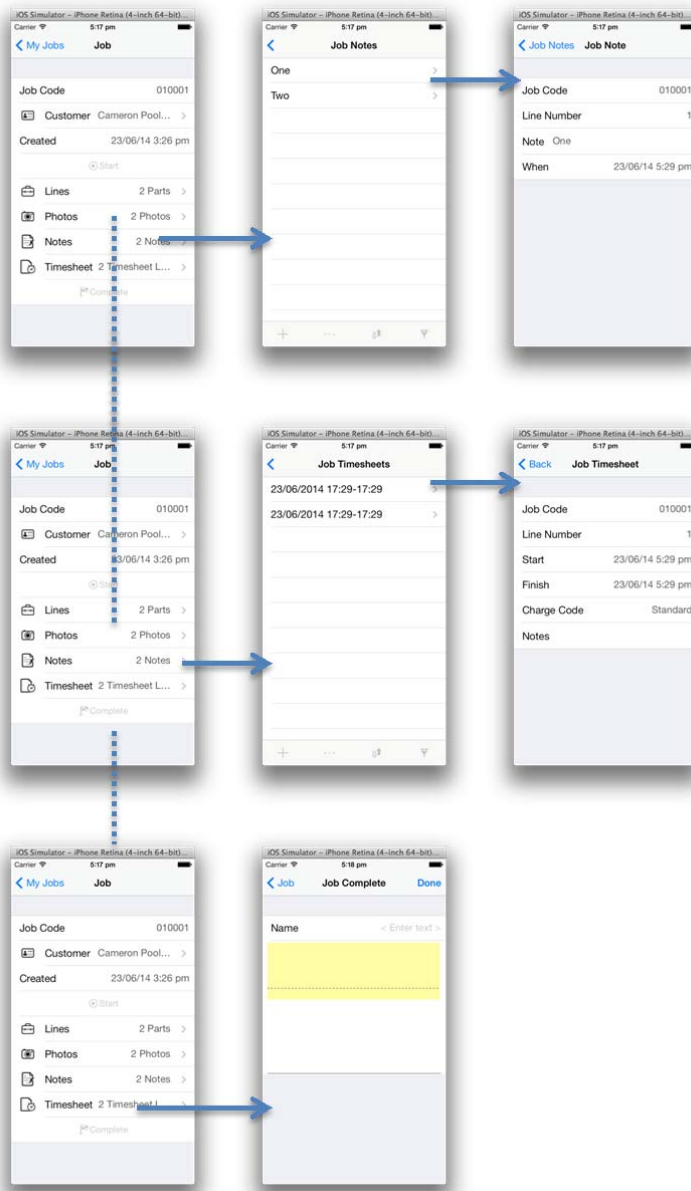
Completing the job:

1. Tap the complete
 2. Enter a name.
 3. Sign in the signature box.
 4. Tap OK.
- The status of the job will now be complete.

Screen Navigation

In this diagram, a solid arrow indicates a transition, while a dashed line indicates the same screen.





Database Design

There are 7 tables in the database on the device. The key thing to note is that the job details have a parent-child/master-detail relationship with the job_header table being the parent/master, whilst the other job tables are the child/detail tables.

Table Name	Description	References	Referenced By
job_header	Record details about a job.	customer	job_detail, job_note, job_photo, job_timesheet
job_detail	Record details of parts (products) used in completing a job.	job_header, product	
job_notes	Record free-text notes against a job.	job_header	
job_photo	Record details of photos taken for a particular job.	job_header	
job_timesheet	Record time entered against a particular job.	job_header	
customer	Customer detail. All information needed to contact the customer, as well as the customer rep (cm_rep).		job_header
product	Product list. All possible products that can be used when completing a job.		job_detail



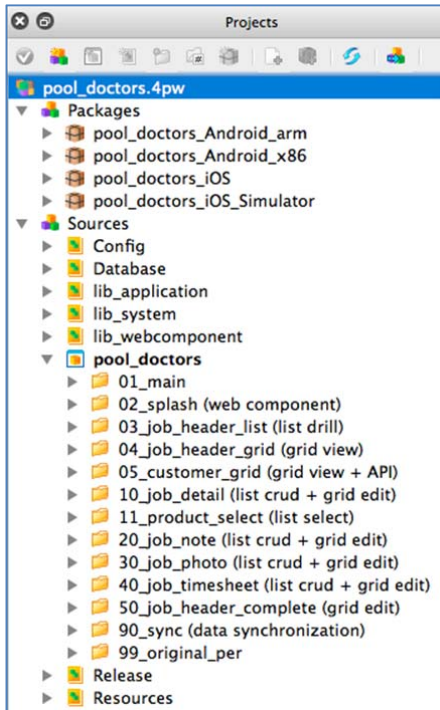
The Sync Process

The sync process:

- Refreshes the product and customer tables.
- Downloads any new jobs for the rep to the job_header table.
- Uploads completed job data from the 5 job-related tables.
- Deletes any completed job data from the device that is more than 7 days old.

Please note that this is a simple synchronization strategy. For production purposes, we would consider adding some more complex logic to ensure that two devices are not updating details against the same job and thus overriding data when both are then uploaded.

The Project Workspace



Project Workspace Entry	Description
pool_doctors_Android_arm	The package entry that should be used to deploy the application to an Android device with an ARM-based processor.
pool_doctors_Android_x86	The package entry that should be used to deploy the application to an Android device with an x86 based processor.
pool_doctors_iOS	The package entry that should be used to deploy the application to an iOS Device (iPhone, iPad Mini, iPad).
pool_doctors_iOS_Simulator	The package entry that should be used to deploy the application to an iOS Simulator.
config	Contains the fglprofile that contains run-time settings.
database	Contains the database schema file (.4db)
lib_application	Contains library routines that are considered unique to this application and/or database.
lib_system	Contains library routines that could be used in other applications.

Project Workspace Entry	Description
lib_webcomponent	Contains web components and library routines used with these web components.
pool_doctors	The Pool Doctors application. <i>* Note: the numbers used below here exist solely to place the virtual folders in the desired order</i>
01_main	The first 4gl executed. Creates the database if it is the first time the application is run.
02_splash (web component)	The first screen that the user sees. The “splash screen”. Uses a web component in lib_webcomponent.
03_job_header_list (list drill)	The list screen that lists all the jobs that have been downloaded onto the device. Essentially lists the rows in the job_header table.
04_job_header_grid (grid view)	The grid view that lists details about the selected job. Essentially displays one row of data from the job_header table
05_customer_grid (grid view+ API)	The grid view that lists details about the customer for the selected job. Essentially displays one row of data from the customer table. A number of device API’s can be used to call, SMS, email the customer as well as use a web component in lib_webcomponent to show the customers location on a map.
10_job_detail (list crud + grid edit)	The list screen that records the job_detail entries entered for the job, and the grid screen used to view and edit one job_detail entry. The entry of the product used illustrates the use of a BUTTONEDIT to call the product select zoom window, or alternatively shows how the product can be selected by using barcode reader.
11_product_select (list select)	The list screen that lists all the products in the product table. Using this screen the user is able to select a product that is then returned to the calling module.
20_job_note (list crud + grid edit)	The list screen that records the job_note entries entered for the job, and the grid screen used to view and edit one job_note entry.



Project Workspace Entry	Description
30_job_photo (list crud + grid edit)	The list screen that records the job_photo entries entered for the job, and the grid screen used to view and edit one job_photo entry. This includes the API call to take a photo using the device, and display it in the list and grid screen.
40_job_timesheet (list crud + grid edit)	The list screen that records the job_timesheet entries entered for the job, and the grid screen used to view and edit one job_timesheet entry.
50_job_header_complete (grid edit)	The screen called when the user taps the Complete button to record that the job is complete. This screen makes use of the signature web component which is in the lib_webcomponent folder.
90_sync (data synchronization)	The 4gl file with the job synchronization functions.
99_original_per	The 4fd form files in .per format for those that prefer the .per form file format.
release	Release notes, readme file, and so on.
resource	The User Interface resource file including the stylesheet (.4st), the action defaults (.4ad), and the toolbar (.4tb),



Videos

A number of videos have been produced to walk through areas of code. There is a rough correspondence between each video and a folder in the project workspace.

Title	Project Folder	Description
A Genero Mobile App		We have created an app that highlights how a Genero Mobile app takes full advantage of being on a mobile device. This preview shows you the app that is developed for the Pool Doctors video series.
Introduction	The project workspace	An introduction to the Pool Doctors application and the structure of the .4pw Project Workspace file
Application Start	01_main + Database	An overview of the code executed at the start of the application. In particular, looking at the initialization and the code that creates the database the first time the application is run.
Splash and Web Component	02_splash	An overview of the code behind the main splash screen. This code uses an INPUT statement in conjunction with a web component to display an image, and record actions selected by the user.
List Drill	03_job_header_list	An overview of the code that lists the jobs on the device. This code uses a DISPLAY ARRAY to list the jobs in the job_header table, and allows the user to select a job to view more detail about the job

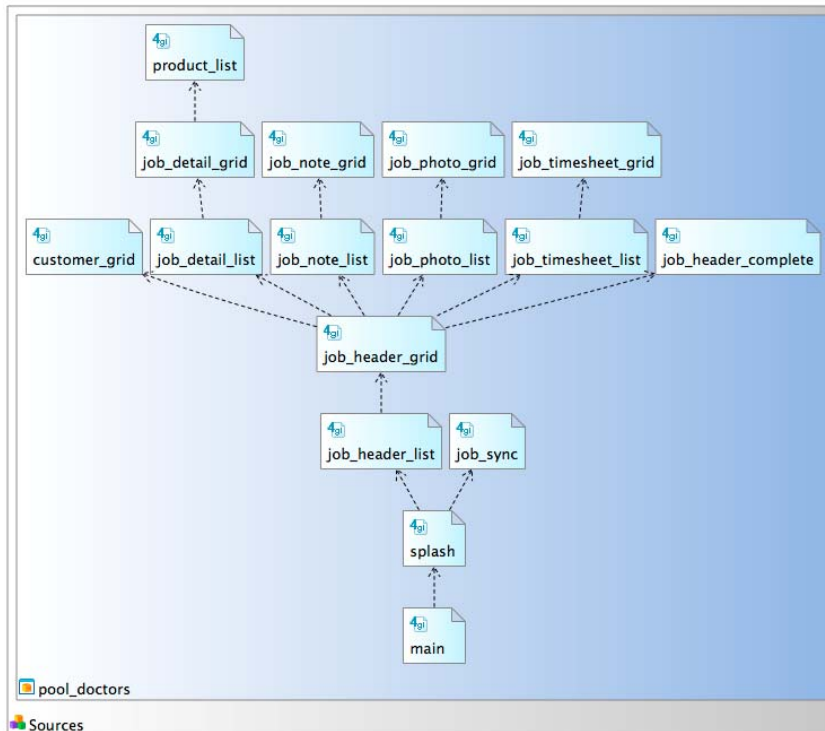
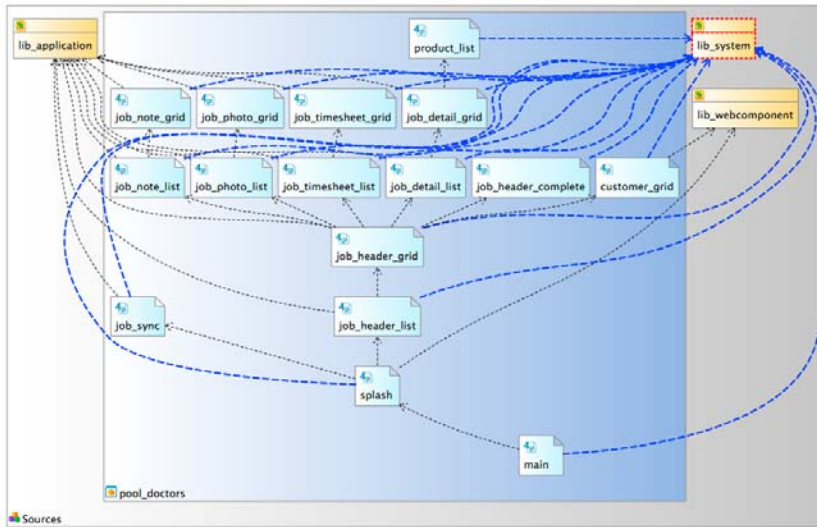
Title	Project Folder	Description
Grid View	04_job_header_grid + 05_customer_grid	An overview of the code that displays the details for an individual job. This code uses a MENU to display the details about a job from the job_header table. The user can then select some actions for the job. The same pattern is repeated in the customer grid code to display the details about a customer from the customer table.
List CRUD	10_job_detail + 20_job_note + 30_job_photo + 40_job_timesheet (list 4fd and 4gl)	An overview of the code that lists further details about a job, and allows the user to add job details, delete job details, or select a job to edit or view job details. The code uses a DISPLAY ARRAY to list the job details. The video concentrates on the listing of the information contained in the job_detail table, and the same coding pattern is repeated for the job_note, job_photo, and job_timesheet tables and screens.
Grid Edit	10_job_detail + 20_job_note + 30_job_photo + 40_job_timesheet (grid 4fd and 4gl)	An overview of the code that for a selected job detail allows the user to view or edit an existing job detail record as well as be used to add a new job detail record. The code uses a MENU to view the job detail, or an INPUT to add or edit a job detail record. The video concentrates on the editing and viewing of a record in the job_detail table, and the same coding pattern is repeated for the job_note, job_photo, and job_timesheet tables and screens.

Title	Project Folder	Description
List Select	11_product_select	<p>An overview of the code that lists the products in the product table and allows the user to select a product record that is passed back to the calling module.</p> <p>The code uses a DISPLAY ARRAY to list the product records from the product table.</p>
Web Component	50_job_header_complete	<p>An overview of the code that records the capture of the customer's name and signature when a job is complete.</p> <p>The code uses an INPUT and the signature web component.</p>
Device API	05_customer_grid	<p>An overview of the code that interacts with the device.</p> <p>From the customer screen, this includes the code to call/SMS/email the customer. From the job_detail screen, this includes the code to read a barcode. From the job_photo screen, this includes the code to take a photo.</p>
Web Component Browser	lib_webcomponent (browser.4gl and browser.per)	<p>An overview of the code that uses the empty web component to display a URL. From the customer screen, this includes the code to display a customer's location using google maps. From the splash screen, this includes the code to display a video or show a document.</p>
UI Library	Lib_system (lib_ui.4gl)	<p>This video gives an overview of the lib_ui.4gl file in the lib_system folder. This is primarily code to give the application a consistent user interface, and allow for easy maintenance of future user interface changes.</p>

Title	Project Folder	Description
Error Handling	Lib_system (lib_error.4gl)	This video gives an overview of the lib_error.4gl file in the lib_system folder. This is primarily code to handle unexpected run-time errors nicely by providing a dialog to the user, and recording in a log file details about the error that can be used for analysis.
UI Resources	Resource	An overview of the stylesheet (*.4st), action defaults (.4ad), and toolbar (.4tb) files in the Resources folder.
Data Synchronization	90_sync	An overview of the process of data synchronization.
Back End Processes		The Pool Doctors Genero Mobile app has a synchronization process interacts with a database held on a remote server. This video introduces the web services that are called from within job_sync.4gl that read from this database and shows how these web services can be run from a browser to view the output. It introduces the Genero Web Client enquiry application that interfaces the database (https://demo.4js.com/gas/wa/r/pool_doctors/enquiry) and how this program can be used to view the database changes as a result of the Pool Doctors Genero Mobile app running and then synchronizing with this database.
Dependency Diagrams		The dependency diagram can be used in Genero Studio to give an overview of the structure of the 4gl code. This video shows the dependency diagram for the Pool Doctors Genero Mobile application.

Dependency Diagram

The Dependency Diagram from within Genero Studio illustrates the relationship between the various .4gl files in the application. The two screenshots below are of the dependency diagram, the second has had the libraries hidden to provide clarity





The back-end database

Data held in the back-end database can be viewed through a Genero Web Client application which we will call the enquiry program:

https://demo.4js.com/gas/wa/r/pool_doctors/enquiry

To understand the flow of data to/from this back-end database, do the following:

Enquiry Program (in browser)	Pool Doctors app (on device)
1. In a browser, run the enquiry program. Note the last entry.	
	2. In the Pool Doctors app, tap Settings. Tap Create Random Job. Note the job number created. Tap OK.
3. In the enquiry program, click Refresh. Note the last entry. This should correspond to the job just created.	
	4. In the Pool Doctors app, view your job list (My Jobs). The new job is not yet listed.
	5. Tap the sync button. The new job is downloaded to your device. It is visible in the job list (My Jobs). It has a status of New.
	6. In the Pool Doctors app, open the job, tap Start, enter some data against the job, (in the parts (Lines), Notes, Photo, or Timesheet sections) and tap Complete. The job status changes to complete.
7. In the enquiry program, click Refresh. Examine the Job List. Note that the status of the job still states New.	
	8. In the Pool Doctors app, tap the Sync icon. The newly-completed job will be uploaded to the back-end database.
9. In the enquiry program, click Refresh. Note the job now has a status of Complete.	



Note that some of the web services used are also visible in a browser. Enter the following URL and note the response:

https://demo.4js.com/gas/ws/r/pool_doctors/service/getProducts