Hi Roger

Please see below

**From:** Weiss, Roger: Absa [<mailto:Roger.Weiss@absa.co.za>]   
**Sent:** Tuesday, 30 April 2019 09:29  
**To:** Claire Bowman <[Claire.Bowman@nacelleaviation.com](mailto:Claire.Bowman@nacelleaviation.com)>  
**Cc:** Kirsty Barkhuizen <[kirsty@nacelleaviation.com](mailto:kirsty@nacelleaviation.com)>  
**Subject:** API Integration

Hi Claire,

As Kirsty mentioned on the WhatsApp group, I want to start integrating the app with the API, actually get some basic calls going in in the next couple of days.

On our call 2 weeks ago where you helped me out with some bookings on cert,  you briefly walked me through a couple of calls, while we tried to make a booking for me.  I didn’t write of  the info down, so not that information is lost.  (Esp. in transferring the ID from one call to the next)

For this first phase,  we’re just replicating in the Xamarin app what’s currently in the store so we can reach feature parity first.  The attached document pretty much outlines the scope of the API calls I need to get going.

Is the attached document that Kirsty sent me a month ago with all its API calls still correct?

Yes they are – and with the check in flow you don’t need to worry about transferring ID and keeping state this has already been done with these calls on tibco – you just have to pass a convesationId that comes back

There are a couple of things unclear to me from the attached doc?

* When does use /findbooking and when /retrievebookingix and when /bookings?

So DDCI calls only work for bookings that are eligible for check-in (24 hour prior to scheduled departure – obviously this is a problem if you want to store your booking on the app but you flying in a couple of months. So the retrievebookingix will retrieve the booking from Sabre regardless if it is eligible for check-in or not and allow it to be saved in the app.  This call is made for this booking every time you go into the app in case you changed your flight. Findbooking can then be called when you know the flight on your booking could be eligible for checkin – 24 hours to check-in

* The attached document says that /retrievebookingix requires a **deviceId**, where does the deviceId come from?

When you first download the app we store the device id and type of device –you can then add preferences against this phones KMA app

When you call retrivebookingIX and pass this device id it is used to store the booking to this phones KMA app

* Are the **SessionToken** is hard coded like they describe in the document?

These sessionTokens are used for other internal calls (they do change between cert/prod)

* For the /RegisterDeviceId call,  where does the **DeviceIdentity** come from?

When you first download the app we store the device id and type of device –you can then add preferences against this phones KMA app- this includes the keys needs for push notifications

* The document describes the following return data  for these calls, but it doesn’t describe their purpose or timing:
  + /RegisterDeviceId,   -register the app to the device (also type of device)
  + /SaveDevicePreference – on the app can choose push notifications, gate changes, delays ( I think this was pushed out of scope)
  + /GetDeviceNotificationPreference – retrieves the device preferences saved
  + /SaveDeviceBooking – this saved the booking to the device ( I believe the retrievebookingIx replaces this call as it does this now)
* The attached doc, describes the return data for the **opsapi** calls ,but it doesn’t say when it must they must get called.  Eg /GetFlightSchedule – the app should not need to use GetflightSchedule – think this was just a call that was available
* Seems like the app must do quite a few calls to populate a single card on the UI eg. /GetFlightTerminal + /GetFlightGate +/ GetFlightStatus, rather than just a single call to return the flight info. In other words the app is quite chatty with backend.

The app keeps checking for these things as they change at different times – but yes there is nothing stopping us added another call that returns this information all at once

Is there a wiki page with more comprehensive info or stories that describe which calls happen on which screens?

Unfortunately not, but I also think this can be defined on the new app a lot better. They were stuck with limitations of when and how to refresh. It might be worth having a session where walk thought the current app and we can talk about the scenarios highlighted in this document. Especially around the check-in rules.

In my opinion this still needs an orchestrated level which should make KMA app interaction with the API simpler. I am looking at  this for the  whatspp and maybe it is smoothing we should do only once?

To get the API layer going, what do think is the order of the calls should tackle which make logical sense?

-          [api.kulula.com/retrievebookingix](https://protect-za.mimecast.com/s/VCc_C8qA64twj36gInASa8?domain=api.kulula.com)

-          [api.kulula.com/findbooking](https://protect-za.mimecast.com/s/KT93C98BQ4hYmnkjsEpsaT?domain=api.kulula.com)

-          [api.kulula.com/Update](https://protect-za.mimecast.com/s/TbdvC0gpVQcr2vGzi23A0P?domain=api.kulula.com) Passenger

-          [api.kulula.com/viewseatmap](https://protect-za.mimecast.com/s/YHBzCg5KLzCYl6AJs3XA6S?domain=api.kulula.com)

-          [api.kulula.com/Selectseats](https://protect-za.mimecast.com/s/8X09Cj2MqDsyjEn0I1tyFn?domain=api.kulula.com)

-          [api.kulula.com/checkin](https://protect-za.mimecast.com/s/3bp8Ck5KZECqnlOvI8d2Dg?domain=api.kulula.com)

-          [api.kulula.com/notifications/pass/getboardingpass](https://protect-za.mimecast.com/s/B7-3ClOL5GtzoG2lSVFo8b?domain=api.kulula.com)

Regards