**08 Claims Team - Code Walkthough ntt-service part1 - 20250307 1336-1**

0:02  
OK.

0:03  
So essentially the coverages portion that we currently deal with respect to claims goes only as far as the benefits that they have with respect to hospital and the health screening benefit which is also kind of a coverage for them, OK.

0:20  
That when we talk about certificates, what we mean is there are intake experiences wherein we actually display the certificate as a PDF, right.

0:31  
We give a link and when they click on that link, we open up a BLOB which is their certificate which would have their benefit details.

0:39  
That is the extent to which we actually cover when it comes to coverage or a benefits way, OK.

0:48  
The rest of the portion will be covered by a different team access, benefits and coverage.

0:53  
Just ABC team, they are planning to have the their KT sessions once they claim KT sessions are over next week.

1:04  
OK, Thanks Lokesh.

1:07  
Got it.

1:08  
Yeah.

1:12  
I'm going to share my screen to let me know if it's visible to this.

1:22  
Thank you.

1:22  
All right.

1:23  
So any questions from yesterday's or previous sessions guys?

1:42  
Sorry.

1:46  
OK, All right, so today we are going to look at the intake flow.

1:50  
OK, so intake flow is nothing, but from claim Center when they click on file A claim based on which sort of product it is, we navigate.

1:59  
I mean, we actually make the yes once again, like before starting the KT like I have seen like one more repository with the CIA intake that is helpful to us or I I need the repository name.

2:17  
Do ping me and I'll check it out.

2:19  
OK, OK, thank you.

2:23  
All right.

2:23  
So in intake, when we actually click on the file acclaim option in claim center, we actually navigate them to the entity intake pages, right?

2:35  
It depends on which product that they have product category that they have selected, right?

2:41  
So it could be critical illness or it could be accident or hospital or it could be for a HSB claim.

2:47  
So based on that, we route them to the appropriate controllers right now to start with, whenever they click on file a claim for CI or an accident or hospital, the first screen that we show is a policy pop up screen.

3:01  
And this screen is shown only if the user has than one policies, right?

3:05  
How do we determine that we have all the information once the user logs in?

3:10  
We have the information in AH cloud product details, and this AH experience data controller actually taps into that specific data and determines the list of policies that the user has and then returns back to the UI, right?

3:25  
And let me go to the data.

3:29  
Yeah, so this is the controller and this get policy data is the endpoint that would be invoked for policy selection.

3:35  
And when you get into this, there will be logic around identifying the coverages, the status and the dependence who are part of the specific policy.

3:46  
All those would be retrieved and it will be sent back to UI so that you'll be able to see that policy pop up screen, right?

3:54  
And from there, the user would be selecting the policy under which they would need to file a claim.

4:00  
Or if the user has only one policy, we actually bypass this policy selection screen and we directly initiate the session.

4:09  
OK, so for once we initiate the session again based on which our product it is because we have different layouts I believe we know could have explored this, but in UI we have layouts for each and every intake.

4:22  
It would be more or less the same but you would see variations from 1 intake to another because it was a UX enhancement as and when we had something we did for that specific intake.

4:34  
For example CIA intake might have the save and close button at the bottom right corner.

4:38  
But then when we started accident, we got an heads up from ADA team that it's better to have a close icon at the top right corner instead of keeping a button at the bottom right, right.

4:49  
So things like that, it would be minute changes, but there will be changes in between these intakes, right?

4:57  
That being said, from back end perspective, the steps are more or less the same.

5:01  
I'm going to explain it for one intake and you can probably explore it for rest of the intakes and reach out if you have any questions.

5:08  
OK, so the first call is for initiate session.

5:11  
So that is going to be routed to California data Controller and the create session.

5:17  
Now what it does is we actually initialize a specific intake.

5:21  
Right now we also have a draft in place.

5:25  
Say for example in claim center, it is possible that the user has a draft and they want to resume the draft.

5:30  
So when we invoke this create intake session, we check whether the user has actually tried coming in through a resume flow or is it a new flow, right?

5:40  
How do we identify it?

5:41  
It's based on the intake ID, whether it is passed or not.

5:44  
If it is not empty, we will check whether the intake actually belongs to that specific user or not and then take them to the place where they closed at last, right.

5:56  
So that is where we would be taking them.

5:58  
And if the user is for a new starts a new intake, then we are going to baseline everything, right.

6:04  
So we are going to create first session ID, then identify the entity product and under tracking data we also have all these attributes.

6:13  
So these attributes are for an automated controller that we are currently working on to reduce the manual intervention.

6:19  
Say for example, at the end of the intake, if there is a complication and it got stuck in the status, instead of proceeding to the entity submitted or claim created status, we actually use that automated controller which currently my team member Mubasam is working on.

6:36  
So we use these attributes to determine how far the intake has progressed.

6:41  
Okay, so for that we would need to baseline all this data at the beginning itself for setting all these attributes to false.

6:49  
And then we would also try to extract the policy specific details.

6:53  
So in previous call which is get policy details, we extract all the policies and base details of those policies.

7:00  
Right?

7:01  
Now when it comes to initiate session, we need to extract the exact policy folder details so that we can pre populate those information in the UI, right?

7:10  
We have the first screen in CIA intake and accident intake, this policy folder information, right?

7:15  
We would be extracting those information.

7:17  
Then we would also extract details around their dependent name so that we can pre populate the information in the second page, right?

7:25  
And then we also extract their address information which we would be utilizing in payment preferences page, right.

7:32  
So all those would be retrieved and also put into AHCI intake collection.

7:37  
OK, so that is the collection, that's the master collection that we have for saving all the information that we collect from the user and to drive this intake experience on the whole, right?

7:48  
So once we gather all those information, we are going to send back a response to UI saying that this is the intake ID that has been created, OK?

7:56  
So that's what is going to happen within this, they create intake session, right?

8:02  
So get and process policy details.

8:05  
It will be so convoluted because it is being reused for all the type of intakes, especially when it is HSB or if it is a hospital intake, we also extract their additional benefits, right?

8:17  
Or we also validate whether it is HSB eligible or not or which I mean what kind of preference the user has with respect to that specific policy.

8:27  
All those would be expected through this specific class.

8:31  
OK, Now coming back to the initiate session, once we send it back saying that this is the intake ID created, we are not going to update the status.

8:40  
If you remember when we when I explained the intake flow, I told you that there are statuses like these are all the possible statuses, OK, So draft then comes submitted, then it is entity submitted or entity submit failed and then we have claim created.

9:12  
These are all the possible statuses for a claim inside this collection.

9:17  
OK, now draft is straightforward.

9:20  
It just means that it is a draft.

9:21  
Submitted is a intermediary status.

9:24  
Say for example, user has clicked on submit, when we are processing that specific submission in our back end, it will be in this state, right?

9:33  
And then we would be making a call to coordinate in order to generate the form.

9:37  
You'll be waiting for acknowledgement.

9:38  
And all those would be happening when the intake is in this state, right?

9:42  
So once we get back the acknowledgement, we fire up the submit claim.

9:46  
Once we have successfully submitted the claim, that is when the status would move to entity submitted, right?

9:51  
If there was a failure when we submit that specific claim, the status would be entity submit failed.

9:56  
OK.

9:57  
Now quick note to our AMS partners and the ANH team is if at all a draft gets stuck in submitted.

10:06  
Currently we are working on an automated controller and I'll be explaining it once we have implemented after this iteration is completed.

10:15  
But essentially it will actually intervene and retry to recreate the environment to retry all these aspects, anything that failed in between, right?

10:24  
So the manual intervention would be very very minimal from my experience.

10:29  
What I have seen is every three weeks or so, there will be one such instance wherein one got stuck probably because there was a batch that was running in quadient and all our retry mechanisms failed.

10:40  
OK, so that in those cases you would need to manually intermede and submit the, I mean mimic as if the user has submitted the intake again, right?

10:51  
So there is an endpoint and it would be available in AMS special.

10:58  
Inside this caution right APIs, you would see the resubmit trigger endpoints that we have in place which can be leveraged using the ingress IP we have.

11:08  
This is an ultimate option.

11:10  
I mean, if we have no other option, only then you would need to go go through this and I have given some specifics to Sheetal and Sheetal, I would need your help to put put it down as a knowledge article.

11:24  
OK, right.

11:27  
Thank you, Sheetal.

11:27  
So this is something that he would share probably once all this KTS are done, I'll also explain the automated scheduler that we have in place to reduce this manual intervention.

11:37  
And he would also be sharing that knowledge article by then hopefully.

11:41  
Let's see.

11:42  
OK now apart from this.

11:45  
Now first we have created the session and I have explained you all the possible statuses.

11:49  
Now the further steps are more or less the same.

11:53  
OK, so let me quickly log in into one of the IDs.

12:04  
I can walk you through the process.

12:32  
OK, so this is the prescript page.

12:34  
So by OK, let me actually start it over and show you the network tab as well.

12:42  
OK, I'm going to start a fresh claim.

12:55  
And if you look at this right now, we would have bypassed the policy selection page.

13:04  
Why?

13:04  
Because this policy selection data event, right?

13:07  
If you look at the preview, it has only one policy.

13:11  
So we have bypassed that policy selection and we immediately initiated this California intake session event, right?

13:17  
So essentially what this intake session is going to respond to versus after creating the session ID, it is going to return back as this session ID.

13:26  
This is the intake ID that we'll be using throughout this flow to identify this specific user, right?

13:32  
And this, so this intake ID is something that we have in place to more or less to confuse people who are trying to intervene or trying to manipulate our system, OK?

13:44  
So this is the session ID that we would be actually using.

13:47  
And this is something temporary.

13:48  
We assign it to our intake, which would be used more or less like camouflage.

13:55  
OK, If someone is trying to access specific intake through this, it will work only during specific timings and only when this specific user is trying to access it.

14:05  
If someone else is trying to access it, it will not allow them to do so, right?

14:09  
And after this specific intake session, we are landing the user on prescript page, right?

14:14  
Once we get a success message saying that the session was created.

14:18  
Now for this prescript to load, we obviously need some data, right?

14:22  
And that is where all the GET calls come.

14:24  
OK, so in this controller you would notice that all the get user data is actually based on the section name.

14:33  
OK, so all this get user data have the same specific structure.

14:38  
When it comes to the request, it is going to have an intake ID.

14:41  
It is going to have the section name, step name and based on the section name, we actually route the user to the appropriate service layer to retrieve the content, right?

14:52  
So how do we retrieve the content?

14:53  
So for content to be retrieved, we have a specific flow which is nothing but first it is going to go to AH component config.

15:03  
OK, now here for each and every step name and section name, we have the appropriate page and content identifier, right?

15:11  
So if you remember when I was explaining the hydration flow in our claims view service, I would have highlighted the fact that we actually rely on a different collection to identify page and content identifier and then frame the AEM request, retrieve the data and send it back to the template, right?

15:31  
Similarly here we use this collection to look up which all AEM data is needed.

15:37  
It could be one or more and we consolidate the A request and then make an AEM call and once we get back the AEM response, we put it as part of the response.

15:48  
So if you look at our response, it will always have standard of two pieces.

15:52  
One is the response data node and the other one would be AEM data node.

15:57  
The AEM data is going to have the content that is necessary in order to showcase the a the content for the specific page and the response data would have all the information that the user entered in previous step or any data that we would need to pre populate.

16:15  
All this would be part of this response data.

16:18  
So till now we are still in pre script page right?

16:22  
So let's move on to the next page, which would be policy holder.

16:26  
If you look at this, we get all this data, right?

16:28  
So these data would be available as part of our response data.

16:35  
This is a common content.

16:37  
This would have that information.

16:40  
You see all this information coming from the back end and the AEM content is coming in this specific node.

16:46  
It is the same format for all the intakes, all the steps right now when you try to, I mean whenever you click on next right in all these intakes, there will be a save data event.

16:57  
Right now for save data, we have different, OK, we have different endpoints created.

17:04  
And if you look at the California data controller, you would see that there is a specific save endpoint for each and every page, right?

17:12  
It could be policy holder or patient info, diagnosis information or physician.

17:16  
It is specific to that intake alone and the request body.

17:20  
The format obviously is going to change, right?

17:23  
So now let's look at our next.

17:26  
I'm going to select this.

17:28  
I'm going to.

17:30  
So if you look at this now, you see the save data event.

17:34  
The payload is going to be this data, right?

17:38  
Whatever SSN that we entered in the first page and the preview is going to be this.

17:45  
We are just saying that it was successfully updated.

17:48  
That's the response that we expect from the saved data right now.

17:51  
For the next page, which is the diagnosis page, if you look at the response information, we have this BLOB ID and a document which is nothing but our certificate.

18:02  
So this is what I was referring to certificate, right?

18:05  
If we have a valid certificate attached to this specific policy, we would be showing this link right?

18:10  
When we click on it, it is going to open up as a BLOB which is the certificate that we have for this specific user.

18:17  
OK.

18:17  
Now next one is let's move on to the next page.

18:28  
Yes, Yeah, just a question, whenever we are doing the next, are we triggering a save call for each particular page?

18:36  
Yes, OK, yes.

18:40  
And that is where all these endpoints comes into picture, right.

18:43  
So for each and every step, that is a save call.

18:46  
And inside the save, we would be saving the data.

18:50  
And also say for example, there might be places where we would use the information that the user has entered to determine the list of documents that they might need, right?

18:59  
Or we might drive different page experience based on the data that they have feeded in.

19:05  
In this specific step.

19:07  
All those would be taken care inside this save method of the service layer.

19:13  
Save data service layer.

19:17  
OK, now let's go on.

19:35  
Now coming to the upload portion, right now, the way that we showcase these required documents is based on conditions, right?

19:45  
How do we determine what documents has to be shown?

19:48  
We have a specific logic in order to retrieve this information.

19:52  
So how do we do with this?

19:54  
We have a specific look up in AH config collection right?

19:58  
For three intakes we have dock upload and for all the three intakes we have those look UPS.

20:04  
OK, it would be this default hospital right?

20:08  
So we are saying.

20:09  
If the user was hospitalized, then these are all the required doc list, right?

20:13  
So what is doc 42?

20:15  
How do we look that up?

20:16  
We have a entity doc look up wherein we would have added an entry for each and every doc.

20:22  
So say for example tomorrow, you would need to modify any of these.

20:25  
So you might need to make an update in this specific collection entity doc look up or the appropriate conflict of that specific intake.

20:39  
So if it is accident, if if the user selected motor vehicle, we are saying that these two are the required documents, right?

20:45  
So we actually extract all this information and send it over to the UI.

20:49  
OK, so now if you look at this get user data event in the response data, you see the required documents is dock 1 and dock 27.

20:58  
So what is the significance of this?

21:00  
This is more of a polite nudge to the user requesting them to upload the documents and tag it accordingly, right?

21:08  
So say for example I upload the specific document, right?

21:16  
So we actually insist that the user select or tag that specific document.

21:23  
OK, now see when I click next, we do not allow unless they tag it.

21:27  
So they would need to tag it either with this or this, or they can use the other option to enter any specific information that they want to enter or tag the document task, right?

21:36  
So this tag is not something that we send to entity.

21:39  
It is more or less a polite way of informing the user that, hey, it's better if you upload these documents right?

21:46  
Now how does this document get uploaded if you click on view, it will actually download the document that you uploaded and also you have an option to delete the document right So how often this happens is it happens in AH dock management controller.

22:03  
OK, so this is common for both claim center dock upload as well As for any other dock upload within different intakes right?

22:11  
So we have an upload dock data which actually runs through the virus scan and also sanitizes the document and finally it uploads the document to a block.

22:23  
OK now if you look at the network tab you would notice that in the upload doc event the response would be a document ID.

22:32  
This is a temporary document ID that we generate when we upload because we allow multi dock upload.

22:37  
In order to tag which responses what in the UI, we use this specific temp doc ID.

22:43  
OK so this is how the upload works and when we download again the call gets triggered to AH doc management controller and it cross verifies if the user is who he says he is right?

22:54  
Because 1 user will not should not be able to access the document uploaded by different users.

22:59  
So we have those kind of validations here and once that is validated we will be sending back the stream and it will be downloaded from the UI.

23:07  
OK now delete again.

23:10  
It means that it just going to delete the BLOB and also deletes the entry.

23:15  
Now when we upload all these documents right, all these documents get saved in AH doc data collection.

23:23  
AH docs data collection.

23:26  
This is where it gets saved.

23:28  
OK, let me just and when you delete it, the entry will be deleted from this collection also, right?

23:34  
It deletes from the BLOB and also it deletes from this collection as well.

23:40  
So this is all about the dock upload and also we have an option temporary ID also location.

23:48  
Yes we will be OK it is wide.

23:52  
After the transaction it is wide but we still wanted to save it for cross verification purpose.

23:57  
We have noticed that people have raised concerns around dock upload just one or two users once in a while and we suspected it could be because of their bandwidth.

24:07  
But still we wanted to to save that information just in case if we want to triage it using the logs.

24:15  
OK.

24:16  
And when we come to this page, we have this routing number validation.

24:21  
So whenever we enter a routing number, it is going to make a call and this specific call goes to financial some controller.

24:30  
Let's start with yes, financial institution controller.

24:33  
So here this what we do is we actually intern call an RMT endpoint which goes to it is not MG DAS, it is Midas.

24:45  
OK, So and I have added that endpoint also as part of this external system calls under RMT, you would notice there is an institution, sorry, this financial institution call and see this is the response that we would get back.

25:05  
And this is the organization name which we passed down to UI and it in turn is being displayed in the UI.

25:15  
OK, The user can also select by mail option, right?

25:18  
So by for by mail, we actually pre populate all this information from where do we get this information?

25:23  
We get it from the first step where we actually extract the policy related policy folder address information, save it to the intake right from there, we actually get this information.

25:33  
And if the user wants to update this, we update, right?

25:38  
We use this the updated information to send it sender claim.

25:43  
But we will not be making any update to their actual address on file, OK.

25:48  
The address that they enter here, the updated address, we will be using that only while submitting the claim.

25:53  
We are not updating their address anywhere, OK.

25:59  
And then we have communication preference step again.

26:02  
This communication preference step would come in only if they have not, either they should, they would have not given their E consent or for this specific product, the user has opted out of e-mail communication, right?

26:17  
Only then we will be showing this communication preference, else it will be bypassed.

26:20  
OK, now either you can leave it as such or you can opt in and then proceed right?

26:27  
At this point, say for example, the user has updated the communication preference, we do not update the communication preference right away, right?

26:34  
That gets processed only at the final submit.

26:39  
OK, what is the final submit?

26:41  
Final submit is nothing but in this part right when they click on submit clay that is when it is going to be updated.

26:48  
Now if you notice here, there is the specific link that we want the users to click open right.

26:53  
So we have a condition in UI to make sure that the user actually clicks on that link.

26:58  
OK, You see we are not allowing them to submit unless they click on this link right?

27:03  
And we do not know if they are going to read through all this or not, but we need to make sure that they are clicking on that link and only then they'll be able to submit.

27:11  
I believe the ANH team would have a feature somewhere saying that this would be removed and instead this content would be added dynamically to this specific spot.

27:23  
I think so.

27:24  
So again, probably you guys would figure out a way to develop that.

27:29  
But yeah, eventually this link would be removed.

27:33  
OK.

27:34  
Now, once the user clicks on this final submit, that is when the final submit process happens.

27:39  
Now the final submit, as we looked at earlier, right, it has a whole bunch of operations to it, which is generating the claim form, sending it to entity.

27:52  
And based on the acknowledgement, sorry, we update the consent and the communication preference.

27:58  
Then we wait for an acknowledgement.

28:00  
Once the acknowledgement comes in, we submit the claim.

28:04  
All those right?

28:05  
How does that happen?

28:07  
So I think we are at time, I don't want to get into the submit flow just yet.

28:13  
Tomorrow we will sorry.

28:14  
The next session we will cover the final submit.

28:17  
Rest all are straightforward and this is a pure Java app.

28:22  
I assume you guys will be able to debug.

28:25  
I have already shared collections for two intakes and we have put together the rest of the two intakes.

28:31  
But I wanted to make sure that it works fine in in our local and I have not missed anything.

28:36  
So once I have verified that I will give an updated collection to you guys or I just be in hospital.

28:43  
We have added it but I wanted to cross a way before I share it.

28:48  
I don't want to give you a broken collection.

28:54  
OK.

28:54  
So any questions?

28:59  
Hey, Lokesh, Yeah, this is Huabe.

29:01  
I have a question about the dock upload.

29:03  
So you mentioned the docker type and my question is related to the Docker version.

29:09  
And do we have any business case we need to maintain the version of the document for for one specific document type?

29:17  
And how if, if we have how to manage it?

29:22  
You mean the type of document that the user uploads, PDF, Excel, things like that.

29:27  
Or when you select a document tab, you upload the document.

29:32  
And my question is later probably we will update the how to say the format of the document, maybe we call it a version one, version 2.

29:41  
And do we have this kind of A use case and how to manage it?

29:45  
No, we do not have versioning of documents.

29:49  
Whatever document the user uploads, we push it to our BLOB storage and once the user submits, we retrieve the document back from BLOB storage and send it back to entity.

30:00  
Once entity confirms that it is uploaded, we can delete it from our side.

30:05  
Because we we currently we do not have a feature wherein we allow the user to look at the documents that they have uploaded post their intake.

30:14  
Once they submit the intake, that's it, right.

30:17  
So I believe down the line probably we would have a feature wherein we would be showcasing the documents that they uploaded during the intake process or so.

30:25  
But as of now we do not have that capability.

30:30  
OK, thank you.

30:31  
Lokesh, I have another question.

30:33  
Just you show that browser, I see there is a production code Gidd, yes, is ANH level.

30:44  
That's a specific CI units level.

30:48  
Critical units, sorry.

30:54  
OK, So GID is a critical illness product code.

30:57  
So for under critical illness we have three different products and under accident we have two different products and hospital has one product and then there is cancer and then there is a special HSB benefit.

31:08  
Right now under critical illness there is CNIC, i.e.

31:12  
and GID.

31:12  
So GID is one of the code and I believe I have, I already added the product code mapping if it is not already there, probably in the next cut off my OneNote you would have that information as well.

31:27  
OK, thanks Lokesh.

31:28  
Yeah, got it.

31:30  
Thank you.

31:30  
Thank you.

31:31  
And I remember seeing a question and yeah, it has been answered.

31:36  
Yes, the NMB limit is accurate.

31:40  
Looks I'm on this side.

31:41  
For example, we are asking the user to upload the death certificate, right And user has uploaded a different, different document.

31:50  
So how we are validating that document is we are not validating any documents.

31:56  
We say whatever they upload unless it is it has virus OK and we send it over to NTT.

32:02  
The NTT OPS team would then be checking their documents, processing their claim.

32:06  
All those happens only on entity site to which we have no visibility to.

32:11  
And after that we will get the status entity submitted.

32:17  
No, we get entity submitted as soon as we submit a claim.

32:20  
The we update the status to entity submitted once we submit the claim to entity right.

32:26  
The if the submission fails, we update it as entity submit failed.

32:29  
I will get to that part of probably in our next session.

32:33  
I have to drop now guys if you have any other questions ping me OK?