**PENDING TASK**

**1. ACH dragged element task(Today it will be done)**

**2. Convert doc file to PDF(Did by Pradeep but not working properly)(Need two days)**

**3. Download doc file from google drive(PDF file is done)(If second task done then it will also be done in Google drive(One day Task))**

**4. Implement upload pdf file from Google drive on getSignature page is pending (Create doc template is done)(4 hours)**

**5. Implement new UI privided by Chithira**

**6. In Task createQuoteSignatureRequest -> if someone request from salesforce with JWT\_TOKEN then athenticate for the UI application(1 day with testing)**

**7. Backend signatureDetail page should be same as react js signatureDetail page (3 hours)**

**8. Give option to staff to delete esignatureRequest entity(4 hours)**

**9. Need detail of doc template on React js app(3 to 4 hour)**

**10 While adding recipients we have two optiion "Signer" and "CC". Needs to handle CC option(3 hour)**

**Step to implement Create Quote Signature Request in Esign**

1. **New Html Quote Template**

 A. User will come and create the quote template with HTML editor

 B. And then quote will be saved as html value in DB

 C. We will groovy expression language to resolve dynamic values in Html



2. **Create Quote Signature Request PDF from Html Template**

Steps

1. Create he needs to get “jwtToken” by login with api URL

[https://acme.invoicingforsalesforce.com/esign\_backend/api/login(POST)](https://acme.invoicingforsalesforce.com/esign_backend/api/login%28POST%29)

**Request Json**

{ "username":"api\_user", "password":"xxxxxxxx", "domain":"acme"}

**In Response it will get jwtToken**

{

 "successful": true,

 "currentUser": {

 "username": "test\_admin",

 "authorities": [

 {

 "authority": "ROLE\_ADMIN"

 }

 ],

 "accountNonExpired": true,

 "accountNonLocked": true,

 "credentialsNonExpired": true,

 "enabled": true,

 "firstName": "Acme",

 "lastName": "Acme",

 "domainPrefix": "Acme",

 "workPhone": "8529637415",

 "emailPrimary": "deepakgpt1010@gmail.com"

 },

 "jwtToken": "eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJ0ZXN0X2FkbWluIn0.fQ5MmK9OafT-BLvCLS0KocXQgMTf-mgnkIbb80QziMqQD9p4KsYVkBQHcrFKtvqfR4afm5FdTiiJZyn7nY9Dsg"

}

1. After getting jwtToken he will request our create Quote Signature Request Api URL by appending header with **X-AUTH-TOKEN=#jwtToken#**

<https://acme.invoicingforsalesforce.com/esign_backend/api/signature/createQuoteGetSignature>

OR

https:////acme.invoicingforsalesforce.com/esign\_backend/api/signature/createQuoteGetSignature?templateId=10

**Requested JSON**

{

 "quotelineItems": [

 {

 "attributes": {

 "type": "QuoteLineItem",

 "url":

"/services/data/v43.0/sobjects/QuoteLineItem/0QL0L000003gIELWA2"

 },

 "Id": "0QL0L000003gIELWA2",

 "LineNumber": "00000107",

 "ListPrice": 15000,

 "Product2Id": "01tE0000001XhCAIA0",

 "blusyn\_\_UnBundled\_Products\_\_c": "0Q00L000001tKbaSAE",

 "Quantity": 1,

 "UnitPrice": 15000,

 "HasSchedule": false,

 "Subtotal": 15000,

 "TotalPrice": 15000,

 "QuoteId": "0Q00L000001tKbaSAE",

 "Product2": {

 "attributes": {

 "type": "Product2",

 "url":

"/services/data/v43.0/sobjects/Product2/01tE0000001XhCAIA0"

 },

 "Name": "GenWatt Propane 100kW",

 "ProductCode": "GC3020",

 "Id": "01tE0000001XhCAIA0"

 }

 },

 {

 "attributes": {

 "type": "QuoteLineItem",

 "url":

"/services/data/v43.0/sobjects/QuoteLineItem/0QL0L000003gIELWA2"

 },

 "Id": "0QL0L000003gIELWA2",

 "LineNumber": "00000107",

 "ListPrice": 15000,

 "Product2Id": "01tE0000001XhCAIA0",

 "blusyn\_\_UnBundled\_Products\_\_c": "0Q00L000001tKbaSAE",

 "Quantity": 1,

 "UnitPrice": 15000,

 "HasSchedule": false,

 "Subtotal": 15000,

 "TotalPrice": 15000,

 "QuoteId": "0Q00L000001tKbaSAE",

 "Product2": {

 "attributes": {

 "type": "Product2",

 "url":

"/services/data/v43.0/sobjects/Product2/01tE0000001XhCAIA0"

 },

 "Name": "GenWatt Propane 100kW",

 "ProductCode": "GC3020",

 "Id": "01tE0000001XhCAIA0"

 }

 }

 ],

 "quote": {

 "attributes": {

 "type": "Quote",

 "url": "/services/data/v43.0/sobjects/Quote/0Q00L000001tKbaSAE"

 },

 "Id": "0Q00L000001tKbaSAE",

 "AccountId": "0010L00001thpJYQAY",

 "AdditionalAddress": null,

 "BillingAddress": {

 "city": "Gladstone",

 "country": "United States",

 "geocodeAccuracy": null,

 "latitude": null,

 "longitude": null,

 "postalCode": "64118",

 "state": "Missouri",

 "street": "Central"

 },

 "BillingName": "Spectrum solutions",

 "ContactId": "0030L00001ycRy4QAE",

 "Description": "TestDescription",

 "Discount": 0,

 "Email": "smithl@spectrum.com",

 "ExpirationDate": "2018-08-30",

 "Fax": "0000",

 "GrandTotal": 25000,

 "LineItemCount": 2,

 "OpportunityId": "0060L00000ohmZqQAI",

 "Phone": "(658) 745 5496",

 "Name": "TestQuote",

 "QuoteNumber": "00000024",

 "QuoteToAddress": null,

 "ShippingAddress": {

 "city": "Gladstone",

 "country": "United States",

 "geocodeAccuracy": null,

 "latitude": null,

 "longitude": null,

 "postalCode": "64118",

 "state": "Missouri",

 "street": "central"

 },

 "Status": "Draft",

 "Subtotal": 25000,

 "IsSyncing": false,

 "Tax": 0,

 "TotalPrice": 25000,

 "Account": {

 "attributes": {

 "type": "Account",

 "url":

"/services/data/v43.0/sobjects/Account/0010L00001thpJYQAY"

 },

 "Name": "Spectrum solutions",

 "Id": "0010L00001thpJYQAY"

 },

 "Contact": [

 {

 "attributes": {

 "type": "Contact",

 "url":

"/services/data/v43.0/sobjects/Contact/0030L00001ycRy4QAE"

 },

 "Name": "Lynn Smith",

 "Email": "deepak@blusynergy.com",

 "Phone": "(658) 745 5496",

 "Id": "0030L00001ycRy4QAE"

 },

 {

 "attributes": {

 "type": "Contact",

 "url":

"/services/data/v43.0/sobjects/Contact/0030L00001ycRy4QAE"

 },

 "Name": "Jon Smith",

 "Email": "deepak@blusynergy.com",

 "Phone": "(658) 745 5496",

 "Id": "0030L00001ycRy4QAE"

 }

 ],

 "Opportunity": {

 "attributes": {

 "type": "Opportunity",

 "url":

"/services/data/v43.0/sobjects/Opportunity/0060L00000ohmZqQAI"

 },

 "Name": "Spectrum Solutions",

 "Id": "0060L00000ohmZqQAI"

 }

 }

}

**HOW AM I PROCESSING ABOVE RESPONSE**

Steps

1. Getting html template from DB for a org
2. Then replacing template groovy expression value with requested JSON value
3. Then converting that html into pdf
4. Then finding coordinates of /s1/,/s2/ … word in generated PDF
5. After getting coordinated , am replacing /s1/,/s2/ … with “Signature” word in PDF
6. After Getting pdf I am converting pdf pages into images as you know
7. Now I am creating dragged element on the PDF pages according to found coordinates

**Response JSON**

{

 "isError": false,

 "esignUrl": "http://acme.localhost:8080/login/authenticateByToken?token=eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJ0ZXN0X2FkbWluIn0.fQ5MmK9OafT-BLvCLS0KocXQgMTf-mgnkIbb80QziMqQD9p4KsYVkBQHcrFKtvqfR4afm5FdTiiJZyn7nY9Dsg&eSignId=305",

 "esignId": 305,

 "message": "Signature template has been successfully saved"

}

Sales Force developer will show response esignUrl in iframe



