Скрипт 1. Автоматичне визначення підрозділу ініціатора та email

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Опис задачі (Що необхідно зробити, Як буде тестуватись, Очікуваний результат)** | | | | |
| При створенні карточки автоматично має заповнюватись:  1) поле OrgRPEmail даними із орг структури Ініціатора документу, якщо такі дані є в орг структурі.  2) поле Branch даними структурного підрозділу рівня ‘1’ Ініціатора (назва структурного підрозділу). При цьому Ініціатор може вибрати інший підрозділ рівня <1> з даного поля, якщо інформацію потрібно змінити | | | | |
| **Опис сценарію (що потрібно реалізувати за допомогою JS. Обов’язково використовуючи «ЯКЩО» -> «ТО» -> «ІНАКШЕ» і тригером (умови(a), або дія при яких має виконатись сценарій)** | | | | |
|  | | | | |
| **Атрибути картки, які задіяні в JS** | | | | |
| **Назва** | **Код** | **Тип в SP** | **Сист.атрибут** | **Код довідника** |

Скрипт 2. Зміна властивостей атрибутів

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Опис задачі (Що необхідно зробити, Як буде тестуватись, Очікуваний результат)** | | | | |
| При призначенні завдання EnterResults наступні поля мають ставати видимими та обов’язковими для заповнення:  SpesificationResult (Результат розгляду ТУ засіданням)  NumberProtocol (Номер протоколу)  DateProtocol (Дата протоколу)  Якщо Відповідальний за задачу не заповнює якесь поле при виконанні завдання, то система має інформувати:  Внесіть значення в поле <найменування поля>  Після виконання завдання в поля неможливо внести зміни. | | | | |
| **Опис сценарію (що потрібно реалізувати за допомогою JS. Обов’язково використовуючи «ЯКЩО» -> «ТО» -> «ІНАКШЕ» і тригером (умови(a), або дія при яких має виконатись сценарій)** | | | | |
|  | | | | |
| **Атрибути картки, які задіяні в JS** | | | | |
| **Назва** | **Код** | **Тип в SP** | **Сист.атрибут** | **Код довідника** |

Скрипт 3. Зміна властивостей атрибутів

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Опис задачі (Що необхідно зробити, Як буде тестуватись, Очікуваний результат)** | | | | |
| При призначенні завдання RegisterSpesification наступні поля мають ставати видимими та обов’язковими для заповнення:  RegDate  RegNumber  Registraion  Якщо Відповідальний за задачу не заповнює якесь поле при виконанні завдання, то система має інформувати:  Внесіть значення в поле <найменування поля>  Після виконання завдання в поля неможливо внести зміни. | | | | |
| **Опис сценарію (що потрібно реалізувати за допомогою JS. Обов’язково використовуючи «ЯКЩО» -> «ТО» -> «ІНАКШЕ» і тригером (умови(a), або дія при яких має виконатись сценарій)** | | | | |
|  | | | | |
| **Атрибути картки, які задіяні в JS** | | | | |
| **Назва** | **Код** | **Тип в SP** | **Сист.атрибут** | **Код довідника** |

Скрипт 4. Зміна властивостей атрибутів

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Опис задачі (Що необхідно зробити, Як буде тестуватись, Очікуваний результат)** | | | | |
| При призначенні завдання MakeDatePayment наступні поля мають ставати видимими та обов’язковими для заповнення:  PaymentOption (Вид оплати)  Роль за розрізом.  Якщо Відповідальний за задачу не заповнює якесь поле при виконанні завдання, то система має інформувати:  Внесіть значення в поле <найменування поля>  Після виконання завдання в поля неможливо внести зміни. | | | | |
| **Опис сценарію (що потрібно реалізувати за допомогою JS. Обов’язково використовуючи «ЯКЩО» -> «ТО» -> «ІНАКШЕ» і тригером (умови(a), або дія при яких має виконатись сценарій)** | | | | |
|  | | | | |
| **Атрибути картки, які задіяні в JS** | | | | |
| **Назва** | **Код** | **Тип в SP** | **Сист.атрибут** | **Код довідника** |

Скрипт 5. Зміна властивостей атрибутів

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Опис задачі (Що необхідно зробити, Як буде тестуватись, Очікуваний результат)** | | | | |
| При призначенні завдання ReceiptFunds наступні поля мають ставати видимими та обов’язковими для заповнення:  InvoiceStatus ( Статус оплати Замовником)  Роль за розрізом.  Якщо Відповідальний за задачу не заповнює якесь поле при виконанні завдання, то система має інформувати:  Внесіть значення в поле <найменування поля>  Після виконання завдання в поля неможливо внести зміни. | | | | |
| **Опис сценарію (що потрібно реалізувати за допомогою JS. Обов’язково використовуючи «ЯКЩО» -> «ТО» -> «ІНАКШЕ» і тригером (умови(a), або дія при яких має виконатись сценарій)** | | | | |
|  | | | | |
| **Атрибути картки, які задіяні в JS** | | | | |
| **Назва** | **Код** | **Тип в SP** | **Сист.атрибут** | **Код довідника** |

Скрипт 6. Визначення ролі за розрізом

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Опис задачі (Що необхідно зробити, Як буде тестуватись, Очікуваний результат)** | | | | |
| Необхідно реалізувати автоматичне визначення Фінансової служби РФ, Технічної служби РФ (роль із розрізом) на основі даних структурного підрозділу із поля Branch.  Додаткова інформація: В системі налаштована роль Фінансова РФ, Технічна служба РФ із розрізами (для кожної філії – окремі відповідальні). Також в карточку додане приховане поле «Розрізи» (Sections) із системним атрибутом «Розрізи». | | | | |
| **Опис сценарію (що потрібно реалізувати за допомогою JS. Обов’язково використовуючи «ЯКЩО» -> «ТО» -> «ІНАКШЕ» і тригером (умови(a), або дія при яких має виконатись сценарій)** | | | | |
|  | | | | |
| **Атрибути картки, які задіяні в JS** | | | | |
| **Назва** | **Код** | **Тип в SP** | **Сист.атрибут** | **Код довідника** |
|  |  |  |  |  |

Скрипт 6. Передача ТУ для ознайомлення в зовнішню систему

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Опис задачі (Що необхідно зробити, Як буде тестуватись, Очікуваний результат)** | | | | | | | |
| Необхідно налаштувати передачу рахунку (тип документу - Spesification) для ознайомлення при виконанні завдання SendOutDoc.  // Відправлення на підпис в зовнішній сервіс eSign договору  //-------------------------------  function setDataForESIGN() {  debugger;  var registrationDate = EdocsApi.getAttributeValue('RegDate').value;  var registrationNumber = EdocsApi.getAttributeValue('RegNumber').value;  var caseType = EdocsApi.getAttributeValue('DocType').value;  var caseKind = EdocsApi.getAttributeValue('DocKind').text;  var name = '';  if (caseKind) {  name += caseKind;  } else {  name += caseType;  }  name += ' №' + (registrationNumber ? registrationNumber : CurrentDocument.id) + (!registrationDate ? '' : (' від ' + moment(registrationDate)  .format('DD.MM.YYYY')));  doc = {  DocName: name,  extSysDocId: CurrentDocument.id,  ExtSysDocVersion: CurrentDocument.version,  docType: "invoice",  docDate: registrationDate,  docNum: registrationNumber,  "File": "",  "parties": [  {  "taskType": "ToSign",  "taskState": "Done",  "legalEntityCode": EdocsApi.getAttributeValue('OrgCode').value,  "contactPersonEmail": EdocsApi.getAttributeValue('OrgRPEmail').value,  "signatures": []  },  {  "taskType": "ToSign",  "taskState": "NotAssigned",  "legalEntityCode": EdocsApi.getAttributeValue('ContractorCode').value,  "contactPersonEmail": EdocsApi.getAttributeValue('ContractorRPEmail').value,  "expectedSignatures": []  }],  "additionalAttributes": [  {  "code": "docDate",  "type": "dateTime",  "value": registrationDate  },  {  "code": "docNum",  "type": "string",  "value": registrationNumber  }  ],  sendingSettings: {  attachFiles: 'fixed', //, можна також встановлювати 'firstOnly' - Лише файл із першої зафіксованої вкладки(Головний файл), або 'all' - всі файли, 'fixed' - усі зафіксовані  attachSignatures: 'signatureAndStamp' // -'signatureAndStamp'Типи “Підпис” або “Печатка”, можна також встановити 'all' - усі типи цифрових підписів  }  }  EdocsApi.setAttributeValue({code: 'JSON', value: JSON.stringify(doc)});  }  function onTaskExecuteSendOutDoc(routeStage) {  debugger;  if (routeStage.executionResult == 'rejected') {  return;  }  setDataForESIGN();  var idnumber = EdocsApi.getAttributeValue('DocId');  var methodData = {  extSysDocId: idnumber.value  };  routeStage.externalAPIExecutingParams = {  externalSystemCode: 'ESIGN1', // код зовнішньої системи  externalSystemMethod: 'integration/importDoc', // метод зовнішньої системи  data: methodData, // дані, що очікує зовнішня система для заданого методу  executeAsync: true // виконувати завдання асинхронно  }    }  function onTaskCommentedSendOutDoc(caseTaskComment){  debugger;  var orgCode = EdocsApi.getAttributeValue('OrgCode').value;  var orgShortName= EdocsApi.getAttributeValue('OrgShortName').value;  if(!orgCode || !orgShortName){  return;  }  var idnumber = EdocsApi.getAttributeValue('DocId');  var methodData = {  extSysDocId:idnumber.value,  eventType:"CommentAdded",  comment: caseTaskComment.comment,  partyCode: orgCode,  userTitle: CurrentUser.name,  partyName: orgShortName,  occuredAt: new Date()  };  caseTaskComment.externalAPIExecutingParams = {  externalSystemCode: 'ESIGN1', // код зовнішньої системи  externalSystemMethod: 'integration/processEvent', // метод зовнішньої системи  data: methodData, // дані, що очікує зовнішня система для заданого методу  executeAsync: true // виконувати завдання асинхронно  }  } | | | | | | | |
| **Опис сценарію (що потрібно реалізувати за допомогою JS. Обов’язково використовуючи «ЯКЩО» -> «ТО» -> «ІНАКШЕ» і тригером (умови(a), або дія при яких має виконатись сценарій)** | | | | | | | |
| ЯКЩО призначене завдання SendOutDoc, ТО документ передається на підпис в зовнішню систему. | | | | | | | |
| **Інформування користувача** | | | | | | | |
| **Передача даних в зовн.систему** | | | | | | | |
| **Код зовн.системи** | | ESIGN1 | | **Метод зовн.системи** |  | | |
|  |  |  |  | **Завдання, яке активує передачу** | SendOutDoc | |
| **Атрибути картки, які задіяні в JS** | | | | | | | |
| **Назва** | | **Код** | | **Тип в SP** | **Сист.атрибут** | **Код довідника** | |