Скрипт 1. Автоматичне визначення email ініціатора рахунку та підрозділу

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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** |
| Email контактної особи Організації | OrgRPEmail | Багаторядковий текст | - |  |

Скрипт 2. Вирахування ПДВ

|  |
| --- |
| **Опис задачі (Що необхідно зробити, Як буде тестуватись, Очікуваний результат)** |
| Реалізувати обчислення суми ПДВ та суми без ПДВ за договір.  Приклад:  // Вирахування суми за послугу  function onChangeContract () {  var VATpercentage = 0;  var attrVATAmount = EdocsApi.getAttributeValue('ContractVATAmount');  var attrVATpercentage = EdocsApi.getAttributeValue('ContractVATPercent');  var attrContractAmount = EdocsApi.getAttributeValue('ContractAmount');  var attrAmountOutVAT = EdocsApi.getAttributeValue('ContractOutVAT');  switch (attrVATpercentage.value) {  case '20%': // if (x === 'если сумма НДС=20%')  var VATpercentage = 1.2;  break  case '7%': // if (x === 'если сумма НДС=7%')  var VATpercentage = 1.07;  break  }  if (attrVATpercentage.value === null || attrContractAmount.value === null) {  // если нет ставки НДС и суммы, то укажем ноль в сумме НДС и без НДС  attrVATAmount.value = 0;  attrAmountOutVAT.value = 0;  } else if (VATpercentage == 0) {  attrVATAmount.value = 0;  attrAmountOutVAT.value = attrContractAmount.value;  } else {  attrAmountOutVAT.value = Math.floor(100 \* attrContractAmount.value / VATpercentage) / 100;  attrVATAmount.value = attrContractAmount.value - attrAmountOutVAT.value;  }  EdocsApi.setAttributeValue(attrVATAmount);  EdocsApi.setAttributeValue(attrAmountOutVAT);  }  function onChangeVATpercentage() {  onChangeContract();  } |

Скрипт 3. Заповнення значення поля суми договору прописом

|  |
| --- |
| **Опис задачі (Що необхідно зробити, Як буде тестуватись, Очікуваний результат)** |
| Необхідно налаштувати, щоб при внесенні числового значення в поля ContractAmount в поле VATAmmountDescription автоматично записувалась сума прописом.  Якщо в поле ContractAmount внесено значення, то в полі VATAmmountDescription автоматично заповнюється сума прописом.  Наприклад, ContractAmount =13 735,10.В поле VATAmmountDescription автоматично записується інформація ‘тринадцять тисяч сімсот тридцять п'ять гривень, десять копійок’. |
| **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)** |
|  |

Скрипт 4. Заповнення інформації про додаткового підписанта

|  |
| --- |
| **Опис задачі (Що необхідно зробити, Як буде тестуватись, Очікуваний результат)** |
| Необхідно налаштувати, щоб при виборі в полі OrgAgentSurname2 Підписанта в полях OrgAgent2, OrgAgentPosition2, ActsOnBasisOrg2 заповнювалась інформація про уповноважену особу із довідника контрагентів.  При заповненні карточки Виконавцем в полі OrgAgentSurname2 вибирається уповноважена особа із випадаючого списку. Після виборі уповноваженої особи поля карточки OrgAgent2, OrgAgentPosition2 (посада), PositionOrgAgent2 (посада родовий відмінок) , ActsOnBasisOrg2 (на підставі чого діє) заповнюються інформацією про дану уповноважену особу із довідника контрагентів.  При зміні Виконавцем значення в полі OrgAgentSurname2 в полях карточки OrgAgent2, OrgAgentPosition2, PositionOrgAgent2, ActsOnBasisOrg2 змінюється інформацію на актуальну для вибраної уповноваженої особи.  Якщо в полі вибирається особа не із довідника контрагентів, то має виводитись помилка «Внесіть коректного підписанта» |

Скрипт 5. Заповнення ініціалів підписантів

|  |
| --- |
| **Опис задачі (Що необхідно зробити, Як буде тестуватись, Очікуваний результат)** |
| Інформація в полі InitialAgent1 має заповнюватись даними <Анастасія МОНАСТИРСЬКА>. Поле із даними ПІБ - OrgAgentSurname1.  Інформація в полі InitialAgent2 має заповнюватись даними <Анастасія МОНАСТИРСЬКА>. Поле із даними ПІБ - OrgAgentSurname2  Якщо значення в полях OrgAgentSurname1 та OrgAgentSurname2 пусті, то поля InitialAgent1 та InitialAgent2 теж мають бути пустими. |

Скрипт 6. Визначення ролі за розрізом

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Опис задачі (Що необхідно зробити, Як буде тестуватись, Очікуваний результат)** | | | | |
| Необхідно реалізувати автоматичне визначення Фінансової служби РФ, Юридичної служби РФ (роль із розрізом) на основі даних структурного підрозділу із поля Branch.  Додаткова інформація: В системі налаштована роль Фінансова служба РФ, Юридична 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 7. Зміна властивостей атрибутів

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Опис задачі (Що необхідно зробити, Як буде тестуватись, Очікуваний результат)** | | | | |
| При виконанні завдання SignPaperContract або RegisterContract (тут виконавець буде за розрізрозом (можливо, це важливо)) поля RegDate та RegNumber мають стати видимими та обовязковими для заповнення.  Якщо Відповідальний за задачу не заповнює дані поля при виконанні завдання, то система має інформувати:  Внесіть номер та дату реєстрації договору. | | | | |
| **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 8. Інформування замовника про дату та номер реєстрації договору

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Опис задачі (Що необхідно зробити, Як буде тестуватись, Очікуваний результат)** | | | | |
| Після позитивного виконання завдань RegisterContract в зовнішню систему Єсайн має передаватись коментар:  ('DocKind') прийнято та зареєстровано за № '+ EdocsApi.getAttributeValue('RegNumber').value + ' від ' + moment(new Date(EdocsApi.getAttributeValue('RegDate').value)).format('DD.MM.YYYY'). | | | | |
| **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 9. Передача договору на підписання в зовнішню систему

|  |
| --- |
| **Опис задачі (Що необхідно зробити, Як буде тестуватись, Очікуваний результат)** |
| Необхідно налаштувати передачу договору (тип документу - Contract) на підпис в зовнішню систему на подію onTaskExecute 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: "Contract",  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  }  } |