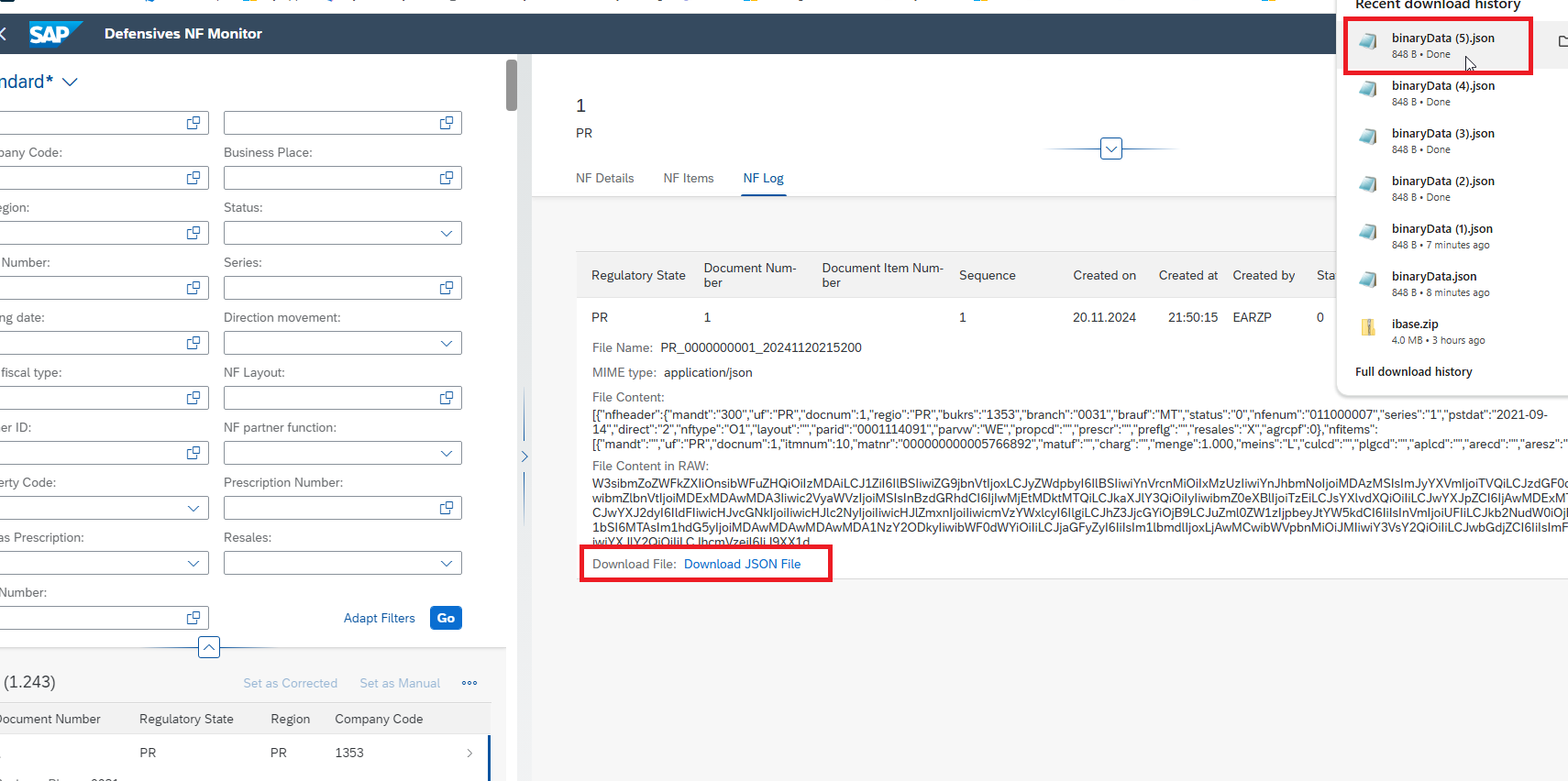
Download File via Fiori Extension:  


Table where the Files are:

A screenshot of a computer

Description automatically generated

|  |
| --- |
| @AccessControl.authorizationCheck: #NOT\_REQUIRED  @EndUserText.label: 'Nota Fiscal Log for Monitor App'  @Metadata.ignorePropagatedAnnotations: true  @ObjectModel.usageType:{  serviceQuality: #A,  sizeCategory: #L,  dataClass: #TRANSACTIONAL  }  **define** **view** **entity** /BAY0/O2C\_I\_NF\_LOG\_V2  **as** **select** **from** /bay0/o2c\_deflog **as** NfLog  **association** **to** **parent** /BAY0/O2C\_I\_NF\_HDR\_V2 **as** \_NfHeader **on** **$projection.**Docnum **=** \_NfHeader**.**Docnum  **and** **$projection.**Uf **=** \_NfHeader**.**Uf  **{**  **key** NfLog**.**uf **as** Uf**,**  **key** NfLog**.**docnum **as** Docnum**,**  **key** NfLog**.**itmnum **as** Itmnum**,**  **key** NfLog**.**sequen **as** Sequen**,**  NfLog**.**credat **as** Credat**,**  NfLog**.**cretim **as** Cretim**,**  NfLog**.**crenam **as** Crenam**,**  NfLog**.**status **as** Status**,**  NfLog**.**descr **as** Descr**,**  NfLog**.**filename **as** Filename**,**  **//The Yellow @Semantics below works only in odata v4 in S/4 2021+**  ~~@Semantics.mimeType: true~~  NfLog**.**mimetype **as** MimeType**,**  NfLog**.**filex **as** FileString**,**  ~~@Semantics.largeObject:{~~  ~~mimeType: 'MimeType',~~  ~~fileName: 'Filename',~~  ~~contentDispositionPreference: #ATTACHMENT~~  ~~}~~  NfLog**.**file\_raw **as** FileRaw**,**  /\* Associations \*/  \_NfHeader  **}** |

|  |
| --- |
| @AccessControl.authorizationCheck: #NOT\_REQUIRED  @EndUserText.label: 'Nota Fiscal Log for Monitor App'  @Metadata.ignorePropagatedAnnotations: true  @Metadata.allowExtensions: true  **define** **view** **entity** /BAY0/O2C\_C\_NF\_LOG\_V2  **as** **projection** **on** /BAY0/O2C\_I\_NF\_LOG\_V2 **as** \_NfLog  **{**  **key** Uf**,**  **key** Docnum**,**  **key** Itmnum**,**  **key** Sequen**,**  Credat**,**  Cretim**,**  Crenam**,**  Status**,**  Descr**,**  Filename**,**  MimeType**,**  FileString**,**  FileRaw**,**  /\* Associations \*/  \_NfHeader **:** **redirected** **to** **parent** /BAY0/O2C\_C\_NF\_HDR\_V2  **}** |

|  |
| --- |
| @Metadata.layer: #CORE  //@UI.presentationVariant: [ { sortOrder: [ { by: 'Docnum', direction: #DESC }, { by: 'Itmnum', direction: #ASC } ] } ]  @Search.searchable: false  **annotate** **entity** /BAY0/O2C\_C\_NF\_LOG\_V2 **with**  **{**  @UI.lineItem: [{ position: 10 }]  Uf**;**  @UI.lineItem: [{ position: 20 }]  Docnum**;**  @UI.lineItem: [{ position: 30 }]  Itmnum**;**  @UI.lineItem: [{ position: 40 }]  Sequen**;**  @UI.lineItem: [{ position: 50 }]  Credat**;**  @UI.lineItem: [{ position: 60 }]  Cretim**;**  @UI.lineItem: [{ position: 70 }]  Crenam**;**  @UI.lineItem: [{ position: 80 }]  Status**;**  @UI.lineItem: [{ position: 90, importance: #HIGH }]  Descr**;**  @UI.lineItem: [{ position: 100, importance: #HIGH }]  Filename**;**  @UI.lineItem: [{ position: 110 }]  MimeType**;**  @UI.lineItem: [{ position: 120 }]  FileString**;**  @UI.lineItem: [{ position: 130 }]  FileRaw**;**  **}** |

Create Custom Column via Guided Development Tool:

A screenshot of a computer

Description automatically generated

Download File with the following JS Code:

|  |
| --- |
| onDownloadJson: function (oEvent) {              try {                  // Get the binary content from the event source                  var base64String = oEvent.getSource().getTarget();                  // Decode the base64 string into a JSON string                  var jsonString = atob(base64String);                  // Parse the JSON string into a JSON object                  var jsonObject = JSON.parse(jsonString);                  // Convert the JSON object to a Blob                  var jsonBlob = new Blob([JSON.stringify(jsonObject, null, 2)], { type: "application/json" });                  // Create a temporary anchor element for download                  var tempAnchor = document.createElement("a");                  tempAnchor.href = URL.createObjectURL(jsonBlob);                  tempAnchor.download = "binaryData.json"; // Set the download filename                  // Append the anchor to the body (not visible to the user)                  document.body.appendChild(tempAnchor);                  // Trigger the download                  tempAnchor.click();                  // Clean up: Remove the temporary anchor                  document.body.removeChild(tempAnchor);              } catch (e) {                  // Handle errors (e.g., invalid JSON)                  console.error("Error processing the binary JSON:", e);                  sap.m.MessageToast.show("Failed to process and download the binary JSON.");              }          }, |