GUÍA PARA EL NUEVO NUEVITO <3

# Linux

## Comandos

pwd Imprime la ruta actual donde nos encontramos

ls -l Lista detallada

cd Cambia directo a ruta home

./ Hace referencia a ruta actual

file [] []es la ruta o el archivo en específico, nos indica de que tipo se trata

mkdir Crea el directorio

-p Crea directorios padres, ej: mkdir –p abuelo/papa/hijo

-v Muestra en línea de comandos lo que se está haciendo

rmdir Borra el directorio

-pv Funcionan igual

Touch Crea el archivo, ej: touch ejercicio1.txt, no es su función principal

Eliminar todos los archivos .txt de una carpeta y sus subcarpetas:

find . -name "\*.txt" -type f –delete

* **$  sudo find /home/nombre\_usuario/Escritorio -name “\*.jpg”**: Busca todos los archivos del Escritorio con extensión .jpg.
* **$  sudo find . -name “\*hola\*”**: Busca todos los archivos que contienen la palabra “hola” en el nombre.
* **$ sudo find . ! -name “\*hola\*”**: Busca todos los archivos que NO contienen la palabra “hola” en el nombre.
* **$ sudo find . -name “\*hola\*” -a “\*caracola\*”**: Busca todos los archivos que contienen la palabra “hola” y “caracola” en el nombre.
* **$ sudo find . -name “\*hola\*” -o “\*caracola\*”**: Busca todos los archivos que contienen la palabra “hola” o “caracola” en el nombre.
* **$  sudo find . -iname “\*hola\*”**: Busca todos los archivos que contienen la palabra “hola” en el nombre tanto en mayúsculas como en minúsculas.
* **$ sudo find . -user nombre\_usuario**: Busca todos los archivos de un usuario determinado (nombre\_usuario).
* **$ sudo find . -size +1000k**: Busca todos los archivos de tamaño mayor a 1000 kb.
* **$ sudo find . -amin -30**: Busca todos los archivos a los que se accedió en los últimos 30 minutos.
* **$ sudo find . -atime 365**: Busca todos los archivos a los que se accedió hace un año exactamente.
* **$ sudo find . -name “\*.jpg” -exec rm {} \;**: Busca todos los archivos con extensión .jpg y los borra.
* find / -name "[0-9]\*" (todo lo que empieze con un dígito)
* find / -name "[Mm]\*" (todo lo que empieze con un la letra M o m)
* find / -name "[a-m]\*.txt" (todo lo que empieze entre a y m y termine en ".txt")

# Liferay

## Levantar servidor de cero

1.- Descomprimir bundle.

2.-Copiar carpeta osgi de bgp-workspace (configs y modules) a osgi del servidor Liferay.

3.- Compilar startup-environment. (Dejarlo en modules no en deploy)

4.- Poner últimos properties. (Comentar prisma para que no vaya lento)

5.- Parchar Liferay (Anexar manual para parche).

6.- Forzar inicio del login en caso de que mande error 404, accediente a la siguiente ruta:

localhost:8080/web/guest/home?p\_p\_id=com\_liferay\_login\_web\_portlet\_LoginPortlet&p\_p\_lifecycle=0&p\_p\_state=maximized&saveLastPath=false&\_com\_liferay\_login\_web\_portlet\_LoginPortlet\_login=ss%40liferay.com

7.- Poner licencia.

8.- Levantar server.

9.- Importar páginas. (Privadas, públicas) .zip .lar (no aceptar .rar)

10.- Deploy a los módulos.

11.- Importar roles.

12.- Importar certificado… Wiki (how to connect to data power). <http://devops/gitlab/bgp-bel/documentacion/wikis/configuracion-de-conexion-con-datapower>

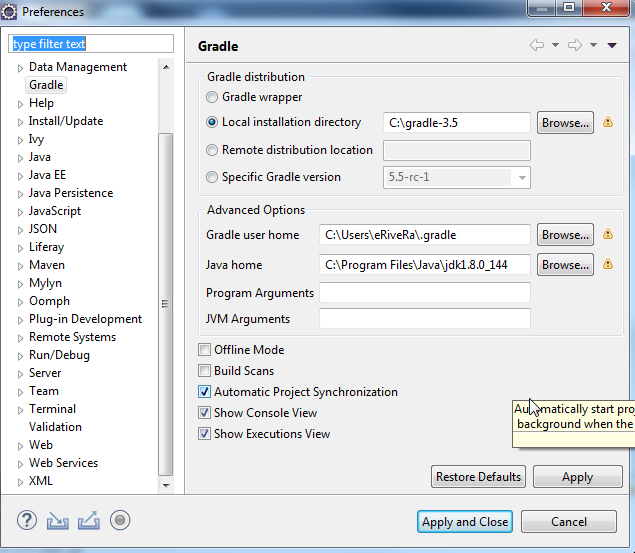
13.- Agregar en el server: -Dorg.apache.cxf.Logger=org.apache.cxf.common.logging.Log4jLogger -Duser.timezone=America/Panama.

14.- Revisar que la ruta de nuestro servidor Liferay esté correcta en .gradle.

15.- Instalar Lombok.jar en Eclipse (copiar el jar en la carpeta de eclipse y darle doble click).

16.- Si queremos revisar log.debug en nuestra consola, debemos agregar 2 archivos log4j.dtd y portal-log4j-ext.xml a la siguiente ruta: <CARPETA-DE-LIFERAY>/<CARPETA-DE-TOMCAT>/webapps/ROOT/WEB-INF/classes/META-INF/ (<http://devops/gitlab/bgp-bel/documentacion/wikis/configuracion-de-logs-del-portal>)

17.- Verificar en eclipse la siguiente configuración para Gradle:



Agregar un rol BGP en properties.bundles (cuando se instalo desde sp49 si fue desde32 no)

## Borrar usuarios de Liferay

Panel de Control / Usuarios / Usuarios y Organizaciones

1.- Desactivar todos menos Test

2.- Ir a inactivos y borrar todos

3.- Importar roles nuevamente

## Exportar LARs

Importante borrar VIEW\_DISABLE\_ACCESS de portlet-data-permissions.xml

## Orden del Front-End al Back-End (Ejemplo International Report)

Partimos desde el controlador JS ubicado en:

C:\Users\eRiveRa\comercial\international-reports-web\src\main\resources\META-INF\resources\js\international-reports.controller.js

1.- El controlador manda a llamar una función inicial…

vm.setup = function() {

vm.getInitialState();

};

Dentro de esa función cargamos empresas afiliadas de la cuenta y llamamos a otra función que solicitará los reportes iniciales…

var getMovements = function() {

…

var xhr = InternationalReportsService.getMovements();

…

}

Así pasamos del controlador al servicio:

C:\Users\eRiveRa\comercial\international-reports-web\src\main\resources\META-INF\resources\js\international-reports.service.js

2.- Acá se solicitará el servicio API REST

function getMovements() {

return serviceInvoker.**invoke**("GET", WS\_PATH + "initial-movements/");

}

Terminamos con front-end de la vista y pasamos al back-end:

C:\Users\eRiveRa\comercial\international-reports-web\src\main\java\com\bgeneral\bel\international\reports\controller\InternationalReportController.java

3.- Trabajaremos con JSON y a través de un filtro (ReportFilter) solicitaremos al servicio una lista con los movimientos…

@GET

@Path("/international-reports/initial-movements")

@Consumes(MediaType.***APPLICATION\_JSON***)

@Produces(MediaType.***APPLICATION\_JSON***)

@HasPermission(action = PermissionsConstants.***VIEW***,

portletId = InternationalReportsKeys.***PORTLET\_NAME***)

**public** InterReportInfoWrapper getInternationalReport()

**throws** BGPException {

***log***.debug("Incio de ultimas transacciones internacionales");

ReportFilter reportFilter = **new** ReportFilter();

reportFilter.setRecordNumber(

NumberUtils.*toInt*(

PropsUtil.*get*(BGPropsKeys.***BG\_BEL\_MAX\_MOVEMENT\_COMMERCIAL\_INITIAL\_STATE***)));

reportFilter.setTransactionScope("T");

List<InternationalReport> list =

internationalReportApi.getInternationalReportList(reportFilter);

Llamamos al método getInternationalReportList así que vamos a:

C:\Users\eRiveRa\comercial\commercial-reports-service\commercial-reports-service\src\main\java\com\bgeneral\bel\commercial\reports\impl\InternationalReportService.java

@Override

**public** List<InternationalReport> getInternationalReportList(ReportFilter reportFilter)

**throws** BGPException {

***log***.debug("Ingresando al getInternationalReportList");

ConsultarReporteTransferenciaInternacionalMsjSolTipo req =

internationalReportMapper

.toConsultarReporteTransferenciaInternacionalMsjSolTipo(reportFilter);

ConsultarReporteTransferenciaInternacionalMsjRespTipo resp =

transferenciaWs.consultarReporteTransferenciaInternacional(req);

ObjectMapper obj = **new** ObjectMapper();

**try** {

obj.writeValueAsString(resp);

} **catch** (JsonProcessingException e) {

***log***.error("Json Parser Error", e);

}

handleWsResponse(resp.getSTATUS());

***log***.debug("Saliendo al metodo");

**return** internationalReportMapper.toInternationalReport(resp.getBODY()); }

Mandamos a llamar al Mapper en las funciones:

ConsultarReporteTransferenciaInternacionalMsjSolTipo

toInternationalReport…

C:\Users\eRiveRa\comercial\commercial-reports-service\commercial-reports-service\src\main\java\com\bgeneral\bel\commercial\reports\mapper\InternationalReportMapper.java

Y la respuesta del servicio en:

C:\Users\eRiveRa\comercial\ws-api-services\transferencia-ws-api\src\main\java\com\bgeneral\bel\ws\transferencia\TransferenciaPortTypeSOAP.java

# HTML

## Columnas

<div class="row" ng-if="ctrl.productWrapper.product.type == accCtrl.productType.checkingAccount">

<div class="col-sm-12" ng-if="accCtrl.overdraftDetailData || accCtrl.overdraftDetail.length > 0">

NOTA:

Siempre usamos un row padre de los col-sm para que los márgenes queden igual, Flau habló de que uno resta 15 y otro suma 15 al margen y padding

## Scroll

Overflow hidden

Overflow-x: scroll

## @media

body {

background-color: lightblue;

}

@media screen and (min-width: 200px) {

body {

background-color: lightgreen;

}

}

@media screen and (min-width: 400px) {

body {

background-color: lavender;

}

}

@media screen and (min-width: 600px) {

body {

background-color: lightblue;

}

}

## Span para mostar/ocultar

<ui-select-match placeholder="{{'select' | t}}">

{{ $select.selected.maskedNumber }} <span ng-if="$select.selected.alias">&middot;</span> {{$select.selected.alias}} <br>

{{$select.selected.productOwner.name}}

</ui-select-match>

## Mostrar/Ocultar en tablet

<span class="hidden-tablet ng-scope" ng-if="true"> · </span>

## Centrar verticalmente

.bgp-vertical-center-cell {

display: flex;

align-items: center;

float: none;

}

## Alinear contenido verticalmente al fondo

<div class="row bgp-align-row-content-bottom">

<div class="col-sm-6 bgp-product-detail-label">

<summary>{{"product-info-payment-date" | t}}:</summary>

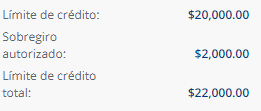
</div>

<div class="col-sm-6 bgp-product-detail-info">

{{ctrl.productWrapper.product.detail.nextPaymentDate | date:'dd-MMM-yyyy'}}

</div>

</div>



## Etiqueta con popover

Pagos recibidos:

C:\Users\eRiveRa\Desktop\2018-10-15 11_17_08-Pagos recibidos - Banco General.png

<div>

<label

id="paymentChannelLabel"

for="paymentChannel">

{{"received-payments-reports-received-in" | t}}

</label>

<span

class="fa fa-info-circle cursor-pointer bgp-detail-label"

popover-class="bgp-popover-m"

popover-placement="right"

popover-trigger="'mouseenter'"

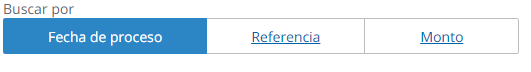
uib-popover="{{'received-payments-reports-received-in-tooltip' | t}}">

</span>

</div>

## Ng-switch on (cambiar el botón dependiendo el tipo de búsqueda)

Pagos recibidos:



<div class="col-sm-6 col-md-4 col-lg-4 pts" ng-switch on="ctrl.tempFilterValues.searchType">

<div ng-switch-when="A">

<div

ng-include="'/o/received-payments-reports-web/templates/partials/search-filter/received-payments-reports.date-filter.html'"/>

</div>

<div ng-switch-when="C">

<div

ng-include="'/o/received-payments-reports-web/templates/partials/search-filter/received-payments-reports.reference-filter.html'"/>

</div>

<div ng-switch-when="B">

<div

ng-include="'/o/received-payments-reports-web/templates/partials/search-filter/received-payments-reports.amount-filter.html'"/>

</div>

</div>

# Angular

## Encontrar una directiva

Html -> comercial-accounts-select

JS -> commercialAccountsSelect

## Utilizar variables de un controlador dentro de otro…

var BGPLoginPasswordController = function(

$log,

$window,

bgpUrls,

loginValidateService,

$state,

handleStateUrlService,

$controller,

BgpFieldRegexService) {

$log.debug("[Liferay/Angular/BGPLoginPasswordController] Initializing...");

var vm = this;

$controller('BGPBaseController', { vm: vm }); 🡨 Así se hace

vm.secretImageURL = "";

vm.loginRemember = false;

## Servicios del tema

### handleStateUrlService

Parámetros:

1->state al que direccionará

2-> información que enviará

3-> información que tenía, en caso de dar anterior, conservarla…

vm.nextPage = function() {

vm.showLoading = true;

var req = {

params: {

transactionWrapper: vm.flow.params.transactionWrapper,

collector: vm.flow.params.collector,

context: vm.flow.params.context,

state: vm.flow.params.state

}

};

if (vm.flow.params.customsFee) {

req.params.customsFee = vm.flow.params.customsFee;

}

var xhr = collectorPaymentService.nextStep(req);

xhr.then(function(flow) {

handleStateUrlService.go(flow.data.params.state, flow.data, vm.flow);

});

xhr.catch(function() {

$state.go("collector\_payment\_error");

vm.showLoading = false;

});

};

### queryParamsService (Obtener valor desde URL)

1.- Inyectar ”queryParamsService” al controlador

2.- Crear una variable para obtener el valor:

var ejemplo = 'sin-data-en-url';

3.- Llamar al método get de queryParamsService y como parámetro decimos el valor a buscar:

var existeVariable = queryParamsService.get('uutime');

if (existeVariable) {

ejemplo = 'valor-desde-url-es-' + existeVariable;

}

console.log(ejemplo);

4.- URL de ejemplo con variable uutime:

<http://localhost:8080/group/guest/lineas-de-credito?uutime=1546883417.425#/index>

5.- Si existe uutime en URL mostrará:

valor-desde-url-es-1546883417.425

si no:

sin-data-en-url

## JSON dentro de JSON

vm.flow.transaction.affiliation = {

beneficiary: {

contractNumber: "909090",

name: "Nombre"

},

description: "descripcion"

};

Con lista:

var Person = {

name: 'John',

age: 21,

alive: true,

siblings: [

{

name: 'Andrew',

age: 23,

alive: true

},

{

name: 'Christine',

age: 19,

alive: true

}

]

}

## Filtros

### Mes (Número del 0 al 11, lo devuelve como cadena, ej: 4 -> Mayo)

{{"credit-card-statement-detail" | t}}

{{ctrl.clonedDateFilter.month | monthName}}

{{ctrl.clonedDateFilter.year}}

### Porcentaje

{{ ctrl.interest | cent }}

### Fecha

{{ctrl.productWrapper.product.detail.lastStatementBankDate | date: 'MMMM y'}}

Ej: marzo 2014

{{ctrl.productWrapper.product.detail.lastStatementBankDate | date : "dd-MMM-yyyy"}}

Ej: 05-marzo-2014

### OrderBy (más de 1 elemento)

<tr ng-repeat="friend in friends | orderBy:['name', '-age']">

### Filtro dentro de filtro (properties)

{{'product-credit-card-up-to-date-payment-2' | t : [(ctrl.productWrapper.product.detail.lastPaymentDate | date : "dd-MMM-yyyy"), (ctrl.productWrapper.product.balance.lastPaymentBalance | currency)]}}

## Escape de caracteres especiales

<p

class="info mbn font-weight-semibold"

ng-bind-html="transaction.details.destinationAccountFriendlyName">

</p>

## Agregar Routes al controlador

1.- Verificar que se agregó el llamado al routes.js en el Portlet.java del módulo… ej:

package com.bgeneral.bel.notification.web.portlet;

import com.bgeneral.bel.notification.web.util.NotificationsKeys;

import com.liferay.portal.kernel.portlet.bridges.mvc.MVCPortlet;

import javax.portlet.Portlet;

import org.osgi.service.component.annotations.Component;

@Component(immediate = true, property = {"com.liferay.portlet.display-category=bgp.bel",

"com.liferay.portlet.instanceable=false", "javax.portlet.display-name= BGP - Notifications",

"javax.portlet.name=" + NotificationsKeys.PORTLET\_NAME,

"javax.portlet.init-param.template-path=/", "javax.portlet.init-param.view-template=/view.jsp",

"javax.portlet.resource-bundle=content.Language",

"javax.portlet.security-role-ref=power-user,user",

"com.liferay.portlet.footer-portlet-javascript=/js/notifications.controller.js",

"com.liferay.portlet.footer-portlet-javascript=/js/notifications.service.js",

"com.liferay.portlet.footer-portlet-javascript=/js/notifications.routes.js",

"com.liferay.portlet.footer-portlet-javascript=/js/notifications.email.delete.controller.js"},

service = Portlet.class)

public class NotificationsPortlet extends MVCPortlet {

}

2.-Crear archivo routes.js… ej:

(function (win) {

"use strict"

function NotificationsRouter($stateProvider) {

var states = [

"index",

"error"

];

\_.map(states, function(state) {

var stateName = "notifications\_" + state;

$stateProvider.state(stateName, {

url:"/" + state,

views: {

"NotificationsView": {

templateUrl: win.baseAssetsURL +

"notifications-config" +

"/templates/notifications." +

state + ".html"

}

},

params: {

data: null,

historyStateData: false

}

})

});

}

NotificationsRouter.$inject = ["$stateProvider"];

angular

.module(win.appName)

.config(NotificationsRouter);

}(window));

NOTA: Esa ruta está definida en el bnd.bnd: Web-ContextPath: /notifications-config

3.- Llamar al Router en el view.jsp… ej:

<div class="col-sm-9 pll">

<div ui-view="NotificationsView" onclick="void(0)"></div>

</div>

4.- Llamar al state.go en el controlador:

vm.getCompanies = function() {

parallelLoaderFactory.addRequest();

var xhr = notificationsService.getCompanies();

xhr.then(function(response) {

$state.go("notifications\_index");

vm.groupCompanies = response.data;

## Mostrar servicio no disponible cuando no carga un servicio

1.- Verificar que tenemos $window, parallelLoaderFactory, bgpUrls y exceptionHandlerService en controlador.

2.- Agregar el siguiente método en el controlador:

// Gets if the bel loader should be shown at any given moment

vm.getLoaderState = function() {

return parallelLoaderFactory.getLoaderState();

};

3.- Revisar que tenemos en la vista (view.jsp):

<loading-component display="ctrl.getLoaderState()"></loading-component>

4.- Antes del then agregamos:

parallelLoaderFactory.addRequest();

5.- Agregamos el siguiente catch a la llamada al servicio:

xhr.catch(function (exception) {

vm.error = exceptionHandlerService.handleError(

exception, vm.serviceTitleKey);

parallelLoaderFactory.disableLoader();

if (vm.error) {

$window.location.href = bgpUrls.SERVICE\_FAILURE;

}

$log.debug("Error al consultar movimientos");

});

NOTA: En notifications-config-web redirecciona a una página dentro del mismo módulo, este redirecciona a la página principal… :

4.- Agregamos el siguiente finally:

xhr.finally(function() {

parallelLoaderFactory.removeRequest();

});

## Mostrar cintillo de servicio no disponible

1.- Creamos VARIABLE y la inicializamos false…

2.- En controlador al cargar el servicio la volvemos true si falla el servicio:

xhr.catch(function(exception) {

vm.overdraftDetailData = true;

vm.error = exceptionHandlerService.handleError(

exception,

vm.serviceTitleKey

);

if (vm.error) {

$log.debug("error getOverdraftDetail");

}

3.- En vista mostramos el siguiente div:

<div

class="alert alert-warning fnt-size-md"

role="alert"

ng-if="accCtrl.overdraftDetailData ">

<i class="bgp-alert\_\_icon bgp-alert\_\_icon--warning fnt-size-x-lg mrs"></i>

{{"common-warning-message-service" | t}}

</div>

## Mostrar alerta de servicio no disponible

1.- Verificar que tenemos cargado alertNotifyService en el controlador

var InternationalReportsController = function($log, $state, $window, exceptionHandlerService,

InternationalReportsService, modalService, parallelLoaderFactory, translateService,

alertNotifyService, bgpUrls, tFilter)

InternationalReportsController.$inject = [

"$log",

"$state",

"$window",

"exceptionHandlerService",

"InternationalReportsService",

"modalService",

"parallelLoaderFactory",

"translateService",

"alertNotifyService",

"bgpUrls",

"tFilter"

];

2.- Agregamos el siguiente catch al servicio:

xhr.catch(function(exception) {

vm.error = exceptionHandlerService.handleError(

exception, vm.serviceTitleKey);

alertNotifyService.showWarningT('common-warning-message-service');

movement.open=false;

});

NOTA:En este caso de internacionales se mandó el false para cerrar el detalle en la vista, que solo se muestra cuando movement.open esigual a true

## Manejo de variables en directivas (camel y kebab case)

1.- En la directiva:

return {

restrict: 'E',

scope: {

selectedCompany: '=',

companies: '=',

groupName: '=',

showGroupSelection: '=',

withPlaceholder: '='

},

2.- En donde se manda a llamar:

<companies-selector

selected-company="ctrl.companyId"

companies="ctrl.clientData.companies"

group-name="ctrl.clientData.groupName"

show-group-selection=true

with-placeholder=true>

</companies-selector>

## Botón para redireccionar a un pago

HTML:

<div class="row"

ng-if="ctrl.productWrapper.product.status.id == ctrl.statusLoan.vigente">

<div class="col-sm-offset-6 col-sm-6 text-left mtm">

<button

class="btn btn-default"

ng-if="ctrl.productWrapper.portletPermissions.PAY\_LOAN"

ng-click="ctrl.descriptionRedirect('cuentas-propias?destination='

+ ctrl.product.accountNumber)">

{{"product-info-make-payment" | t}}

</button>

</div>

</div>

JS:

vm.descriptionRedirect = function (link) {

$window.location.href = bgpUrls.DEFAULT + link;

};

## Usar los textos del properties dentro de un filtro

Llamar a tFilter..

/\* globals MainApp \*/

(function(win) {

"user strict";

function OptionalFieldFilter($window, tFilter) {

return function (value) {

if (\_.isEmpty(value)) {

return tFilter("international-reports-detail-no-data");

} else {

return value;

}

};

}

OptionalFieldFilter.$inject = ['$window', tFilter];

win.MainApp.Filters.filter('optionalFieldDefault', OptionalFieldFilter);

}(window));

## Validaciones dentro de 2-way data binding

<label ng-class="{'text-color-brand-red': [ctrl.numberOfRecords] <= 0}">

{{ctrl.numberOfRecords == 1

? "special-reports-reports-search-found-one"

: "special-reports-reports-search-found-many"

| t : [ctrl.numberOfRecords]}}

</label>

En Seven hicieron esto (podría servir a futuro)…:(

<label

ng-if="ctrl.achTransationsNumber == 1"

ng-bind-html="::'manage-users-search-single-result'|t">

</label>

<label

ng-if="ctrl.achTransationsNumber > 1 &&

ctrl.achTransationsNumber < ctrl.maxNumberOfResults"

ng-bind-html="'manage-users-search-results'|t:[ctrl.achTransationsNumber]">

</label>

<label

ng-if="ctrl.achTransationsNumber >= ctrl.maxNumberOfResults"

ng-bind-html="'manage-users-search-results'|t:[(ctrl.maxNumberOfResults) + '+']">

</label>

## Realizar funciones en el formulario en caso de que sea válido

<form

name="ctrl.selectionForm"

class="form form-vertical"

novalidate

ng-submit="ctrl.selectionForm.$valid ? ctrl.validateCharge() && ctrl.nextStep() : ctrl.showErrorNotification()">

## Funciones anidadas

vm.setup = function() {

vm.templateCache()

.then(vm.getDefaultValues)

.then(vm.requestAccount);

};

## Setear hora 00:00:00 para búsquedas

var starDate = new Date(vm.filter.startDate)

.setHours(0, 0, 0);

var endDate = new Date(vm.filter.endDate)

.setHours(0, 0, 0);

## Encontrar patrones en una cadena

var localIndex = objPerItem.index;

var expresion = /[\s]+/gi;

var array\_emparejamientos = objPerItem.tempFriendlyName.match(expresion);

## Comparación para mostrar el menor o mayor

var smallAmount = (vm.amountFilter.minAmount < vm.amountFilter.maxAmount) ?

vm.amountFilter.minAmount : vm.amountFilter.maxAmount;

var bigAmount = (vm.amountFilter.maxAmount > vm.amountFilter.minAmount) ?

vm.amountFilter.maxAmount : vm.amountFilter.minAmount;

amountFilter.minAmount = smallAmount;

amountFilter.maxAmount = bigAmount;

var smallDate = (dateFilter.sinceDate < dateFilter.toDate) ?

dateFilter.sinceDate : dateFilter.toDate;

var bigDate = (dateFilter.toDate > dateFilter.sinceDate) ?

dateFilter.toDate : dateFilter.sinceDate;

dateFilter.sinceDate = smallDate;

dateFilter.toDate = bigDate;

## Cambio de estilos

### Números

ng-class="{

'text-color-brand-red': movement.product\_detail\_movement.comission < 0,

'text-color-brand-blue': movement.product\_detail\_movement.comission >= 0}"

### Letras

ng-class="{

'text-color-brand-red': movement.natureMovement == 'D',

'text-primary': movement.natureMovement != 'D'}"

### Más de una condición

ng-class="{

'text-brand-green-variation-1' : movement.detail.transactionresult.transactionState.description

== ctrl.transactionState.success,

'text-color-brand-blue': movement.detail.transactionresult.transactionState.description

== ctrl.transactionState.process,

'text-color-brand-red': movement.detail.transactionresult.transactionState.description

== 'ctrl.transactionState.rejected'}"

## Loadash.js

### \_.reject y .find

vm.messageCategories = response.data;

$state.transitionTo("contact\_us\_index");

var others = \_.find(vm.messageCategories,

function (category) { return category.id === "O" });

var suggestions = \_.find(vm.messageCategories,

function (category) { return category.id === "SU" });

var newMessageCategories = \_.reject(vm.messageCategories,

function (category) { return category.id === "O" || category.id === "SU"});

newMessageCategories.push(suggestions);

newMessageCategories.push(others);

vm.messageCategories = newMessageCategories;

### indexOf

Devuelve el índice del ítem encontrado dentro de la lista…

## Moment.js

Probar en: http://jsfiddle.net/remus/rLjQx/

var date = new Date(2018, 10, 26);

var todaymoment = moment().startOf('day');

var weekfromNowMoment = moment(todaymoment).add(8, 'day').startOf('day');

var inputDate = moment(date).startOf('day');

var filteredString = weekfromNowMoment.from(todaymoment);

var jelou = todaymoment.isSame(inputDate);

alert(filteredString);

## Componentes

1.- Añadir componentes en app.js

**diff --git a/src/theme/js/app/app.js b/src/theme/js/app/app.js**

**index 73d41d4f..ac6d763e 100755**

**--- a/src/theme/js/app/app.js**

**+++ b/src/theme/js/app/app.js**

@@ -16,6 +16,7 @@

win.MainApp.Services = angular.module(appName + ".services", []);

win.MainApp.Filters = angular.module(appName + ".filters", []);

win.MainApp.Factories = angular.module(appName + ".factories", []);

+ win.MainApp.Components = angular.module(appName + ".components", []);

/\*\*

\* Setup Common Modules

@@ -27,6 +28,7 @@

appName + ".services",

appName + ".filters",

appName + ".factories",

+ appName + ".components",

"autoComplete",

'ngSanitize',

'ui.router',

2.- Añadir componentes en el Portlet.java

@@ -22,7 +22,11 @@ import javax.portlet.Portlet;

"javax.portlet.security-role-ref=power-user,user",

"com.liferay.portlet.footer-portlet-javascript=/js/line-of-credit-reports.routes.js",

"com.liferay.portlet.footer-portlet-javascript=/js/line-of-credit-reports.controller.js",

- "com.liferay.portlet.footer-portlet-javascript=/js/line-of-credit-reports.service.js"},

+ "com.liferay.portlet.footer-portlet-javascript=/js/line-of-credit-reports.service.js",

+ "com.liferay.portlet.footer-portlet-javascript="

+ + "/js/line-of-credit-reports.movements.component.js",

+ },

3.- Ejemplos de uso

line-of-credit-reports.movements.component.js

(function (win) {

"use strict";

// Controller

var lineOfCreditMovsController = function() {

var ctrlLineMovs = this;

}

// Components attributes

var lineMovs = {

templateUrl: '/o/line-of-credit-reports-web/templates/components/line-of-credit-reports-movements.html',

bindings: {

util: '='

},

controller: lineOfCreditMovsController,

controllerAs: 'ctrlLineMovs'

};

win.MainApp.Components.component("lineMovs", lineMovs);

}(window));

line-of-credit-reports-movements.html

<div class="mbs pts">

<div ng-if="true">

<div class="alert alert-warning fnt-size-md" role="alert">

<i class="bgp-alert\_\_icon bgp-alert\_\_icon--warning fnt-size-x-lg mrs"></i>

{{"common-warning-message-service" | t}}

</div>

</div>

<div class="row" ng-if="true">

<div class="col-sm-6 mbs">

<label ng-class="{'text-color-brand-red': [ctrl.numberOfRecords] <= 0}">

{{"special-reports-reports-search-found-many" | t : [ctrl.numberOfRecords]}}

</label>

</div>

</div>

….

Llamado al componente:

<line-movs util="ctrl"></line-movs>

# SCSS

## Crear grid de columnas a partir de col-sm

.accordion-col-89 {

@extend .col-sm-12;

width: 89%;

}

@media (min-width: 1200px) {

.accordion-col-89 {

width: 91.8%;

}

}

@media (max-width: 991px) {

.accordion-col-89 {

width: 86%;

}

}

.accordion-col-11 {

@extend .col-sm-12;

width: 11%;

}

@media (min-width: 1200px) {

.accordion-col-11 {

width: 8.2%;

}

}

@media (max-width: 991px) {

.accordion-col-11 {

width: 14%;

}

}

# Jasper

## Guía de Excel (en inglés)

These are the tips I've learned on how to create a report that is expected to be exported to Excel.  Most of my reports now days, have a user selection of 'PDF or Excel'.  So the reports are built for both PDF format and for Excel format.

Reports need to be designed differently depending on how the user is going to view/use them. I find PDF is easier than Excel. Excel has taken some trial and error to get satisfactory results.

With Excel you have to make certain all the columns line up, so you don't get a whole bunch of different sized columns in Excel. Export a poorly laid out report to Excel, and this is clearly seen. This post doesn't cover that effort.

For exporting to Excel, my objective is to keep things clean and tidy. This is because the user wants the data in Excel so that they can slice, dice, and sort, group, filter, and play with it until they see/get what they need. They don't want to do clean up before they can do that. In Excel, I have one tab for data, and another tab for footnotes. In the data tab I have one row for the report name / company logo. The next row is for column headers. Then the rows for data.

**Now in Jaspersoft Studio...**

The Column Header section is where I place the table for the data. Having the data in a table, allows the user when viewing the data in HTML to sort the data, etc. The data table, has one row for column headers, one row for data.

In the Summary section, I show the user all their input selections, explain any abbreviations or calculations, show who ran the report and when, etc. I also have the report name and company logo at the top of this section.  I don't put them in the Column Header or the Page Header (unless the report also has PDF output) because it kind of junks up the look of the Excel data page.  Place a Break Palette Element at the very top of the Summary. Mine is at x: 0 px, y: 0 px, Height 1px, Position Type: Fix Relative To Top.  At the top of the Summary section, I have a page break (which is 1 pixel high).  All Other elements on the summary page start on or lower than Y position 1, so as to not overlap with the page break.

I also have a No Data section, which is a copy of the Summary page with the addition of a big "There was no data found for the selections made" notice, and the addition of the report name and company logo. With this No Data section, if the user made selections that have no resulting data, they see a notice telling them that, and they can see their selections. Often times an empty report, to a user, is seen as the system or the report not working. So I want to tell them that it was their choices that led them here. The Main Report has to have the same dataset query as the table, for this to work.

In Main Report, Report properties, Page Format (for this one example report I am looking at), my Page Format has the following properties:

* Format: Letter is what I always use
* Width: When I have a report that does both PDF and Excel, the the width is 792.
* Height: When I have  report that does both PDF and Excel, then the height is 612
* Note: Excel doesn't care what your margins here are.  PDF likes these just fine.
* Units: pixel.  Much easier to work in than inches
* Page Orientation: Landscape
* Margins: 36 for all margins.

Also, in Report properties, I have made these selections:

* When No Data Type: No Data Section
* Summary On A New Page: checked
* Summary with Page Header and Footer: checked --I could have this unchecked, and in the Summary section, I would have to add the report name and the company logo.
* Ignore Pagination: After much experiementation, I have decided that unchecked is best, otherwise when doing a PDF or Excel format report, the PDF does not page properly, and can't be recovered from.  It is the lessor of two evils (until Jaspersoft fixes this in a new version).

Also, in the Main Report Properties, in Advanced, Properties, I have added these (which end up near the top of the Source XML):

**<property** name = "net.sf.jasperreports.export.xls.one.page.per.sheet" value="true"**/>**

* Without the above, JS only puts as many data rows on a single tab as fit within the Page Height.

**<property** name="net.sf.jasperreports.export.xls.remove.empty.space.between.rows" value="true"**/>**

* The above gets rid of one row with a very large row height at the end of the data tab. Don't understand why that one very high row is created, but this is needed to get rid of it. Of course having this means that you can't (easily) have row spacing in the footnotes.

**<property** name="net.sf.jasperreports.export.xls.remove.empty.space.between.columns" value="true"**/>**

* The above gets rid of empty columns, such as caused by margins, or by poor layout.

**<property** name="net.sf.jasperreports.export.xls.white.page.background" value="false"**/>**

* The above shows the Excel grid lines, otherwise you get solid white, which kind of ruins the look of Excel (for a data dump).

**<property** name="net.sf.jasperreports.export.xls.detect.cell.type" value="true"**/>**

* The above tells Excel to treat dates as dates, numbers as numbers, instead of all being treated as text when it is set to "false" (which is the default).
* ***Note:*** Java's custom date formats are different from Excel's custom date formats, so a date format might look good in Excel but bad in Java/HTML, or vis-a-versa, so take care with custom date formats.

**<property** name="net.sf.jasperreports.export.xls.ignore.graphics" value="false"**/>**

* The above tells Excel to not delete the company logo graphic. "true" (which is the default) deletes graphics.

**<property** name="net.sf.jasperreports.page.break.no.pagination" value="apply"**/>**

* The above, as far as I know, has to be typed or cut-n-pasted directly into the Source XML. I have not found a place to select it under any Properties. I am not certain why it is needed, but it is.

**<property** name="net.sf.jasperreports.export.xls.sheet.names.all" value="Data/Footnotes"**/>**

* The above tells Excel, to name the first tab "Data", and the second tab "Footnotes". Instead of "Data", I would recommend something more descriptive, maybe the report name.

**<property** name="net.sf.jasperreports.export.xls.freeze.row" value="3"**/>**

* The above tells Excel to freeze the top 2 rows. Once in Excel therefore, the report name/company logo row, and the column headers row is frozen, and the user can scroll through thousands of records and still see them. Without this, the user would have had to do it themselves.

**<property** name="net.sf.jasperreports.print.keep.full.text" value="true"**/>**

* The above keeps JS from truncating values wider than the explicity set colunmn widths. In Excel this is seen as wrapping.

**<property** name="net.sf.jasperreports.exports.xls.font.size.fix.enabled" value="false"**/>**

* The above keeps JS from reducing the font size by 1 when exporting to Excel. For example, without this, an Arial 10 in Studio, in Server when exported to Excel ends up as an Arial 9 font.

When looking at your report, the look in Preview is different than the look in JasperReports Server.

To keep the pagination, when running the report from the server, then Save/Export as Excel or XLSX. Excel Paginated or XLSX Paginated reintroduce Excel paging (which you don't want).

## Indicaciones

* Tablas se manejan como listas, objetos como parámetro CommercialReport.
* Cuando mando un objeto con un array dentro, debo crear una clase para almacenar esa lista y pasarla como otro parámetro (ejemplo: ReceivedPayments).
* ColumnHeader se imprime en todas las hojas del PDF, Details solo cuando aparece por primera vez y ya.

## Ubicación de reportes (Jasper)

### Breakpoint

**return** builder.buildReportDetails(elementList, docType);

Así sabemos que builder se manda a llamar al darle click a imprimir cualquier reporte.

### Headers

BGPHeader -> Ahorro, corriente, TC prepagada

BGPHCreditCard –> TC

### Associated

BGPLastMov -> Ahorro, Corriente

BGPMasterCardMovements y BGPCardMovements -> TC

### Transaction

BGPPaymentDetail -> Pagos

BGPPerformedList -> Realizadas

BGPPerformedDetail -> Detalle de Realizadas

BGPScheduledDetail -> Programadas

BGPTransactionHistory -> Historial de programadas

BGPStockConfirmations -> Confirmaciones bursátiles

BGPStockConfirmation -> Imagen de Confirmación bursátil

BGPTransfer y TransferDetail-> Cuentas propias, Terceros, Internacionales, Recargas…

### Associated (Excel)

BGRCardExcelReport -> TC

## Mostrar caracteres especiales

String nameCompany = ReflectionUtil.getPropertyString(filter, "nameCompany");

String amp = "\u0026amp;";

intFilter.setNameCompany(nameCompany.replace("&", amp));

## Ubicación de reportes (Builders y Mappers)

### Detalles del producto (Al seleccionar cualquier producto de la pestaña Cuentas)

HeadderMapper.Java -> se usa en algunos, no en todos…

Cada producto tiene su reporte, ej: SavingAccountDetailBuilder.java

### Últimos Movimientos (Al seleccionar cualquier producto de la pestaña Cuentas)

HeadderMapper.Java -> es el mismo para todos…

MovementsBuilder.Java -> es el mismo para todos… incluye estados de cuenta…

BankStateMapper -> Mapeado de estado de cuenta (ahorro, corriente y tarjeta de crédito)

### Transacciones

TransactionDetailBuilder -> Detalle en cuentas (adenda, TC, etc)

TransactionDetailTransferBuilder -> Propias, Terceros, Internacionales

## Manejo de líneas en el reporte

Si le pongo Float se mueve más abajo si el campo de arriba se vuelve más grande, si le pongo fixed relative to top se queda en la misma posición…

## Frames que desaparecen si no viene información

Frame Padre e hijos -> Float

Hijos -> Remove Line When Blank para que desaparezca y el de abajo suba de posición

## 2 celdas en la misma columna

Text Field -> Line Spacing -> 1\_1\_2



Diferentes tipos de Stretch Type: ContainerHeight sería la opción para centrar verticalmente (utilizado en: desembolsado el)

## Mapear campos desde el properties

import com.liferay.portal.kernel.language.LanguageUtil;

import com.liferay.portal.kernel.util.LocaleUtil;

import org.osgi.service.component.annotations.Component;

listItem.setStatus(ReflectionUtil.getPropertyString(list, STATUS));

String statusDescription = ReflectionUtil.getPropertyString(list,

RESULT, DESCRIPTION);

listItem

.setAuditState(LanguageUtil.get(LocaleUtil.getDefault(), statusDescription));

## Mostrar caracteres especiales

String nameCompany = ReflectionUtil.getPropertyString(filter, "nameCompany");

String amp = "\u0026amp;";

intFilter.setNameCompany(nameCompany.replace("&", amp));

## Primera letra mayúscula

String numToMonth = getMonth(ReflectionUtil.getPropertyInteger(filter, "month"));

filterWrap.setToMonth(numToMonth.substring(0, 1).toUpperCase() + numToMonth.substring(1));

// Obtener mes

public String getMonth(int month) {

return new DateFormatSymbols().getMonths()[month - 1];

}

## No usar Jasper, imprimir desde vista (jsPDF)

vm.printDetail = function() {

var today = new Date();

vm.dateAsString = $filter('date')(today, "dd-MMM-yyyy - hh:mm a");

var footer = tFilter('stock-market-confirmations-copyright-text', [today.getFullYear()]).toString();

var footerSbp = tFilter('stock-market-confirmaciones-footer-sbp').toString();

var footerSbpLink = tFilter('stock-market-confirmaciones-footer-sbp-link').toString();

var img1 = new Image();

var img2 = new Image();

img1.onload = function() {

var pageHeight = 215.9; // 8.5" in mm

var pageWidth = 279.4; // 11" in mm

var doc = new jsPDF('l', 'mm', 'letter');

doc.addImage(img1, 'PNG', 10, 10, 90, 12);

doc.addImage(img2, 'PNG', 232, 35, 28, 10);

doc.setLineWidth(0.5);

doc.setDrawColor(255, 121, 0);

doc.line(10, 25, pageWidth - 10, 25);

doc.setDrawColor(246, 174, 57);

doc.line(20, 35, pageWidth - 20, 35);

doc.setDrawColor(115, 115, 115);

doc.line(10, pageHeight - 20, pageWidth - 10, pageHeight - 20);

doc.setDrawColor(44, 134, 198);

doc.line(71, pageHeight - 9, 136, pageHeight - 9);

doc.setFontSize(12);

doc.setTextColor(35, 75, 127);

doc.setFont('sans-serif');

doc.text(vm.dateAsString, pageWidth - 58, 20);

doc.setFontSize(16);

doc.setTextColor(0, 56, 130);

doc.text(tFilter('stock-market-confirmations-modal-title').toString() + ' #' + vm.confirmationId, 20, 32);

doc.setFontSize(6);

doc.setTextColor(0, 0, 0);

doc.setFont('monospace');

doc.text(vm.details, 40, 20);

doc.setFontSize(10);

doc.setFont('sans-serif');

doc.setTextColor(0, 0, 0);

doc.text(footer, 10, pageHeight - 15);

doc.text(footerSbp, 10, pageHeight - 10);

doc.setTextColor(44, 134, 198);

doc.text(footerSbpLink, 71, pageHeight - 10);

if (window.navigator && window.navigator.msSaveOrOpenBlob) {

window.navigator.msSaveOrOpenBlob(doc.output('blob'));

}

else {

window.open(doc.output('bloburl'), '\_blank');

}

};

img1.src = "/o/stock-market-confirmations-web/images/bgp-logo-primary.png";

img2.src = "/o/stock-market-confirmations-web/images/bg-valores.png";

};

# VSCode

## User settings:

### Confuguración general

{

"workbench.colorTheme": "Tomorrow Night Blue",

"window.zoomLevel": -1,

"files.autoSave": "onFocusChange",

"editor.tabSize": 2,

"explorer.confirmDelete": false,

"workbench.statusBar.visible": true,

"workbench.activityBar.visible": true,

"editor.minimap.enabled": false,

"java.errors.incompleteClasspath.severity": "ignore",

}

## Worskpace settings:

### Ignorar archivos o carpetas en el area de trabajo

{

  "folders": [

    {

      "path": "C:\\Users\\eRiveRa\\comercial"

    }

  ],

  "settings": {

    "files.exclude": {

      "\*\*/node\_modules": true,

      "\*\*/build": true,

      "\*\*/bundles": true,

      "\*\*/bin": true,

      "\*\*/\*.zip": true,

      "\*\*/\*.rar": true,

      "\*\*/.classpath": true,

      "\*\*/.metadata": true,

      "\*\*/.settings": true,

      "\*\*/.sonarlint": true,

      "\*\*/.vscode": true,

      "\*\*/.recommenders": true,

      "\*\*/.gradle": true,

      "\*\*/.gitlab": true,

      "\*\*/.gitattributes": true,

      "\*\*/.gitignore": true,

      "\*\*/.git": true,

      "\*\*/.project": true,

      "\*\*/\*.yml": true,

      "\*\*/.svn": true,

      "\*\*/.hg": true,

      "\*\*/CVS": true,

      "\*\*/.DS\_Store": true

    }

  },

}

# Servicios

## Construcción de nuevos servicios

### Error: Null Pointer

2018-09-26 15:19:39.698 WARN  [http-nio-8080-exec-8][code\_jsp:172] {code="404", msg="ProxyServlet: /o/bgp-theme/js/angular-md5.min.js.map", uri=/o/bgp-theme/js/angular-md5.min.js.map}  
2018-09-26 15:19:39.738 WARN  [http-nio-8080-exec-4][code\_jsp:172] {code="404", msg="ProxyServlet: /o/bgp-theme/css/style.css.map", uri=/o/bgp-theme/css/style.css.map}  
2018-09-26 15:21:06.912 WARN  [http-nio-8080-exec-5][PhaseInterceptorChain:449] Interceptor for {http://xmlns.bgeneral.com/ServicioIntegracion/DominioLineaCredito/LineaCredito/V1}LineaCreditoPortTypeSOAPService#{http://xmlns.bgeneral.com/ServicioIntegracion/DominioLineaCredito/LineaCredito/V1}ConsultarReporteLineaCredito has thrown exception, unwinding now  
java.lang.NullPointerException

SOLUCIÓN: Actualizar el properties, tal vez falte:

bg.bel.ws.address.esb.credit.line=${bg.bel.ws.address}/DominioLineaCredito/LineaCredito/V1/LineaCreditoESBPortSOAP

# Java

## Mappers

UserMapper -> toProfile, toUser (información del usuario en hilo)

ProductWSMapper -> mapProductFields (detalle cuentas del consolidado)

PersonalTransferController -> newInitialStateParams(cuentas origen y destino)

DashboardProduct -> addAccount (se setea la info de los productos que llega al consolidado)

ChargeUtil -> getChargeAccounts (cuentas para mostrar en combo de morosidad)

TransactionDTOMapper -> toTransaction (lista de realizadas)

TransactionPerformedServiceImpl -> getDetailedTransaction (detalles de las realizadas)

## Crear una excepción

modified: common-api/src/main/java/com/bgeneral/bel/common/exception/ApiExceptionManager.java

modified: common-api/src/main/java/com/bgeneral/bel/common/model/WSStatus.java

modified: common-api/src/main/java/com/bgeneral/bel/common/util/AbstractGenericService.java

modified: common-api/src/main/java/com/bgeneral/bel/common/util/RestApiError.java

Untracked files:

(use "git add <file>..." to include in what will be committed)

common-api/src/main/java/com/bgeneral/bel/common/exception/AnnuityChargeException.java

## common-api/src/main/java/com/bgeneral/bel/common/exception/HasConditionException.java

## Métodos útiles

### ProductService (Obtener cuenta en base a número o posición en el arreglo)

Product getProductById(String id) **throws** BGPException;

Product getProductByPos(**int** pos) **throws** BGPException;

## Iterar un objeto y mapear otro en base a una condición

List<Product> credit = **new** ArrayList<>();

**for** (Product pro : getProductSummary()) {

**if** (!pro.isCreditAccount()) {

**continue**;

}

credit.add(pro);

}

## Ordenar una lista

Collections.*sort*(companies, **new** Comparator<Company>() {

@Override

**public** **int** compare(Company c1, Company c2) {

**return** c1.getName().compareTo(c2.getName());

}

});

## Iterar un enum

**public** **static** TransferState getState(String value) {

**for** (TransferState e : TransferState.*values*()) {

**if** (e.state.equals(value)) {

**return** e;

}

}

**return** ***EMPTY***;

## (ProductTypeUtil) - Métodos para dar formato a montos y enmascarar cuentas

dto.setComission(

CommonUtil.*formatNumber*(transaccion.getMontoComision(), BGStringPool.***DOLLAR***));

**if** (subTipoProducto.getCodigo().equals(***BGP\_PRODUCTCARD***)) {

dto.setDestinationAccountNumber(

ProductTypeUtil.*getMaskCreditCardNumber*(cuentaDestino.getNumeroCuenta()));

} **else** {

dto.setDestinationAccountNumber(

ProductTypeUtil.*getMaskedAccount*(cuentaDestino.getNumeroCuenta()));

}

## Mapeo

annualFee.setFee(resp.getBODY().getCobroAnualidad().get(0).getValorCobro());

CommonUtil.*formatNumber*(product.getAvailableBalance(), "$")).toString()

.append(LanguageUtil.*get*(CommonUtil.***DEFAULT\_LOCALE***, "common-account-text"))

## Enmascaramiento de cuentas

**public** **static** String getMaskedTicketNumber(String value) {

**if** (Validator.*isBlank*(value)) {

**return** value;

}

String valueformated = StringUtils.*leftPad*(value, ***CREDIT\_CARD\_SIZE\_FIFTEEN***, "0");

**return** valueformated.replaceAll("(\\d{3})(\\d{2})(\\w\*)", "$1-$2-$3");

}

\d digitos, \w alfanumérico, leftPad pone 0 a la izquierda hasta llenar los 15.

## Formato a números y fechas para mandar al Jasper

NumberFormat formatter = **new** DecimalFormat("#,##0.00");

BigDecimal amount =

ReflectionUtil.*getPropertyBigDecimal*(inputObject, ***MOVEMENT***, ***AMOUNT\_MOVEMENT***);

**if** ("D".equals(natureMovement)) {

transactionReport.setLblAmount("-$" + formatter.format(amount.floatValue()));

} **else** {

transactionReport.setLblAmount("$" + formatter.format(amount.floatValue()));

}

SimpleDateFormat formatDate = **new** SimpleDateFormat("dd-MMM-yyyy", **new** Locale("es", "pa"));

Date effectiveDate =

ReflectionUtil.*getPropertyDate*(inputObject, ***MOVEMENT***, ***EFFECTIVE\_DATE***);

transactionReport.setDateCreated(formatDate.format(effectiveDate));

## Manejo de Errores

RestApiError.java

WSStatus.java

## Inyección de código

Revisar cambios entre DashboardPerformedTransactionsController de comercial con personal…

## Comparar BigDecimals

if (reportWapper.getPayments().getPaymentsTotal()

.compareTo(reportTemp.getPaymentsTotal()) != 0) {

-1 es menor, 0 iguales, 1 es mayor

## Java 8 (Streams List Collector Example)

A Java 8 example to show you how to convert a Stream to a List via Collectors.toList

Java8Example1.java

package com.mkyong.java8;

import java.util.List;

import java.util.stream.Collectors;

import java.util.stream.Stream;

public class Java8Example1 {

public static void main(String[] args) {

Stream<String> language = Stream.of("java", "python", "node");

//Convert a Stream to List

List<String> result = language.collect(Collectors.toList());

result.forEach(System.out::println);

}

}

Copy

output

java

python

node

Copy

Yet another example, filter a number 3 and convert it to a List.

Java8Example2.java

package com.mkyong.java8;

import java.util.List;

import java.util.stream.Collectors;

import java.util.stream.Stream;

public class Java8Example2 {

public static void main(String[] args) {

Stream<Integer> number = Stream.of(1, 2, 3, 4, 5);

List<Integer> result2 = number.filter(x -> x != 3).collect(Collectors.toList());

result2.forEach(x -> System.out.println(x));

}

}

Copy

output

1

2

4

5

# JSP

## Inicializar variable en controlador JS tomando un valor desde Java

### controller.js

//Page Permissions for product menu

vm.pagePermissions = {};

### view.jsp

<div ng-controller="BGPDashboardController as dashboardCtrl"

ng-init="dashboardCtrl.pagePermissions.ownAccounts=

'<%= PagePermissions.CUENTAS\_PROPIAS.getName() %>';

dashboardCtrl.pagePermissions.payments=

'<%= PagePermissions.PAGOS.getName() %>';

dashboardCtrl.pagePermissions.minPayment=

'<%= PagePermissions.PAGO\_MINIMO.getName() %>';

dashboardCtrl.pagePermissions.deposit=

'<%= PagePermissions.DEPOSIT.getName() %>';

dashboardCtrl.pagePermissions.debtBalance=

'<%= PagePermissions.SALDO\_ADEUDADO.getName() %>';

dashboardCtrl.pagePermissions.doPayment=

'<%= PagePermissions.REALIZAR\_PAGO.getName() %>';">

<loading-component display="dashboardCtrl.showLoading"></loading-component>

### PagePermisions.java

<%@page import="com.bgeneral.bel.dashboard.account.util.PagePermissions"%>

<%@ include file="/init.jsp" %>

<div ng-controller="BGPDashboardController as dashboardCtrl"

ng-init="dashboardCtrl.pagePermissions.ownAccounts=

'<%= PagePermissions.CUENTAS\_PROPIAS.getName() %>';

dashboardCtrl.pagePermissions.payments=

'<%= PagePermissions.PAGOS.getName() %>';

dashboardCtrl.pagePermissions.minPayment=

'<%= PagePermissions.PAGO\_MINIMO.getName() %>';

dashboardCtrl.pagePermissions.deposit=

'<%= PagePermissions.DEPOSIT.getName() %>';

dashboardCtrl.pagePermissions.debtBalance=

'<%= PagePermissions.SALDO\_ADEUDADO.getName() %>';

dashboardCtrl.pagePermissions.doPayment=

'<%= PagePermissions.REALIZAR\_PAGO.getName() %>';

dashboardCtrl.pagePermissions.detail=

'<%= PagePermissions.DETALLE\_DE\_CUENTA.getName() %>';">

<loading-component display="dashboardCtrl.showLoading"></loading-component>

<ng-include src="'/o/dashboard-web/templates/dashboard.html'"></ng-include>

</div>

## Inicializar una variable validando una condición en la URL

### View.jsp

<%@ include file="/init.jsp"%>

<%

String initialRoute = "";

if(themeDisplay.getURLCurrent().contains("/editar-pago")){

initialRoute = "collector\_payment\_edit";

} else {

initialRoute = "collector\_payment\_selection";

}

%>

<script>

var initialRoute = "<%= initialRoute %>";

</script>

<div class="ng-scope">

<div class="bgp-common-module">

<div ng-controller="BGPCollectorPaymentController as CollectorPaymentCtrl">

<div class="row">

<div class="col-sm-12">

<h4 ng-bind-html="CollectorPaymentCtrl.pageTitle() | t" class="bgp-title\_\_bordered--bottom">

</h4>

</div>

</div>

<div ui-view="BGPCollectorPaymentView">

<loading-component display="CollectorPaymentCtrl.showLoading"></loading-component>

</div>

</div>

</div>

</div>

### Controller.js

if (typeof initialRoute !== 'undefined') {

vm.initialRoute = initialRoute;

}

$state.go(vm.initialRoute);

### Routes.js

var states = [

"selection",

"data",

"edit",

"validate",

"authenticate",

"confirmation",

"error",

"no\_deposit\_accounts",

"unavailable"

];

\_.map(states, function(state) {

var stateName = "collector\_payment\_" + state;

# GIT

## Comandos

### Cambiar dirección remota

git remote set-url origin \*URL.git\*

# BGP-Theme

## Parámetros en el properties

C:\Users\eRiveRa\Desktop\2018-11-06 13_21_01-Detalle de cuenta - Banco General.png

product-info-line-of-credit-breakdown-title = Desglose de {0} ({1})

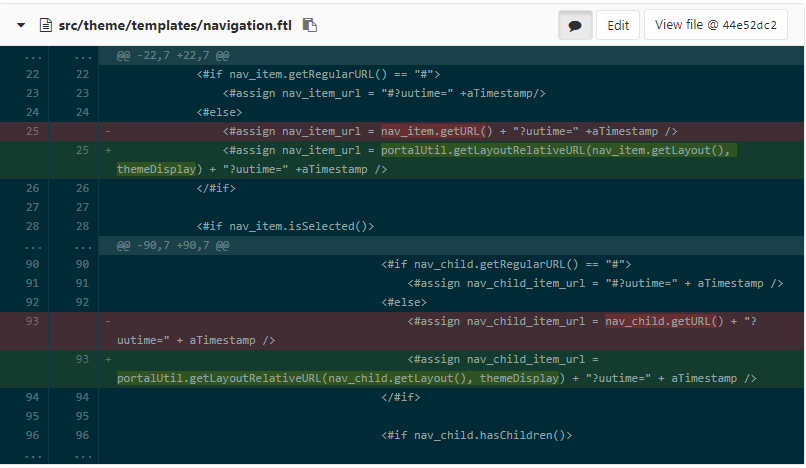
<div class="text-color-brand-blue fnt-size-md font-weight-semibold">

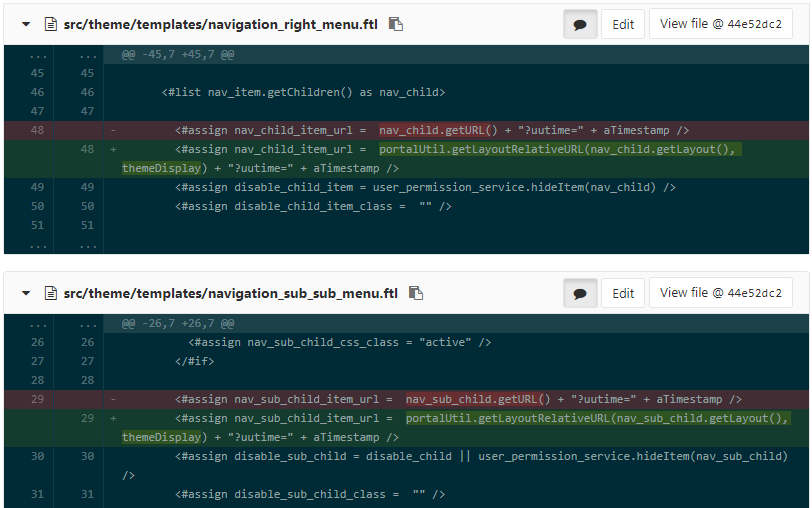
{{"product-info-line-of-credit-breakdown-title" | t:[ctrl.productWrapper.product.disbursementList[0].operationAlias,

ctrl.productWrapper.product.disbursementList[0].maskedNumber]}}

</div>

## Rutas relativas





# Caracteres especiales

## Unicode

Á : \u00C1

á : \u00E1

É : \u00C9

é : \u00E9

Í : \u00CD

í : \u00ED

Ó : \u00D3

ó : \u00F3

Ú : \u00DA

ú : \u00FA

Ü : \u00DC

ü : \u00FC

Ñ : \u00D1

ñ : \u00F1

" : \u0022

< : \u003C

> : \u003E

í : \u00ED

" : \u0022

' : \u0027

© : \u00A9

® : \u00AE

€ : \u20AC

¼ : \u00BC

½ : \u00BD

¾ : \u00BE