Complex Splunk Query Notes.txt

github.com/tim-rafferty/splunk-queries/blob/main/Complex Splunk Query Notes.txt

tim-rafferty/**splunk-queries**

Repo for useful Splunk queries and dashboards that I want to remember for future use

1 Contributor 0 Issues 1 Star 0 Forks

1
2
3
4
5
6
7
8
9
10
11
12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

101

102

103

104

105

106

107

108

109

110

111

112

113

114

115

116

117

118

119

120

121

122

123

124

125

126

127

128

129

130

131

132

133

134

135

136

137

138

139

140

141

142

143

144

145

146

147

148

149

150

151

152

153

WF Package Completed Delays

---

index=blend-sensitive-prod parsed.tenant=wellsfargo

(kubernetes.container_name=disclosures OR kubernetes.container_name=blend-wellsfargo-mediumpriorityworkers)

```
(parsed.data.packageType=CLOSING_PACKAGE OR
parsed.event.params.contextSelectors.packageType=CLOSING_PACKAGE)

(parsed.httpMethod=POST OR parsed.httpMethod=PATCH OR "In disclosures event handler" OR

"Sending disclosures package status event to the events pipeline" OR "Sending disclosures
recipient status event to the events pipeline")

(parsed.data.status=COMPLETED OR parsed.data.status=SIGNED OR
parsed.data.status=CANCELLED OR parsed.data.status=VIEWED OR

parsed.event.params.value=COMPLETED OR parsed.event.params.value=SIGNED OR
parsed.event.params.value=CANCELLED OR parsed.event.params.value=VIEWED)

(parsed.data.applicationId=* OR parsed.loanId=*)

(parsed.originalUrl="/api/events/envelopes" OR parsed.originalUrl="/api/disclosures/*" OR
"DisclosuresExternal.status")

| rex "(packageI....)(?P<Package_UUID>\w{8}\S\w{4}\S\w{4}\S\w{4}\S\w{12})"

| rex "(applicationId...|loanId...)(?P<Loan_UUID>\w{8}\S\w{4}\S\w{4}\S\w{4}\S\w{12})"

| stats earliest(eval(if(searchmatch("SIGNED"), _time, none))) as Package_Signed,

earliest(eval(if(searchmatch("VIEWED"), _time, none))) as Package_Viewed,

latest(eval(if(searchmatch("COMPLETED"), _time, none))) as Package_Completed,

latest(eval(if(searchmatch("CANCELLED"), _time, none))) as Package_Cancelled by
Package_UUID Loan_UUID

| eval Duration_Secs=Package_Completed-Package_Signed

| eval Package_Completed=strftime(Package_Completed,"%m/%d/%y %H:%M:%S.%3N")

| eval Package_Cancelled=strftime(Package_Cancelled,"%m/%d/%y %H:%M:%S.%3N")

| eval Package_Signed=strftime(Package_Signed,"%m/%d/%y %H:%M:%S.%3N")

| eval Package_Viewed=strftime(Package_Viewed,"%m/%d/%y %H:%M:%S.%3N")

| eval HourAgo=relative_time(now(),"-1h")

| eval HourAgo=strftime(HourAgo,"%m/%d/%y %H:%M:%S.%3N")

| eval Duration_Mins=round(Duration_Secs/60,0)
```

| where Package_Signed!="" AND Package_Viewed!="" AND (isnull(Package_Completed) OR Duration_Mins>45)

AND isnull(Package_Cancelled) AND Package_Signed<HourAgo AND Package_Viewed<HourAgo

| fields Loan_UUID, Package_UUID, Package_Signed, Package_Viewed, Package_Completed, Package_Cancelled, Duration_Mins

---

1. Filters logs related to disclosures processing for closing packages in Wells Fargo.

2. Extracts relevant package and loan UUIDs.

3. Captures timestamps for key events: Signed, Viewed, Completed, and Cancelled.

4. Calculates delay duration between signing and completion.

5. Identifies delayed or incomplete packages:

- Packages signed & viewed ✅

- Not completed OR took longer than 45 minutes to complete ⏳

- Not canceled ❌

- Signed/viewed at least an hour ago 🕑

6. Outputs a report with loan ID, package ID, timestamps, and delay duration.

This query helps in identifying workflow bottlenecks where closing packages are getting stuck after signing and viewing.

WF Submitted Apps Not Exported

(index=blend-sensitive-mulesoft-prod OR index=blend-sensitive-prod) wellsfargo

(service="blend-wells-cb-connector-api" OR service="blend-sys-api" OR service="blend-consumer-banking-prc-api" OR kubernetes.container_name=blend-wellsfargo-mediumpriorityworkers)

(message.statusCode=200 OR message.tracepoint=END OR message.payload.eventType="applicationUpdated" OR message.text="*PATCHING*" OR message.eventData.data.trigger.type=EXPORTED OR message.tracepoint="blend-api-post-status-updates" OR parsed.loanType="PERSONAL_LOAN")

| rex "(trackingId....|applications.|loanId...)(?P<Loan_UUID>\w{8}\S\w{4}\S\w{4}\S\w{4}\S\w{12})"

| stats earliest(eval(if(searchmatch("Event received from Blend"), _time, none))) as Export_Triggered, latest(eval(if(searchmatch("Response from wellsFargo after loan export") OR searchmatch("Loan export successfully completed") OR searchmatch("export-status") OR searchmatch("Blend API Patch Loan LosId") OR searchmatch("EXPORTED") OR searchmatch("Update BONS status to SUCCEEDED") OR searchmatch("Loan is not eligible for TWN on submit"), _time, none))) as Export_Completed by Loan_UUID

| eval Export_Duration_In_Secs=Export_Completed-Export_Triggered

| eval Export_Completed=strftime(Export_Completed,"%m/%d/%y %H:%M:%S.%3N")

| eval Export_Triggered=strftime(Export_Triggered,"%m/%d/%y %H:%M:%S.%3N")

| eval HourAgo=relative_time(now(),"-1h")

| eval HourAgo=strftime(HourAgo,"%m/%d/%y %H:%M:%S.%3N")

| eval Export_Duration_In_Mins=round(Export_Duration_In_Secs/60,0)

| where Export_Triggered!="" AND (isnull(Export_Completed) OR Export_Duration_In_Mins>240) AND Export_Triggered<HourAgo

| fields Loan_UUID Export_Triggered Export_Completed Export_Duration_In_Mins

---

- This query identifies loan applications that were submitted but not exported successfully.
- It extracts loan UUIDs, tracks export start & completion times, and calculates export duration.
- It then filters out loans that either failed to export or took longer than 4 hours.
- The final output presents only the Loan UUID, export timestamps, and export duration.

US Bank Mobile Screenshot

---

index=blend-sensitive-prod (kubernetes.container_name=disclosures OR kubernetes.container_name=files) parsed.document.primaryFile.url=*

[search index="blend-sensitive-prod" "usbank/documents/" parsed.apiRequestId=*

[search index="blend-sensitive-prod" parsed.deployment=usbank parsed.originalUrl="/v1/internal/api/activities/document-screenshot"

| dedup parsed.apiRequestId

| table parsed.apiRequestId]

| rename parsed.keys{} as parsed.document.primaryFile.url

| dedup parsed.document.primaryFile.url

| table parsed.document.primaryFile.url]

| rename parsed.document.displayFileName as fileType, parsed.document.fileName as fileName, parsed.document.primaryFile.createdBy as userId, parsed.loanId as loanId, parsed.tenant as tenant, parsed.document._id as docId

| dedup docId | table _time loanId docId fileName fileType userId tenant | sort - _time

---

1. Finds events in blend-sensitive-prod related to disclosures and files.

2. Filters document URLs that are linked to document-screenshot API activities.

3. Extracts relevant metadata about documents such as file name, type, user, loan ID, and tenant.

4. Ensures only unique documents are included in the results.

5. Sorts the documents by timestamp, showing the most recent ones first.

WF CRV Integration Failures

---

index=blend-sensitive-prod parsed.level=error kubernetes.container_name="wells-consumer-banking"

parsed.statusText="Bad Request" parsed.message="validation error"

[search index=blend-sensitive-prod POST "/wells-fargo-personal-loan/applications" "914191e433594e01bbff397189a7ded1"

"wells-consumer-banking" "mTLS validation failed but tenant has permissive configuration"

| dedup parsed.dd.trace_id | table parsed.dd.trace_id] | rename parsed.errors{}.field{} as err_flds,

parsed.errors{}.location as err_location, parsed.errors{}.messages{} as err_info, parsed.statusText as err_status,

parsed.status as err_code, parsed.message as err_msg, parsed.name as err_name

| table _time err_flds err_location err_info err_status err_code err_msg err_name

---

1. Finds error logs (level=error) from the wells-consumer-banking container.

2. Filters errors related to validation failures where the status is "Bad Request".

3. Matches these errors with specific mTLS validation failures from another set of logs (via subsearch).

4. Extracts relevant fields, renames them for clarity, and displays them in a structured table.

US Bank Package Failed Creation

---

index=blend-sensitive-prod-blend-bailey-web parsed.task.processId="Disclosures*" parsed.tenant=usbank parsed.task.loanRequest.requirement.disclosuresPackageId=*

| rename parsed.task.loanRequest.requirement.disclosuresPackageId as packageID | search packageID=*

[search index=blend-sensitive-prod "kubernetes.container_name"=disclosures "parsed.tenant"=usbank "parsed.message"="Status for package * is FAILED_TO_CREATE"

| rex "Status for package (?P<packageID>\w{8}\S\w{4}\S\w{4}\S\w{4}\S\w{12}) is FAILED_TO_CREATE"

| table packageID]

| dedup parsed.loanId

| table _time parsed.tenant parsed.loanId parsed.envelopeId parsed.task.requestId

| sort - _time

The query provides a list of loan requests where disclosure package creation failed, along with relevant details like loan ID, envelope ID, and request ID. It ensures that only failed packages (as identified by the subsearch) are included.

Mr Cooper mrcCustomerId Failed

---

index="blend-sensitive-prod-blend-bailey-web" parsed.loanId=* parsed.borrowerId=*

parsed.tenant=mrcooper kubernetes.container_name=blend-bailey-web

| rename parsed.borrowerId as partyId | join partyId

[search index="blend-sensitive-prod-blend-bailey-web" error mrcCustomerId parsed.tenant=mrcooper "/api/external/parties/*"

| rex "/api/external/parties/(?P<partyId>\w{8}\S\w{4}\S\w{4}\S\w{4}\S\w{12})" | dedup partyId | table partyId parsed.err.message]

| dedup parsed.loanId

| table _time parsed.tenant parsed.err.message parsed.loanId partyId

---

1. It searches for loan transactions (loanId and borrowerId) for Mr. Cooper within blend-bailey-web logs.

2. It joins with another search that finds mrcCustomerId failures related to /api/external/parties/.

3. It extracts the partyId (UUID format) from the API request URL.

4. It removes duplicates and presents a clean output showing loan ID, borrower ID, error message, and timestamp.

Keybank ESOCKETTIMEDOUT

---

index="blend-sensitive-prod" ESOCKETTIMEDOUT parsed.tenant=key
parsed.details.event.data.eventType="applicationExportRequested"
"parsed.dd.service"="beehive-retryworkers"

| rex "los-integrations.blendlabs.com:443...(?P<error>\w{5}..\w{15})"

| dedup parsed.details.event.data.data.loanId

| rename parsed.details.event.data.data.loanId as loanId,
parsed.details.event.data.data.exportRequestReason as exportReason,
parsed.details.event.data.eventType as eventType, parsed.tenant as client

| table _time client loanId error exportReason eventType

---

1. Searches logs in the blend-sensitive-prod index for Keybank transactions with ESOCKETTIMEDOUT errors.

2. Filters logs to only include application export requests handled by the beehive-retryworkers service.

3. Extracts a specific error code from logs where "los-integrations.blendlabs.com:443" is present.

4. Removes duplicate loan IDs to prevent multiple entries for the same loan.

5. Renames fields to make the results more readable.

6. Displays the final output in a table format with relevant details.
