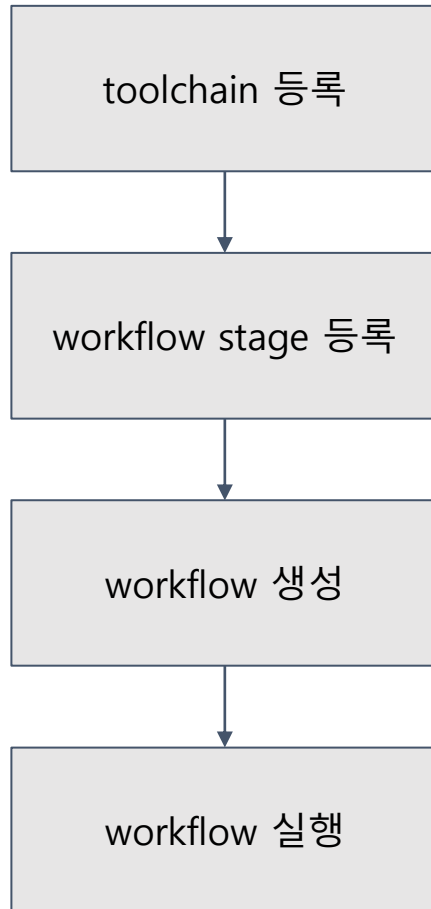


1. toolchain
2. workflow stage(젠킨스 스테이지)
3. workflow
4. Software catalog

workflow 생성 및 실행 순서



연동될 OSS정보들을 등록

- 연동되어 사용될 jenkins, gitlab, argoCD, nexus, tumblebug 등
- Workflow용 oss인 jenkins는 workflow관련 기능을 위해 필수

workflow를 구성할 stage 등록(선택)

- 템플릿 개념의 stage 등록 후 workflow에서 사용 가능
- 필수 아님

등록된 stage들을 조합, 직접 입력하여 workflow 생성

ui 또는 api를 활용하여 workflow 실행

1-1. 툴체인 목록

1. 목록

API : (GET) /config/oss/list
Desc : 툴체인 목록을 조회한다.

연동사용 OSS(nexus, jenkins 등)을 위한 정보를 입력 및 조회
db등의 정보 이외에는 직접 등록 후 사용한다.

툴체인 목록

전체 (5)

검색어를 입력하세요.

+ 새 툴체인 등록

TOOL NAME ⇅	NAME ⇅	Description ⇅	URL ⇅	생성일 ⇅	명령
GITLAB	gitlab(10.10.30...	Manifest Repository	http://10.10.30.21:18082/	2023-09-13 15:21:09	<div>수정</div> <div>삭제</div>
JENKINS	jenkins(10.10.3...	WorkFlow	http://10.10.30.22:9800	2023-09-13 15:22:16	<div>수정</div> <div>삭제</div>
NEXUS	nexus(10.10.30...	Image/Package Repository	http://10.10.30.21:18081/	2023-10-26 14:01:02	<div>수정</div> <div>삭제</div>
NEXUS	nexus(210.217.1...		http://210.217.178.130:8081/	2023-11-16 13:27:08	<div>수정</div> <div>삭제</div>
TUMBLEBUG	asdfasdf		http://tb.tb-cb.kro.kr:1323	2024-06-13 03:57:10	<div>수정</div> <div>삭제</div>

1-2. 툴체인 등록 (중복체크 / 연결확인)

1. 중복체크

API : (GET) /config/oss/duplicate

Param : (QueryParam)

ossName=tumblebug&

ossUrl=http:%2F%2Ftb.tb-cb.kro.kr:1323&

ossUsername=default

Desc : 등록될 툴체인 명 중복 검사를 한다.

1. 연결확인

API : (POST) /config/oss/connection/check

Param : (Request Body)

```
{
  "ossCd": "TUMBLEBUG",
  "ossName": "tumblebug",
  "ossDesc": "",
  "ossUrl": "http://tb.tb-cb.kro.kr:1323",
  "ossUsername": "default",
  "ossPassword": "default",
  "ossToken": ""
}
```

Desc : 등록될 툴체인의 ID / PW로 연결확인을 한다.

툴체인 등록

구분 *

툴체인을 선택하세요

툴체인 명 *

툴체인 명을 입력하세요

URL

URL을 입력하세요

사용자명 *

사용자명을 입력하세요

비밀번호 *

비밀번호

툴체인 토큰

툴체인 토큰을 입력하세요

중복체크

연결 확인

툴체인 설명

설명을 입력하세요

취소

등록

<http://app.tb-cb.kro.kr/>

1-2. 툴체인 등록 (툴체인 등록)

3. 툴체인 등록

API : (POST) /config/oss

Param : (Request Body)

```
{
  "ossCd":"TUMBLEBUG",
  "ossName":"tumblebug",
  "ossDesc": "",
  "ossUrl":"http://tb.tb-cb.kro.kr:1323",
  "ossUsername":"default",
  "ossPassword":"ZGVmYXVsdA==",
  "ossToken":""
}
```

Desc : 툴체인을 등록한다.

툴체인 등록

구분 *

툴체인을 선택하세요

툴체인 명 *

툴체인 명을 입력하세요

URL

URL을 입력하세요

사용자명 *

사용자명을 입력하세요

비밀번호 *

비밀번호

툴체인 토큰

툴체인 토큰을 입력하세요

중복체크

연결 확인

툴체인 설명

설명을 입력하세요

취소

등록

<http://app.tb-cb.kro.kr/>

2-1. 젠킨스 스테이지 목록

1. 목록

API : (GET) /wf/pipeline/list

Desc : 젠킨스 스테이지 목록을 조회한다.

젠킨스 스테이지를 조합한 jenkins job(workflow) 구성을 위한 스테이지 목록 리스트업

젠킨스 스테이지 목록		
전체 (10)	검색어를 입력하세요.	+ 새 스테이지 등록
구분	이름	액션
SHELL_SCRIPT	날짜확인	<button>수정</button> <button>삭제</button>
GITLAB_CHECKOUT	gitlab-checkout	<button>수정</button> <button>삭제</button>
DEPLOY_CLUSTER	deploy-cluster	<button>수정</button> <button>삭제</button>
FLOW DATE CHECK	Date Check	<button>수정</button> <button>삭제</button>
SPIDER INFO CHECK	Spider Info Check	<button>수정</button> <button>삭제</button>
INFRASTRUCTURE CREATE	Infrastructure Create	<button>수정</button> <button>삭제</button>
INFRASTRUCTURE RUNNING STATUS	Infrastructure Running Status	<button>수정</button> <button>삭제</button>
TOMCAT INSTALL	Tomcat install vm servers	<button>수정</button> <button>삭제</button>
ASDF	test stage	<button>수정</button> <button>삭제</button>
TUMBLEBUG	tumblebug-Test	<button>수정</button> <button>삭제</button>

2-2. 젠킨스 스테이지 등록

1. 스테이지 구분 추가

API : (POST) /common/group/Pipeline/code

Param : (Request Body)

```
{
  "commonGroupCd":"Pipeline",
  "commonCd":"TUMBLEBUG",
  "codeName":"TUMBLEBUG",
  "codeDesc":"Add StageCd",
  "codeOrder":13
}
```

Desc : 스테이지 구분을 추가한다.
(여러 스테이지를 관리할 상위 depth 개념)

1. 중복체크

API : (GET) /wf/pipeline/name/duplicate

Param : (Query Param)

pipelineCd={스테이지구분}&pipelineName={스테이지명}

Desc : 스테이지 구분에 종속된 스테이지 명 중복 검사를 한다.

새 스테이지 등록

스테이지 구분 *

☐ SHELL_SCRIPT ☐ TEST ☐ GITLAB_CHECKOUT ☐ DEPLOY_CLUSTER ☐ DATE_CHECK
☐ FLOW_DATE_CHECK ☐ SPIDER_INFO_CHECK ☐ INFRASTRUCTURE_CREATE
☐ INFRASTRUCTURE_RUNNING_STATUS ☐ TOMCAT_INSTALL ☐ GESTT

추가할 스테이지 구분 입력



스테이지 명 *

스테이지 명을 입력해주세요.

중복체크

스테이지 내용 *

1 스테이지 내용을 입력해주세요.

취소

등록

<http://app.tb-cb.kro.kr/>

2-2. 젠킨스 스테이지 등록

3. 스테이지 등록

API : (POST) /wf/pipeline

Param : (Request Body)

```
{
  "pipelineCd":"TUMBLEBUG",
  "pipelineName":"tumblebug-Test",
  "pipelineScript": "stage('tumblebug') {
    \Wn      steps {
    \Wn      echo '>>>>STAGE:
TUMBLEBUG'\Wn \Wn      // 스크립트를 작성해주세요.\Wn      }
    \Wn      }
  "newStageCd":"tumblebug"
}
```

Desc : 스테이지를 등록한다.

새 스테이지 등록

스테이지 구분 *

☐ SHELL_SCRIPT ☐ TEST ☐ GITLAB_CHECKOUT ☐ DEPLOY_CLUSTER ☐ DATE_CHECK
☐ FLOW_DATE_CHECK ☐ SPIDER_INFO_CHECK ☐ INFRASTRUCTURE_CREATE
☐ INFRASTRUCTURE_RUNNING_STATUS ☐ TOMCAT_INSTALL ☐ GESTT

추가할 스테이지 구분 입력 +

스테이지 명 *

스테이지 명을 입력해주세요.

중복체크

스테이지 내용 *

1 스테이지 내용을 입력해주세요.

취소

등록

<http://app.tb-cb.kro.kr/>

3-1. 워크플로우 목록

1. 목록

API : (GET) /workflow/list

Desc : 워크플로우 목록을 조회한다.

스테이지를 조합하여 구성된 workflow 목록을 조회

워크플로우 목록			
전체 (5)		<input type="text" value="검색어를 입력하세요."/> <button>워크플로우 등록</button>	
워크플로우명	분류	젠킨스	Action
test-workflow	test	jenkins(10.10.30.22) (http://10.10.30.22:9800)	<button>상세</button> <button>삭제</button>
workflow-test-6	test	jenkins(10.10.30.22) (http://10.10.30.22:9800)	<button>상세</button> <button>삭제</button>
workflow-test-5	deploy	jenkins(10.10.30.22) (http://10.10.30.22:9800)	<button>상세</button> <button>삭제</button>
workflow-test2	test	jenkins(10.10.30.22) (http://10.10.30.22:9800)	<button>상세</button> <button>삭제</button>
workflow-test1	test	jenkins(10.10.30.22) (http://10.10.30.22:9800)	<button>상세</button> <button>삭제</button>

3-2. 워크플로우 생성

1. 중복체크
- API : (GET) /workflow/name/duplicate
- Param : (Query Param)
workflowName=workflow-test
- Desc : 등록될 워크플로우 명 중복 검사를 한다.

1. 스크립트 생성
- API : (POST) /workflow/jenkins/pipeline/default
- Param : (Request Body)
{
 "pipelineParam":[{"paramKey":"","paramValue":""}],
 "workflowName":"workflow-test",
 "workflowPurpose":"test",
 "jenkinsId":2
}
- Desc : 기본 스크립트를 생성한다.

새 워크플로우 등록

워크플로우 명 *

test-workflow

중복 체크

목적 *

테스트용

젠킨스 주소 *

jenkins(10.10.30.22)(http://10.10.30.22:9800)

파이프라인 *

스크립트 생성

1 //It was created by the Devops portal.
2 pipeline {
3 agent any
4
5 environment {
6 MC_SPIDER_REST_URI = http://tb.tb-cb.kro.kr:1323
7 CB_TUMBLEBUG_SWAGGER_URI = http://tb.tb-cb.kro.kr:1323/tumblebug/api/index.html
8 MCIS_NAME = "\${MCIS_NAME}"
9 }
10
11 stages {
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
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

key

value

파라미터 Key 입력

파라미터 Value 입력

추가

취소

확인

http://app.tb-cb.kro.kr/

3-2. 워크플로우 생성

3. 확인 (생성)

API : (POST) /workflow

Param : (Request Body)

```
{
  "workflowName": "workflow-test",
  "workflowPurpose": "test",
  "jenkinsId": 2,
  "pipelineParam": [
    {
      "paramKey": "MCIS_NAME",
      "paramValue": "vm_name"
    }
  ],
  "pipelineScript": "/*It was created by the Devops portal.\n\npipeline {\n  agent any\n  \n  environment {\n    MC_SPIDER_REST_URI = http://tb.tb-cb.kro.kr:1323\n    CB_TUMBLEBUG_SWAGGER_URI = http://tb.tb-cb.kro.kr:1323/tumblebug/api/index.html\n  }\n  \n  stages {\n    stage('tumblebug') {\n      steps {\n        echo '>>>>STAGE: TUMBLEBUG'\n        // 스크립트를 작성해주세요\n        echo 'MCIS NAME >> ' W'${MCIS_NAME}W'\n      }\n    }\n  }\n}\n\n",
  "pipelines": [
    {
      "pipelineScript": "/*It was created by the Devops portal.\n\npipeline {\n  agent any\n  \n  environment {\n    MC_SPIDER_REST_URI = http://tb.tb-cb.kro.kr:1323\n    CB_TUMBLEBUG_SWAGGER_URI = http://tb.tb-cb.kro.kr:1323/tumblebug/api/index.html\n  }\n  \n  stages {\n    stage('tumblebug') {\n      steps {\n        echo '>>>>STAGE: TUMBLEBUG'\n        // 스크립트를 작성해주세요\n        echo 'MCIS NAME >> ' W'${MCIS_NAME}W'\n      }\n    }\n  }\n}\n\n",
      "pipelineOrder": 0,
      "isDefaultScript": true,
      "defaultScriptTag": "DEFAULT_START"
    },
    {
      "pipelineScript": " stage('tumblebug') {\n      steps {\n        echo '>>>>STAGE: TUMBLEBUG'\n        // 스크립트를 작성해주세요\n        echo 'MCIS NAME >> ' W'${MCIS_NAME}W'\n      }\n    }\n\n",
      "pipelineOrder": 1,
      "isDefaultScript": true,
      "defaultScriptTag": "DEFAULT_END"
    }
  ]
}
```

Desc : 워크플로우 생성과 함께 Jenkins Job이 생성된다

새 워크플로우 등록

워크플로우명 *

test-workflow

중복 체크

목적 *

테스트용

젠킨스 주소 *

jenkins(10.10.30.22)(http://10.10.30.22:9800)

파이프라인 *

스크립트 생성

```
1 //It was created by the Devops portal.
2 pipeline {
3   agent any
4
5   environment {
6     MC_SPIDER_REST_URI = http://tb.tb-cb.kro.kr:1323
7     CB_TUMBLEBUG_SWAGGER_URI = http://tb.tb-cb.kro.kr:1323/tumblebug/api/index.html
8     MCIS_NAME = "${MCIS_NAME}"
9   }
10
11   stages {
12
13   }
14 }
```

GITLAB_CHECKOUT

SHELL_SCRIPT

FLOW DATE CHECK

INFRASTRUCTURE CREATE

TEST

INFRASTRUCTURE RUNNING STATUS

SPIDER INFO CHECK

TOMCAT INSTALL

DATE CHECK

ASDF

GESTT

DEPLOY_CLUSTER

key

파라미터 Key 입력

value

파라미터 Value 입력

추가

취소

확인

http://app.tb-cb.kro.kr/

4. 워크플로우 실행

1. 실행

API : (POST) /workflow/run, (GET) /workflow/{workflowIdx}/run

Param : (Request Body)

```
{
  "workflowId":47,"
  pipelineParam": [
    {
      "paramKey":"MCIS_NAME",
      "paramValue":"test",
    },
  ],
}
```

Desc : 등록된 워크플로우를 실행한다.

작업 중 페이지(0613 기준)에서는 수정화면에 실행버튼 존재

등록된 workflow에 따라 실행 시 추가로 필요한 데이터가 존재
할 수 있음(param)

GET방식 실행 시 설정된 value 기본값으로 실행

Ex) tomcat 9.0 설치 workflow

- MCIS, NAMESPACE 등 필요

- 개발서버기준 vm 삭제

API : (POST) /workflow/run

Param : (Request Body)

```
{
  "workflowId":47,
  "pipelineParam": [
    {
      "paramKey":"MCIS", "paramValue":"{mcisname}",
    },
    {
      "paramKey":"NAMESPACE", "paramValue":"{namespace}",
    }
  ],
}
```

Ex2) 현재시간 반환 workflow, health check 등

- 단순실행(GET) 가능

새 워크플로우 등록

워크플로우 명

workflow-test

중복 체크

목적 *

테스트용

젠킨스 주소 *

jenkins(10.10.30.22)(http://10.10.30.22:9800)

파이프라인 *

스크립트 생성

```
1 //It was created by the Devops portal.
2 pipeline {
3   agent any
4
5   environment {
6     MC_SPIDER_REST_URI = http://tb.tb-cb.kro.kr:1323
7     CB_TUMBLEBUG_SWAGGER_URI = http://tb.tb-cb.kro.kr:1323/tumblebug/api/index.html
8   }
9   stages {
10
11
```

TUMBLEBUG

```
1 stage('tumblebug') {
2   steps {
3     echo '>>>>>STAGE: TUMBLEBUG'
4
5     // 스크립트 실행 후 세요.
6     echo 'MCIS NAME >> ' "${MCIS_NAME}"
7   }
8 }
9
10
```

```
1 }
2
3
```

key

MCIS_NAME

value

vm_name

추가

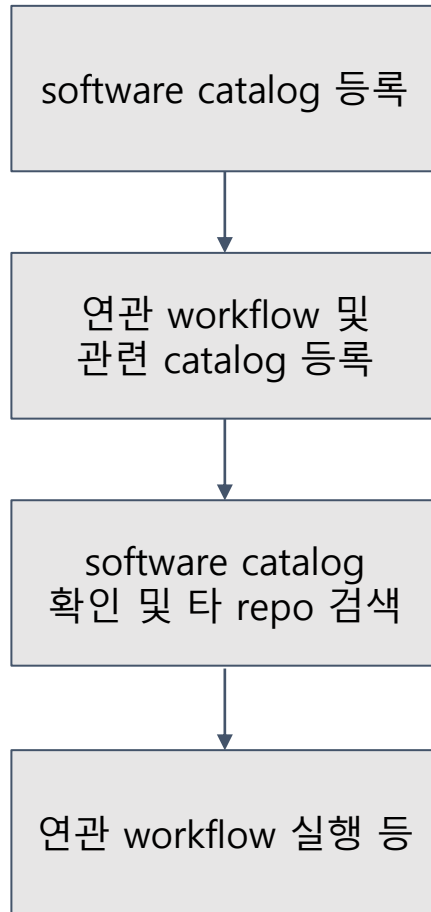
<http://app.tb-cb.kro.kr/#/projects/workflow/edit/45>

취소

수정

실행

software catalog 확인 및 연관 workflow 실행



software catalog 등록

software catalog 연계 workflow 및 catalog 등록(optional)
- 해당 software 설치 workflow 실행, 재시작 등 연관 workflow 등록
- 연관 catalog 등록

software catalog 확인 및 타 repo 검색

연관 workflow 실행, 또는 연관 catalog 정보 확인 등
- 해당 software 설치 workflow 실행 등 등록된 연관 workflow 사용
- 검색된 외부 sw를 catalog에 등록 등

5. Software catalog

1. Software catalog 등록

API : (post) /workflow/name/duplicate

```
{
  "scTitle": "test",
  "scVersion": "test",
  "sclcon": "test",
  "scSummary": "test",
  "scUri": "test",
  "scCategory": "test",
  "scReference": "test",
  "scDescription": "test"
}
```

Desc : m-cmp에서 관리할 catalog를 등록, 등록 후 idx 반환

1. Software catalog 연관 workflow/catalog 등록

API : (get) /catalog/software/relation/sw-catalog/{catalogIdx}/{workflowIdx}
(get) /catalog/software/relation/workflow/{catalogIdx} /refCatalogIdx}

Desc : 해당 catalog에 연관된 workflow와 catalog를 등록하여
기 생성된 workflow 및 catalog를 해당 catalog 내용에 노출








1. Software catalog 조회

API : (post) /catalog/software/{keyword}
Desc : catalog 목록을 조회한다.

2. Software catalog 연관 workflow 실행

- workflow 실행 참조

Software catalog

16	 openJdk 21 non-production builds of OpenJDK 17	JAVA ...
15	 openjdk 1.8 non-production builds of OpenJDK 1.8	JAVA ...
14	 openJdk 11 non-production builds of OpenJDK 11	JAVA ...
13	 mongoDB document DB	NoSQL ...
12	 redis In memory DB	NoSQL ...
11	 openJdk 17 non-production builds of OpenJDK 17	JAVA ...
10	 Apache Tomcat The Apache Tomcat® software is an open source implementation https://tomcat.apache.org/ The Apache Tomcat® software is an open source implementation of the Jakarta Servlet, Jakarta Pages, Jakarta Expression Language, Jakarta WebSocket, Jakarta Annotations and Jakarta Authentication specifications. These specifications are part of the Jakarta EE platform. The Jakarta EE platform is the evolution of the Java EE platform. Tomcat 10 and later implement specifications developed as part of Jakarta EE. Tomcat 9 and earlier implement specifications developed as part of Java EE. The Apache Tomcat software is developed in an open and participatory environment and released under the Apache License version 2. The Apache Tomcat project is intended to be a collaboration of the best-of-breed developers from around the world. We invite you to participate in this open development project. To learn more about getting involved, click here. 연동 workflow <ul style="list-style-type: none">tomcat 9.0.2 install ↗tomcat restart ↗	WAS ...

Search...

dockerHub search

검색된 관련 ContainerImage가 없습니다.

artifactHub search

검색된 관련 HelmChart가 없습니다.

5. Software catalog

1. artifactHub package 조회

API : (get) /applications/search/artifacthub/package/{keyword}

Desc : artifactHub의 packag를 검색한다.

1. dockerHub catalog 조회

API : (get) /applications/search/dockerhub/catalog/{keyword}

Desc : dockerHub의 catalog(containerImage)를 검색한다.

1. Nexus repository 조회(temp)

API : (get) /applications/search/nexus/{repository}/{keyword}


Desc : 틀체인관리에서 연동한 nexus repository에 등록된 내용을 검색한다.

2. 검색된 외부 저장소결과 중 선택하여 M-cmp catalog에 등록

API : -


Desc : -


Software catalog


10  Apache Tomcat
The Apache Tomcat® software is an open source implementation

WAS ...


dockerHub search


 tomcat
Apache Tomcat is an open
sourc...


 tomee
Apache TomEE is an all-Apache ...

 bitnami/tomcat
Bitnami container image for To...

artifactHub search

 tomcat
Apache Tomcat is an open-
sourc...

 tomcat
Apache Tomcat is an open-
sourc...

 tomcat
Apache Tomcat is an open-
sourc...

감사합니다.
