UX-DX Design Document (MANISH)

# GET DASHBOARD DETAILS [DCG2-1794]

## REQUEST URL

GET byoc.url​/api​/v1​/pod?pageNumber=1&pageSize=10

* The pageNumber and pageSize query parameter is used to perform pagination.

## API RESPONSE (Click on the embedded file to view the schema.)

## 

The API response was changed to accommodate.

* Project > Targets > Pods(Deployments) Hierarchy. Each User has multiple projects, each project can have multiple targets and each target can have multiple deployments.
* The status of the target is a max priority status of the underlying pods.
* Pagination support.

## COMPARISON ID LOGIC MODIFIED

* The previous logic was changed to make sure all the pods in a single multicontainer deployment share the same comparison Id.

* 

## PAGINATION SUPPORT

Pagination was introduced in this API using the following logic.

Here the pagination is being performed on the BYOC Side, and not in the database end, since the Pagination is to be performed on a list which is formed by a combination of tables. It was much simpler to perform pagination in backend that figuring out how many deployements do we need to fetch from the DB to get required number of projects.



# GET PROJECTS [DCG2-2080]

## REQUEST URL

GET byoc.url​/api/v1/projects/details?pageNumber=1&pageSize=10

GET byoc.url​/api/v1/projects/?pageNumber=1&pageSize=10

* The pageNumber and pageSize query parameter is used to perform pagination.

## API RESPONSE

**For GET Project** (Click on the embedded file to view the schema.)



**For GET Project Details** (Click on the embedded file to view the schema.)



## PAGINATION SUPPORT

Here since we are only fetching the data from a single table, pagination is much simpler if done from the database end. We are using Spring Boot and Spring Data JPA Support for Pagination using pagable items.



# SEARCH PROJECTS [DCG2-2080]

## REQUEST URL

GET byoc.url/api/v1/projects/search?searchKeyWord=abv?pageNumber=1&pageSize=10

## API RESPONSE

Same as GET Project API.

## PAGINATION SUPPORT

Same as GET Project API.

# SEARCH BYOC Containers API (NOT BEING USED : MOVED TO BYOS)

# GET BYOC Containers API (NOT BEING USED : MOVED TO BYOS)

# STOP AND REMOVE API [DCG2-1914]

## REQUEST URL

DELETE executionms.url/api/v1/target/{comparison\_id}?is\_stopped=true

* If is\_stopped = true (STOP the target)
* If is\_stopped = false (REMOVE the target)

## STOP VS REMOVE

* STOP Terminates the target and also removes the target from the dashboard. (Relaunch is not in scope as of now).
* REMOVE Both Terminates and removes the target from dashboard.

## OS-INT-MS BEHAVIOR

* On Both Remove and Stop we terminated the pod in the openshift cluster. There was no support of pause and restart both from fabric8 APIs and openshift/Kubernetes API, So we just terminated the pod in both case.

## DATABASE BEHAVIOR

* A new field (is\_stopped) was introduced in the gen2\_deployments table, to distinguish between manually stopped vs removed/terminated pods.
* On Stopping the target, we loop over all it’s deployments and mark this field as true. Since the is\_terminated flag is still false, the pods appears in the dashboard.
* On Removing the target, we loop over all it’s deployments and then soft delete the target i.e set the is\_terminated=true in db for that deployment.

## REMOVE ALL DEPLOYMENTS IN PROJECT

DELETE executionms.url/api/v1/project/{project\_id}

* This API will remove all the deployments in the project from the dashboard and will terminate the openshift pods as well.

## ADDITIONAL NOTES - Helm STOP/REMOVE is not covered in this.

# ASSIGN AND UNASSIGN API [DCG2-2349, DCG2-2350]

## APIs (Changes in the highlighted section from my end).

* Create Container (BYOC)
  + Creates the container if doesn’t exist.
  + Also calls the configms assign API.
* Delete Project (BYOC)
  + Calls configms unassign API for all the containers.
* Assign Project To Container (Config MS)
  + Creates the project <-> container relationship.
  + Calls the BYOS service to make project <-> resource\_name(has tag\_name) <-> container\_id.
* Unassign Project From Container (Config MS)
  + Removes the project <-> container relationship.
  + Calls the BYOS Removes the resource\_name <-> project\_id relationship.

## BEHAVIOR OF UI.

* In case of assigning of resources to project, UI has to call createContainer, if container\_id for the resource (IN GET RESOURCE API) is null.
* UI has to call assignAPI of configMs if container\_id is not null.
* UI also calls the createContainer API while importing resource from

## ADDITIONAL NOTES - Helm Assign/Unassign is not covered in this.