



BrAPI V2.1 & 2022 Development Plan

BrAPI Hackathon April 2022

BrAPI V2.1

- 46 issues closed (majority are small changes)
- Waiting for the results from the working groups
 - GET /variantmatrix
 - PUT /calls
 - GET /events

BrAPI V2.1

Issue #460:

'referenceID' was changed to 'referenceId'. This follows the best practice of using camel case for all data field names. Even though this change is a small one, it has been applied across most of the major endpoints.

Issue #480:

'programDbId' is added as a new search parameter for 34 different endpoints. This change is to support multi-tenant systems that have multiple programs data stored in the same database. In those cases, programDbId is essential for almost every query for data to make sure there is no cross contamination.

Issue #462:

Plate is now a high level entity with its own set of endpoints. In V2.0, Sample had a 'plateDbId' field, but there was no corresponding object in the spec. Plate is now available with meta data about the group of samples it holds.

BrAPI V2.1

Issues #392 and #396:

Images and Observations can now be deleted. The endpoints 'POST /delete/images' and 'POST /delete/observations' were added to accomplish this. The HTTP method POST was used because it will allow for bulk deletes, instead of DELETE which will only allow one item deleted at a time. These are the first two endpoints in BrAPI that will allow deleting, but this may become standard for all high level entities in the spec. Note for server implementation: A request to delete does not mean data should be removed from the database. It is recommended to mark records as "inactive" so the can be recovered and the delete can be undone.

BrAPI V2.1

Issues #239 and #481 :

Pedigree is now a high level entity with its own set of endpoints. This replaces the endpoints '/germplasm/germplasmDbId/pedigree' and '/germplasm/germplasmDbId/progeny'. The 'GET /pedigree' endpoint will allow for more flexibility for retrieving nodes of a pedigree tree, allowing for multiple levels of parents and children to be returned in a single request. 'POST /pedigree' allows users to transfer detailed pedigree trees between systems without relying on pedigree string parsing.

Pedigree Updates

Search Parameters

germplasmDbId	synonyms
germplasmName	instituteCode
germplasmPUI	programDbId
accessionNumber	programName
commonCropName	trialDbId
binomialName	trialName
genus	studyDbId
species	studyName
collection	externalReference
familyCode	

Option Flags

- includeParents
- includeProgeny
- includeSiblings
- includeFullTree
- pedigreeDepth
- progenyDepth

Pedigree Updates

```
"data": [ {  
  "breedingMethodDbId": "ffcce7ef",  
  "breedingMethodName": "Male Backcross",  
  "crossingProjectDbId": "625e745a",  
  "crossingYear": 2005,  
  "defaultDisplayName": "A0000003",  
  "familyCode": "F0000203",  
  "germplasmDbId": "1098ebaf",  
  "germplasmName": "A0021004",  
  "germplasmPUI": "http://pui.per/accession/A0000003",  
  "pedigreeString": " A0000592/A0000002",  
  "parents": [ { "germplasmDbId": "b66958de", "germplasmName": "A0000592", "parentType": "MALE" }, ... ],  
  "progeny": [ { "germplasmDbId": "e8d5dad7", "germplasmName": "A0021011", "parentType": "FEMALE" }, ... ],  
  "siblings": [ { "germplasmDbId": "334f53a3", "germplasmName": "A0021005"}, ... ]  
}, ... ]
```

2022 Development Plans

Standard Maintenance

- Upgrade the BrAPI Test Server with V2.1
- Upgrade BRAVA with V2.1
- Updates to brapi.org
 - Get Started
 - Compatible Software
 - Governance

Goals from 2021

- Extract the BrAPI data model and define outside REST
- Build an Interactive UML diagram
- BrAPI Models in GraphQL

2022 Development Plans

Documentation

- Governance
- Tutorial examples
- Big data in BrAPI best practices
- BrAPI extension guidelines
- BrAPI Wiki improvements
- BrAPI Manuscript V2