

Hands on task: Custom dataset types

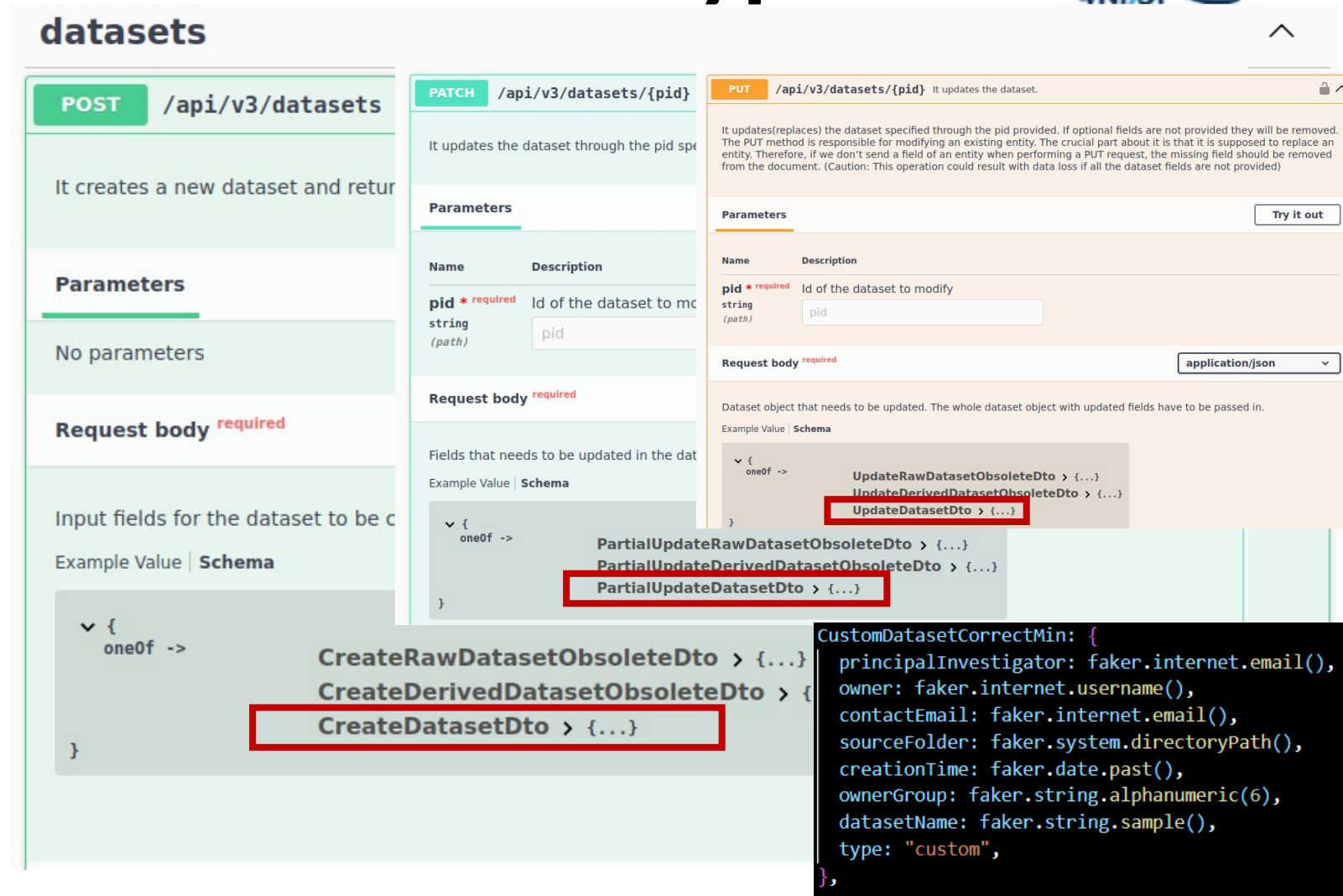
- Dataset instances have a type; up to now only Raw or Derived
- New capability: Definition of custom types
 - Definitions by .env variable or default in „datasetTypes.json“
 - Raw and Derived are always available
- Use case example: Dataset collections

```
{ } datasetTypes.example.json > ...  
1 {  
2   "Custom": "custom"  
3 }
```

```
MONGODB_COLLECTION="Dataset"  
ES_MAX_RESULT=100000  
ES_FIELDS_LIMIT=400000  
ES_INDEX="dataset"  
ES_PORT=9200  
ES_HOST="https://localhost:9200"  
ES_USERNAME="elastic"  
ES_PASSWORD="duo-password"  
ES_REFRESH=<"wait_for"|"false">  
  
LOGGERS_CONFIG_FILE="loggers.json"  
DATASET_TYPES_FILE="datasetTypes.json"  
PROPOSAL_TYPES_FILE="proposalTypes.json"
```

Hands on task: Custom dataset types

- Interface: Custom types require adherence to new unified API schema
- Old formats remain available for now
- Planned API v4 will sunset obsolete formats



The screenshot displays the API documentation for the 'datasets' endpoint. It includes three main sections: POST, PATCH, and PUT.

POST /api/v3/datasets: It creates a new dataset and returns it. Parameters: None. Request body: required. Example Value | Schema: `oneOf -> CreateRawDatasetObsoleteDto > {...}`, `CreateDerivedDatasetObsoleteDto > {...}`, `CreateDatasetDto > {...}`.

PATCH /api/v3/datasets/{pid}: It updates the dataset through the pid specified. Parameters: Name (Description), pid (required, string, path) (Id of the dataset to modify). Request body: required. Fields that need to be updated in the dataset. Example Value | Schema: `oneOf -> PartialUpdateRawDatasetObsoleteDto > {...}`, `PartialUpdateDerivedDatasetObsoleteDto > {...}`, `PartialUpdateDatasetDto > {...}`.

PUT /api/v3/datasets/{pid}: It updates the dataset. It updates(replaces) the dataset specified through the pid provided. If optional fields are not provided they will be removed. The PUT method is responsible for modifying an existing entity. The crucial part about it is that it is supposed to replace an entity. Therefore, if we don't send a field of an entity when performing a PUT request, the missing field should be removed from the document. (Caution: This operation could result with data loss if all the dataset fields are not provided). Parameters: Name (Description), pid (required, string, path) (Id of the dataset to modify). Request body: required. Dataset object that needs to be updated. The whole dataset object with updated fields have to be passed in. Example Value | Schema: `oneOf -> UpdateRawDatasetObsoleteDto > {...}`, `UpdateDerivedDatasetObsoleteDto > {...}`, `UpdateDatasetDto > {...}`.

CustomDatasetCorrectMin:

```
CustomDatasetCorrectMin: {
  principalInvestigator: faker.internet.email(),
  owner: faker.internet.username(),
  contactEmail: faker.internet.email(),
  sourceFolder: faker.system.directoryPath(),
  creationTime: faker.date.past(),
  ownerGroup: faker.string.alphanumeric(6),
  datasetName: faker.string.sample(),
  type: "custom",
},
```