# Project set up:

Information that would apply to all samples collected by the project. Can be set up in the office. Might also want to define some place name and keyword vocabularies to promote consistency…. Selecting a subset of the iSamples Material type, Specimen type and SampledFeature type vocabularies (and any extensions being used) that apply to the project. Project needs to decide if these will be assigned by team in the field at the collection event or individual sample level.

* iSamples Context Category for all samples (if applicable). . ("has\_context\_category") For open context work, probably   
  { "label": "Site of past human activities",  
   "identifier": "https://w3id.org/isample/vocabulary/sampledfeature/0.9/pasthumanoccupationsite" }  
  There might be extension vocabularies with more granular terms useful to the project; these might be assigned at the sampling event or individual sample level.
* Generic keywords for all samples related to project ("keywords": [ { "keyword": }, …]) Assume that later processing will add “scheme\_name”, “scheme\_uri”, and/or “keyword\_uri” as appropriate.
* Definition of sampling protocols or policies used (if applicable) (“complies\_with”)
* Feature of interest, if applies to all samples ("produced\_by": { "has\_feature\_of\_interest": })
* Project name, contact information, identifier ("produced\_by": { "responsibility": {"role" : "Project" , …..}} )
* Generic label for the project sample collection area that applies to all samples ("produced\_by": { "samplingSite": { "label": … }})
* Generic description of the project sample collection area that applies to all samples ("produced\_by": { "samplingSite": { "description": … }}).
* Geospatial coordinates for the project area; lat, long for the centroid of the project sample collection area ("location": { "elevation": , "latitude": , "longitude": } )
* Place names, generic place names for the project sample collection area ("produced\_by": { "samplingSite": { "place\_name": … }})
* Identification of any permits under which project work is being done ("produced\_by": { "authorized\_by": [ "” ] })

Persons in collector (or other) role ("produced\_by": { "responsibility": {"role" : "collector" , …Agent..}} ) can be prepopulated Agent objects:

Agent:

“name”:

“affiliation”:

“contact\_information”:

“identifier”: (ORCID preferred)

Other roles might be defined for the collection event (“produced\_by”: {“responsibility”: {“role”:.. Agent ..}} ). See draft role vocabulary https://github.com/isamplesorg/metadata/blob/main/notes/vocabulary/RoleVocabulary.xlsx

# Collection Event

Based on idea that sets of samples will be collected under circumstances that can be bundled, e.g. time interval, investigator, procedure, location. Base properties for these would be recorded in the field at start of event. Project needs to define granularity of events. These are all in the "produced\_by": section of the iSamples metadata.

* REQUIRED: Label for team to identify the event (“label”: )
* REQUIRED: Description of sampling event (“description”: )
* REQUIRED: Collector- provide name, app picks up rest of Agent information
* REQUIRED: Time stamp—project decides—start time, end time, interval with start and end, just the date (“result\_time”: ) (automate?)
* Sampling site, content in the “produced\_by”: {“sampling\_site”: ) object
  + REQUIRED: Label to identify the site for people (“label”: )

Optional

* Context category (sampled feature), material type, and specimen type assignments from controlled vocabulary.
* update what specifically is being sampled ("has\_feature\_of\_interest": ). Could replace or append
* other roles… other “responsibility” Agents if applicable, e.g. assistants
* Append keywords for all samples related to the event ("keywords": [ { "keyword": }, …]) Assume that later processing will add “scheme\_name”, “scheme\_uri”, and/or “keyword\_uri” as appropriate.
* sampling site, optional content in the “produced\_by”: {“sampling\_site”: ) object
  + detailed information on precisely where the sampling event is taking place ("produced\_by": { "sampling\_site": { "description": … }}), append to generic description if there is one already
  + specific lat, long for the event site, e.g. from GPS, ("location": { "elevation": , "latitude": , "longitude": } )
  + specific place name(s) that would be appended in the place name array, e.g. trench number { "samplingSite": { "place\_name": [] }}

# As samples are collected,

* REQUIRED: identifier—assign appropriate token for each sample that will be the unique part of its URI. Tools might do this with some automatic algorithm. This might be a string formatted according to project conventions, with punctuation/syntax that will be modified in the final URI. This original ‘people-friendly’ string would then be recorded as “alternate\_identifier” with “scheme\_name” something like ‘local project’. (“sample\_identifier”)
* REQUIRED: Label for sample; this should be something informative to identify sample in the context of the site (“label”)
* REQUIRED: Description of the sample, describing what, why, other details specific to the sample (“description”)
* REQUIRED: Check that Context category (sampled feature), material type, and specimen type assignments from controlled vocabulary are appropriate, update if necessary ("has\_context\_category": , "has\_material\_category": , "has\_specimen\_category": )

Optional

* keywords to append to the project- and event-level keywords ("keywords": [ { "keyword": }, …] ). Assume that later processing will add “scheme\_name”, “scheme\_uri”, and/or “keyword\_uri” as appropriate.
* specific location information, e.g. in coordinates specific to the site or a particular trench, these get appended to “samping\_site”: {“description”:}. If each sample has a lat long location, record in “sampling\_site”:{“location”: }
* specific place names to append to “sampling\_site”: {“place\_name”:[] } array.
* sampling site label and description update. (“sampling\_site”:{“label”: , ”description”: }
* sample-specific features of interest update ("has\_feature\_of\_interest": ). Could replace or append