

## REST APIs

### Model Inference REST API

#### Create/Post

- Create a new model for inference
  - Requires a key to do
  - In post, include model, model name (must be unique to current models), and additional metadata (cpu/gpu, optimized/or not, version, and anything else)
  - Returns: Json format, Successful, Not successful (and why? invalid model type etc., invalid key, not unique name)
- 

#### Read/Get

- Inference
  - Include model name (defaults to a specific one if not given), type of output (full, probs, object name, classification for bin, bounding box etc.), and image
  - Returns: json formatted output containing what was requested
    - Full result example: {'probabilities': [.9, .05, 0, .05], 'error\_msg': None, 'bounding\_box': [(1,5), (100, 200)], 'classification': 'water bottle', 'bin\_classification': 'recycle'}
- Batch inference
  - Include model name (defaults to a specific one if not given), type of output, and list of images
  - Returns: json formatted output containing what was requested
    - Full result example: [{'probabilities': [.9, .05, 0, .05], 'error\_msg': None, 'bounding\_box': [(1,5), (100, 200)], 'classification': 'water bottle', 'bin\_classification': 'recycle'}, ..., other image results]
- Get list of models that are usable
  - returns: json format, list of model names and metadata for them
- Download model
  - Include model name
  - Returns model and metadata of model, or failure to give (does not exist etc.)

#### Update/Put

- Update a model with a new model
  - Requires a key to do
  - In put, include model, model name to replace, and additional metadata (cpu/gpu, optimized/or not, version, and anything else)
  - Returns: Json format, Successful, Not successful (and why? invalid model type etc., invalid key, not unique name)
- Update a model metadata
  - Requires a key to do
  - In put, include, model name, and metadata to replace (cpu/gpu, optimized/or not, version, and anything else)

- Returns: Json format, Successful, Not successful (and why? invalid model type etc., invalid key, name does not exist)
- Update a model metadata tag

#### Delete/Delete

- Deletes a model
  - Requires a key to do
  - In put, model name
  - Returns: Json format, Successful, Not successful (and why? invalid model name, invalid key)
- Delete metadata tag
  - Requires a key to do
  - In put, model name and tag name for metadata
  - Returns: Json format, Successful, Not successful (and why? invalid model name, invalid key, metadata tag does not exist)

### Training (annotate) Dataset REST API

ALL CALLS should require a developer key of sorts

#### Create/Post

- Create a new entry
  - Include, image and possibly annotations and other metadata of interests
- Create a bunch of new entries
  - Include, list of images, annotations if any, and metadata

#### Read/Get

- Get number of entries
  - Can be filtered in request
- Get an entry
  - Include a unique identifier (index etc.)
- Get all entries given search
  - Include a searchable value to get entries
- Get entry with no or least annotation confirmations
- Get entry with most annotations
  - Can specify class

#### Update/Put

- Fully replace annotations given image unique identifier
- Mix annotations given image unique identifier
- Edit metadata of image given unique identifier
- Edit metadata tag

#### Delete/Delete

- Remove metadata
- Remove metadata tag

- Remove annotations
- Remove image and all info on it
-