Vertex AI API

Train high-quality custom machine learning models with minimal machine learning expertise and effort.

Service: aiplatform.googleapis.com

To call this service, we recommend that you use the Google-provided client libraries (https://cloud.google.com/apis/docs/client-libraries-explained). If your application needs to use your own libraries to call this service, use the following information when you make the API requests.

Discovery document

A Discovery Document (https://developers.google.com/discovery/v1/reference/apis) is a machine-readable specification for describing and consuming REST APIs. It is used to build client libraries, IDE plugins, and other tools that interact with Google APIs. One service may provide multiple discovery documents. This service provides the following discovery documents:

- https://aiplatform.googleapis.com/$discovery/rest?version=v1 (https://aiplatform.googleapis.com/$discovery/rest?version=v1)
- https://aiplatform.googleapis.com/$discovery/rest?version=v1beta1 (https://aiplatform.googleapis.com/$discovery/rest?version=v1beta1)

Service endpoint

A service endpoint (https://cloud.google.com/apis/design/glossary#api_service_endpoint) is a base URL that specifies the network address of an API service. One service might have multiple service endpoints. This service has the following service endpoints and all URIs below are relative to these service endpoints:

- `https://aiplatform.googleapis.com`
- `https://africa-south1-aiplatform.googleapis.com`
- `https://asia-east1-aiplatform.googleapis.com`

- `https://asia-east2-aiplatform.googleapis.com`
- `https://asia-northeast1-aiplatform.googleapis.com`
- `https://asia-northeast2-aiplatform.googleapis.com`
- `https://asia-northeast3-aiplatform.googleapis.com`
- `https://asia-south1-aiplatform.googleapis.com`
- `https://asia-southeast1-aiplatform.googleapis.com`
- `https://asia-southeast2-aiplatform.googleapis.com`
- `https://australia-southeast1-aiplatform.googleapis.com`
- `https://australia-southeast2-aiplatform.googleapis.com`
- `https://europe-central2-aiplatform.googleapis.com`
- `https://europe-north1-aiplatform.googleapis.com`
- `https://europe-southwest1-aiplatform.googleapis.com`
- `https://europe-west1-aiplatform.googleapis.com`
- `https://europe-west2-aiplatform.googleapis.com`
- `https://europe-west3-aiplatform.googleapis.com`
- `https://europe-west4-aiplatform.googleapis.com`
- `https://europe-west6-aiplatform.googleapis.com`
- `https://europe-west8-aiplatform.googleapis.com`
- `https://europe-west9-aiplatform.googleapis.com`
- `https://europe-west12-aiplatform.googleapis.com`
- `https://me-central1-aiplatform.googleapis.com`
- `https://me-central2-aiplatform.googleapis.com`
- `https://me-west1-aiplatform.googleapis.com`
- `https://northamerica-northeast1-aiplatform.googleapis.com`
- `https://northamerica-northeast2-aiplatform.googleapis.com`

- `https://southamerica-east1-aiplatform.googleapis.com`
- `https://southamerica-west1-aiplatform.googleapis.com`
- `https://us-central1-aiplatform.googleapis.com`
- `https://us-east1-aiplatform.googleapis.com`
- `https://us-east4-aiplatform.googleapis.com`
- `https://us-south1-aiplatform.googleapis.com`
- `https://us-west1-aiplatform.googleapis.com`
- `https://us-west2-aiplatform.googleapis.com`
- `https://us-west3-aiplatform.googleapis.com`
- `https://us-west4-aiplatform.googleapis.com`
- `https://us-east5-aiplatform.googleapis.com`

See Feature availability (/vertex-ai/docs/general/locations#feature-availability) for the supported features for each region.

REST Resource: v1.media (/vertex-ai/docs/reference/rest/v1/media)

Methods	

upload (/vertex-ai/docs/reference/rest/v1/media/upload)	`POST /v1/{parent}/ragFiles:upload`
	`POST /upload/v1/{parent}/ragFiles:upload`
	Upload a file into a RagCorpus.

REST Resource: v1.projects.locations

(/vertex-ai/docs/reference/rest/v1/projects.locations)

Methods	
`augmentPrompt` (/vertex-ai/docs/reference/rest/v1/projects.locations/augmentPrompt)	`POST /v1/{parent}:augmentPrompt` Given an input prompt, it returns augmented prompt from vertex rag store to guide LLM towards generating grounded responses.
`corroborateContent` (/vertex-ai/docs/reference/rest/v1/projects.locations/corroborateContent)	`POST /v1/{parent}:corroborateContent` Given an input text, it returns a score that evaluates the factuality of the text.
`evaluateInstances` (/vertex-ai/docs/reference/rest/v1/projects.locations/evaluateInstances)	`POST /v1/{location}:evaluateInstances` Evaluates instances based on a given metric.
`retrieveContexts` (/vertex-ai/docs/reference/rest/v1/projects.locations/retrieveContexts)	`POST /v1/{parent}:retrieveContexts` Retrieves relevant contexts for a query.

REST Resource: v1.projects.locations.batchPredictionJobs

(/vertex-ai/docs/reference/rest/v1/projects.locations.batchPredictionJobs)

Methods	
`cancel` (/vertex-ai/docs/reference/rest/v1/projects.locations.batchPredictionJobs/cancel)	`POST /v1/{name}:cancel` Cancels a BatchPredictionJob.

Methods	
create (/vertex-ai/docs/reference/rest/v1/projects.locations.batchPredictionJobs/create)	`POST /v1/{parent}/batchPredictionJobs` Creates a BatchPredictionJob.
delete (/vertex-ai/docs/reference/rest/v1/projects.locations.batchPredictionJobs/delete)	`DELETE /v1/{name}` Deletes a BatchPredictionJob.
get (/vertex-ai/docs/reference/rest/v1/projects.locations.batchPredictionJobs/get)	`GET /v1/{name}` Gets a BatchPredictionJob
list (/vertex-ai/docs/reference/rest/v1/projects.locations.batchPredictionJobs/list)	`GET /v1/{parent}/batchPredictionJobs` Lists BatchPredictionJobs in a Location.

REST Resource: v1.projects.locations.cachedContents

(/vertex-ai/docs/reference/rest/v1/projects.locations.cachedContents)

Methods	
create (/vertex-ai/docs/reference/rest/v1/projects.locations.cachedContents/create)	`POST /v1/{parent}/cachedContents` Creates cached content, this call will initialize the cached content in the data storage, and users need to pay for the cache data storage.

Methods	
delete (/vertex-ai/docs/reference/rest/v1/projects.locations.cachedContents/delete)	`DELETE /v1/{name}` Deletes cached content
get (/vertex-ai/docs/reference/rest/v1/projects.locations.cachedContents/get)	`GET /v1/{name}` Gets cached content configurations
list (/vertex-ai/docs/reference/rest/v1/projects.locations.cachedContents/list)	`GET /v1/{parent}/cachedContents` Lists cached contents in a project
patch (/vertex-ai/docs/reference/rest/v1/projects.locations.cachedContents/patch)	`PATCH /v1/{cachedContent.name}` Updates cached content configurations

REST Resource: v1.projects.locations.customJobs

(/vertex-ai/docs/reference/rest/v1/projects.locations.customJobs)

Methods	
cancel (/vertex-ai/docs/reference/rest/v1/projects.locations.customJobs/cancel)	`POST /v1/{name}:cancel` Cancels a CustomJob.

Methods

`create` (/vertex-ai/docs/reference/rest/v1/projects.locations.customJobs/create)	`POST /v1/{parent}/customJobs` Creates a CustomJob.
`delete` (/vertex-ai/docs/reference/rest/v1/projects.locations.customJobs/delete)	`DELETE /v1/{name}` Deletes a CustomJob.
`get` (/vertex-ai/docs/reference/rest/v1/projects.locations.customJobs/get)	`GET /v1/{name}` Gets a CustomJob.
`list` (/vertex-ai/docs/reference/rest/v1/projects.locations.customJobs/list)	`GET /v1/{parent}/customJobs` Lists CustomJobs in a Location.

REST Resource: v1.projects.locations.datasets

(/vertex-ai/docs/reference/rest/v1/projects.locations.datasets)

Methods

`create` (/vertex-ai/docs/reference/rest/v1/projects.locations.datasets/create)	`POST /v1/{parent}/datasets` Creates a Dataset.

Methods

`delete` (/vertex-ai/docs/reference/rest/v1/projects.locations.datasets/delete)	`DELETE /v1/{name}` Deletes a Dataset.
`export` (/vertex-ai/docs/reference/rest/v1/projects.locations.datasets/export)	`POST /v1/{name}:export` Exports data from a Dataset.
`get` (/vertex-ai/docs/reference/rest/v1/projects.locations.datasets/get)	`GET /v1/{name}` Gets a Dataset.
`import` (/vertex-ai/docs/reference/rest/v1/projects.locations.datasets/import)	`POST /v1/{name}:import` Imports data into a Dataset.
`list` (/vertex-ai/docs/reference/rest/v1/projects.locations.datasets/list)	`GET /v1/{parent}/datasets` Lists Datasets in a Location.
`patch` (/vertex-ai/docs/reference/rest/v1/projects.locations.datasets/patch)	`PATCH /v1/{dataset.name}` Updates a Dataset.
`searchDataItems` (/vertex-ai/docs/reference/rest/v1/projects.locations.datasets/searchDataItems)	`GET /v1/{dataset}:searchDataItems` Searches DataItems in a Dataset.

REST Resource: v1.projects.locations.datasets.annotationSpecs (/vertex-ai/docs/reference/rest/v1/projects.locations.datasets.annotationSpecs)

Methods	
`get` (/vertex-ai/docs/reference/rest/v1/projects.locations.datasets.annotationSpecs/get)	**`GET /v1/{name}`** Gets an AnnotationSpec.

REST Resource: v1.projects.locations.datasets.dataItems (/vertex-ai/docs/reference/rest/v1/projects.locations.datasets.dataItems)

Methods	
`list` (/vertex-ai/docs/reference/rest/v1/projects.locations.datasets.dataItems/list)	**`GET /v1/{parent}/dataItems`** Lists DataItems in a Dataset.

REST Resource: v1.projects.locations.datasets.dataItems.annotations (/vertex-ai/docs/reference/rest/v1/projects.locations.datasets.dataItems.annotations)

Methods	
`list` (/vertex-ai/docs/reference/rest/v1/projects.locations.datasets.dataItems.annotations/list)	**`GET /v1/{parent}/annotations`** Lists Annotations belongs to a dataitem.

REST Resource: v1.projects.locations.datasets.datasetVersions

(/vertex-ai/docs/reference/rest/v1/projects.locations.datasets.datasetVersions)

Methods

`create` (/vertex-ai/docs/reference/rest/v1/projects.locations.datasets.datasetVersions/create)	**`POST /v1/{parent}/datasetVersions`** Create a version from a Dataset.
`delete` (/vertex-ai/docs/reference/rest/v1/projects.locations.datasets.datasetVersions/delete)	**`DELETE /v1/{name}`** Deletes a Dataset version.
`get` (/vertex-ai/docs/reference/rest/v1/projects.locations.datasets.datasetVersions/get)	**`GET /v1/{name}`** Gets a Dataset version.
`list` (/vertex-ai/docs/reference/rest/v1/projects.locations.datasets.datasetVersions/list)	**`GET /v1/{parent}/datasetVersions`** Lists DatasetVersions in a Dataset.
`patch` (/vertex-ai/docs/reference/rest/v1/projects.locations.datasets.datasetVersions/patch)	**`PATCH /v1/{datasetVersion.name}`** Updates a DatasetVersion.
`restore` (/vertex-ai/docs/reference/rest/v1/projects.locations.datasets.datasetVersions/restore)	**`GET /v1/{name}:restore`** Restores a dataset version.

REST Resource: v1.projects.locations.datasets.savedQueries

(/vertex-ai/docs/reference/rest/v1/projects.locations.datasets.savedQueries)

Methods	
`delete` (/vertex-ai/docs/reference/rest/v1/projects.locations.datasets.savedQueries/delete)	`DELETE /v1/{name}` Deletes a SavedQuery.
`list` (/vertex-ai/docs/reference/rest/v1/projects.locations.datasets.savedQueries/list)	`GET /v1/{parent}/savedQueries` Lists SavedQueries in a Dataset.

REST Resource: v1.projects.locations.deploymentResourcePools (/vertex-ai/docs/reference/rest/v1/projects.locations.deploymentResourcePools)

Methods	
`create` (/vertex-ai/docs/reference/rest/v1/projects.locations.deploymentResourcePools/create)	`POST /v1/{parent}/deploymentResourcePools` Create a DeploymentResourcePool.
`delete` (/vertex-ai/docs/reference/rest/v1/projects.locations.deploymentResourcePools/delete)	`DELETE /v1/{name}` Delete a DeploymentResourcePool.

Methods	
get (/vertex-ai/docs/reference/rest/v1/projects.locations.deploymentResourcePools/get)	`GET /v1/{name}` Get a DeploymentResourcePool.
list (/vertex-ai/docs/reference/rest/v1/projects.locations.deploymentResourcePools/list)	`GET /v1/{parent}/deploymentResourcePools` List DeploymentResourcePools in a location.
patch (/vertex-ai/docs/reference/rest/v1/projects.locations.deploymentResourcePools/patch)	`PATCH /v1/{deploymentResourcePool.name}` Update a DeploymentResourcePool.
queryDeployedModels (/vertex-ai/docs/reference/rest/v1/projects.locations.deploymentResourcePools/queryDeployedModels)	`GET /v1/{deploymentResourcePool}:queryDeployedModels` List DeployedModels that have been deployed on this DeploymentResourcePool.

REST Resource: v1.projects.locations.endpoints

(/vertex-ai/docs/reference/rest/v1/projects.locations.endpoints)

Methods	
computeTokens (/vertex-ai/docs/reference/rest/v1/projects.locations.endpoints/computeTokens)	`POST /v1/{endpoint}:computeTokens` Return a list of tokens based on the input text.

Methods	
`countTokens` (/vertex-ai/docs/reference/rest/v1/projects.locations.endpoints/countTokens)	`POST /v1/{endpoint}:countTokens` Perform a token counting.
`create` (/vertex-ai/docs/reference/rest/v1/projects.locations.endpoints/create)	`POST /v1/{parent}/endpoints` Creates an Endpoint.
`delete` (/vertex-ai/docs/reference/rest/v1/projects.locations.endpoints/delete)	`DELETE /v1/{name}` Deletes an Endpoint.
`deployModel` (/vertex-ai/docs/reference/rest/v1/projects.locations.endpoints/deployModel)	`POST /v1/{endpoint}:deployModel` Deploys a Model into this Endpoint, creating a DeployedModel within it.
`directPredict` (/vertex-ai/docs/reference/rest/v1/projects.locations.endpoints/directPredict)	`POST /v1/{endpoint}:directPredict` Perform an unary online prediction request to a gRPC model server for Vertex first-party products and frameworks.
`directRawPredict` (/vertex-ai/docs/reference/rest/v1/projects.locations.endpoints/directRawPredict)	`POST /v1/{endpoint}:directRawPredict` Perform an unary online prediction request to a gRPC model server for custom containers.
`explain` (/vertex-ai/docs/reference/rest/v1/projects.locations.endpoints/explain)	`POST /v1/{endpoint}:explain` Perform an online explanation.

Methods

`generateContent` (/vertex-ai/docs/reference/rest/v1/projects.locations.endpoints/generateContent)	`POST /v1/{model}:generateContent` Generate content with multimodal inputs.
`get` (/vertex-ai/docs/reference/rest/v1/projects.locations.endpoints/get)	`GET /v1/{name}` Gets an Endpoint.
`list` (/vertex-ai/docs/reference/rest/v1/projects.locations.endpoints/list)	`GET /v1/{parent}/endpoints` Lists Endpoints in a Location.
`mutateDeployedModel` (/vertex-ai/docs/reference/rest/v1/projects.locations.endpoints/mutateDeployedModel)	`POST /v1/{endpoint}:mutateDeployedModel` Updates an existing deployed model.
`patch` (/vertex-ai/docs/reference/rest/v1/projects.locations.endpoints/patch)	`PATCH /v1/{endpoint.name}` Updates an Endpoint.
`predict` (/vertex-ai/docs/reference/rest/v1/projects.locations.endpoints/predict)	`POST /v1/{endpoint}:predict` Perform an online prediction.
`predictLongRunning` (/vertex-ai/docs/reference/rest/v1/projects.locations.endpoints/predictLongRunning)	`POST /v1/{endpoint}:predictLongRunning`

Methods	
rawPredict (/vertex-ai/docs/reference/rest/v1/projects.locations.endpoints/rawPredict)	`POST /v1/{endpoint}:rawPredict` Perform an online prediction with an arbitrary HTTP payload.
serverStreamingPredict (/vertex-ai/docs/reference/rest/v1/projects.locations.endpoints/serverStreamingPredict)	`POST /v1/{endpoint}:serverStreamingPredict` Perform a server-side streaming online prediction request for Vertex LLM streaming.
streamGenerateContent (/vertex-ai/docs/reference/rest/v1/projects.locations.endpoints/streamGenerateContent)	`POST /v1/{model}:streamGenerateContent` Generate content with multimodal inputs with streaming support.
streamRawPredict (/vertex-ai/docs/reference/rest/v1/projects.locations.endpoints/streamRawPredict)	`POST /v1/{endpoint}:streamRawPredict` Perform a streaming online prediction with an arbitrary HTTP payload.
undeployModel (/vertex-ai/docs/reference/rest/v1/projects.locations.endpoints/undeployModel)	`POST /v1/{endpoint}:undeployModel` Undeploys a Model from an Endpoint, removing a DeployedModel from it, and freeing all resources it's using.
update (/vertex-ai/docs/reference/rest/v1/projects.locations.endpoints/update)	`POST /v1/{endpoint.name}:update` Updates an Endpoint with a long running operation.

REST Resource: v1.projects.locations.featureGroups

(/vertex-ai/docs/reference/rest/v1/projects.locations.featureGroups)

Methods

`create` (/vertex-ai/docs/reference/rest/v1/projects.locations.featureGroups/create)	`POST /v1/{parent}/featureGroups` Creates a new FeatureGroup in a given project and location.
`delete` (/vertex-ai/docs/reference/rest/v1/projects.locations.featureGroups/delete)	`DELETE /v1/{name}` Deletes a single FeatureGroup.
`get` (/vertex-ai/docs/reference/rest/v1/projects.locations.featureGroups/get)	`GET /v1/{name}` Gets details of a single FeatureGroup.
`getIamPolicy` (/vertex-ai/docs/reference/rest/v1/projects.locations.featureGroups/getIamPolicy)	`POST /v1/{resource}:getIamPolicy` Gets the access control policy for a resource.
`list` (/vertex-ai/docs/reference/rest/v1/projects.locations.featureGroups/list)	`GET /v1/{parent}/featureGroups` Lists FeatureGroups in a given project and location.
`patch` (/vertex-ai/docs/reference/rest/v1/projects.locations.featureGroups/patch)	`PATCH /v1/{featureGroup.name}` Updates the parameters of a single FeatureGroup.

Methods	
`setIamPolicy` (/vertex-ai/docs/reference/rest/v1/projects.locations.featureGroups/setIamPolicy)	`POST /v1/{resource}:setIamPolicy` Sets the access control policy on the specified resource.
`testIamPermissions` (/vertex-ai/docs/reference/rest/v1/projects.locations.featureGroups/testIamPermissions)	`POST /v1/{resource}:testIamPermissions` Returns permissions that a caller has on the specified resource.

REST Resource: v1.projects.locations.featureGroups.features

(/vertex-ai/docs/reference/rest/v1/projects.locations.featureGroups.features)

Methods	
`batchCreate` (/vertex-ai/docs/reference/rest/v1/projects.locations.featureGroups.features/batchCreate)	`POST /v1/{parent}/features:batchCreate` Creates a batch of Features in a given FeatureGroup.
`create` (/vertex-ai/docs/reference/rest/v1/projects.locations.featureGroups.features/create)	`POST /v1/{parent}/features` Creates a new Feature in a given FeatureGroup.
`delete` (/vertex-ai/docs/reference/rest/v1/projects.locations.featureGroups.features/delete)	`DELETE /v1/{name}` Deletes a single Feature.

Methods	
get (/vertex-ai/docs/reference/rest/v1/projects.locations.featureGroups.features/get)	`GET /v1/{name}` Gets details of a single Feature.
list (/vertex-ai/docs/reference/rest/v1/projects.locations.featureGroups.features/list)	`GET /v1/{parent}/features` Lists Features in a given FeatureGroup.
patch (/vertex-ai/docs/reference/rest/v1/projects.locations.featureGroups.features/patch)	`PATCH /v1/{feature.name}` Updates the parameters of a single Feature.

REST Resource: v1.projects.locations.featureOnlineStores

(/vertex-ai/docs/reference/rest/v1/projects.locations.featureOnlineStores)

Methods	
create (/vertex-ai/docs/reference/rest/v1/projects.locations.featureOnlineStores/create)	`POST /v1/{parent}/featureOnlineStores` Creates a new FeatureOnlineStore in a given project and location.
delete (/vertex-ai/docs/reference/rest/v1/projects.locations.featureOnlineStores/delete)	`DELETE /v1/{name}` Deletes a single FeatureOnlineStore.

Methods

get (/vertex-ai/docs/reference/rest/v1/projects.locations.featureOnlineStores/get)	`GET /v1/{name}` Gets details of a single FeatureOnlineStore.
getIamPolicy (/vertex-ai/docs/reference/rest/v1/projects.locations.featureOnlineStores/getIamPolicy)	`POST /v1/{resource}:getIamPolicy` Gets the access control policy for a resource.
list (/vertex-ai/docs/reference/rest/v1/projects.locations.featureOnlineStores/list)	`GET /v1/{parent}/featureOnlineStores` Lists FeatureOnlineStores in a given project and location.
patch (/vertex-ai/docs/reference/rest/v1/projects.locations.featureOnlineStores/patch)	`PATCH /v1/{featureOnlineStore.name}` Updates the parameters of a single FeatureOnlineStore.
setIamPolicy (/vertex-ai/docs/reference/rest/v1/projects.locations.featureOnlineStores/setIamPolicy)	`POST /v1/{resource}:setIamPolicy` Sets the access control policy on the specified resource.
testIamPermissions (/vertex-ai/docs/reference/rest/v1/projects.locations.featureOnlineStores/testIamPermissions)	`POST /v1/{resource}:testIamPermissions` Returns permissions that a caller has on the specified resource.

REST Resource: v1.projects.locations.featureOnlineStores.featureViews

(/vertex-ai/docs/reference/rest/v1/projects.locations.featureOnlineStores.featureViews)

Methods

`create` (/vertex-ai/docs/reference/rest/v1/projects.locations.featureOnlineStores.featureViews/create)	`POST /v1/{parent}/featureViews` Creates a new FeatureView in a given FeatureOnlineStore.
`delete` (/vertex-ai/docs/reference/rest/v1/projects.locations.featureOnlineStores.featureViews/delete)	`DELETE /v1/{name}` Deletes a single FeatureView.
`fetchFeatureValues` (/vertex-ai/docs/reference/rest/v1/projects.locations.featureOnlineStores.featureViews/fetchFeatureValues)	`POST /v1/{featureView}:fetchFeatureValues` Fetch feature values under a FeatureView.
`get` (/vertex-ai/docs/reference/rest/v1/projects.locations.featureOnlineStores.featureViews/get)	`GET /v1/{name}` Gets details of a single FeatureView.
`getIamPolicy` (/vertex-ai/docs/reference/rest/v1/projects.locations.featureOnlineStores.featureViews/getIamPolicy)	`POST /v1/{resource}:getIamPolicy` Gets the access control policy for a resource.

Methods

`list` (/vertex-ai/docs/reference/rest/v1/projects.locations.featureOnlineStores.featureViews/list)	`GET /v1/{parent}/featureViews` Lists FeatureViews in a given FeatureOnlineStore.
`patch` (/vertex-ai/docs/reference/rest/v1/projects.locations.featureOnlineStores.featureViews/patch)	`PATCH /v1/{featureView.name}` Updates the parameters of a single FeatureView.
`searchNearestEntities` (/vertex-ai/docs/reference/rest/v1/projects.locations.featureOnlineStores.featureViews/searchNearestEntities)	`POST /v1/{featureView}:searchNearestEntities` Search the nearest entities under a FeatureView.
`setIamPolicy` (/vertex-ai/docs/reference/rest/v1/projects.locations.featureOnlineStores.featureViews/setIamPolicy)	`POST /v1/{resource}:setIamPolicy` Sets the access control policy on the specified resource.
`sync` (/vertex-ai/docs/reference/rest/v1/projects.locations.featureOnlineStores.featureViews/sync)	`POST /v1/{featureView}:sync` Triggers on-demand sync for the FeatureView.
`testIamPermissions` (/vertex-ai/docs/reference/rest/v1/projects.locations.featureOnlineStores.featureViews/testIamPermissions)	`POST /v1/{resource}:testIamPermissions` Returns permissions that a caller has on the specified resource.

REST Resource: v1.projects.locations.featureOnlineStores.featureViews.featureViewSyncs

(/vertex-ai/docs/reference/rest/v1/projects.locations.featureOnlineStores.featureViews.featureViewSyncs)

Methods	
`get` (/vertex-ai/docs/reference/rest/v1/projects.locations.featureOnlineStores.featureViews.featureViewSyncs/get)	`GET /v1/{name}` Gets details of a single FeatureViewSync.
`list` (/vertex-ai/docs/reference/rest/v1/projects.locations.featureOnlineStores.featureViews.featureViewSyncs/list)	`GET /v1/{parent}/featureViewSyncs` Lists FeatureViewSyncs in a given FeatureView.

REST Resource: v1.projects.locations.featurestores

(/vertex-ai/docs/reference/rest/v1/projects.locations.featurestores)

Methods	
`batchReadFeatureValues` (/vertex-ai/docs/reference/rest/v1/projects.locations.featurestores/batchReadFeatureValues)	`POST /v1/{featurestore}:batchReadFeatureValues` Batch reads Feature values from a Featurestore.
`create` (/vertex-ai/docs/reference/rest/v1/projects.locations.featurestores/create)	`POST /v1/{parent}/featurestores` Creates a new Featurestore in a given project and location.

Methods

delete (/vertex-ai/docs/reference/rest/v1/projects.locations.featurestores/delete)	`DELETE /v1/{name}` Deletes a single Featurestore.
get (/vertex-ai/docs/reference/rest/v1/projects.locations.featurestores/get)	`GET /v1/{name}` Gets details of a single Featurestore.
getIamPolicy (/vertex-ai/docs/reference/rest/v1/projects.locations.featurestores/getIamPolicy)	`POST /v1/{resource}:getIamPolicy` Gets the access control policy for a resource.
list (/vertex-ai/docs/reference/rest/v1/projects.locations.featurestores/list)	`GET /v1/{parent}/featurestores` Lists Featurestores in a given project and location.
patch (/vertex-ai/docs/reference/rest/v1/projects.locations.featurestores/patch)	`PATCH /v1/{featurestore.name}` Updates the parameters of a single Featurestore.
searchFeatures (/vertex-ai/docs/reference/rest/v1/projects.locations.featurestores/searchFeatures)	`GET /v1/{location}/featurestores:searchFeatures` Searches Features matching a query in a given project.
setIamPolicy (/vertex-ai/docs/reference/rest/v1/projects.locations.featurestores/setIamPolicy)	`POST /v1/{resource}:setIamPolicy` Sets the access control policy on the specified resource.

Methods	
testIamPermissions (/vertex-ai/docs/reference/rest/v1/projects.locations.featurestores/testIamPermissions)	`POST /v1/{resource}:testIamPermissions` Returns permissions that a caller has on the specified resource.

REST Resource: v1.projects.locations.featurestores.entityTypes

(/vertex-ai/docs/reference/rest/v1/projects.locations.featurestores.entityTypes)

Methods	
create (/vertex-ai/docs/reference/rest/v1/projects.locations.featurestores.entityTypes/create)	`POST /v1/{parent}/entityTypes` Creates a new EntityType in a given Featurestore.
delete (/vertex-ai/docs/reference/rest/v1/projects.locations.featurestores.entityTypes/delete)	`DELETE /v1/{name}` Deletes a single EntityType.
deleteFeatureValues (/vertex-ai/docs/reference/rest/v1/projects.locations.featurestores.entityTypes/deleteFeatureValues)	`POST /v1/{entityType}:deleteFeatureValues` Delete Feature values from Featurestore.
exportFeatureValues (/vertex-ai/docs/reference/rest/v1/projects.locations.featurestores.entityTypes/exportFeatureValues)	`POST /v1/{entityType}:exportFeatureValues` Exports Feature values from all the entities of a target EntityType.

Methods

`get` (/vertex-ai/docs/reference/rest/v1/projects.locations.featurestores.entityTypes/get)	`GET /v1/{name}` Gets details of a single EntityType.
`getIamPolicy` (/vertex-ai/docs/reference/rest/v1/projects.locations.featurestores.entityTypes/getIamPolicy)	`POST /v1/{resource}:getIamPolicy` Gets the access control policy for a resource.
`importFeatureValues` (/vertex-ai/docs/reference/rest/v1/projects.locations.featurestores.entityTypes/importFeatureValues)	`POST /v1/{entityType}:importFeatureValues` Imports Feature values into the Featurestore from a source storage.
`list` (/vertex-ai/docs/reference/rest/v1/projects.locations.featurestores.entityTypes/list)	`GET /v1/{parent}/entityTypes` Lists EntityTypes in a given Featurestore.
`patch` (/vertex-ai/docs/reference/rest/v1/projects.locations.featurestores.entityTypes/patch)	`PATCH /v1/{entityType.name}` Updates the parameters of a single EntityType.
`readFeatureValues` (/vertex-ai/docs/reference/rest/v1/projects.locations.featurestores.entityTypes/readFeatureValues)	`POST /v1/{entityType}:readFeatureValues` Reads Feature values of a specific entity of an EntityType.

Methods

`setIamPolicy` (/vertex-ai/docs/reference/rest/v1/projects.locations.featurestores.entityTypes/setIamPolicy)	`POST /v1/{resource}:setIamPolicy` Sets the access control policy on the specified resource.
`streamingReadFeatureValues` (/vertex-ai/docs/reference/rest/v1/projects.locations.featurestores.entityTypes/streamingReadFeatureValues)	`POST /v1/{entityType}:streamingReadFeatureValues` Reads Feature values for multiple entities.
`testIamPermissions` (/vertex-ai/docs/reference/rest/v1/projects.locations.featurestores.entityTypes/testIamPermissions)	`POST /v1/{resource}:testIamPermissions` Returns permissions that a caller has on the specified resource.
`writeFeatureValues` (/vertex-ai/docs/reference/rest/v1/projects.locations.featurestores.entityTypes/writeFeatureValues)	`POST /v1/{entityType}:writeFeatureValues` Writes Feature values of one or more entities of an EntityType.

REST Resource: v1.projects.locations.featurestores.entityTypes.features

(/vertex-ai/docs/reference/rest/v1/projects.locations.featurestores.entityTypes.features)

Methods

`batchCreate` (/vertex-ai/docs/reference/rest/v1/projects.locations.featurestores.entityTypes.features/batchCreate)	`POST /v1/{parent}/features:batchCreate` Creates a batch of Features in a given EntityType.
`create` (/vertex-ai/docs/reference/rest/v1/projects.locations.featurestores.entityTypes.features/create)	`POST /v1/{parent}/features` Creates a new Feature in a given EntityType.
`delete` (/vertex-ai/docs/reference/rest/v1/projects.locations.featurestores.entityTypes.features/delete)	`DELETE /v1/{name}` Deletes a single Feature.
`get` (/vertex-ai/docs/reference/rest/v1/projects.locations.featurestores.entityTypes.features/get)	`GET /v1/{name}` Gets details of a single Feature.
`list` (/vertex-ai/docs/reference/rest/v1/projects.locations.featurestores.entityTypes.features/list)	`GET /v1/{parent}/features` Lists Features in a given EntityType.
`patch` (/vertex-ai/docs/reference/rest/v1/projects.locations.featurestores.entityTypes.features/patch)	`PATCH /v1/{feature.name}` Updates the parameters of a single Feature.

REST Resource: v1.projects.locations.hyperparameterTuningJobs

(/vertex-ai/docs/reference/rest/v1/projects.locations.hyperparameterTuningJobs)

Methods

`cancel` (/vertex-ai/docs/reference/rest/v1/projects.locations.hyperparameterTuningJobs/cancel)	`POST /v1/{name}:cancel` Cancels a HyperparameterTuningJob.
`create` (/vertex-ai/docs/reference/rest/v1/projects.locations.hyperparameterTuningJobs/create)	`POST /v1/{parent}/hyperparameterTuningJobs` Creates a HyperparameterTuningJob
`delete` (/vertex-ai/docs/reference/rest/v1/projects.locations.hyperparameterTuningJobs/delete)	`DELETE /v1/{name}` Deletes a HyperparameterTuningJob.
`get` (/vertex-ai/docs/reference/rest/v1/projects.locations.hyperparameterTuningJobs/get)	`GET /v1/{name}` Gets a HyperparameterTuningJob
`list` (/vertex-ai/docs/reference/rest/v1/projects.locations.hyperparameterTuningJobs/list)	`GET /v1/{parent}/hyperparameterTuningJobs` Lists HyperparameterTuningJobs in a Location.

REST Resource: v1.projects.locations.indexEndpoints

(/vertex-ai/docs/reference/rest/v1/projects.locations.indexEndpoints)

Methods	
`create` (/vertex-ai/docs/reference/rest/v1/projects.locations.indexEndpoints/create)	`POST /v1/{parent}/indexEndpoints` Creates an IndexEndpoint.
`delete` (/vertex-ai/docs/reference/rest/v1/projects.locations.indexEndpoints/delete)	`DELETE /v1/{name}` Deletes an IndexEndpoint.
`deployIndex` (/vertex-ai/docs/reference/rest/v1/projects.locations.indexEndpoints/deployIndex)	`POST /v1/{indexEndpoint}:deployIndex` Deploys an Index into this IndexEndpoint, creating a DeployedIndex within it.
`get` (/vertex-ai/docs/reference/rest/v1/projects.locations.indexEndpoints/get)	`GET /v1/{name}` Gets an IndexEndpoint.
`list` (/vertex-ai/docs/reference/rest/v1/projects.locations.indexEndpoints/list)	`GET /v1/{parent}/indexEndpoints` Lists IndexEndpoints in a Location.
`mutateDeployedIndex` (/vertex-ai/docs/reference/rest/v1/projects.locations.indexEndpoints/mutateDeployedIndex)	`POST /v1/{indexEndpoint}:mutateDeployedIndex` Update an existing DeployedIndex under an IndexEndpoint.

Methods	
patch (/vertex-ai/docs/reference/rest/v1/projects.locations.indexEndpoints/patch)	`PATCH /v1/{indexEndpoint.name}` Updates an IndexEndpoint.
undeployIndex (/vertex-ai/docs/reference/rest/v1/projects.locations.indexEndpoints/undeployIndex)	`POST /v1/{indexEndpoint}:undeployIndex` Undeploys an Index from an IndexEndpoint, removing a DeployedIndex from it, and freeing all resources it's using.

REST Resource: v1.projects.locations.indexes

(/vertex-ai/docs/reference/rest/v1/projects.locations.indexes)

Methods	
create (/vertex-ai/docs/reference/rest/v1/projects.locations.indexes/create)	`POST /v1/{parent}/indexes` Creates an Index.
delete (/vertex-ai/docs/reference/rest/v1/projects.locations.indexes/delete)	`DELETE /v1/{name}` Deletes an Index.
get (/vertex-ai/docs/reference/rest/v1/projects.locations.indexes/get)	`GET /v1/{name}` Gets an Index.

Methods	
`list` (/vertex-ai/docs/reference/rest/v1/projects.locations.indexes/list)	`GET /v1/{parent}/indexes` Lists Indexes in a Location.
`patch` (/vertex-ai/docs/reference/rest/v1/projects.locations.indexes/patch)	`PATCH /v1/{index.name}` Updates an Index.
`removeDatapoints` (/vertex-ai/docs/reference/rest/v1/projects.locations.indexes/removeDatapoints)	`POST /v1/{index}:removeDatapoints` Remove Datapoints from an Index.
`upsertDatapoints` (/vertex-ai/docs/reference/rest/v1/projects.locations.indexes/upsertDatapoints)	`POST /v1/{index}:upsertDatapoints` Add/update Datapoints into an Index.

REST Resource: v1.projects.locations.metadataStores

(/vertex-ai/docs/reference/rest/v1/projects.locations.metadataStores)

Methods	
`create` (/vertex-ai/docs/reference/rest/v1/projects.locations.metadataStores/create)	`POST /v1/{parent}/metadataStores` Initializes a MetadataStore, including allocation of resources.

Methods	
`delete` (/vertex-ai/docs/reference/rest/v1/projects.locations.metadataStores/delete)	`DELETE /v1/{name}` Deletes a single MetadataStore and all its child resources (Artifacts, Executions, and Contexts).
`get` (/vertex-ai/docs/reference/rest/v1/projects.locations.metadataStores/get)	`GET /v1/{name}` Retrieves a specific MetadataStore.
`list` (/vertex-ai/docs/reference/rest/v1/projects.locations.metadataStores/list)	`GET /v1/{parent}/metadataStores` Lists MetadataStores for a Location.

REST Resource: v1.projects.locations.metadataStores.artifacts

(/vertex-ai/docs/reference/rest/v1/projects.locations.metadataStores.artifacts)

Methods	
`create` (/vertex-ai/docs/reference/rest/v1/projects.locations.metadataStores.artifacts/create)	`POST /v1/{parent}/artifacts` Creates an Artifact associated with a MetadataStore.
`delete` (/vertex-ai/docs/reference/rest/v1/projects.locations.metadataStores.artifacts/delete)	`DELETE /v1/{name}` Deletes an Artifact.

Methods	
get (/vertex-ai/docs/reference/rest/v1/projects.locations.metadataStores.artifacts/get)	`GET /v1/{name}` Retrieves a specific Artifact.
list (/vertex-ai/docs/reference/rest/v1/projects.locations.metadataStores.artifacts/list)	`GET /v1/{parent}/artifacts` Lists Artifacts in the MetadataStore.
patch (/vertex-ai/docs/reference/rest/v1/projects.locations.metadataStores.artifacts/patch)	`PATCH /v1/{artifact.name}` Updates a stored Artifact.
purge (/vertex-ai/docs/reference/rest/v1/projects.locations.metadataStores.artifacts/purge)	`POST /v1/{parent}/artifacts:purge` Purges Artifacts.
queryArtifactLineageSubgraph (/vertex-ai/docs/reference/rest/v1/projects.locations.metadataStores.artifacts/queryArtifactLineageSubgraph)	`GET /v1/{artifact}:queryArtifactLineageSubgraph` Retrieves lineage of an Artifact represented through Artifacts and Executions connected by Event edges and returned as a LineageSubgraph.

REST Resource: v1.projects.locations.metadataStores.contexts

(/vertex-ai/docs/reference/rest/v1/projects.locations.metadataStores.contexts)

Methods

addContextArtifactsAndExecutions (/vertex-ai/docs/reference/rest/v1/projects.locations.metadataStores.contexts/addContextArtifactsAndExecutions)	`POST /v1/{context}:addContextArtifactsAndExecutions` Adds a set of Artifacts and Executions to a Context.
addContextChildren (/vertex-ai/docs/reference/rest/v1/projects.locations.metadataStores.contexts/addContextChildren)	`POST /v1/{context}:addContextChildren` Adds a set of Contexts as children to a parent Context.
create (/vertex-ai/docs/reference/rest/v1/projects.locations.metadataStores.contexts/create)	`POST /v1/{parent}/contexts` Creates a Context associated with a MetadataStore.
delete (/vertex-ai/docs/reference/rest/v1/projects.locations.metadataStores.contexts/delete)	`DELETE /v1/{name}` Deletes a stored Context.
get (/vertex-ai/docs/reference/rest/v1/projects.locations.metadataStores.contexts/get)	`GET /v1/{name}` Retrieves a specific Context.
list (/vertex-ai/docs/reference/rest/v1/projects.locations.metadataStores.contexts/list)	`GET /v1/{parent}/contexts` Lists Contexts on the MetadataStore.

Methods	
`patch` (/vertex-ai/docs/reference/rest/v1/projects.locations.metadataStores.contexts/patch)	`PATCH /v1/{context.name}` Updates a stored Context.
`purge` (/vertex-ai/docs/reference/rest/v1/projects.locations.metadataStores.contexts/purge)	`POST /v1/{parent}/contexts:purge` Purges Contexts.
`queryContextLineageSubgraph` (/vertex-ai/docs/reference/rest/v1/projects.locations.metadataStores.contexts/queryContextLineageSubgraph)	`GET /v1/{context}:queryContextLineageSubgraph` Retrieves Artifacts and Executions within the specified Context, connected by Event edges and returned as a LineageSubgraph.
`removeContextChildren` (/vertex-ai/docs/reference/rest/v1/projects.locations.metadataStores.contexts/removeContextChildren)	`POST /v1/{context}:removeContextChildren` Remove a set of children contexts from a parent Context.

REST Resource: v1.projects.locations.metadataStores.executions

(/vertex-ai/docs/reference/rest/v1/projects.locations.metadataStores.executions)

Methods	
`addExecutionEvents` (/vertex-ai/docs/reference/rest/v1/projects.locations.metadataStores.executions/addExecutionEvents)	`POST /v1/{execution}:addExecutionEvents` Adds Events to the specified Execution.

Methods

`create` (/vertex-ai/docs/reference/rest/v1/projects.locations.metadataStores.executions/create)	`POST /v1/{parent}/executions` Creates an Execution associated with a MetadataStore.
`delete` (/vertex-ai/docs/reference/rest/v1/projects.locations.metadataStores.executions/delete)	`DELETE /v1/{name}` Deletes an Execution.
`get` (/vertex-ai/docs/reference/rest/v1/projects.locations.metadataStores.executions/get)	`GET /v1/{name}` Retrieves a specific Execution.
`list` (/vertex-ai/docs/reference/rest/v1/projects.locations.metadataStores.executions/list)	`GET /v1/{parent}/executions` Lists Executions in the MetadataStore.
`patch` (/vertex-ai/docs/reference/rest/v1/projects.locations.metadataStores.executions/patch)	`PATCH /v1/{execution.name}` Updates a stored Execution.
`purge` (/vertex-ai/docs/reference/rest/v1/projects.locations.metadataStores.executions/purge)	`POST /v1/{parent}/executions:purge` Purges Executions.

Methods	
queryExecutionInputsAndOutputs (/vertex-ai/docs/reference/rest/v1/projects.locations.metadataStores.executions/queryExecutionInputsAndOutputs)	`GET /v1/{execution}:queryExecutionInputsAndOutputs` Obtains the set of input and output Artifacts for this Execution, in the form of LineageSubgraph that also contains the Execution and connecting Events.

REST Resource: v1.projects.locations.metadataStores.metadataSchemas

(/vertex-ai/docs/reference/rest/v1/projects.locations.metadataStores.metadataSchemas)

Methods	
create (/vertex-ai/docs/reference/rest/v1/projects.locations.metadataStores.metadataSchemas/create)	`POST /v1/{parent}/metadataSchemas` Creates a MetadataSchema.
get (/vertex-ai/docs/reference/rest/v1/projects.locations.metadataStores.metadataSchemas/get)	`GET /v1/{name}` Retrieves a specific MetadataSchema.
list (/vertex-ai/docs/reference/rest/v1/projects.locations.metadataStores.metadataSchemas/list)	`GET /v1/{parent}/metadataSchemas` Lists MetadataSchemas.

REST Resource: v1.projects.locations.migratableResources

(/vertex-ai/docs/reference/rest/v1/projects.locations.migratableResources)

Methods	
batchMigrate (/vertex-ai/docs/reference/rest/v1/projects.locations.migratableResources/batchMigrate)	`POST /v1/{parent}/migratableResources:batchMigrate` Batch migrates resources from ml.googleapis.com, automl.googleapis.com, and datalabeling.googleapis.com to Vertex AI.
search (/vertex-ai/docs/reference/rest/v1/projects.locations.migratableResources/search)	`POST /v1/{parent}/migratableResources:search` Searches all of the resources in automl.googleapis.com, datalabeling.googleapis.com and ml.googleapis.com that can be migrated to Vertex AI's given location.

REST Resource: v1.projects.locations.modelDeploymentMonitoringJobs

(/vertex-ai/docs/reference/rest/v1/projects.locations.modelDeploymentMonitoringJobs)

Methods	
create (/vertex-ai/docs/reference/rest/v1/projects.locations.modelDeploymentMonitoringJobs/create)	`POST /v1/{parent}/modelDeploymentMonitoringJobs` Creates a ModelDeploymentMonitoringJob.
delete (/vertex-ai/docs/reference/rest/v1/projects.locations.modelDeploymentMonitoringJobs/delete)	`DELETE /v1/{name}` Deletes a ModelDeploymentMonitoringJob.

Methods

get (/vertex-ai/docs/reference/rest/v1/projects.locations.modelDeploymentMonitoringJobs/get)	`GET /v1/{name}` Gets a ModelDeploymentMonitoringJob.
list (/vertex-ai/docs/reference/rest/v1/projects.locations.modelDeploymentMonitoringJobs/list)	`GET /v1/{parent}/modelDeploymentMonitoringJobs` Lists ModelDeploymentMonitoringJobs in a Location.
patch (/vertex-ai/docs/reference/rest/v1/projects.locations.modelDeploymentMonitoringJobs/patch)	`PATCH /v1/{modelDeploymentMonitoringJob.name}` Updates a ModelDeploymentMonitoringJob.
pause (/vertex-ai/docs/reference/rest/v1/projects.locations.modelDeploymentMonitoringJobs/pause)	`POST /v1/{name}:pause` Pauses a ModelDeploymentMonitoringJob.
resume (/vertex-ai/docs/reference/rest/v1/projects.locations.modelDeploymentMonitoringJobs/resume)	`POST /v1/{name}:resume` Resumes a paused ModelDeploymentMonitoringJob.
searchModelDeploymentMonitoringStatsAnomalies (/vertex-ai/docs/reference/rest/v1/projects.locations.modelDeploymentMonitoringJobs/searchModelDeploymentMonitoringStatsAnomalies)	`POST /v1/{modelDeploymentMonitoringJob}:searchModelDeploymentMonitoringStatsAnomalies` Searches Model Monitoring Statistics generated within a given time window.

REST Resource: v1.projects.locations.models

(/vertex-ai/docs/reference/rest/v1/projects.locations.models)

Methods

`copy` (/vertex-ai/docs/reference/rest/v1/projects.locations.models/copy)	`POST /v1/{parent}/models:copy` Copies an already existing Vertex AI Model into the specified Location.
`delete` (/vertex-ai/docs/reference/rest/v1/projects.locations.models/delete)	`DELETE /v1/{name}` Deletes a Model.
`deleteVersion` (/vertex-ai/docs/reference/rest/v1/projects.locations.models/deleteVersion)	`DELETE /v1/{name}:deleteVersion` Deletes a Model version.
`export` (/vertex-ai/docs/reference/rest/v1/projects.locations.models/export)	`POST /v1/{name}:export` Exports a trained, exportable Model to a location specified by the user.
`get` (/vertex-ai/docs/reference/rest/v1/projects.locations.models/get)	`GET /v1/{name}` Gets a Model.
`getIamPolicy` (/vertex-ai/docs/reference/rest/v1/projects.locations.models/getIamPolicy)	`POST /v1/{resource}:getIamPolicy` Gets the access control policy for a resource.

Methods

`list` (/vertex-ai/docs/reference/rest/v1/projects.locations.models/list)	`GET /v1/{parent}/models` Lists Models in a Location.
`listCheckpoints` (/vertex-ai/docs/reference/rest/v1/projects.locations.models/listCheckpoints)	`GET /v1/{name}:listCheckpoints` Lists checkpoints of the specified model version.
`listVersions` (/vertex-ai/docs/reference/rest/v1/projects.locations.models/listVersions)	`GET /v1/{name}:listVersions` Lists versions of the specified model.
`mergeVersionAliases` (/vertex-ai/docs/reference/rest/v1/projects.locations.models/mergeVersionAliases)	`POST /v1/{name}:mergeVersionAliases` Merges a set of aliases for a Model version.
`patch` (/vertex-ai/docs/reference/rest/v1/projects.locations.models/patch)	`PATCH /v1/{model.name}` Updates a Model.
`setIamPolicy` (/vertex-ai/docs/reference/rest/v1/projects.locations.models/setIamPolicy)	`POST /v1/{resource}:setIamPolicy` Sets the access control policy on the specified resource.
`testIamPermissions` (/vertex-ai/docs/reference/rest/v1/projects.locations.models/testIamPermissions)	`POST /v1/{resource}:testIamPermissions` Returns permissions that a caller has on the specified resource.

Methods	
`updateExplanationDataset` (/vertex-ai/docs/reference/rest/v1/projects.locations.models/updateExplanationDataset)	`POST /v1/{model}:updateExplanationDataset` Incrementally update the dataset used for an examples model.
`upload` (/vertex-ai/docs/reference/rest/v1/projects.locations.models/upload)	`POST /v1/{parent}/models:upload` Uploads a Model artifact into Vertex AI.

REST Resource: v1.projects.locations.models.evaluations

(/vertex-ai/docs/reference/rest/v1/projects.locations.models.evaluations)

Methods	
`get` (/vertex-ai/docs/reference/rest/v1/projects.locations.models.evaluations/get)	`GET /v1/{name}` Gets a ModelEvaluation.
`import` (/vertex-ai/docs/reference/rest/v1/projects.locations.models.evaluations/import)	`POST /v1/{parent}/evaluations:import` Imports an externally generated ModelEvaluation.
`list` (/vertex-ai/docs/reference/rest/v1/projects.locations.models.evaluations/list)	`GET /v1/{parent}/evaluations` Lists ModelEvaluations in a Model.

REST Resource: v1.projects.locations.models.evaluations.slices

(/vertex-ai/docs/reference/rest/v1/projects.locations.models.evaluations.slices)

Methods	
`batchImport` (/vertex-ai/docs/reference/rest/v1/projects.locations.models.evaluations.slices/batchImport)	**`POST /v1/{parent}:batchImport`** Imports a list of externally generated EvaluatedAnnotations.
`get` (/vertex-ai/docs/reference/rest/v1/projects.locations.models.evaluations.slices/get)	**`GET /v1/{name}`** Gets a ModelEvaluationSlice.
`list` (/vertex-ai/docs/reference/rest/v1/projects.locations.models.evaluations.slices/list)	**`GET /v1/{parent}/slices`** Lists ModelEvaluationSlices in a ModelEvaluation.

REST Resource: v1.projects.locations.nasJobs

(/vertex-ai/docs/reference/rest/v1/projects.locations.nasJobs)

Methods	
`cancel` (/vertex-ai/docs/reference/rest/v1/projects.locations.nasJobs/cancel)	**`POST /v1/{name}:cancel`** Cancels a NasJob.

Methods	
create (/vertex-ai/docs/reference/rest/v1/projects.locations.nasJobs/create)	`POST /v1/{parent}/nasJobs` Creates a NasJob
delete (/vertex-ai/docs/reference/rest/v1/projects.locations.nasJobs/delete)	`DELETE /v1/{name}` Deletes a NasJob.
get (/vertex-ai/docs/reference/rest/v1/projects.locations.nasJobs/get)	`GET /v1/{name}` Gets a NasJob
list (/vertex-ai/docs/reference/rest/v1/projects.locations.nasJobs/list)	`GET /v1/{parent}/nasJobs` Lists NasJobs in a Location.

REST Resource: v1.projects.locations.nasJobs.nasTrialDetails

(/vertex-ai/docs/reference/rest/v1/projects.locations.nasJobs.nasTrialDetails)

Methods	
get (/vertex-ai/docs/reference/rest/v1/projects.locations.nasJobs.nasTrialDetails/get)	`GET /v1/{name}` Gets a NasTrialDetail.

Methods	
`list` (/vertex-ai/docs/reference/rest/v1/projects.locations.nasJobs.nasTrialDetails/list)	`GET /v1/{parent}/nasTrialDetails` List top NasTrialDetails of a NasJob.

REST Resource: v1.projects.locations.notebookExecutionJobs

(/vertex-ai/docs/reference/rest/v1/projects.locations.notebookExecutionJobs)

Methods	
`create` (/vertex-ai/docs/reference/rest/v1/projects.locations.notebookExecutionJobs/create)	`POST /v1/{parent}/notebookExecutionJobs` Creates a NotebookExecutionJob.
`delete` (/vertex-ai/docs/reference/rest/v1/projects.locations.notebookExecutionJobs/delete)	`DELETE /v1/{name}` Deletes a NotebookExecutionJob.
`get` (/vertex-ai/docs/reference/rest/v1/projects.locations.notebookExecutionJobs/get)	`GET /v1/{name}` Gets a NotebookExecutionJob.
`list` (/vertex-ai/docs/reference/rest/v1/projects.locations.notebookExecutionJobs/list)	`GET /v1/{parent}/notebookExecutionJobs` Lists NotebookExecutionJobs in a Location.

REST Resource: v1.projects.locations.notebookRuntimeTemplates

(/vertex-ai/docs/reference/rest/v1/projects.locations.notebookRuntimeTemplates)

Methods

`create` (/vertex-ai/docs/reference/rest/v1/projects.locations.notebookRuntimeTemplates/create)	**`POST /v1/{parent}/notebookRuntimeTemplates`** Creates a NotebookRuntimeTemplate.
`delete` (/vertex-ai/docs/reference/rest/v1/projects.locations.notebookRuntimeTemplates/delete)	**`DELETE /v1/{name}`** Deletes a NotebookRuntimeTemplate.
`get` (/vertex-ai/docs/reference/rest/v1/projects.locations.notebookRuntimeTemplates/get)	**`GET /v1/{name}`** Gets a NotebookRuntimeTemplate.
`getIamPolicy` (/vertex-ai/docs/reference/rest/v1/projects.locations.notebookRuntimeTemplates/getIamPolicy)	**`POST /v1/{resource}:getIamPolicy`** Gets the access control policy for a resource.
`list` (/vertex-ai/docs/reference/rest/v1/projects.locations.notebookRuntimeTemplates/list)	**`GET /v1/{parent}/notebookRuntimeTemplates`** Lists NotebookRuntimeTemplates in a Location.

Methods	
`patch` (/vertex-ai/docs/reference/rest/v1/projects.locations.notebookRuntimeTemplates/patch)	`PATCH /v1/{notebookRuntimeTemplate.name}` Updates a NotebookRuntimeTemplate.
`setIamPolicy` (/vertex-ai/docs/reference/rest/v1/projects.locations.notebookRuntimeTemplates/setIamPolicy)	`POST /v1/{resource}:setIamPolicy` Sets the access control policy on the specified resource.
`testIamPermissions` (/vertex-ai/docs/reference/rest/v1/projects.locations.notebookRuntimeTemplates/testIamPermissions)	`POST /v1/{resource}:testIamPermissions` Returns permissions that a caller has on the specified resource.

REST Resource: v1.projects.locations.notebookRuntimes

(/vertex-ai/docs/reference/rest/v1/projects.locations.notebookRuntimes)

Methods	
`assign` (/vertex-ai/docs/reference/rest/v1/projects.locations.notebookRuntimes/assign)	`POST /v1/{parent}/notebookRuntimes:assign` Assigns a NotebookRuntime to a user for a particular Notebook file.
`delete` (/vertex-ai/docs/reference/rest/v1/projects.locations.notebookRuntimes/delete)	`DELETE /v1/{name}` Deletes a NotebookRuntime.

Methods	
`get` (/vertex-ai/docs/reference/rest/v1/projects.locations.notebookRuntimes/get)	`GET /v1/{name}` Gets a NotebookRuntime.
`list` (/vertex-ai/docs/reference/rest/v1/projects.locations.notebookRuntimes/list)	`GET /v1/{parent}/notebookRuntimes` Lists NotebookRuntimes in a Location.
`start` (/vertex-ai/docs/reference/rest/v1/projects.locations.notebookRuntimes/start)	`POST /v1/{name}:start` Starts a NotebookRuntime.
`stop` (/vertex-ai/docs/reference/rest/v1/projects.locations.notebookRuntimes/stop)	`POST /v1/{name}:stop` Stops a NotebookRuntime.
`upgrade` (/vertex-ai/docs/reference/rest/v1/projects.locations.notebookRuntimes/upgrade)	`POST /v1/{name}:upgrade` Upgrades a NotebookRuntime.

REST Resource: v1.projects.locations.operations

(/vertex-ai/docs/reference/rest/v1/projects.locations.operations)

Methods	
cancel (/vertex-ai/docs/reference/rest/v1/projects.locations.operations/cancel)	`POST /v1/{name}:cancel` Starts asynchronous cancellation on a long-running operation.
delete (/vertex-ai/docs/reference/rest/v1/projects.locations.operations/delete)	`DELETE /v1/{name}` Deletes a long-running operation.
get (/vertex-ai/docs/reference/rest/v1/projects.locations.operations/get)	`GET /v1/{name}` Gets the latest state of a long-running operation.
list (/vertex-ai/docs/reference/rest/v1/projects.locations.operations/list)	`GET /v1/{name}/operations` Lists operations that match the specified filter in the request.
wait (/vertex-ai/docs/reference/rest/v1/projects.locations.operations/wait)	`POST /v1/{name}:wait` Waits until the specified long-running operation is done or reaches at most a specified timeout, returning the latest state.

REST Resource: v1.projects.locations.persistentResources

(/vertex-ai/docs/reference/rest/v1/projects.locations.persistentResources)

Methods	
create (/vertex-ai/docs/reference/rest/v1/projects.locations.persistentResources/create)	`POST /v1/{parent}/persistentResources` Creates a PersistentResource.
delete (/vertex-ai/docs/reference/rest/v1/projects.locations.persistentResources/delete)	`DELETE /v1/{name}` Deletes a PersistentResource.
get (/vertex-ai/docs/reference/rest/v1/projects.locations.persistentResources/get)	`GET /v1/{name}` Gets a PersistentResource.
list (/vertex-ai/docs/reference/rest/v1/projects.locations.persistentResources/list)	`GET /v1/{parent}/persistentResources` Lists PersistentResources in a Location.
patch (/vertex-ai/docs/reference/rest/v1/projects.locations.persistentResources/patch)	`PATCH /v1/{persistentResource.name}` Updates a PersistentResource.
reboot (/vertex-ai/docs/reference/rest/v1/projects.locations.persistentResources/reboot)	`POST /v1/{name}:reboot` Reboots a PersistentResource.

REST Resource: v1.projects.locations.pipelineJobs

(/vertex-ai/docs/reference/rest/v1/projects.locations.pipelineJobs)

Methods

batchCancel (/vertex-ai/docs/reference/rest/v1/projects.locations.pipelineJobs/batchCancel)	`POST /v1/{parent}/pipelineJobs:batchCancel` Batch cancel PipelineJobs.
batchDelete (/vertex-ai/docs/reference/rest/v1/projects.locations.pipelineJobs/batchDelete)	`POST /v1/{parent}/pipelineJobs:batchDelete` Batch deletes PipelineJobs The Operation is atomic.
cancel (/vertex-ai/docs/reference/rest/v1/projects.locations.pipelineJobs/cancel)	`POST /v1/{name}:cancel` Cancels a PipelineJob.
create (/vertex-ai/docs/reference/rest/v1/projects.locations.pipelineJobs/create)	`POST /v1/{parent}/pipelineJobs` Creates a PipelineJob.
delete (/vertex-ai/docs/reference/rest/v1/projects.locations.pipelineJobs/delete)	`DELETE /v1/{name}` Deletes a PipelineJob.
get (/vertex-ai/docs/reference/rest/v1/projects.locations.pipelineJobs/get)	`GET /v1/{name}` Gets a PipelineJob.
list (/vertex-ai/docs/reference/rest/v1/projects.locations.pipelineJobs/list)	`GET /v1/{parent}/pipelineJobs` Lists PipelineJobs in a Location.

REST Resource: v1.projects.locations.publishers.models

(/vertex-ai/docs/reference/rest/v1/projects.locations.publishers.models)

Methods

computeTokens (/vertex-ai/docs/reference/rest/v1/projects.locations.publishers.models/computeTokens)	**`POST /v1/{endpoint}:computeTokens`** Return a list of tokens based on the input text.
countTokens (/vertex-ai/docs/reference/rest/v1/projects.locations.publishers.models/countTokens)	**`POST /v1/{endpoint}:countTokens`** Perform a token counting.
generateContent (/vertex-ai/docs/reference/rest/v1/projects.locations.publishers.models/generateContent)	**`POST /v1/{model}:generateContent`** Generate content with multimodal inputs.
predict (/vertex-ai/docs/reference/rest/v1/projects.locations.publishers.models/predict)	**`POST /v1/{endpoint}:predict`** Perform an online prediction.
predictLongRunning (/vertex-ai/docs/reference/rest/v1/projects.locations.publishers.models/predictLongRunning)	**`POST /v1/{endpoint}:predictLongRunning`**
rawPredict (/vertex-ai/docs/reference/rest/v1/projects.locations.publishers.models/rawPredict)	**`POST /v1/{endpoint}:rawPredict`** Perform an online prediction with an arbitrary HTTP payload.

Methods	
`serverStreamingPredict` (/vertex-ai/docs/reference/rest/v1/projects.locations.publishers.models/serverStreamingPredict)	`POST /v1/{endpoint}:serverStreamingPredict` Perform a server-side streaming online prediction request for Vertex LLM streaming.
`streamGenerateContent` (/vertex-ai/docs/reference/rest/v1/projects.locations.publishers.models/streamGenerateContent)	`POST /v1/{model}:streamGenerateContent` Generate content with multimodal inputs with streaming support.
`streamRawPredict` (/vertex-ai/docs/reference/rest/v1/projects.locations.publishers.models/streamRawPredict)	`POST /v1/{endpoint}:streamRawPredict` Perform a streaming online prediction with an arbitrary HTTP payload.

REST Resource: v1.projects.locations.ragCorpora

(/vertex-ai/docs/reference/rest/v1/projects.locations.ragCorpora)

Methods	
`create` (/vertex-ai/docs/reference/rest/v1/projects.locations.ragCorpora/create)	`POST /v1/{parent}/ragCorpora` Creates a RagCorpus.
`delete` (/vertex-ai/docs/reference/rest/v1/projects.locations.ragCorpora/delete)	`DELETE /v1/{name}` Deletes a RagCorpus.

Methods	
`get` (/vertex-ai/docs/reference/rest/v1/projects.locations.ragCorpora/get)	`GET /v1/{name}` Gets a RagCorpus.
`list` (/vertex-ai/docs/reference/rest/v1/projects.locations.ragCorpora/list)	`GET /v1/{parent}/ragCorpora` Lists RagCorpora in a Location.
`patch` (/vertex-ai/docs/reference/rest/v1/projects.locations.ragCorpora/patch)	`PATCH /v1/{ragCorpus.name}` Updates a RagCorpus.

REST Resource: v1.projects.locations.ragCorpora.ragFiles

(/vertex-ai/docs/reference/rest/v1/projects.locations.ragCorpora.ragFiles)

Methods	
`delete` (/vertex-ai/docs/reference/rest/v1/projects.locations.ragCorpora.ragFiles/delete)	`DELETE /v1/{name}` Deletes a RagFile.
`get` (/vertex-ai/docs/reference/rest/v1/projects.locations.ragCorpora.ragFiles/get)	`GET /v1/{name}` Gets a RagFile.

Methods	
import (/vertex-ai/docs/reference/rest/v1/projects.locations.ragCorpora.ragFiles/import)	`POST /v1/{parent}/ragFiles:import` Import files from Google Cloud Storage or Google Drive into a RagCorpus.
list (/vertex-ai/docs/reference/rest/v1/projects.locations.ragCorpora.ragFiles/list)	`GET /v1/{parent}/ragFiles` Lists RagFiles in a RagCorpus.

REST Resource: v1.projects.locations.reasoningEngines

(/vertex-ai/docs/reference/rest/v1/projects.locations.reasoningEngines)

Methods	
create (/vertex-ai/docs/reference/rest/v1/projects.locations.reasoningEngines/create)	`POST /v1/{parent}/reasoningEngines` Creates a reasoning engine.
delete (/vertex-ai/docs/reference/rest/v1/projects.locations.reasoningEngines/delete)	`DELETE /v1/{name}` Deletes a reasoning engine.
get (/vertex-ai/docs/reference/rest/v1/projects.locations.reasoningEngines/get)	`GET /v1/{name}` Gets a reasoning engine.

Methods	
`list` (/vertex-ai/docs/reference/rest/v1/projects.locations.reasoningEngines/list)	`GET /v1/{parent}/reasoningEngines` Lists reasoning engines in a location.
`patch` (/vertex-ai/docs/reference/rest/v1/projects.locations.reasoningEngines/patch)	`PATCH /v1/{reasoningEngine.name}` Updates a reasoning engine.
`query` (/vertex-ai/docs/reference/rest/v1/projects.locations.reasoningEngines/query)	`POST /v1/{name}:query` Queries using a reasoning engine.
`streamQuery` (/vertex-ai/docs/reference/rest/v1/projects.locations.reasoningEngines/streamQuery)	`POST /v1/{name}:streamQuery` Streams queries using a reasoning engine.

REST Resource: v1.projects.locations.schedules

(/vertex-ai/docs/reference/rest/v1/projects.locations.schedules)

Methods	
`create` (/vertex-ai/docs/reference/rest/v1/projects.locations.schedules/create)	`POST /v1/{parent}/schedules` Creates a Schedule.

Methods	
`delete` (/vertex-ai/docs/reference/rest/v1/projects.locations.schedules/delete)	`DELETE /v1/{name}` Deletes a Schedule.
`get` (/vertex-ai/docs/reference/rest/v1/projects.locations.schedules/get)	`GET /v1/{name}` Gets a Schedule.
`list` (/vertex-ai/docs/reference/rest/v1/projects.locations.schedules/list)	`GET /v1/{parent}/schedules` Lists Schedules in a Location.
`patch` (/vertex-ai/docs/reference/rest/v1/projects.locations.schedules/patch)	`PATCH /v1/{schedule.name}` Updates an active or paused Schedule.
`pause` (/vertex-ai/docs/reference/rest/v1/projects.locations.schedules/pause)	`POST /v1/{name}:pause` Pauses a Schedule.
`resume` (/vertex-ai/docs/reference/rest/v1/projects.locations.schedules/resume)	`POST /v1/{name}:resume` Resumes a paused Schedule to start scheduling new runs.

REST Resource: v1.projects.locations.specialistPools

(/vertex-ai/docs/reference/rest/v1/projects.locations.specialistPools)

Methods

create (/vertex-ai/docs/reference/rest/v1/projects.locations.specialistPools/create)	`POST /v1/{parent}/specialistPools` Creates a SpecialistPool.
delete (/vertex-ai/docs/reference/rest/v1/projects.locations.specialistPools/delete)	`DELETE /v1/{name}` Deletes a SpecialistPool as well as all Specialists in the pool.
get (/vertex-ai/docs/reference/rest/v1/projects.locations.specialistPools/get)	`GET /v1/{name}` Gets a SpecialistPool.
list (/vertex-ai/docs/reference/rest/v1/projects.locations.specialistPools/list)	`GET /v1/{parent}/specialistPools` Lists SpecialistPools in a Location.
patch (/vertex-ai/docs/reference/rest/v1/projects.locations.specialistPools/patch)	`PATCH /v1/{specialistPool.name}` Updates a SpecialistPool.

REST Resource: v1.projects.locations.studies

(/vertex-ai/docs/reference/rest/v1/projects.locations.studies)

Methods	
create (/vertex-ai/docs/reference/rest/v1/projects.locations.studies/create)	`POST /v1/{parent}/studies` Creates a Study.
delete (/vertex-ai/docs/reference/rest/v1/projects.locations.studies/delete)	`DELETE /v1/{name}` Deletes a Study.
get (/vertex-ai/docs/reference/rest/v1/projects.locations.studies/get)	`GET /v1/{name}` Gets a Study by name.
list (/vertex-ai/docs/reference/rest/v1/projects.locations.studies/list)	`GET /v1/{parent}/studies` Lists all the studies in a region for an associated project.
lookup (/vertex-ai/docs/reference/rest/v1/projects.locations.studies/lookup)	`POST /v1/{parent}/studies:lookup` Looks a study up using the user-defined display_name field instead of the fully qualified resource name.

REST Resource: v1.projects.locations.studies.trials

(/vertex-ai/docs/reference/rest/v1/projects.locations.studies.trials)

Methods

`addTrialMeasurement` (/vertex-ai/docs/reference/rest/v1/projects.locations.studies.trials/addTrialMeasurement)	`POST /v1/{trialName}:addTrialMeasurement` Adds a measurement of the objective metrics to a Trial.
`checkTrialEarlyStoppingState` (/vertex-ai/docs/reference/rest/v1/projects.locations.studies.trials/checkTrialEarlyStoppingState)	`POST /v1/{trialName}:checkTrialEarlyStoppingState` Checks whether a Trial should stop or not.
`complete` (/vertex-ai/docs/reference/rest/v1/projects.locations.studies.trials/complete)	`POST /v1/{name}:complete` Marks a Trial as complete.
`create` (/vertex-ai/docs/reference/rest/v1/projects.locations.studies.trials/create)	`POST /v1/{parent}/trials` Adds a user provided Trial to a Study.
`delete` (/vertex-ai/docs/reference/rest/v1/projects.locations.studies.trials/delete)	`DELETE /v1/{name}` Deletes a Trial.
`get` (/vertex-ai/docs/reference/rest/v1/projects.locations.studies.trials/get)	`GET /v1/{name}` Gets a Trial.
`list` (/vertex-ai/docs/reference/rest/v1/projects.locations.studies.trials/list)	`GET /v1/{parent}/trials` Lists the Trials associated with a Study.

Methods	
listOptimalTrials (/vertex-ai/docs/reference/rest/v1/projects.locations.studies.trials/listOptimalTrials)	`POST /v1/{parent}/trials:listOptimalTrials` Lists the pareto-optimal Trials for multi-objective Study or the optimal Trials for single-objective Study.
stop (/vertex-ai/docs/reference/rest/v1/projects.locations.studies.trials/stop)	`POST /v1/{name}:stop` Stops a Trial.
suggest (/vertex-ai/docs/reference/rest/v1/projects.locations.studies.trials/suggest)	`POST /v1/{parent}/trials:suggest` Adds one or more Trials to a Study, with parameter values suggested by Vertex AI Vizier.

REST Resource: v1.projects.locations.tensorboards

(/vertex-ai/docs/reference/rest/v1/projects.locations.tensorboards)

Methods	
batchRead (/vertex-ai/docs/reference/rest/v1/projects.locations.tensorboards/batchRead)	`GET /v1/{tensorboard}:batchRead` Reads multiple TensorboardTimeSeries' data.
create (/vertex-ai/docs/reference/rest/v1/projects.locations.tensorboards/create)	`POST /v1/{parent}/tensorboards` Creates a Tensorboard.

Methods	
delete (/vertex-ai/docs/reference/rest/v1/projects.locations.tensorboards/delete)	`DELETE /v1/{name}` Deletes a Tensorboard.
get (/vertex-ai/docs/reference/rest/v1/projects.locations.tensorboards/get)	`GET /v1/{name}` Gets a Tensorboard.
list (/vertex-ai/docs/reference/rest/v1/projects.locations.tensorboards/list)	`GET /v1/{parent}/tensorboards` Lists Tensorboards in a Location.
patch (/vertex-ai/docs/reference/rest/v1/projects.locations.tensorboards/patch)	`PATCH /v1/{tensorboard.name}` Updates a Tensorboard.
readSize (/vertex-ai/docs/reference/rest/v1/projects.locations.tensorboards/readSize)	`GET /v1/{tensorboard}:readSize` Returns the storage size for a given TensorBoard instance.
readUsage (/vertex-ai/docs/reference/rest/v1/projects.locations.tensorboards/readUsage)	`GET /v1/{tensorboard}:readUsage` Returns a list of monthly active users for a given TensorBoard instance.

REST Resource: v1.projects.locations.tensorboards.experiments

(/vertex-ai/docs/reference/rest/v1/projects.locations.tensorboards.experiments)

Methods

`batchCreate` (/vertex-ai/docs/reference/rest/v1/projects.locations.tensorboards.experiments/batchCreate)	`POST /v1/{parent}:batchCreate` Batch create TensorboardTimeSeries that belong to a TensorboardExperiment.
`create` (/vertex-ai/docs/reference/rest/v1/projects.locations.tensorboards.experiments/create)	`POST /v1/{parent}/experiments` Creates a TensorboardExperiment.
`delete` (/vertex-ai/docs/reference/rest/v1/projects.locations.tensorboards.experiments/delete)	`DELETE /v1/{name}` Deletes a TensorboardExperiment.
`get` (/vertex-ai/docs/reference/rest/v1/projects.locations.tensorboards.experiments/get)	`GET /v1/{name}` Gets a TensorboardExperiment.
`list` (/vertex-ai/docs/reference/rest/v1/projects.locations.tensorboards.experiments/list)	`GET /v1/{parent}/experiments` Lists TensorboardExperiments in a Location.
`patch` (/vertex-ai/docs/reference/rest/v1/projects.locations.tensorboards.experiments/patch)	`PATCH /v1/{tensorboardExperiment.name}` Updates a TensorboardExperiment.
`write` (/vertex-ai/docs/reference/rest/v1/projects.locations.tensorboards.experiments/write)	`POST /v1/{tensorboardExperiment}:write` Write time series data points of multiple TensorboardTimeSeries in multiple TensorboardRun's.

REST Resource: v1.projects.locations.tensorboards.experiments.runs

(/vertex-ai/docs/reference/rest/v1/projects.locations.tensorboards.experiments.runs)

Methods

batchCreate (/vertex-ai/docs/reference/rest/v1/projects.locations.tensorboards.experiments.runs/batchCreate)	`POST /v1/{parent}/runs:batchCreate` Batch create TensorboardRuns.
create (/vertex-ai/docs/reference/rest/v1/projects.locations.tensorboards.experiments.runs/create)	`POST /v1/{parent}/runs` Creates a TensorboardRun.
delete (/vertex-ai/docs/reference/rest/v1/projects.locations.tensorboards.experiments.runs/delete)	`DELETE /v1/{name}` Deletes a TensorboardRun.
get (/vertex-ai/docs/reference/rest/v1/projects.locations.tensorboards.experiments.runs/get)	`GET /v1/{name}` Gets a TensorboardRun.
list (/vertex-ai/docs/reference/rest/v1/projects.locations.tensorboards.experiments.runs/list)	`GET /v1/{parent}/runs` Lists TensorboardRuns in a Location.

Methods

`patch` (/vertex-ai/docs/reference/rest/v1/projects.locations.tensorboards.experiments.runs/patch)	**`PATCH /v1/{tensorboardRun.name}`** Updates a TensorboardRun.
`write` (/vertex-ai/docs/reference/rest/v1/projects.locations.tensorboards.experiments.runs/write)	**`POST /v1/{tensorboardRun}:write`** Write time series data points into multiple TensorboardTimeSeries under a TensorboardRun.

REST Resource: v1.projects.locations.tensorboards.experiments.runs.timeSeries (/vertex-ai/docs/reference/rest/v1/projects.locations.tensorboards.experiments.runs.timeSeries)

Methods

`create` (/vertex-ai/docs/reference/rest/v1/projects.locations.tensorboards.experiments.runs.timeSeries/create)	**`POST /v1/{parent}/timeSeries`** Creates a TensorboardTimeSeries.
`delete` (/vertex-ai/docs/reference/rest/v1/projects.locations.tensorboards.experiments.runs.timeSeries/delete)	**`DELETE /v1/{name}`** Deletes a TensorboardTimeSeries.

Methods

`exportTensorboardTimeSeries` (/vertex-ai/docs/reference/rest/v1/projects.locations.tensorboards.experiments.runs.timeSeries/exportTensorboardTimeSeries)	`POST /v1/{tensorboardTimeSeries}:exportTensorboardTimeSeries` Exports a TensorboardTimeSeries' data.
`get` (/vertex-ai/docs/reference/rest/v1/projects.locations.tensorboards.experiments.runs.timeSeries/get)	`GET /v1/{name}` Gets a TensorboardTimeSeries.
`list` (/vertex-ai/docs/reference/rest/v1/projects.locations.tensorboards.experiments.runs.timeSeries/list)	`GET /v1/{parent}/timeSeries` Lists TensorboardTimeSeries in a Location.
`patch` (/vertex-ai/docs/reference/rest/v1/projects.locations.tensorboards.experiments.runs.timeSeries/patch)	`PATCH /v1/{tensorboardTimeSeries.name}` Updates a TensorboardTimeSeries.
`read` (/vertex-ai/docs/reference/rest/v1/projects.locations.tensorboards.experiments.runs.timeSeries/read)	`GET /v1/{tensorboardTimeSeries}:read` Reads a TensorboardTimeSeries' data.
`readBlobData` (/vertex-ai/docs/reference/rest/v1/projects.locations.tensorboards.experiments.runs.timeSeries/readBlobData)	`GET /v1/{timeSeries}:readBlobData` Gets bytes of TensorboardBlobs.

REST Resource: v1.projects.locations.trainingPipelines

(/vertex-ai/docs/reference/rest/v1/projects.locations.trainingPipelines)

Methods

`cancel` (/vertex-ai/docs/reference/rest/v1/projects.locations.trainingPipelines/cancel)	**`POST /v1/{name}:cancel`** Cancels a TrainingPipeline.
`create` (/vertex-ai/docs/reference/rest/v1/projects.locations.trainingPipelines/create)	**`POST /v1/{parent}/trainingPipelines`** Creates a TrainingPipeline.
`delete` (/vertex-ai/docs/reference/rest/v1/projects.locations.trainingPipelines/delete)	**`DELETE /v1/{name}`** Deletes a TrainingPipeline.
`get` (/vertex-ai/docs/reference/rest/v1/projects.locations.trainingPipelines/get)	**`GET /v1/{name}`** Gets a TrainingPipeline.
`list` (/vertex-ai/docs/reference/rest/v1/projects.locations.trainingPipelines/list)	**`GET /v1/{parent}/trainingPipelines`** Lists TrainingPipelines in a Location.

REST Resource: v1.projects.locations.tuningJobs

(/vertex-ai/docs/reference/rest/v1/projects.locations.tuningJobs)

Methods

`cancel` (/vertex-ai/docs/reference/rest/v1/projects.locations.tuningJobs/cancel)	`POST /v1/{name}:cancel` Cancels a TuningJob.
`create` (/vertex-ai/docs/reference/rest/v1/projects.locations.tuningJobs/create)	`POST /v1/{parent}/tuningJobs` Creates a TuningJob.
`get` (/vertex-ai/docs/reference/rest/v1/projects.locations.tuningJobs/get)	`GET /v1/{name}` Gets a TuningJob.
`list` (/vertex-ai/docs/reference/rest/v1/projects.locations.tuningJobs/list)	`GET /v1/{parent}/tuningJobs` Lists TuningJobs in a Location.
`rebaseTunedModel` (/vertex-ai/docs/reference/rest/v1/projects.locations.tuningJobs/rebaseTunedModel)	`POST /v1/{parent}/tuningJobs:rebaseTunedModel` Rebase a TunedModel.

REST Resource: v1.publishers.models

(/vertex-ai/docs/reference/rest/v1/publishers.models)

Methods

`get` (/vertex-ai/docs/reference/rest/v1/publishers.models/get)	`GET /v1/{name}` Gets a Model Garden publisher model.

REST Resource: v1beta1.media (/vertex-ai/docs/reference/rest/v1beta1/media)

Methods

upload (/vertex-ai/docs/reference/rest/v1beta1/media/upload)	**POST /v1beta1/{parent}/ragFiles:upload** **POST /upload/v1beta1/{parent}/ragFiles:upload** Upload a file into a RagCorpus.

REST Resource: v1beta1.projects.locations

(/vertex-ai/docs/reference/rest/v1beta1/projects.locations)

Methods

augmentPrompt (/vertex-ai/docs/reference/rest/v1beta1/projects.locations/augmentPrompt)	**POST /v1beta1/{parent}:augmentPrompt** Given an input prompt, it returns augmented prompt from vertex rag store to guide LLM towards generating grounded responses.
corroborateContent (/vertex-ai/docs/reference/rest/v1beta1/projects.locations/corroborateContent)	**POST /v1beta1/{parent}:corroborateContent** Given an input text, it returns a score that evaluates the factuality of the text.
deploy (/vertex-ai/docs/reference/rest/v1beta1/projects.locations/deploy)	**POST /v1beta1/{destination}:deploy** Deploys a model to a new endpoint.
deployPublisherModel (/vertex-ai/docs/reference/rest/v1beta1/projects.locations/deployPublisherModel) **(deprecated)**	**POST /v1beta1/{destination}:deployPublisherModel** Deploys publisher models.

Methods	
`evaluateDataset` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations/evaluateDataset)	`POST /v1beta1/{location}:evaluateDataset` Evaluates a dataset based on a set of given metrics.
`evaluateInstances` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations/evaluateInstances)	`POST /v1beta1/{location}:evaluateInstances` Evaluates instances based on a given metric.
`getRagEngineConfig` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations/getRagEngineConfig)	`GET /v1beta1/{name}` Gets a RagEngineConfig.
`retrieveContexts` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations/retrieveContexts)	`POST /v1beta1/{parent}:retrieveContexts` Retrieves relevant contexts for a query.
`updateRagEngineConfig` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations/updateRagEngineConfig)	`PATCH /v1beta1/{ragEngineConfig.name}` Updates a RagEngineConfig.

REST Resource: v1beta1.projects.locations.batchPredictionJobs

(/vertex-ai/docs/reference/rest/v1beta1/projects.locations.batchPredictionJobs)

Methods

`cancel` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.batchPredictionJobs/cancel)	`POST /v1beta1/{name}:cancel` Cancels a BatchPredictionJob.
`create` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.batchPredictionJobs/create)	`POST /v1beta1/{parent}/batchPredictionJobs` Creates a BatchPredictionJob.
`delete` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.batchPredictionJobs/delete)	`DELETE /v1beta1/{name}` Deletes a BatchPredictionJob.
`get` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.batchPredictionJobs/get)	`GET /v1beta1/{name}` Gets a BatchPredictionJob
`list` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.batchPredictionJobs/list)	`GET /v1beta1/{parent}/batchPredictionJobs` Lists BatchPredictionJobs in a Location.

REST Resource: v1beta1.projects.locations.cachedContents

(/vertex-ai/docs/reference/rest/v1beta1/projects.locations.cachedContents)

Methods	
create (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.cachedContents/create)	`POST /v1beta1/{parent}/cachedContents` Creates cached content, this call will initialize the cached content in the data storage, and users need to pay for the cache data storage.
delete (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.cachedContents/delete)	`DELETE /v1beta1/{name}` Deletes cached content
get (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.cachedContents/get)	`GET /v1beta1/{name}` Gets cached content configurations
list (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.cachedContents/list)	`GET /v1beta1/{parent}/cachedContents` Lists cached contents in a project
patch (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.cachedContents/patch)	`PATCH /v1beta1/{cachedContent.name}` Updates cached content configurations

REST Resource: v1beta1.projects.locations.customJobs

(/vertex-ai/docs/reference/rest/v1beta1/projects.locations.customJobs)

Methods

`cancel` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.customJobs/cancel)	`POST /v1beta1/{name}:cancel` Cancels a CustomJob.
`create` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.customJobs/create)	`POST /v1beta1/{parent}/customJobs` Creates a CustomJob.
`delete` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.customJobs/delete)	`DELETE /v1beta1/{name}` Deletes a CustomJob.
`get` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.customJobs/get)	`GET /v1beta1/{name}` Gets a CustomJob.
`list` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.customJobs/list)	`GET /v1beta1/{parent}/customJobs` Lists CustomJobs in a Location.

REST Resource: v1beta1.projects.locations.datasets

(/vertex-ai/docs/reference/rest/v1beta1/projects.locations.datasets)

Methods	
`assemble` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.datasets/assemble)	`POST /v1beta1/{name}:assemble` Assembles each row of a multimodal dataset and writes the result into a BigQuery table.
`assess` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.datasets/assess)	`POST /v1beta1/{name}:assess` Assesses the state or validity of the dataset with respect to a given use case.
`create` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.datasets/create)	`POST /v1beta1/{parent}/datasets` Creates a Dataset.
`delete` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.datasets/delete)	`DELETE /v1beta1/{name}` Deletes a Dataset.
`export` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.datasets/export)	`POST /v1beta1/{name}:export` Exports data from a Dataset.
`get` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.datasets/get)	`GET /v1beta1/{name}` Gets a Dataset.
`import` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.datasets/import)	`POST /v1beta1/{name}:import` Imports data into a Dataset.

Methods	
list (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.datasets/list)	`GET /v1beta1/{parent}/datasets` Lists Datasets in a Location.
patch (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.datasets/patch)	`PATCH /v1beta1/{dataset.name}` Updates a Dataset.
searchDataItems (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.datasets/searchDataItems)	`GET /v1beta1/{dataset}:searchDataItems` Searches DataItems in a Dataset.

REST Resource: v1beta1.projects.locations.datasets.annotationSpecs

(/vertex-ai/docs/reference/rest/v1beta1/projects.locations.datasets.annotationSpecs)

Methods	
get (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.datasets.annotationSpecs/get)	`GET /v1beta1/{name}` Gets an AnnotationSpec.

REST Resource: v1beta1.projects.locations.datasets.dataItems

(/vertex-ai/docs/reference/rest/v1beta1/projects.locations.datasets.dataItems)

Methods	
list (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.datasets.dataItems/list)	**`GET /v1beta1/{parent}/dataItems`** Lists DataItems in a Dataset.

REST Resource: v1beta1.projects.locations.datasets.dataItems.annotations

(/vertex-ai/docs/reference/rest/v1beta1/projects.locations.datasets.dataItems.annotations)

Methods	
list (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.datasets.dataItems.annotations/list)	**`GET /v1beta1/{parent}/annotations`** Lists Annotations belongs to a dataitem.

REST Resource: v1beta1.projects.locations.datasets.datasetVersions

(/vertex-ai/docs/reference/rest/v1beta1/projects.locations.datasets.datasetVersions)

Methods	
create (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.datasets.datasetVersions/create)	**`POST /v1beta1/{parent}/datasetVersions`** Create a version from a Dataset.

Methods	
`delete` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.datasets.datasetVersions/delete)	`DELETE /v1beta1/{name}` Deletes a Dataset version.
`get` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.datasets.datasetVersions/get)	`GET /v1beta1/{name}` Gets a Dataset version.
`list` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.datasets.datasetVersions/list)	`GET /v1beta1/{parent}/datasetVersions` Lists DatasetVersions in a Dataset.
`patch` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.datasets.datasetVersions/patch)	`PATCH /v1beta1/{datasetVersion.name}` Updates a DatasetVersion.
`restore` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.datasets.datasetVersions/restore)	`GET /v1beta1/{name}:restore` Restores a dataset version.

REST Resource: v1beta1.projects.locations.datasets.savedQueries

(/vertex-ai/docs/reference/rest/v1beta1/projects.locations.datasets.savedQueries)

Methods	
delete (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.datasets.savedQueries/delete)	`DELETE /v1beta1/{name}` Deletes a SavedQuery.
list (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.datasets.savedQueries/list)	`GET /v1beta1/{parent}/savedQueries` Lists SavedQueries in a Dataset.

REST Resource: v1beta1.projects.locations.deploymentResourcePools

(/vertex-ai/docs/reference/rest/v1beta1/projects.locations.deploymentResourcePools)

Methods	
create (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.deploymentResourcePools/create)	`POST /v1beta1/{parent}/deploymentResourcePools` Create a DeploymentResourcePool.
delete (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.deploymentResourcePools/delete)	`DELETE /v1beta1/{name}` Delete a DeploymentResourcePool.

Methods	
get (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.deploymentResourcePools/get)	`GET /v1beta1/{name}` Get a DeploymentResourcePool.
list (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.deploymentResourcePools/list)	`GET /v1beta1/{parent}/deploymentResourcePools` List DeploymentResourcePools in a location.
patch (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.deploymentResourcePools/patch)	`PATCH /v1beta1/{deploymentResourcePool.name}` Update a DeploymentResourcePool.
queryDeployedModels (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.deploymentResourcePools/queryDeployedModels)	`GET /v1beta1/{deploymentResourcePool}:queryDeployedModels` List DeployedModels that have been deployed on this DeploymentResourcePool.

REST Resource: v1beta1.projects.locations.endpoints

(/vertex-ai/docs/reference/rest/v1beta1/projects.locations.endpoints)

Methods	
computeTokens (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.endpoints/computeTokens)	`POST /v1beta1/{endpoint}:computeTokens` Return a list of tokens based on the input text.

Methods

`countTokens` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.endpoints/countTokens)	`POST /v1beta1/{endpoint}:countTokens` Perform a token counting.
`create` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.endpoints/create)	`POST /v1beta1/{parent}/endpoints` Creates an Endpoint.
`delete` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.endpoints/delete)	`DELETE /v1beta1/{name}` Deletes an Endpoint.
`deployModel` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.endpoints/deployModel)	`POST /v1beta1/{endpoint}:deployModel` Deploys a Model into this Endpoint, creating a DeployedModel within it.
`directPredict` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.endpoints/directPredict)	`POST /v1beta1/{endpoint}:directPredict` Perform an unary online prediction request to a gRPC model server for Vertex first-party products and frameworks.
`directRawPredict` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.endpoints/directRawPredict)	`POST /v1beta1/{endpoint}:directRawPredict` Perform an unary online prediction request to a gRPC model server for custom containers.
`explain` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.endpoints/explain)	`POST /v1beta1/{endpoint}:explain` Perform an online explanation.

Methods

`generateContent` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.endpoints/generateContent)	`POST /v1beta1/{model}:generateContent` Generate content with multimodal inputs.
`get` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.endpoints/get)	`GET /v1beta1/{name}` Gets an Endpoint.
`getIamPolicy` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.endpoints/getIamPolicy)	`POST /v1beta1/{resource}:getIamPolicy` Gets the access control policy for a resource.
`list` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.endpoints/list)	`GET /v1beta1/{parent}/endpoints` Lists Endpoints in a Location.
`mutateDeployedModel` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.endpoints/mutateDeployedModel)	`POST /v1beta1/{endpoint}:mutateDeployedModel` Updates an existing deployed model.
`patch` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.endpoints/patch)	`PATCH /v1beta1/{endpoint.name}` Updates an Endpoint.
`predict` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.endpoints/predict)	`POST /v1beta1/{endpoint}:predict` Perform an online prediction.

Methods

rawPredict (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.endpoints/rawPredict)	`POST /v1beta1/{endpoint}:rawPredict` Perform an online prediction with an arbitrary HTTP payload.
serverStreamingPredict (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.endpoints/serverStreamingPredict)	`POST /v1beta1/{endpoint}:serverStreamingPredict` Perform a server-side streaming online prediction request for Vertex LLM streaming.
setIamPolicy (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.endpoints/setIamPolicy)	`POST /v1beta1/{resource}:setIamPolicy` Sets the access control policy on the specified resource.
streamGenerateContent (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.endpoints/streamGenerateContent)	`POST /v1beta1/{model}:streamGenerateContent` Generate content with multimodal inputs with streaming support.
streamRawPredict (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.endpoints/streamRawPredict)	`POST /v1beta1/{endpoint}:streamRawPredict` Perform a streaming online prediction with an arbitrary HTTP payload.
testIamPermissions (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.endpoints/testIamPermissions)	`POST /v1beta1/{resource}:testIamPermissions` Returns permissions that a caller has on the specified resource.

Methods	
undeployModel (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.endpoints/undeployModel)	**POST /v1beta1/{endpoint}:undeployModel** Undeploys a Model from an Endpoint, removing a DeployedModel from it, and freeing all resources it's using.
update (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.endpoints/update)	**POST /v1beta1/{endpoint.name}:update** Updates an Endpoint with a long running operation.

REST Resource: v1beta1.projects.locations.endpoints.chat

(/vertex-ai/docs/reference/rest/v1beta1/projects.locations.endpoints.chat)

Methods	
completions (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.endpoints.chat/completions)	**POST /v1beta1/{endpoint}/chat/completions** Exposes an OpenAI-compatible endpoint for chat completions.

REST Resource: v1beta1.projects.locations.exampleStores

(/vertex-ai/docs/reference/rest/v1beta1/projects.locations.exampleStores)

Methods	
create (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.exampleStores/create)	**POST /v1beta1/{parent}/exampleStores** Create an ExampleStore.

Methods

`delete` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.exampleStores/delete)	`DELETE /v1beta1/{name}` Delete an ExampleStore.
`fetchExamples` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.exampleStores/fetchExamples)	`POST /v1beta1/{exampleStore}:fetchExamples` Get Examples from the Example Store.
`get` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.exampleStores/get)	`GET /v1beta1/{name}` Get an ExampleStore.
`list` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.exampleStores/list)	`GET /v1beta1/{parent}/exampleStores` List ExampleStores in a Location.
`patch` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.exampleStores/patch)	`PATCH /v1beta1/{exampleStore.name}` Update an ExampleStore.
`removeExamples` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.exampleStores/removeExamples)	`POST /v1beta1/{exampleStore}:removeExamples` Remove Examples from the Example Store.

Methods	
searchExamples (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.exampleStores/searchExamples)	`POST /v1beta1/{exampleStore}:searchExamples` Search for similar Examples for given selection criteria.
upsertExamples (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.exampleStores/upsertExamples)	`POST /v1beta1/{exampleStore}:upsertExamples` Create or update Examples in the Example Store.

REST Resource: v1beta1.projects.locations.extensions

(/vertex-ai/docs/reference/rest/v1beta1/projects.locations.extensions)

Methods	
delete (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.extensions/delete)	`DELETE /v1beta1/{name}` Deletes an Extension.
execute (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.extensions/execute)	`POST /v1beta1/{name}:execute` Executes the request against a given extension.
get (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.extensions/get)	`GET /v1beta1/{name}` Gets an Extension.

Methods	
`import` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.extensions/import)	`POST /v1beta1/{parent}/extensions:import` Imports an Extension.
`list` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.extensions/list)	`GET /v1beta1/{parent}/extensions` Lists Extensions in a location.
`patch` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.extensions/patch)	`PATCH /v1beta1/{extension.name}` Updates an Extension.
`query` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.extensions/query)	`POST /v1beta1/{name}:query` Queries an extension with a default controller.

REST Resource: v1beta1.projects.locations.featureGroups

(/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featureGroups)

Methods	
`create` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featureGroups/create)	`POST /v1beta1/{parent}/featureGroups` Creates a new FeatureGroup in a given project and location.

Methods

delete (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featureGroups/delete)	`DELETE /v1beta1/{name}` Deletes a single FeatureGroup.
get (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featureGroups/get)	`GET /v1beta1/{name}` Gets details of a single FeatureGroup.
getIamPolicy (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featureGroups/getIamPolicy)	`POST /v1beta1/{resource}:getIamPolicy` Gets the access control policy for a resource.
list (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featureGroups/list)	`GET /v1beta1/{parent}/featureGroups` Lists FeatureGroups in a given project and location.
patch (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featureGroups/patch)	`PATCH /v1beta1/{featureGroup.name}` Updates the parameters of a single FeatureGroup.
setIamPolicy (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featureGroups/setIamPolicy)	`POST /v1beta1/{resource}:setIamPolicy` Sets the access control policy on the specified resource.

Methods	
testIamPermissions (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featureGroups/testIamPermissions)	**POST /v1beta1/{resource}:testIamPermissions** Returns permissions that a caller has on the specified resource.

REST Resource: v1beta1.projects.locations.featureGroups.featureMonitors

(/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featureGroups.featureMonitors)

Methods	
create (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featureGroups.featureMonitors/create)	**POST /v1beta1/{parent}/featureMonitors** Creates a new FeatureMonitor in a given project, location and FeatureGroup.
delete (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featureGroups.featureMonitors/delete)	**DELETE /v1beta1/{name}** Deletes a single FeatureMonitor.
get (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featureGroups.featureMonitors/get)	**GET /v1beta1/{name}** Gets details of a single FeatureMonitor.

Methods	
list (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featureGroups.featureMonitors/list)	`GET /v1beta1/{parent}/featureMonitors` Lists FeatureGroups in a given project and location.
patch (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featureGroups.featureMonitors/patch)	`PATCH /v1beta1/{featureMonitor.name}` Updates the parameters of a single FeatureMonitor.

REST Resource: v1beta1.projects.locations.featureGroups.featureMonitors.featureMonitorJobs

(/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featureGroups.featureMonitors.featureMonitorJobs)

Methods	
create (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featureGroups.featureMonitors.featureMonitorJobs/create)	`POST /v1beta1/{parent}/featureMonitorJobs` Creates a new feature monitor job.
get (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featureGroups.featureMonitors.featureMonitorJobs/get)	`GET /v1beta1/{name}` Get a feature monitor job.

Methods	
list (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featureGroups.featureMonitors.featureMonitorJobs/list)	`GET /v1beta1/{parent}/featureMonitorJobs` List feature monitor jobs.

REST Resource: v1beta1.projects.locations.featureGroups.features

(/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featureGroups.features)

Methods	
batchCreate (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featureGroups.features/batchCreate)	`POST /v1beta1/{parent}/features:batchCreate` Creates a batch of Features in a given FeatureGroup.
create (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featureGroups.features/create)	`POST /v1beta1/{parent}/features` Creates a new Feature in a given FeatureGroup.
delete (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featureGroups.features/delete)	`DELETE /v1beta1/{name}` Deletes a single Feature.

Methods	
get (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featureGroups.features/get)	`GET /v1beta1/{name}` Gets details of a single Feature.
list (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featureGroups.features/list)	`GET /v1beta1/{parent}/features` Lists Features in a given FeatureGroup.
patch (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featureGroups.features/patch)	`PATCH /v1beta1/{feature.name}` Updates the parameters of a single Feature.

REST Resource: v1beta1.projects.locations.featureOnlineStores

(/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featureOnlineStores)

Methods	
create (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featureOnlineStores/create)	`POST /v1beta1/{parent}/featureOnlineStores` Creates a new FeatureOnlineStore in a given project and location.
delete (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featureOnlineStores/delete)	`DELETE /v1beta1/{name}` Deletes a single FeatureOnlineStore.

Methods

`get` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featureOnlineStores/get)	`GET /v1beta1/{name}` Gets details of a single FeatureOnlineStore.
`getIamPolicy` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featureOnlineStores/getIamPolicy)	`POST /v1beta1/{resource}:getIamPolicy` Gets the access control policy for a resource.
`list` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featureOnlineStores/list)	`GET /v1beta1/{parent}/featureOnlineStores` Lists FeatureOnlineStores in a given project and location.
`patch` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featureOnlineStores/patch)	`PATCH /v1beta1/{featureOnlineStore.name}` Updates the parameters of a single FeatureOnlineStore.
`setIamPolicy` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featureOnlineStores/setIamPolicy)	`POST /v1beta1/{resource}:setIamPolicy` Sets the access control policy on the specified resource.
`testIamPermissions` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featureOnlineStores/testIamPermissions)	`POST /v1beta1/{resource}:testIamPermissions` Returns permissions that a caller has on the specified resource.

REST Resource: v1beta1.projects.locations.featureOnlineStores.featureViews

(/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featureOnlineStores.featureViews)

Methods

`create` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featureOnlineStores.featureViews/create)	`POST /v1beta1/{parent}/featureViews` Creates a new FeatureView in a given FeatureOnlineStore.
`delete` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featureOnlineStores.featureViews/delete)	`DELETE /v1beta1/{name}` Deletes a single FeatureView.
`directWrite` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featureOnlineStores.featureViews/directWrite)	`POST /v1beta1/{featureView}:directWrite` Bidirectional streaming RPC to directly write to feature values in a feature view.
`fetchFeatureValues` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featureOnlineStores.featureViews/fetchFeatureValues)	`POST /v1beta1/{featureView}:fetchFeatureValues` Fetch feature values under a FeatureView.
`get` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featureOnlineStores.featureViews/get)	`GET /v1beta1/{name}` Gets details of a single FeatureView.

Methods

`getIamPolicy` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featureOnlineStores.featureViews/getIamPolicy)	`POST /v1beta1/{resource}:getIamPolicy` Gets the access control policy for a resource.
`list` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featureOnlineStores.featureViews/list)	`GET /v1beta1/{parent}/featureViews` Lists FeatureViews in a given FeatureOnlineStore.
`patch` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featureOnlineStores.featureViews/patch)	`PATCH /v1beta1/{featureView.name}` Updates the parameters of a single FeatureView.
`searchNearestEntities` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featureOnlineStores.featureViews/searchNearestEntities)	`POST /v1beta1/{featureView}:searchNearestEntities` Search the nearest entities under a FeatureView.
`setIamPolicy` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featureOnlineStores.featureViews/setIamPolicy)	`POST /v1beta1/{resource}:setIamPolicy` Sets the access control policy on the specified resource.
`streamingFetchFeatureValues` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featureOnlineStores.featureViews/streamingFetchFeatureValues)	`POST /v1beta1/{featureView}:streamingFetchFeatureValues` Bidirectional streaming RPC to fetch feature values under a FeatureView.

Methods	
`sync` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featureOnlineStores.featureViews/sync)	**`POST /v1beta1/{featureView}:sync`** Triggers on-demand sync for the FeatureView.
`testIamPermissions` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featureOnlineStores.featureViews/testIamPermissions)	**`POST /v1beta1/{resource}:testIamPermissions`** Returns permissions that a caller has on the specified resource.

REST Resource: v1beta1.projects.locations.featureOnlineStores.featureViews.featureViewSyncs

(/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featureOnlineStores.featureViews.featureViewSyncs)

Methods	
`get` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featureOnlineStores.featureViews.featureViewSyncs/get)	**`GET /v1beta1/{name}`** Gets details of a single FeatureViewSync.

Methods	
list (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featureOnlineStores.featureViews.featureViewSyncs/list)	`GET /v1beta1/{parent}/featureViewSyncs` Lists FeatureViewSyncs in a given FeatureView.

REST Resource: v1beta1.projects.locations.featurestores

(/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featurestores)

Methods	
batchReadFeatureValues (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featurestores/batchReadFeatureValues)	`POST /v1beta1/{featurestore}:batchReadFeatureValues` Batch reads Feature values from a Featurestore.
create (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featurestores/create)	`POST /v1beta1/{parent}/featurestores` Creates a new Featurestore in a given project and location.
delete (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featurestores/delete)	`DELETE /v1beta1/{name}` Deletes a single Featurestore.
get (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featurestores/get)	`GET /v1beta1/{name}` Gets details of a single Featurestore.

Methods	
getIamPolicy (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featurestores/getIamPolicy)	`POST /v1beta1/{resource}:getIamPolicy` Gets the access control policy for a resource.
list (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featurestores/list)	`GET /v1beta1/{parent}/featurestores` Lists Featurestores in a given project and location.
patch (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featurestores/patch)	`PATCH /v1beta1/{featurestore.name}` Updates the parameters of a single Featurestore.
searchFeatures (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featurestores/searchFeatures)	`GET /v1beta1/{location}/featurestores:searchFeatures` Searches Features matching a query in a given project.
setIamPolicy (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featurestores/setIamPolicy)	`POST /v1beta1/{resource}:setIamPolicy` Sets the access control policy on the specified resource.
testIamPermissions (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featurestores/testIamPermissions)	`POST /v1beta1/{resource}:testIamPermissions` Returns permissions that a caller has on the specified resource.

REST Resource: v1beta1.projects.locations.featurestores.entityTypes

(/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featurestores.entityTypes)

Methods

`create` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featurestores.entityTypes/create)	**`POST /v1beta1/{parent}/entityTypes`** Creates a new EntityType in a given Featurestore.
`delete` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featurestores.entityTypes/delete)	**`DELETE /v1beta1/{name}`** Deletes a single EntityType.
`deleteFeatureValues` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featurestores.entityTypes/deleteFeatureValues)	**`POST /v1beta1/{entityType}:deleteFeatureValues`** Delete Feature values from Featurestore.
`exportFeatureValues` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featurestores.entityTypes/exportFeatureValues)	**`POST /v1beta1/{entityType}:exportFeatureValues`** Exports Feature values from all the entities of a target EntityType.
`get` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featurestores.entityTypes/get)	**`GET /v1beta1/{name}`** Gets details of a single EntityType.

Methods

`getIamPolicy` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featurestores.entityTypes/getIamPolicy)	`POST /v1beta1/{resource}:getIamPolicy` Gets the access control policy for a resource.
`importFeatureValues` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featurestores.entityTypes/importFeatureValues)	`POST /v1beta1/{entityType}:importFeatureValues` Imports Feature values into the Featurestore from a source storage.
`list` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featurestores.entityTypes/list)	`GET /v1beta1/{parent}/entityTypes` Lists EntityTypes in a given Featurestore.
`patch` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featurestores.entityTypes/patch)	`PATCH /v1beta1/{entityType.name}` Updates the parameters of a single EntityType.
`readFeatureValues` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featurestores.entityTypes/readFeatureValues)	`POST /v1beta1/{entityType}:readFeatureValues` Reads Feature values of a specific entity of an EntityType.
`setIamPolicy` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featurestores.entityTypes/setIamPolicy)	`POST /v1beta1/{resource}:setIamPolicy` Sets the access control policy on the specified resource.

Methods

streamingReadFeatureValues (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featurestores.entityTypes/streamingReadFeatureValues)	`POST /v1beta1/{entityType}:streamingReadFeatureValues` Reads Feature values for multiple entities.
testIamPermissions (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featurestores.entityTypes/testIamPermissions)	`POST /v1beta1/{resource}:testIamPermissions` Returns permissions that a caller has on the specified resource.
writeFeatureValues (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featurestores.entityTypes/writeFeatureValues)	`POST /v1beta1/{entityType}:writeFeatureValues` Writes Feature values of one or more entities of an EntityType.

REST Resource: v1beta1.projects.locations.featurestores.entityTypes.features

(/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featurestores.entityTypes.features)

Methods

batchCreate (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featurestores.entityTypes.features/batchCreate)	`POST /v1beta1/{parent}/features:batchCreate` Creates a batch of Features in a given EntityType.

Methods	
`create` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featurestores.entityTypes.features/create)	`POST /v1beta1/{parent}/features` Creates a new Feature in a given EntityType.
`delete` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featurestores.entityTypes.features/delete)	`DELETE /v1beta1/{name}` Deletes a single Feature.
`get` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featurestores.entityTypes.features/get)	`GET /v1beta1/{name}` Gets details of a single Feature.
`list` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featurestores.entityTypes.features/list)	`GET /v1beta1/{parent}/features` Lists Features in a given EntityType.
`patch` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.featurestores.entityTypes.features/patch)	`PATCH /v1beta1/{feature.name}` Updates the parameters of a single Feature.

REST Resource: v1beta1.projects.locations.hyperparameterTuningJobs

(/vertex-ai/docs/reference/rest/v1beta1/projects.locations.hyperparameterTuningJobs)

Methods	
`cancel` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.hyperparameterTuningJobs/cancel)	`POST /v1beta1/{name}:cancel` Cancels a HyperparameterTuningJob.
`create` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.hyperparameterTuningJobs/create)	`POST /v1beta1/{parent}/hyperparameterTuningJobs` Creates a HyperparameterTuningJob
`delete` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.hyperparameterTuningJobs/delete)	`DELETE /v1beta1/{name}` Deletes a HyperparameterTuningJob.
`get` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.hyperparameterTuningJobs/get)	`GET /v1beta1/{name}` Gets a HyperparameterTuningJob
`list` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.hyperparameterTuningJobs/list)	`GET /v1beta1/{parent}/hyperparameterTuningJobs` Lists HyperparameterTuningJobs in a Location.

REST Resource: v1beta1.projects.locations.indexEndpoints

(/vertex-ai/docs/reference/rest/v1beta1/projects.locations.indexEndpoints)

Methods	
create (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.indexEndpoints/create)	`POST /v1beta1/{parent}/indexEndpoints` Creates an IndexEndpoint.
delete (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.indexEndpoints/delete)	`DELETE /v1beta1/{name}` Deletes an IndexEndpoint.
deployIndex (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.indexEndpoints/deployIndex)	`POST /v1beta1/{indexEndpoint}:deployIndex` Deploys an Index into this IndexEndpoint, creating a DeployedIndex within it.
get (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.indexEndpoints/get)	`GET /v1beta1/{name}` Gets an IndexEndpoint.
list (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.indexEndpoints/list)	`GET /v1beta1/{parent}/indexEndpoints` Lists IndexEndpoints in a Location.
mutateDeployedIndex (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.indexEndpoints/mutateDeployedIndex)	`POST /v1beta1/{indexEndpoint}:mutateDeployedIndex` Update an existing DeployedIndex under an IndexEndpoint.
patch (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.indexEndpoints/patch)	`PATCH /v1beta1/{indexEndpoint.name}` Updates an IndexEndpoint.

Methods	
undeployIndex (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.indexEndpoints/undeployIndex)	**POST /v1beta1/{indexEndpoint}:undeployIndex** Undeploys an Index from an IndexEndpoint, removing a DeployedIndex from it, and freeing all resources it's using.

REST Resource: v1beta1.projects.locations.indexes

(/vertex-ai/docs/reference/rest/v1beta1/projects.locations.indexes)

Methods	
create (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.indexes/create)	**POST /v1beta1/{parent}/indexes** Creates an Index.
delete (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.indexes/delete)	**DELETE /v1beta1/{name}** Deletes an Index.
get (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.indexes/get)	**GET /v1beta1/{name}** Gets an Index.
import (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.indexes/import)	**POST /v1beta1/{name}:import** Imports an Index from an external source (e.g., BigQuery).

Methods

list (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.indexes/list)	**GET /v1beta1/{parent}/indexes** Lists Indexes in a Location.
patch (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.indexes/patch)	**PATCH /v1beta1/{index.name}** Updates an Index.
removeDatapoints (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.indexes/removeDatapoints)	**POST /v1beta1/{index}:removeDatapoints** Remove Datapoints from an Index.
upsertDatapoints (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.indexes/upsertDatapoints)	**POST /v1beta1/{index}:upsertDatapoints** Add/update Datapoints into an Index.

REST Resource: v1beta1.projects.locations.metadataStores

(/vertex-ai/docs/reference/rest/v1beta1/projects.locations.metadataStores)

Methods

create (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.metadataStores/create)	**POST /v1beta1/{parent}/metadataStores** Initializes a MetadataStore, including allocation of resources.

Methods	
delete (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.metadataStores/delete)	`DELETE /v1beta1/{name}` Deletes a single MetadataStore and all its child resources (Artifacts, Executions, and Contexts).
get (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.metadataStores/get)	`GET /v1beta1/{name}` Retrieves a specific MetadataStore.
list (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.metadataStores/list)	`GET /v1beta1/{parent}/metadataStores` Lists MetadataStores for a Location.

REST Resource: v1beta1.projects.locations.metadataStores.artifacts

(/vertex-ai/docs/reference/rest/v1beta1/projects.locations.metadataStores.artifacts)

Methods	
create (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.metadataStores.artifacts/create)	`POST /v1beta1/{parent}/artifacts` Creates an Artifact associated with a MetadataStore.
delete (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.metadataStores.artifacts/delete)	`DELETE /v1beta1/{name}` Deletes an Artifact.

Methods	
`get` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.metadataStores.artifacts/get)	`GET /v1beta1/{name}` Retrieves a specific Artifact.
`list` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.metadataStores.artifacts/list)	`GET /v1beta1/{parent}/artifacts` Lists Artifacts in the MetadataStore.
`patch` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.metadataStores.artifacts/patch)	`PATCH /v1beta1/{artifact.name}` Updates a stored Artifact.
`purge` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.metadataStores.artifacts/purge)	`POST /v1beta1/{parent}/artifacts:purge` Purges Artifacts.
`queryArtifactLineageSubgraph` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.metadataStores.artifacts/queryArtifactLineageSubgraph)	`GET /v1beta1/{artifact}:queryArtifactLineageSubgraph` Retrieves lineage of an Artifact represented through Artifacts and Executions connected by Event edges and returned as a LineageSubgraph.

REST Resource: v1beta1.projects.locations.metadataStores.contexts

(/vertex-ai/docs/reference/rest/v1beta1/projects.locations.metadataStores.contexts)

Methods

addContextArtifactsAndExecutions (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.metadataStores.contexts/addContextArtifactsAndExecutions)	`POST /v1beta1/{context}:addContextArtifactsAndExecutions` Adds a set of Artifacts and Executions to a Context.
addContextChildren (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.metadataStores.contexts/addContextChildren)	`POST /v1beta1/{context}:addContextChildren` Adds a set of Contexts as children to a parent Context.
create (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.metadataStores.contexts/create)	`POST /v1beta1/{parent}/contexts` Creates a Context associated with a MetadataStore.
delete (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.metadataStores.contexts/delete)	`DELETE /v1beta1/{name}` Deletes a stored Context.
get (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.metadataStores.contexts/get)	`GET /v1beta1/{name}` Retrieves a specific Context.
list (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.metadataStores.contexts/list)	`GET /v1beta1/{parent}/contexts` Lists Contexts on the MetadataStore.

Methods	
`patch` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.metadataStores.contexts/patch)	`PATCH /v1beta1/{context.name}` Updates a stored Context.
`purge` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.metadataStores.contexts/purge)	`POST /v1beta1/{parent}/contexts:purge` Purges Contexts.
`queryContextLineageSubgraph` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.metadataStores.contexts/queryContextLineageSubgraph)	`GET /v1beta1/{context}:queryContextLineageSubgraph` Retrieves Artifacts and Executions within the specified Context, connected by Event edges and returned as a LineageSubgraph.
`removeContextChildren` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.metadataStores.contexts/removeContextChildren)	`POST /v1beta1/{context}:removeContextChildren` Remove a set of children contexts from a parent Context.

REST Resource: v1beta1.projects.locations.metadataStores.executions

(/vertex-ai/docs/reference/rest/v1beta1/projects.locations.metadataStores.executions)

Methods

addExecutionEvents (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.metadataStores.executions/addExecutionEvents)	`POST /v1beta1/{execution}:addExecutionEvents` Adds Events to the specified Execution.
create (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.metadataStores.executions/create)	`POST /v1beta1/{parent}/executions` Creates an Execution associated with a MetadataStore.
delete (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.metadataStores.executions/delete)	`DELETE /v1beta1/{name}` Deletes an Execution.
get (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.metadataStores.executions/get)	`GET /v1beta1/{name}` Retrieves a specific Execution.
list (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.metadataStores.executions/list)	`GET /v1beta1/{parent}/executions` Lists Executions in the MetadataStore.
patch (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.metadataStores.executions/patch)	`PATCH /v1beta1/{execution.name}` Updates a stored Execution.

Methods	
`purge` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.metadataStores.executions/purge)	`POST /v1beta1/{parent}/executions:purge` Purges Executions.
`queryExecutionInputsAndOutputs` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.metadataStores.executions/queryExecutionInputsAndOutputs)	`GET /v1beta1/{execution}:queryExecutionInputsAndOutputs` Obtains the set of input and output Artifacts for this Execution, in the form of LineageSubgraph that also contains the Execution and connecting Events.

REST Resource: v1beta1.projects.locations.metadataStores.metadataSchemas

(/vertex-ai/docs/reference/rest/v1beta1/projects.locations.metadataStores.metadataSchemas)

Methods	
`create` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.metadataStores.metadataSchemas/create)	`POST /v1beta1/{parent}/metadataSchemas` Creates a MetadataSchema.
`get` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.metadataStores.metadataSchemas/get)	`GET /v1beta1/{name}` Retrieves a specific MetadataSchema.

Methods	
`list` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.metadataStores.metadataSchemas/list)	`GET /v1beta1/{parent}/metadataSchemas` Lists MetadataSchemas.

REST Resource: v1beta1.projects.locations.migratableResources

(/vertex-ai/docs/reference/rest/v1beta1/projects.locations.migratableResources)

Methods	
`batchMigrate` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.migratableResources/batchMigrate)	`POST /v1beta1/{parent}/migratableResources:batchMigrate` Batch migrates resources from ml.googleapis.com, automl.googleapis.com, and datalabeling.googleapis.com to Vertex AI.
`search` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.migratableResources/search)	`POST /v1beta1/{parent}/migratableResources:search` Searches all of the resources in automl.googleapis.com, datalabeling.googleapis.com and ml.googleapis.com that can be migrated to Vertex AI's given location.

REST Resource: v1beta1.projects.locations.modelDeploymentMonitoringJobs

(/vertex-ai/docs/reference/rest/v1beta1/projects.locations.modelDeploymentMonitoringJobs)

Methods

create (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.modelDeploymentMonitoringJobs/create)	`POST /v1beta1/{parent}/modelDeploymentMonitoringJobs` Creates a ModelDeploymentMonitoringJob.
delete (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.modelDeploymentMonitoringJobs/delete)	`DELETE /v1beta1/{name}` Deletes a ModelDeploymentMonitoringJob.
get (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.modelDeploymentMonitoringJobs/get)	`GET /v1beta1/{name}` Gets a ModelDeploymentMonitoringJob.
list (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.modelDeploymentMonitoringJobs/list)	`GET /v1beta1/{parent}/modelDeploymentMonitoringJobs` Lists ModelDeploymentMonitoringJobs in a Location.
patch (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.modelDeploymentMonitoringJobs/patch)	`PATCH /v1beta1/{modelDeploymentMonitoringJob.name}` Updates a ModelDeploymentMonitoringJob.
pause (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.modelDeploymentMonitoringJobs/pause)	`POST /v1beta1/{name}:pause` Pauses a ModelDeploymentMonitoringJob.

Methods	
resume (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.modelDeploymentMonitoringJobs/resume)	`POST /v1beta1/{name}:resume` Resumes a paused ModelDeploymentMonitoringJob.
searchModelDeploymentMonitoringStatsAnomalies (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.modelDeploymentMonitoringJobs/searchModelDeploymentMonitoringStatsAnomalies)	`POST /v1beta1/{modelDeploymentMonitoringJob}:searchModelDeploymentMonitoringStatsAnomalies` Searches Model Monitoring Statistics generated within a given time window.

REST Resource: v1beta1.projects.locations.modelMonitors

(/vertex-ai/docs/reference/rest/v1beta1/projects.locations.modelMonitors)

Methods	
create (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.modelMonitors/create)	`POST /v1beta1/{parent}/modelMonitors` Creates a ModelMonitor.
delete (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.modelMonitors/delete)	`DELETE /v1beta1/{name}` Deletes a ModelMonitor.
get (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.modelMonitors/get)	`GET /v1beta1/{name}` Gets a ModelMonitor.

Methods	
`list` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.modelMonitors/list)	`GET /v1beta1/{parent}/modelMonitors` Lists ModelMonitors in a Location.
`patch` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.modelMonitors/patch)	`PATCH /v1beta1/{modelMonitor.name}` Updates a ModelMonitor.
`searchModelMonitoringAlerts` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.modelMonitors/searchModelMonitoringAlerts)	`POST /v1beta1/{modelMonitor}:searchModelMonitoringAlerts` Returns the Model Monitoring alerts.
`searchModelMonitoringStats` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.modelMonitors/searchModelMonitoringStats)	`POST /v1beta1/{modelMonitor}:searchModelMonitoringStats` Searches Model Monitoring Stats generated within a given time window.

REST Resource: v1beta1.projects.locations.modelMonitors.modelMonitoringJobs (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.modelMonitors.modelMonitoringJobs)

Methods	
`create` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.modelMonitors.modelMonitoringJobs/create)	`POST /v1beta1/{parent}/modelMonitoringJobs` Creates a ModelMonitoringJob.

Methods	
delete (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.modelMonitors.modelMonitoringJobs/delete)	`DELETE /v1beta1/{name}` Deletes a ModelMonitoringJob.
get (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.modelMonitors.modelMonitoringJobs/get)	`GET /v1beta1/{name}` Gets a ModelMonitoringJob.
list (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.modelMonitors.modelMonitoringJobs/list)	`GET /v1beta1/{parent}/modelMonitoringJobs` Lists ModelMonitoringJobs.

REST Resource: v1beta1.projects.locations.models

(/vertex-ai/docs/reference/rest/v1beta1/projects.locations.models)

Methods	
copy (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.models/copy)	`POST /v1beta1/{parent}/models:copy` Copies an already existing Vertex AI Model into the specified Location.
delete (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.models/delete)	`DELETE /v1beta1/{name}` Deletes a Model.

Methods

`deleteVersion` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.models/deleteVersion)	`DELETE /v1beta1/{name}:deleteVersion` Deletes a Model version.
`export` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.models/export)	`POST /v1beta1/{name}:export` Exports a trained, exportable Model to a location specified by the user.
`get` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.models/get)	`GET /v1beta1/{name}` Gets a Model.
`getIamPolicy` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.models/getIamPolicy)	`POST /v1beta1/{resource}:getIamPolicy` Gets the access control policy for a resource.
`list` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.models/list)	`GET /v1beta1/{parent}/models` Lists Models in a Location.
`listCheckpoints` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.models/listCheckpoints)	`GET /v1beta1/{name}:listCheckpoints` Lists checkpoints of the specified model version.
`listVersions` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.models/listVersions)	`GET /v1beta1/{name}:listVersions` Lists versions of the specified model.

Methods

mergeVersionAliases (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.models/mergeVersionAliases)	`POST /v1beta1/{name}:mergeVersionAliases` Merges a set of aliases for a Model version.
patch (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.models/patch)	`PATCH /v1beta1/{model.name}` Updates a Model.
setIamPolicy (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.models/setIamPolicy)	`POST /v1beta1/{resource}:setIamPolicy` Sets the access control policy on the specified resource.
testIamPermissions (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.models/testIamPermissions)	`POST /v1beta1/{resource}:testIamPermissions` Returns permissions that a caller has on the specified resource.
updateExplanationDataset (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.models/updateExplanationDataset)	`POST /v1beta1/{model}:updateExplanationDataset` Incrementally update the dataset used for an examples model.
upload (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.models/upload)	`POST /v1beta1/{parent}/models:upload` Uploads a Model artifact into Vertex AI.

REST Resource: v1beta1.projects.locations.models.evaluations

(/vertex-ai/docs/reference/rest/v1beta1/projects.locations.models.evaluations)

Methods

`get` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.models.evaluations/get)	**`GET /v1beta1/{name}`** Gets a ModelEvaluation.
`import` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.models.evaluations/import)	**`POST /v1beta1/{parent}/evaluations:import`** Imports an externally generated ModelEvaluation.
`list` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.models.evaluations/list)	**`GET /v1beta1/{parent}/evaluations`** Lists ModelEvaluations in a Model.

REST Resource: v1beta1.projects.locations.models.evaluations.slices

(/vertex-ai/docs/reference/rest/v1beta1/projects.locations.models.evaluations.slices)

Methods

`batchImport` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.models.evaluations.slices/batchImport)	**`POST /v1beta1/{parent}:batchImport`** Imports a list of externally generated EvaluatedAnnotations.

Methods	
`get` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.models.evaluations.slices/get)	**`GET /v1beta1/{name}`** Gets a ModelEvaluationSlice.
`list` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.models.evaluations.slices/list)	**`GET /v1beta1/{parent}/slices`** Lists ModelEvaluationSlices in a ModelEvaluation.

REST Resource: v1beta1.projects.locations.notebookExecutionJobs

(/vertex-ai/docs/reference/rest/v1beta1/projects.locations.notebookExecutionJobs)

Methods	
`create` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.notebookExecutionJobs/create)	**`POST /v1beta1/{parent}/notebookExecutionJobs`** Creates a NotebookExecutionJob.
`delete` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.notebookExecutionJobs/delete)	**`DELETE /v1beta1/{name}`** Deletes a NotebookExecutionJob.

Methods	
get (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.notebookExecutionJobs/get)	`GET /v1beta1/{name}` Gets a NotebookExecutionJob.
list (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.notebookExecutionJobs/list)	`GET /v1beta1/{parent}/notebookExecutionJobs` Lists NotebookExecutionJobs in a Location.

REST Resource: v1beta1.projects.locations.notebookRuntimeTemplates

(/vertex-ai/docs/reference/rest/v1beta1/projects.locations.notebookRuntimeTemplates)

Methods	
create (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.notebookRuntimeTemplates/create)	`POST /v1beta1/{parent}/notebookRuntimeTemplates` Creates a NotebookRuntimeTemplate.
delete (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.notebookRuntimeTemplates/delete)	`DELETE /v1beta1/{name}` Deletes a NotebookRuntimeTemplate.

Methods

`get` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.notebookRuntimeTemplates/get)	`GET /v1beta1/{name}` Gets a NotebookRuntimeTemplate.
`getIamPolicy` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.notebookRuntimeTemplates/getIamPolicy)	`POST /v1beta1/{resource}:getIamPolicy` Gets the access control policy for a resource.
`list` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.notebookRuntimeTemplates/list)	`GET /v1beta1/{parent}/notebookRuntimeTemplates` Lists NotebookRuntimeTemplates in a Location.
`patch` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.notebookRuntimeTemplates/patch)	`PATCH /v1beta1/{notebookRuntimeTemplate.name}` Updates a NotebookRuntimeTemplate.
`setIamPolicy` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.notebookRuntimeTemplates/setIamPolicy)	`POST /v1beta1/{resource}:setIamPolicy` Sets the access control policy on the specified resource.
`testIamPermissions` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.notebookRuntimeTemplates/testIamPermissions)	`POST /v1beta1/{resource}:testIamPermissions` Returns permissions that a caller has on the specified resource.

REST Resource: v1beta1.projects.locations.notebookRuntimes

(/vertex-ai/docs/reference/rest/v1beta1/projects.locations.notebookRuntimes)

Methods

`assign` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.notebookRuntimes/assign)	`POST /v1beta1/{parent}/notebookRuntimes:assign` Assigns a NotebookRuntime to a user for a particular Notebook file.
`delete` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.notebookRuntimes/delete)	`DELETE /v1beta1/{name}` Deletes a NotebookRuntime.
`get` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.notebookRuntimes/get)	`GET /v1beta1/{name}` Gets a NotebookRuntime.
`list` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.notebookRuntimes/list)	`GET /v1beta1/{parent}/notebookRuntimes` Lists NotebookRuntimes in a Location.
`start` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.notebookRuntimes/start)	`POST /v1beta1/{name}:start` Starts a NotebookRuntime.
`stop` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.notebookRuntimes/stop)	`POST /v1beta1/{name}:stop` Stops a NotebookRuntime.

Methods	
upgrade (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.notebookRuntimes/upgrade)	`POST /v1beta1/{name}:upgrade` Upgrades a NotebookRuntime.

REST Resource: v1beta1.projects.locations.operations

(/vertex-ai/docs/reference/rest/v1beta1/projects.locations.operations)

Methods	
cancel (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.operations/cancel)	`POST /v1beta1/{name}:cancel` Starts asynchronous cancellation on a long-running operation.
delete (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.operations/delete)	`DELETE /v1beta1/{name}` Deletes a long-running operation.
get (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.operations/get)	`GET /v1beta1/{name}` Gets the latest state of a long-running operation.
list (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.operations/list)	`GET /v1beta1/{name}/operations` Lists operations that match the specified filter in the request.

Methods	
wait (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.operations/wait)	`POST /v1beta1/{name}:wait` Waits until the specified long-running operation is done or reaches at most a specified timeout, returning the latest state.

REST Resource: v1beta1.projects.locations.persistentResources

(/vertex-ai/docs/reference/rest/v1beta1/projects.locations.persistentResources)

Methods	
create (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.persistentResources/create)	`POST /v1beta1/{parent}/persistentResources` Creates a PersistentResource.
delete (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.persistentResources/delete)	`DELETE /v1beta1/{name}` Deletes a PersistentResource.
get (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.persistentResources/get)	`GET /v1beta1/{name}` Gets a PersistentResource.
list (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.persistentResources/list)	`GET /v1beta1/{parent}/persistentResources` Lists PersistentResources in a Location.

Methods	
patch (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.persistentResources/patch)	`PATCH /v1beta1/{persistentResource.name}` Updates a PersistentResource.
reboot (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.persistentResources/reboot)	`POST /v1beta1/{name}:reboot` Reboots a PersistentResource.

REST Resource: v1beta1.projects.locations.pipelineJobs

(/vertex-ai/docs/reference/rest/v1beta1/projects.locations.pipelineJobs)

Methods	
batchCancel (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.pipelineJobs/batchCancel)	`POST /v1beta1/{parent}/pipelineJobs:batchCancel` Batch cancel PipelineJobs.
batchDelete (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.pipelineJobs/batchDelete)	`POST /v1beta1/{parent}/pipelineJobs:batchDelete` Batch deletes PipelineJobs The Operation is atomic.
cancel (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.pipelineJobs/cancel)	`POST /v1beta1/{name}:cancel` Cancels a PipelineJob.

Methods	
`create` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.pipelineJobs/create)	**`POST /v1beta1/{parent}/pipelineJobs`** Creates a PipelineJob.
`delete` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.pipelineJobs/delete)	**`DELETE /v1beta1/{name}`** Deletes a PipelineJob.
`get` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.pipelineJobs/get)	**`GET /v1beta1/{name}`** Gets a PipelineJob.
`list` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.pipelineJobs/list)	**`GET /v1beta1/{parent}/pipelineJobs`** Lists PipelineJobs in a Location.

REST Resource: v1beta1.projects.locations.publishers.models

(/vertex-ai/docs/reference/rest/v1beta1/projects.locations.publishers.models)

Methods	
`computeTokens` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.publishers.models/computeTokens)	**`POST /v1beta1/{endpoint}:computeTokens`** Return a list of tokens based on the input text.

Methods

`countTokens` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.publishers.models/countTokens)	`POST /v1beta1/{endpoint}:countTokens` Perform a token counting.
`export` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.publishers.models/export)	`POST /v1beta1/{parent}/{name}:export` Exports a publisher model to a user provided Google Cloud Storage bucket.
`fetchPublisherModelConfig` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.publishers.models/fetchPublisherModelConfig)	`GET /v1beta1/{name}:fetchPublisherModelConfig` Fetches the configs of publisher models.
`generateContent` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.publishers.models/generateContent)	`POST /v1beta1/{model}:generateContent` Generate content with multimodal inputs.
`getIamPolicy` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.publishers.models/getIamPolicy)	`POST /v1beta1/{resource}:getIamPolicy` Gets the access control policy for a resource.
`predict` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.publishers.models/predict)	`POST /v1beta1/{endpoint}:predict` Perform an online prediction.

Methods

`rawPredict` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.publishers.models/rawPredict)	`POST /v1beta1/{endpoint}:rawPredict` Perform an online prediction with an arbitrary HTTP payload.
`serverStreamingPredict` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.publishers.models/serverStreamingPredict)	`POST /v1beta1/{endpoint}:serverStreamingPredict` Perform a server-side streaming online prediction request for Vertex LLM streaming.
`setPublisherModelConfig` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.publishers.models/setPublisherModelConfig)	`POST /v1beta1/{name}:setPublisherModelConfig` Sets (creates or updates) configs of publisher models.
`streamGenerateContent` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.publishers.models/streamGenerateContent)	`POST /v1beta1/{model}:streamGenerateContent` Generate content with multimodal inputs with streaming support.
`streamRawPredict` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.publishers.models/streamRawPredict)	`POST /v1beta1/{endpoint}:streamRawPredict` Perform a streaming online prediction with an arbitrary HTTP payload.

REST Resource: v1beta1.projects.locations.ragCorpora

(/vertex-ai/docs/reference/rest/v1beta1/projects.locations.ragCorpora)

Methods

create (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.ragCorpora/create)	`POST /v1beta1/{parent}/ragCorpora` Creates a RagCorpus.
delete (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.ragCorpora/delete)	`DELETE /v1beta1/{name}` Deletes a RagCorpus.
get (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.ragCorpora/get)	`GET /v1beta1/{name}` Gets a RagCorpus.
list (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.ragCorpora/list)	`GET /v1beta1/{parent}/ragCorpora` Lists RagCorpora in a Location.
patch (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.ragCorpora/patch)	`PATCH /v1beta1/{ragCorpus.name}` Updates a RagCorpus.

REST Resource: v1beta1.projects.locations.ragCorpora.ragFiles

(/vertex-ai/docs/reference/rest/v1beta1/projects.locations.ragCorpora.ragFiles)

Methods	
`delete` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.ragCorpora.ragFiles/delete)	`DELETE /v1beta1/{name}` Deletes a RagFile.
`get` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.ragCorpora.ragFiles/get)	`GET /v1beta1/{name}` Gets a RagFile.
`import` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.ragCorpora.ragFiles/import)	`POST /v1beta1/{parent}/ragFiles:import` Import files from Google Cloud Storage or Google Drive into a RagCorpus.
`list` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.ragCorpora.ragFiles/list)	`GET /v1beta1/{parent}/ragFiles` Lists RagFiles in a RagCorpus.

REST Resource: v1beta1.projects.locations.reasoningEngines

(/vertex-ai/docs/reference/rest/v1beta1/projects.locations.reasoningEngines)

Methods	
`create` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.reasoningEngines/create)	`POST /v1beta1/{parent}/reasoningEngines` Creates a reasoning engine.

Methods	
delete (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.reasoningEngines/delete)	`DELETE /v1beta1/{name}` Deletes a reasoning engine.
get (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.reasoningEngines/get)	`GET /v1beta1/{name}` Gets a reasoning engine.
list (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.reasoningEngines/list)	`GET /v1beta1/{parent}/reasoningEngines` Lists reasoning engines in a location.
patch (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.reasoningEngines/patch)	`PATCH /v1beta1/{reasoningEngine.name}` Updates a reasoning engine.
query (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.reasoningEngines/query)	`POST /v1beta1/{name}:query` Queries using a reasoning engine.
streamQuery (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.reasoningEngines/streamQuery)	`POST /v1beta1/{name}:streamQuery` Streams queries using a reasoning engine.

REST Resource: v1beta1.projects.locations.reasoningEngines.sessions

(/vertex-ai/docs/reference/rest/v1beta1/projects.locations.reasoningEngines.sessions)

Methods

`appendEvent` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.reasoningEngines.sessions/appendEvent)	**`POST /v1beta1/{name}:appendEvent`** Appends an event to a given session.
`create` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.reasoningEngines.sessions/create)	**`POST /v1beta1/{parent}/sessions`** Creates a new **`Session`** (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.reasoningEngines.sessions#Session) .
`delete` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.reasoningEngines.sessions/delete)	**`DELETE /v1beta1/{name}`** Deletes details of the specific **`Session`** (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.reasoningEngines.sessions#Session) .
`get` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.reasoningEngines.sessions/get)	**`GET /v1beta1/{name}`** Gets details of the specific **`Session`** (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.reasoningEngines.sessions#Session) .
`list` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.reasoningEngines.sessions/list)	**`GET /v1beta1/{parent}/sessions`** Lists **`Sessions`** (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.reasoningEngines.sessions#Session) in a given reasoning engine.
`patch` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.reasoningEngines.sessions/patch)	**`PATCH /v1beta1/{session.name}`** Updates the specific **`Session`** (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.reasoningEngines.sessions#Session) .

REST Resource: v1beta1.projects.locations.reasoningEngines.sessions.events

(/vertex-ai/docs/reference/rest/v1beta1/projects.locations.reasoningEngines.sessions.events)

Methods	
list (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.reasoningEngines.sessions.events/list)	`GET /v1beta1/{parent}/events` Lists **Events** (/vertex-ai/docs/reference/rest/v1beta1/Event) in a given session.

REST Resource: v1beta1.projects.locations.schedules

(/vertex-ai/docs/reference/rest/v1beta1/projects.locations.schedules)

Methods	
create (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.schedules/create)	`POST /v1beta1/{parent}/schedules` Creates a Schedule.
delete (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.schedules/delete)	`DELETE /v1beta1/{name}` Deletes a Schedule.
get (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.schedules/get)	`GET /v1beta1/{name}` Gets a Schedule.

Methods	
`list` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.schedules/list)	`GET /v1beta1/{parent}/schedules` Lists Schedules in a Location.
`patch` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.schedules/patch)	`PATCH /v1beta1/{schedule.name}` Updates an active or paused Schedule.
`pause` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.schedules/pause)	`POST /v1beta1/{name}:pause` Pauses a Schedule.
`resume` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.schedules/resume)	`POST /v1beta1/{name}:resume` Resumes a paused Schedule to start scheduling new runs.

REST Resource: v1beta1.projects.locations.specialistPools

(/vertex-ai/docs/reference/rest/v1beta1/projects.locations.specialistPools)

Methods	
`create` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.specialistPools/create)	`POST /v1beta1/{parent}/specialistPools` Creates a SpecialistPool.

Methods

delete (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.specialistPools/delete)	`DELETE /v1beta1/{name}` Deletes a SpecialistPool as well as all Specialists in the pool.
get (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.specialistPools/get)	`GET /v1beta1/{name}` Gets a SpecialistPool.
list (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.specialistPools/list)	`GET /v1beta1/{parent}/specialistPools` Lists SpecialistPools in a Location.
patch (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.specialistPools/patch)	`PATCH /v1beta1/{specialistPool.name}` Updates a SpecialistPool.

REST Resource: v1beta1.projects.locations.studies

(/vertex-ai/docs/reference/rest/v1beta1/projects.locations.studies)

Methods

create (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.studies/create)	`POST /v1beta1/{parent}/studies` Creates a Study.

Methods	
`delete` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.studies/delete)	**`DELETE /v1beta1/{name}`** Deletes a Study.
`get` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.studies/get)	**`GET /v1beta1/{name}`** Gets a Study by name.
`list` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.studies/list)	**`GET /v1beta1/{parent}/studies`** Lists all the studies in a region for an associated project.
`lookup` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.studies/lookup)	**`POST /v1beta1/{parent}/studies:lookup`** Looks a study up using the user-defined display_name field instead of the fully qualified resource name.

REST Resource: v1beta1.projects.locations.studies.trials

(/vertex-ai/docs/reference/rest/v1beta1/projects.locations.studies.trials)

Methods	
`addTrialMeasurement` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.studies.trials/addTrialMeasurement)	**`POST /v1beta1/{trialName}:addTrialMeasurement`** Adds a measurement of the objective metrics to a Trial.

Methods

checkTrialEarlyStoppingState (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.studies.trials/checkTrialEarlyStoppingState)	`POST /v1beta1/{trialName}:checkTrialEarlyStoppingState` Checks whether a Trial should stop or not.
complete (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.studies.trials/complete)	`POST /v1beta1/{name}:complete` Marks a Trial as complete.
create (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.studies.trials/create)	`POST /v1beta1/{parent}/trials` Adds a user provided Trial to a Study.
delete (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.studies.trials/delete)	`DELETE /v1beta1/{name}` Deletes a Trial.
get (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.studies.trials/get)	`GET /v1beta1/{name}` Gets a Trial.
list (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.studies.trials/list)	`GET /v1beta1/{parent}/trials` Lists the Trials associated with a Study.
listOptimalTrials (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.studies.trials/listOptimalTrials)	`POST /v1beta1/{parent}/trials:listOptimalTrials` Lists the pareto-optimal Trials for multi-objective Study or the optimal Trials for single-objective Study.

Methods	
stop (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.studies.trials/stop)	`POST /v1beta1/{name}:stop` Stops a Trial.
suggest (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.studies.trials/suggest)	`POST /v1beta1/{parent}/trials:suggest` Adds one or more Trials to a Study, with parameter values suggested by Vertex AI Vizier.

REST Resource: v1beta1.projects.locations.tensorboards

(/vertex-ai/docs/reference/rest/v1beta1/projects.locations.tensorboards)

Methods	
batchRead (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.tensorboards/batchRead)	`GET /v1beta1/{tensorboard}:batchRead` Reads multiple TensorboardTimeSeries' data.
create (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.tensorboards/create)	`POST /v1beta1/{parent}/tensorboards` Creates a Tensorboard.
delete (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.tensorboards/delete)	`DELETE /v1beta1/{name}` Deletes a Tensorboard.

Methods	
get (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.tensorboards/get)	`GET /v1beta1/{name}` Gets a Tensorboard.
list (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.tensorboards/list)	`GET /v1beta1/{parent}/tensorboards` Lists Tensorboards in a Location.
patch (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.tensorboards/patch)	`PATCH /v1beta1/{tensorboard.name}` Updates a Tensorboard.
readSize (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.tensorboards/readSize)	`GET /v1beta1/{tensorboard}:readSize` Returns the storage size for a given TensorBoard instance.
readUsage (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.tensorboards/readUsage)	`GET /v1beta1/{tensorboard}:readUsage` Returns a list of monthly active users for a given TensorBoard instance.

REST Resource: v1beta1.projects.locations.tensorboards.experiments

(/vertex-ai/docs/reference/rest/v1beta1/projects.locations.tensorboards.experiments)

Methods

`batchCreate` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.tensorboards.experiments/batchCreate)	`POST /v1beta1/{parent}:batchCreate` Batch create TensorboardTimeSeries that belong to a TensorboardExperiment.
`create` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.tensorboards.experiments/create)	`POST /v1beta1/{parent}/experiments` Creates a TensorboardExperiment.
`delete` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.tensorboards.experiments/delete)	`DELETE /v1beta1/{name}` Deletes a TensorboardExperiment.
`get` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.tensorboards.experiments/get)	`GET /v1beta1/{name}` Gets a TensorboardExperiment.
`list` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.tensorboards.experiments/list)	`GET /v1beta1/{parent}/experiments` Lists TensorboardExperiments in a Location.
`patch` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.tensorboards.experiments/patch)	`PATCH /v1beta1/{tensorboardExperiment.name}` Updates a TensorboardExperiment.

Methods	
`write` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.tensorboards.experiments/write)	`POST /v1beta1/{tensorboardExperiment}:write` Write time series data points of multiple TensorboardTimeSeries in multiple TensorboardRun's.

REST Resource: v1beta1.projects.locations.tensorboards.experiments.runs

(/vertex-ai/docs/reference/rest/v1beta1/projects.locations.tensorboards.experiments.runs)

Methods	
`batchCreate` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.tensorboards.experiments.runs/batchCreate)	`POST /v1beta1/{parent}/runs:batchCreate` Batch create TensorboardRuns.
`create` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.tensorboards.experiments.runs/create)	`POST /v1beta1/{parent}/runs` Creates a TensorboardRun.
`delete` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.tensorboards.experiments.runs/delete)	`DELETE /v1beta1/{name}` Deletes a TensorboardRun.

Methods	
get (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.tensorboards.experiments.runs/get)	`GET /v1beta1/{name}` Gets a TensorboardRun.
list (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.tensorboards.experiments.runs/list)	`GET /v1beta1/{parent}/runs` Lists TensorboardRuns in a Location.
patch (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.tensorboards.experiments.runs/patch)	`PATCH /v1beta1/{tensorboardRun.name}` Updates a TensorboardRun.
write (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.tensorboards.experiments.runs/write)	`POST /v1beta1/{tensorboardRun}:write` Write time series data points into multiple TensorboardTimeSeries under a TensorboardRun.

REST Resource: v1beta1.projects.locations.tensorboards.experiments.runs.timeSeries

(/vertex-ai/docs/reference/rest/v1beta1/projects.locations.tensorboards.experiments.runs.timeSeries)

Methods

create (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.tensorboards.experiments.runs.timeSeries/create)	`POST /v1beta1/{parent}/timeSeries` Creates a TensorboardTimeSeries.
delete (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.tensorboards.experiments.runs.timeSeries/delete)	`DELETE /v1beta1/{name}` Deletes a TensorboardTimeSeries.
exportTensorboardTimeSeries (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.tensorboards.experiments.runs.timeSeries/exportTensorboardTimeSeries)	`POST /v1beta1/{tensorboardTimeSeries}:exportTensorboardTimeSeries` Exports a TensorboardTimeSeries' data.
get (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.tensorboards.experiments.runs.timeSeries/get)	`GET /v1beta1/{name}` Gets a TensorboardTimeSeries.
list (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.tensorboards.experiments.runs.timeSeries/list)	`GET /v1beta1/{parent}/timeSeries` Lists TensorboardTimeSeries in a Location.
patch (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.tensorboards.experiments.runs.timeSeries/patch)	`PATCH /v1beta1/{tensorboardTimeSeries.name}` Updates a TensorboardTimeSeries.

Methods	
`read` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.tensorboards.experiments.runs.timeSeries/read)	`GET /v1beta1/{tensorboardTimeSeries}:read` Reads a TensorboardTimeSeries' data.
`readBlobData` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.tensorboards.experiments.runs.timeSeries/readBlobData)	`GET /v1beta1/{timeSeries}:readBlobData` Gets bytes of TensorboardBlobs.

REST Resource: v1beta1.projects.locations.trainingPipelines

(/vertex-ai/docs/reference/rest/v1beta1/projects.locations.trainingPipelines)

Methods	
`cancel` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.trainingPipelines/cancel)	`POST /v1beta1/{name}:cancel` Cancels a TrainingPipeline.
`create` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.trainingPipelines/create)	`POST /v1beta1/{parent}/trainingPipelines` Creates a TrainingPipeline.
`delete` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.trainingPipelines/delete)	`DELETE /v1beta1/{name}` Deletes a TrainingPipeline.

Methods	
`get` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.trainingPipelines/get)	`GET /v1beta1/{name}` Gets a TrainingPipeline.
`list` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.trainingPipelines/list)	`GET /v1beta1/{parent}/trainingPipelines` Lists TrainingPipelines in a Location.

REST Resource: v1beta1.projects.locations.tuningJobs

(/vertex-ai/docs/reference/rest/v1beta1/projects.locations.tuningJobs)

Methods	
`cancel` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.tuningJobs/cancel)	`POST /v1beta1/{name}:cancel` Cancels a TuningJob.
`create` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.tuningJobs/create)	`POST /v1beta1/{parent}/tuningJobs` Creates a TuningJob.
`get` (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.tuningJobs/get)	`GET /v1beta1/{name}` Gets a TuningJob.

Methods	
list (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.tuningJobs/list)	`GET /v1beta1/{parent}/tuningJobs` Lists TuningJobs in a Location.
rebaseTunedModel (/vertex-ai/docs/reference/rest/v1beta1/projects.locations.tuningJobs/rebaseTunedModel)	`POST /v1beta1/{parent}/tuningJobs:rebaseTunedModel` Rebase a TunedModel.

REST Resource: v1beta1.projects.modelGardenEula

(/vertex-ai/docs/reference/rest/v1beta1/projects.modelGardenEula)

Methods	
accept (/vertex-ai/docs/reference/rest/v1beta1/projects.modelGardenEula/accept)	`POST /v1beta1/{parent}/modelGardenEula:accept` Accepts the EULA acceptance status of a publisher model.
check (/vertex-ai/docs/reference/rest/v1beta1/projects.modelGardenEula/check)	`POST /v1beta1/{parent}/modelGardenEula:check` Checks the EULA acceptance status of a publisher model.

REST Resource: v1beta1.publishers.models

(/vertex-ai/docs/reference/rest/v1beta1/publishers.models)

Methods

`get` (/vertex-ai/docs/reference/rest/v1beta1/publishers.models/get)	**`GET /v1beta1/{name}`** Gets a Model Garden publisher model.
`list` (/vertex-ai/docs/reference/rest/v1beta1/publishers.models/list)	**`GET /v1beta1/{parent}/models`** Lists publisher models in Model Garden.

Last updated 2025-05-28 UTC.