

Gilhari RESTful JSON Persistence Service APIs

API Route Path Prefix

http://<host>:<port>/gilhari

API	Main URI	Query Parameters (req.query params)	Body Parameters (req.body params)	Comments with example URIs (in bold)
GET	/:version/:class			Query all objects of a particular type (class) http://localhost:80/gilhari/v1/Employee
		filter		Query all qualifying objects of a particular type (class) based on the filter condition http://localhost:80/gilhari/v1/Employee?filter=exempt=true
		deep		Retrieve only the top level objects if deep=false otherwise retrieve related (linked) objects also (default value for 'deep' is true) http://localhost:80/gilhari/v1/Employee?deep=false
		maxObjects		Limit on the number of top-level objects http://localhost:80/gilhari/v1/Employee?maxObjects=10
			operationDetails	With the "operationDetails" tag in the body of a REST request, you may specify some directives (using the "opType" tag) to further refine the operation. Please see some examples below. You may combine multiple directives within one "operationDetails" specification. Please see below a high-level grammar for specifying operationDetails.

			projections	"projections": This opType specifies that the operation should consider only a subset of the attributes of particular types (classes). You may specify "projectionDetails" for multiple types (classes). For example, the following specification means that only attributes "ald" and "aString" of the type (class) "A" should be queried. <pre>{ "operationDetails": [{ "opType": "projections", "projectionsDetails": [{ "type": "A", "attrs": ["ald", "aString"] }] }] }</pre>
--	--	--	-------------	---

			ignore	"ignore": This opType specifies that the operation should ignore (not follow or consider) the referenced branch of the data specified by the type (class) name and the attribute name pair. Useful if the overall operation is of "deep" kind where, by default, the data of the whole object graph is in play but you want to ignore some branch(es). You may specify "references" for multiple branches of mutilpe types (classes). For example, the following specification means that the object branch referenced by the "aCs" attribute in the type (class) "A" should be ignored in this operation. <pre>{ "operationDetails": [{ "opType": "ignore", "references": ["A", "aCs"] }] }</pre>
--	--	--	--------	---

			follow	"follow": This opType specifies that the operation should follow or consider the referenced branch of the data specified by the type (class) name and the attribute name pair. Useful if the overall operation is of "shallow" kind where, by default, the data of only the top level object is in play. You may specify "references" for multiple branches of mutilpe types (classes). For example, the following specification means that the object branch referenced by the "aB" attribute in the type (class) "A" should be followed in this operation. <pre>{ "operationDetails": [{ "opType": "follow", "references": ["A", "aB"] }] }</pre>
--	--	--	--------	--

			filters	"filters": This opType specifies that the operation should consider only that subset of the objects for a specified type (class) which satisfy a given predicate (filter criterion). You may specify multiple predicates for multiple types (classes). For example, the following specification means that the a predicate of "aString > 'aString_0'" should be applied for objects of type "A" and a predicate of "bInt > 100" should be applied for objects of type "B". <pre> { "operationDetails": [{ "opType": "filters", "predicates": [{ "type": "A", "predicate": "aString > 'aString_0'" }, { "type": "B", "predicate": "bInt > 100" }] }] } </pre>
	/:version/:class/getObjectById	filter		Query an object by its primary key (id); filter identifies the primary key http://localhost:80/gilhari/v1/Employee/getObjectById?filter=id=40

		deep		Retrieve only the top level object if deep=false otherwise retrieve related (linked) objects also (default value for 'deep' is true) http://localhost:80/gilhari/v1/Employee/getObjectById?filter=id=40&deep=false
			operationDetails	With the "operationDetails" tag in the body of a REST request, you may specify some directives (using the "opType" tag) to further refine the operation. Please see some examples below. You may combine multiple directives within one "operationDetails" specification. Please see below a high-level grammar for specifying operationDetails.
	/:version/:class/getAggregate	attribute		Query an aggregate value (COUNT, SUM, AVG, MIN, MAX) for an attribute http://localhost:80/gilhari/v1/Employee/getAggregate?attribute=id&aggregateType=COUNT
		aggregateType		COUNT, SUM, AVG, MIN, MAX
		filter		Query the aggregate value over only the qualifying objects based on the filter condition http://localhost:80/gilhari/v1/Employee/getAggregate?attribute=compensation&aggregateType=MAX&filter=exempt=false
	/:version/:class/getObjectById			Retrieves an object with the id specified in the filter clause http://localhost:80/gilhari/v1/Employee/getObjectById?filter=id=16

		deep		Retrieve only the top level objects if deep=false otherwise retrieve related (linked) objects also (default value for 'deep' is true) http://localhost:80/gilhari/v1/Employee/getObjectById?filter=id=16&deep=false
	/:version/:class/startStream	maxObjects		Retrieves the first "maxObjects" number of objects after opening a query in the streaming mode. Object streaming is done inside an implicit transaction. http://localhost:80/gilhari/v1/Employee/startStream?maxObjects=20
		filter		Stream all qualifying objects of a particular type (class) based on the filter condition http://localhost:80/gilhari/v1/Employee/startStream?filter=example&maxObjects=20
		deep		Retrieve only the top level objects if deep=false otherwise retrieve related (linked) objects also (default value for 'deep' is true) http://localhost:80/gilhari/v1/Employee/startStream?filter=example
	/:version/:class/fetchMore	maxObjects		Retrieves the subsequent "maxObjects" number of objects from a previously opened query stream (can be called multiple times until the stream is exhausted) http://localhost:80/gilhari/v1/Employee/fetchMore?maxObjects=10
	/:version/:class/closeStream			Closes the currently open query stream. The implicit transaction is committed. http://localhost:80/gilhari/v1/Employee/closeStream

	/:version/:class/access	attribute	entity	Query (lazy) for a contained (referenced) object indicated by the given 'attribute' name of an entity; The "entity" in the body needs to have only the connecting key value http://localhost:80/gilhari/v1/A/access?attribute=aB
		deep		Retrieve only the top level referenced objects if deep=false otherwise retrieve their related (linked) objects also (default value for 'deep' is true)
	/:version/getObjectModelSummary/now			Get a summary of the information about the underlying object model, such as classes (types), attributes, primary keys, relationships, etc. May be helpful to a client application like an AI LLM to generate proper API calls. http://localhost:80/gilhari/v1/getObjectModelSummary/now
	/:version/health/check			Get the health status of the Gilhari microservice and its uptime. http://localhost:80/gilhari/v1/health/check
	/:version/quit/now			Shuts down the Gilhari microservice; Cleans up database connections and other internal data structures. http://localhost:80/gilhari/v1/quit/now

POST	/:version/:class		entity	Save (Insert) one or more JSON objects; One or more JSON objects that need to be saved are specified using the "entity" tag in the Body of the request. Only the mapped attributes specified for the class in the ORM specification are saved in the database. So, if an original JSON object in the body has more attributes, they will be ignored - the object will essentially be trimmed when saved. http://localhost:80/gilhari/v1/Employee/
			deep	Save only the top level objects if deep=false otherwise save related (linked) objects also (default value for 'deep' is true)
PUT	/:version/:class/updateEntity		entity	Update one or more JSON objects; One or more JSON objects that need to be updated are specified using the "entity" tag in the Body of the request. http://localhost:80/gilhari/v1/Employee/updateEntity
			deep	Update only the top level objects if deep=false otherwise update related (linked) objects also (default value for 'deep' is true)
PATCH	/:version/:class		newValues	Bulk-update all objects of a particular type (class); The new values of the attributes for the JSON objects to be updated must be specified using the "newValue" tag in the Body of the request. The new values are specified in a comma separated name value pair. http://localhost:80/gilhari/v1/Employee
		filter		Bulk-update only the qualifying objects of a particular type (class) based on the filter condition http://localhost:80/gilhari/v1/Employee?filter=id=39

		deep		Bulk-update only the top level objects if deep=false otherwise update related (linked) objects also (default value for 'deep' is true)
			operationDetails	With the "operationDetails" tag in the body of a REST request, you may specify some directives (using the "opType" tag) to further refine the operation. Please see some examples below. You may combine multiple directives within one "operationDetails" specification. Please see below a high-level grammar for specifying operationDetails.
DELETE	/:version/:class/deleteEntity		entity	Delete one or more objects; One or more JSON objects that need to be deleted are specified using the "entity" tag in the Body of the request. Only the primary key attributes of an object need to be specified in the Body. The other attributes may be specified but will be ignored. http://localhost:80/gilhari/v1/Employee/deleteEntity
			deep	Delete only the top level objects if deep=false otherwise delete related (linked) objects also (default value for 'deep' is true)
	/:version/:class			Bulk-delete all objects of a particular type (class) http://localhost:80/gilhari/v1/Employee
		filter		Bulk-delete only the qualifying objects of a particular type (class) based on the filter condition http://localhost:80/gilhari/v1/Employee?filter=exempt=true
		deep		Delete only the top level objects if deep=false otherwise delete related (linked) objects also (default value for 'deep' is true)

			operationDetails	With the "operationDetails" tag in the body of a REST request, you may specify some directives (using the "opType" tag) to further refine the operation. Please see some examples below. You may combine multiple directives within one "operationDetails" specification. Please see below a high-level grammar for specifying operationDetails.
--	--	--	------------------	--

Notes on the filter clause

Please note that the filter clause specifies the predicate (equivalent to the standard WHERE clause in a SQL statement) to qualify the objects to be operated on. So

High-level grammar for specifying operation details in the body of a Gilhari REST request

```
<operationDetails> ::= { "operationDetails": [ comma-separated list of <operationDetail> ] }
```

```
<operationDetail> ::= <projectionsDetail> | <ignoreDetail> | <followDetail> | <filtersDetail>
```

```
<projectionsDetail> ::= { "opType": "projections", "projectionsDetails": [ comma-separated list of <projectionDetail> ] }
```

```
<projectionDetail> ::= { "type": "<typeName>", "attrs": [ <comma-separated list of "<attributeName>"> ] }
```

```
<ignoreDetail> ::= { "opType": "ignore", "references": [ comma-separated list of <reference> ] }
```

```
<reference> ::= "<typeName>", "<referencedAttribName>"
```

```
<followDetail> ::= { "opType": "follow", "references": [ comma-separated list of <reference> ] }
```

```
<filtersDetail> ::= { "opType": "filters", "predicates" : [comma-separated list of <predicate> ] }
```

```
<predicate> ::= { "type": "<typeName>", "predicate": "<predicate string>" }
```

Explanation for the grammar and terms used in specifying operation details

With the "operationDetails" tag in the body of a REST request, you may specify one or more directives (using the "opType" tag) to further refine the operation. You

<projectionsDetail>

projections: This opType specifies that the operation should consider only a subset of the attributes of particular types (classes). You may specify

```
{
  "operationDetails": [
    {
      "opType": "projections",
      "projectionsDetails": [
        {
          "type": "A",
          "attrs": [
            "aId",
            "aString"
          ]
        }
      ]
    }
  ]
}
```

<ignoreDetail>

ignore: This opType specifies that the operation should ignore (not follow or consider) the referenced branch of the data specified by the type (class) name and

```
{
  "operationDetails": [
    {
      "opType": "ignore",
      "references": ["A", "aCs"]
    }
  ]
}
```

<followDetail>

follow: This opType specifies that the operation should follow or consider the referenced branch of the data specified by the type (class) name and the attribute

```
{
  "operationDetails": [
    {
      "opType": "follow",
      "references": ["A", "aB"]
    }
  ]
}
```

<filtersDetail>

filters: This opType specifies that the operation should consider only that subset of the objects for a specified type (class) which satisfy a given predicate (filter)

```
{
  "operationDetails": [
```

```

{
  "opType": "filters",
  "predicates": [
    {
      "type": "A",
      "predicate": "aString > 'aString_0'"
    },
    {
      "type": "B",
      "predicate": "bInt > 100"
    }
  ]
}

```

Some more examples of specifying operation details in the body of a Gilhari REST request

Here is an example of three operation details for "projections", "ignore", and "filters" combined together.

```

{ "operationDetails":
  [
    {
      "opType": "projections",
      "projectionsDetails": [
        {
          "type": "B",
          "attrs": [
            "aId",
            "bId",
            "bInt"
          ]
        }
      ]
    }
  ]
}

```

```
    },
    {
      "type": "A",
      "attrs": [
        "aId",
        "aString"
      ]
    }
  ]
},

{
  "opType": "ignore",
  "references": ["A", "aCs"]
},

{
  "opType": "filters",
  "predicates": [
    {
      "type": "A",
      "predicate": "aString > 'aString_0'"
    },
    {
      "type": "B",
      "predicate": "bInt > 100"
    }
  ]
}
]
```