MapManagerV2: Interfaces

🗺️ Map Manager Interface Documentation

📌 Topics

Topic Name	Message Type	Description
`/map_manager/current_map_info`	`kabam_msgs/msg/MapInfo`	Information about the currently active map
`/map_manager/map_list`	`kabam_msgs/msg/MapList`	List of all available maps
`/map_manager/archived_map_list`	`kabam_msgs/msg/MapList`	List of archived maps
`/map_manager/current_map_waypoints`	`kabam_msgs/msg/MapWaypointList`	Waypoints associated with the current map

🔢 Topic Field Definitions

`/map_manager/current_map_info`

Message Type: `kabam_msgs/msg/MapInfo`

Field	Type	Description
`map_id`	`string`	ID of the map
`map_name`	`string`	Actual map filename (used by the robot)
`map_label`	`string`	Map name shown in UI

updated_at	string	Timestamp of last update
created_at	string	Timestamp of creation
map_manager_state	string	Robot map manager state: `IDLE`, `MAPPING`, `ACTIVE`

Example:

```
1  map_id: "d891fad6-bd99-484c-81b7-db8d1b8b663e"
2  map_name: "new_map2"
3  map_label: "new_map2"
4  updated_at: "2025-05-21 11:28:18"
5  created_at: "2025-05-21 11:28:18"
6  map_manager_state: "ACTIVE"
7
```

`/map_manager/map_list`

Message Type: `kabam_msgs/msg/MapList`

Field	Type	Description
maps	kabam_msgs/MapMetaData[]	List of available map metadata

Each `MapMetaData` includes:

Field	Type	Description
map_name	string	Map filename (used for robot)
map_label	string	UI label for the map
map_id	string	Unique ID of the map

Example:

```
1  maps:
```

```
 2    - map_name: "map_file_name"
 3      map_label: "Map UI Name *"
 4      map_id: "e9bc58b9-dde7-4882-9e59-551f426b0469"
 5    - map_name: "new_map"
 6      map_label: "new_map"
 7      map_id: "22252ee8-98e1-4072-a03e-1613ebf29405"
 8    - map_name: "new_map2"
 9      map_label: "new_map2"
10      map_id: "d891fad6-bd99-484c-81b7-db8d1b8b663e"
11    - map_name: "new_map3"
12      map_label: "new_map3"
13      map_id: "773c7fba-2721-4ba0-92d3-1abbf4e93a6e"
14
```

`/map_manager/archived_map_list`

Same format as `/map_manager/map_list` , but includes only archived maps.

`/map_manager/current_map_waypoints`

Message Type: `kabam_msgs/msg/MapWaypointList`

Field	Type	Description
waypoints	kabam_msgs/MapWaypoint[]	List of waypoints for the current map

Each `MapWaypoint` includes:

Field	Type	Description
name	string	Name of the waypoint
id	string	Unique identifier
frame_id	string	Frame in which pose is defined
pose	geometry_msgs/Pose	Position and orientation
map_id	string	ID of associated map
waypoint_tags	string[]	Tags associated with the waypoint

| | | (`map_entry` , `localisation`) |

Example:

```yaml
1  waypoints:
2    - name: "my_waypoint"
3      id: "7f1500cd-d32d-4e94-a5b0-68641334d3ab"
4      frame_id: ''
5      pose:
6        position:
7          x: 0.0
8          y: 0.0
9          z: 0.0
10       orientation:
11         x: 0.0
12         y: 0.0
13         z: 0.0
14         w: 0.0
15     map_id: "d891fad6-bd99-484c-81b7-db8d1b8b663e"
16     waypoint_tags: []
17
```

🛠️ Services

🗺️ Map Management

Service Name	Service Type	Action
`/map_manager/start_mapping`	`std_srvs/srv/Trigger`	Start mapping process
`/map_manager/stop_mapping`	`std_srvs/srv/Trigger`	Stop mapping process
`/map_manager/save_map`	`kabam_msgs/srv/SaveMap`	Save current map with metadata
`/map_manager/load_map`	`kabam_msgs/srv/LoadMap`	Load a specified map
`/map_manager/delete_map`	`kabam_msgs/srv/UpdateMap`	Delete a selected map

`/map_manager/archive_map`	`kabam_msgs/srv/UpdateMap`	Archive a selected map
`/map_manager/rename_map`	`kabam_msgs/srv/UpdateMap`	Rename an existing map
`/map_manager/update_map_image`	`kabam_msgs/srv/UpdateMap`	Update the map's visual/image content

`/map_manager/start_mapping` **and** `/map_manager/stop_mapping`

Request Fields:

- None

Response Fields:

- `bool success` – Indicates if the operation succeeded.
- `string message` – Informational message.

Example:

```
1  success: True/False
2  message: "Mapping started/stopped" or "Invalid request to start/stop mapping" or "Failed to
   start/stop mapping"
3
```

`/map_manager/save_map`

Request Fields:

- `string map_name` – Name of the map to save.

Response Fields:

- `uint8 result`
  - `0` : RESULT_MAP_SAVED
  - `1` : RESULT_INVALID_REQUEST
  - `2` : RESULT_MAP_NAME_ALREADY_EXISTS
  - `3` : RESULT_UNDEFINED_FAILURE
- `string map_id` – ID of the saved map (only if successful).

Example:

```
1  map_name: "demo_mapping_first_map"
2  ---
3  result: 0
4  map_id: "fbc2e069-adc8-417e-93ca-f5907c8f8f1d"
5
```

`/map_manager/load_map`

Request Fields:

- `string map_id` – ID of the map to load
- `string loc_waypoint_name` – Reserved for future use (optional)

Response Fields:

- `uint8 result`
  - `0` : RESULT_MAP_LOADED
  - `1` : RESULT_MAP_DOES_NOT_EXIST
  - `2` : RESULT_MAP_ALREADY_LOADED
  - `3` : RESULT_INVALID_REQUEST
  - `4` : RESULT_UNDEFINED_FAILURE

Example:

```
1  map_id: "22252ee8-98e1-4072-a03e-1613ebf29405"
2  loc_waypoint_name: ""
3  ---
4  result: 0
5
```

`/map_manager/delete_map` , `/map_manager/archive_map` , `/map_manager/rename_map` , `/map_manager/update_map_image`

Request Fields:

- `string map_id` – Required for all operations
- `sensor_msgs/Image updated_image` – Only for image update
- `string new_map_label` – Only for renaming map
- `bool set_archive` – Set to true to archive, false to unarchive
- `bool delete_map` – Optional, used for delete operation

Response Fields:

- `uint8 result`
  - `0` : RESULT_MAP_UPDATED
  - `1` : RESULT_MAP_DELETED
  - `2` : RESULT_MAP_DOES_NOT_EXIST
  - `3` : RESULT_INVALID_REQUEST
  - `4` : RESULT_UNDEFINED_FAILURE

📍 **Waypoint Management**

Service Name	Service Type	Action
`/map_manager/create_waypoint`	`kabam_msgs/srv/ManageWaypoint`	Create a new waypoint
`/map_manager/delete_waypoint`	`kabam_msgs/srv/ManageWaypoint`	Delete an existing waypoint
`/map_manager/rename_waypoint`	`kabam_msgs/srv/ManageWaypoint`	Rename a waypoint
`/map_manager/reposition_waypoint`	`kabam_msgs/srv/ManageWaypoint`	Change the position of a waypoint
`/map_manager/get_waypoint`	`kabam_msgs/srv/GetWaypoint`	Retrieve detailed info about a waypoint

Services

`/map_manager/create_waypoint`
`/map_manager/delete_waypoint`
`/map_manager/rename_waypoint`

/map_manager/reposition_waypoint

Type: kabam_msgs/srv/ManageWaypoint

Request:

```
1  kabam_msgs/MapWaypoint waypoint_info  # ID can be blank if creating new waypoint
2  bool delete_waypoint
3
```

Response:

```
 1  uint8 RESULT_WAYPOINT_CREATED = 0
 2  uint8 RESULT_MAP_NOT_FOUND = 1
 3  uint8 RESULT_WAYPOINT_NAME_ALREADY_EXISTS_IN_SAME_MAP = 2
 4  uint8 RESULT_WAYPOINT_UPDATED = 3
 5  uint8 RESULT_WAYPOINT_NOT_FOUND = 4
 6  uint8 RESULT_WAYPOINT_DELETED = 5
 7  uint8 RESULT_UNDEFINED_FAILURE = 6
 8
 9  uint8 result
10  kabam_msgs/MapWaypoint waypoint_response  # Includes assigned ID if a waypoint was created
11
```

Message: MapWaypoint

Type: kabam_msgs/msg/MapWaypoint

```
 1  string name             # Provided when creating or renaming a waypoint
 2  string id               # Required when updating an existing waypoint
 3  string frame_id         # Provided when creating or updating waypoint position
 4  geometry_msgs/Pose pose # Required when creating or updating position
 5    geometry_msgs/Point position
 6      float64 x
 7      float64 y
 8      float64 z
 9    geometry_msgs/Quaternion orientation
10      float64 x = 0
11      float64 y = 0
12      float64 z = 0
13      float64 w = 1
14  string map_id # Required when creating a waypoint
15  string[] waypoint_tags  # Provided when creating or updating a waypoint
16
```

Service: GetWaypoint

Type: kabam_msgs/srv/GetWaypoint

Request:

```
1  string map_id
2  string waypoint_id  # Optional; if not given, return all waypoints in map_id
3
```

Response:

```
1  kabam_msgs/MapWaypointList waypoints
2
```

Message: MapWaypointList

Type: `kabam_msgs/msg/MapWaypointList`

```
1  kabam_msgs/MapWaypoint[] waypoints
2
```

🧭 Waypoint Management Guide

This section describes how to interact with the waypoint services exposed by the Map Manager. These services allow users to **create**, **delete**, **rename**, **reposition**, and **retrieve** waypoints on a map.

🔧 General Instructions

- **Creating a waypoint**:
  - Leave the `waypoint_info.id` blank.
  - Provide required fields: `name`, `map_id`, `pose`, `frame_id`, and optionally `waypoint_tags`.
- **Updating (rename or reposition)**:
  - Provide the `id` of the existing waypoint and set the new values in `waypoint_info`.
- **Deleting a waypoint**:
  - Set the `id` of the waypoint to delete.
- **Getting waypoints**:
  - Use `/map_manager/get_waypoint`.
  - Provide `map_id` and optionally `waypoint_id` (leave empty to get all waypoints for that map).

🟦 Result Codes (for ManageWaypoint Response)

- `0` : RESULT_WAYPOINT_CREATED – A new waypoint was successfully created.
- `1` : RESULT_MAP_NOT_FOUND – The specified map was not found.
- `2` : RESULT_WAYPOINT_NAME_ALREADY_EXISTS_IN_SAME_MAP – Name conflict within the same map.
- `3` : RESULT_WAYPOINT_UPDATED – Waypoint was successfully updated.

- **4** : RESULT_WAYPOINT_NOT_FOUND – The specified waypoint could not be found.

- **5** : RESULT_WAYPOINT_DELETED – Waypoint was successfully deleted.

- **6** : RESULT_UNDEFINED_FAILURE – An unknown error occurred.

These codes help you handle the outcome of each request programmatically.