
Shipyard API

Release 0.1.0

Totengeist

Mar 07, 2024

CONTENTS:

1	API Reference	1
1.1	General	1
1.2	Account Management	2
1.3	Ship Management	5
1.4	Save Management	12
1.5	Mod Management	15
1.6	Screenshot Management	17
1.7	Administrative	18
	HTTP Routing Table	23

API REFERENCE

Reference information to help you get started integrating Shipyard functionality into your application via our REST API.

Note:

- This documentation is under construction and incomplete.
 - Bearer tokens listed as “optional” are only optional for clients that support cookies.
-

Table of contents

- *General*
- *Account Management*
- *Ship Management*
- *Save Management*
- *Mod Management*
- *Screenshot Management*
- *Administrative*
 - *User Management*
 - *Tag & Release Management*
 - *Permission & Role Management*

1.1 General

GET /api/v1/version

Get version information for the current Shipyard instance.

This query can be used as a sanity check for new clients.

Example request:

```
POST /api/v1/version HTTP/1.1
Host: example.com
Accept: application/json, text/javascript
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: text/javascript

{
  "app": "Shipyard",
  "version": "0.1.1"
}
```

Status Codes

- 200 OK – no error

1.2 Account Management

POST /api/v1/register

Create a new user account.

Example request:

```
POST /api/v1/register HTTP/1.1
Host: example.com
Accept: application/json, text/javascript
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: text/javascript

{
  "user": {
    "name": "A Standard User",
    "email": "user@example.com",
    "ref": "d5052196e",
    "updated_at": "2024-02-15 19:42:06",
    "created_at": "2024-02-15 19:42:06"
  }
}
```

Query Parameters

- **name** – the username, which can contain spaces and special characters.
- **email** – a valid email address to send the activation link to.
- **password** – a strong password
- **password_confirmation** – the same strong password, to verify it wasn't misspelled

Request Headers

- **Authorization** – optional Bearer token to authenticate

Status Codes

- **200 OK** – no error
- **422 Unprocessable Entity** – the input was invalid or the email was already used

POST /api/v1/activate/(*token*)

Activate a new user account. Unactivated accounts will be unable to *POST /api/v1/login*.

Example request:

```
GET /api/v1/activate/f49c499f7d7b006d66bcc9a5ad11ee5491ecfe95 HTTP/1.1
Host: example.com
Accept: application/json, text/javascript
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "name": "A Standard User",
  "ref": "d5052196e",
  "email": "user@example.com",
  "created_at": "2024-02-15 19:42:06",
  "updated_at": "2024-02-15 20:03:17"
}
```

Parameters

- **token** – the activation token sent to the user's email

Status Codes

- **200 OK** – no error
- **404 Not Found** – the token was not found

POST /api/v1/login

Log in a user.

The `session_id` token should be stored by the client and used as a Bearer token on future requests. Logging the user out should be as simple as deleting the token in the client's storage, but you can also *POST /api/v1/logout* to invalidate the session on the server.

Example request:

```
GET /api/v1/login HTTP/1.1
Host: example.com
Accept: application/json, text/javascript
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: text/javascript

{
  "name": "A Standard User",
  "ref": "d5052196e",
  "email": "user@example.com",
  "created_at": "2024-02-15 19:42:06",
  "updated_at": "2024-02-15 20:03:17",
  "session_id": "1vcf2evvf51t0o9l7n0f38gr24",
  "roles": []
}
```

Query Parameters

- **email** – the user’s email address
- **password** – the user’s password

Status Codes

- **200 OK** – no error
- **401 Unauthorized** – the account doesn’t exist, the password is incorrect, or the account has not been activated

POST /api/v1/logout

Log out the current session.

Example request:

```
POST /api/v1/logout HTTP/1.1
Host: example.com
Accept: application/json, text/javascript
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: text/javascript

{
  "message": "You have been logged out."
}
```

Request Headers

- **Authorization** – optional bearer token to authenticate

Status Codes

- **200 OK** – no error

GET /api/v1/me

Information about the currently logged in user. Use this to check if your session is logged in.

Example request:


```
GET /api/v1/me HTTP/1.1
Host: example.com
Accept: application/json, text/javascript
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "name": "A Standard User",
  "ref": "d5052196e",
  "email": "user@example.com",
  "created_at": "2024-02-15 19:42:06",
  "updated_at": "2024-02-15 20:03:17",
  "session_id": "1vcf2evvf51t0o9l7n0f38gr24",
  "roles": []
}
```

Request Headers

- [Authorization](#) – optional bearer token to authenticate

Status Codes

- [200 OK](#) – no error
- [401 Unauthorized](#) – no user is logged in

1.3 Ship Management

GET /api/v1/ship

A paginated list of ships.

Example request:

```
GET /api/v1/ship HTTP/1.1
Host: example.com
Accept: application/json, text/javascript
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "current_page": 1,
  "data": [
    {
      "ref": "5abe24b6a",
      "title": "Ship 1",
      "description": "An example ship.",
      "downloads": 0,

```

(continues on next page)

(continued from previous page)

```

    "created_at": "2024-02-15 20:56:49",
    "updated_at": "2024-02-15 20:56:49",
    "user": {
      "name": "Test User 1",
      "ref": "a4c836a85"
    },
    "primary_screenshot": []
  },
  {
    "ref": "96a9d0a9e",
    "title": "blanditiis rem nemo",
    "description": "Aut debitis ipsam saepe sed iusto sint. Laboriosam odio_
↳eveniet expedita dolorem et ut. Esse eum molestiae et veritatis ut velit dicta_
↳laudantium.",
    "downloads": 97320,
    "created_at": "2024-02-15 20:58:01",
    "updated_at": "2024-02-15 20:58:01",
    "user": {
      "name": "Test User 1",
      "ref": "a4c836a85"
    },
    "primary_screenshot": [
      {
        "ref": "36a269e9e",
        "description": "Et explicabo perspiciatis libero. Rem illo ea voluptatem_
↳Et vitae aut sapiente perferendis officia repudiandae hic quaerat. Eligendi_
↳consequuntur ut explicabo eveniet aut quo.",
        "created_at": "2024-02-15 20:58:01",
        "updated_at": "2024-02-15 20:58:01"
      }
    ]
  },
  {
    "ref": "713d63cc2",
    "title": "et voluptatem enim",
    "description": "Quo fugit voluptatem soluta voluptate ullam possimus_
↳inventore et. Et voluptates nesciunt vero dolor expedita et excepturi. Odio aut_
↳amet omnis repudiandae.",
    "downloads": 90023,
    "created_at": "2024-02-15 20:58:01",
    "updated_at": "2024-02-15 20:58:01",
    "user": {
      "name": "Alvah Mraz",
      "ref": "362702654"
    },
    "primary_screenshot": [
      {
        "ref": "c9f5d4d03",
        "description": "Vitae nobis aut velit omnis. Dolores similique corporis_
↳quo sunt ut. Nihil ipsum nostrum quo ipsam qui beatae ex. Saepe enim adipisci_
↳dolore eum labore.",
        "created_at": "2024-02-15 20:58:01",

```

(continues on next page)

(continued from previous page)

```

        "updated_at": "2024-02-15 20:58:01"
      }
    ],
    {
      "ref": "9062614bc",
      "title": "at asperiores labore",
      "description": "Deserunt illo qui sunt qui. Fuga fugiat aut rerum alias. Ad_
↪enim suscipit ratione et ea.",
      "downloads": 30458,
      "created_at": "2024-02-15 20:58:02",
      "updated_at": "2024-02-15 20:58:02",
      "user": {
        "name": "Palma Jaskolski",
        "ref": "f64a4aeef"
      },
      "primary_screenshot": [
        {
          "ref": "a049c977d",
          "description": "Omnis amet unde quis dolore inventore. Rerum sint veniam_
↪molestias nihil id asperiores. Sequi dolor libero autem sint corporis similique_
↪provident.",
          "created_at": "2024-02-15 20:58:02",
          "updated_at": "2024-02-15 20:58:02"
        }
      ]
    },
    {
      "ref": "91a537b51",
      "title": "facilis dolore atque",
      "description": "Est qui aut est sunt explicabo quisquam perspiciatis eum_
↪Quod fugit officiis aliquam dolores distinctio maxime ut. Dolores in vitae ut_
↪Neque adipisci qui molestias et quo qui consequatur nihil.",
      "downloads": 55341,
      "created_at": "2024-02-15 20:58:02",
      "updated_at": "2024-02-15 20:58:02",
      "user": {
        "name": "administrator",
        "ref": "c72d235d6"
      },
      "primary_screenshot": [
        {
          "ref": "a2209078f",
          "description": "Nemo iusto aliquid adipisci explicabo explicabo quia non_
↪Nemo sed adipisci non voluptas ab omnis dignissimos. Dolor et qui earum.",
          "created_at": "2024-02-15 20:58:02",
          "updated_at": "2024-02-15 20:58:02"
        }
      ]
    }
  ],
  "first_page_url": "/?page=1",

```

(continues on next page)

(continued from previous page)

```
"from": 1,  
"last_page": 3,  
"last_page_url": "/?page=3",  
"next_page_url": "/?page=2",  
"path": "/",  
"per_page": 5,  
"prev_page_url": null,  
"to": 15,  
"total": 45  
}
```

Query Parameters

- **page** – an optional page number, defaulting to 1
- **per_page** – an optional number of ships per page to return, defaulting to 15 and limited to 100 or less

Status Codes

- **200 OK** – no error

GET /api/v1/ship/(ref)

Information about a specific ship, identified by ref.

Example request:

```
GET /api/v1/ship/5abe24b6a HTTP/1.1  
Host: example.com  
Accept: application/json, text/javascript
```

Example response:

```
HTTP/1.1 200 OK  
Content-Type: application/json  
  
{  
  "ref": "06f5f4644",  
  "title": "odit mollitia enim",  
  "description": "Minima perferendis nam ipsum eveniet alias odio. Blanditiis quos_↵  
↵eius voluptatibus quia non. Qui dignissimos tempore sit at voluptatem debitis_↵  
↵officia. Dolor nulla non blanditiis cumque.",  
  "downloads": 89439,  
  "created_at": "2024-02-15 20:58:05",  
  "updated_at": "2024-02-15 20:58:05",  
  "user": {  
    "name": "Arnulfo Hand",  
    "ref": "4cebc8209"  
  }  
}
```

Parameters

- **ref** – the unique ID of the ship to query

Status Codes

- 200 OK – no error
- 404 Not Found – the ship does not exist

POST /api/v1/ship

Upload a new ship file.

Example request:

```
POST /api/v1/ship HTTP/1.1
Host: example.com
Accept: application/json, text/javascript
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "ref": "19fb7fa39",
  "title": "et ut voluptatem",
  "description": "Ipsa eligendi quia dolorem sit amet illo. Magnam quae voluptas,
  ↳mollitia. Nemo et asperiores adipisci dolor cumque.",
  "downloads": 19519,
  "created_at": "2024-02-15 20:58:03",
  "updated_at": "2024-02-15 20:58:03",
  "user": {
    "name": "Maxie Rice",
    "ref": "01a8a22ec"
  }
}
```

Parameters

- **file** – the ship file being uploaded

Request Headers

- **Authorization** – optional bearer token to authenticate

Status Codes

- 200 OK – no error
- 401 Unauthorized – not logged in or banned from uploading

GET /api/v1/ship/{ref}/download

Download a ship file.

Example request:

```
GET /api/v1/ship/5abe24b6a/download HTTP/1.1
Host: example.com
Accept: application/json, text/javascript
```

Parameters

- **ref** – the unique ID of the ship to query

Status Codes

- 200 OK – no error
- 404 Not Found – the ship does not exist

GET /api/v1/ship/(ref)/screenshots

A list of screenshots for a ship.

Example request:

```
GET /api/v1/ship/8ef20cff9/screenshots HTTP/1.1
Host: example.com
Accept: application/json, text/javascript
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

[
  {
    "ref": "6ddc196f9",
    "description": "Similique amet nulla sed rem. Sunt quia voluptatem ut_
↪consequuntur commodi. Cupiditate ipsum dicta magni est labore recusandae.",
    "created_at": "2024-02-17 00:22:52",
    "updated_at": "2024-02-17 00:22:52",
    "primary": 0
  },
  {
    "ref": "8b198db69",
    "description": "Tempore maiores ut repellendus iusto modi omnis non. Sapiente_
↪maxime assumenda dignissimos enim perferendis earum dolore. Deserunt beatae_
↪ducimus praesentium ipsum ut placeat error.",
    "created_at": "2024-02-17 00:22:52",
    "updated_at": "2024-02-17 00:22:52",
    "primary": 1
  }
]
```

Parameters

- **ref** – the unique ID of the ship to get screenshots for

Status Codes

- 200 OK – no error
- 404 Not Found – the ship does not exist

POST /api/v1/ship/(ref)

Edit an existing ship.

Parameters

- **ref** – the unique ID of the ship to edit

Request Headers

- **Authorization** – optional bearer token to authenticate

Status Codes

- **200 OK** – no error
- **404 Not Found** – the ship does not exist

POST /api/v1/ship/(ref)/upgrade

Replace an existing ship with a new version.

Older versions will still be accessible. This allows users to upgrade ships to support new features added to The Last Starship. The `ref` will continue to point to the older version, but it's page will display a notice that a newer version is available.

Example request:

```
POST /api/v1/ship/19fb7fa39/upgrade HTTP/1.1
Host: example.com
Accept: application/json, text/javascript
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "ref": "8ef20cff9",
  "title": "et ut voluptatem",
  "description": "Ipsa eligendi quia dolore sit amet illo. Magnam quae voluptas. mollitia. Nemo et asperiores adipisci dolor cumque.",
  "downloads": 19519,
  "created_at": "2024-02-15 20:58:03",
  "updated_at": "2024-02-15 20:58:03",
  "user": {
    "name": "Maxie Rice",
    "ref": "01a8a22ec"
  }
}
```

Parameters

- **file** – the ship file being uploaded
- **ref** – the unique ID of the ship to be upgraded

Request Headers

- **Authorization** – optional bearer token to authenticate

Status Codes

- **200 OK** – no error
- **401 Unauthorized** – not logged in or not the owner of the ship
- **404 Not Found** – the ship does not exist

POST /api/v1/ship/(ref)/screenshots

Adds one or more screenshots to an existing ship.

Parameters

- **ref** – the unique ID of the ship to add screenshots to

Request Headers

- **Authorization** – optional bearer token to authenticate

Status Codes

- **200 OK** – no error
- **401 Unauthorized** – not logged in or not the owner of the ship
- **404 Not Found** – the ship does not exist

DELETE /api/v1/ship/(ref)

Delete an existing ship.

Parameters

- **ref** – the unique ID of the ship to delete

Request Headers

- **Authorization** – optional bearer token to authenticate

Status Codes

- **200 OK** – no error
- **401 Unauthorized** – not logged in or not the owner of the ship
- **404 Not Found** – the ship does not exist

1.4 Save Management

GET /api/v1/save

A paginated list of saves.

Example request:

```
GET /api/v1/save HTTP/1.1
Host: example.com
Accept: application/json, text/javascript
```

Example response:

Query Parameters

- **page** – an optional page number, defaulting to 1
- **per_page** – an optional number of saves per page to return, defaulting to 15 and limited to 100 or less

Status Codes

- **200 OK** – no error

GET /api/v1/save/(ref)

Information about a specific save, identified by `ref`.

Example request:

Example response:

Parameters

- **ref** – the unique ID of the save to query

Status Codes

- **200 OK** – no error
- **404 Not Found** – the save does not exist

POST /api/v1/save

Upload a new save file.

Example request:

Example response:

Parameters

- **file** – the save file being uploaded

Request Headers

- **Authorization** – optional bearer token to authenticate

Status Codes

- **200 OK** – no error
- **401 Unauthorized** – not logged in or banned from uploading

GET /api/v1/save/(ref)/download

Download a save file.

Example request:

```
GET /api/v1/ship/5abe24b6a/download HTTP/1.1
Host: example.com
Accept: application/json, text/javascript
```

Parameters

- **ref** – the unique ID of the ship to query

Status Codes

- **200 OK** – no error
- **404 Not Found** – the save does not exist

GET /api/v1/save/(ref)/screenshots

A list of screenshots for a save.

Example request:

Example response:

Parameters

- **ref** – the unique ID of the save to get screenshots for

Status Codes

- **200 OK** – no error
- **404 Not Found** – the save does not exist

POST /api/v1/save/(ref)

Edit an existing save.

Parameters

- **ref** – the unique ID of the save to edit

Request Headers

- **Authorization** – optional bearer token to authenticate

Status Codes

- **200 OK** – no error
- **404 Not Found** – the save does not exist

POST /api/v1/save/(ref)/upgrade

Replace an existing save with a new version.

Older versions will still be accessible. This allows users to upgrade saves to support new features added to The Last Starship. The **ref** will continue to point to the older version, but it's page will display a notice that a newer version is available.

Example request:

Example response:

Parameters

- **file** – the save file being uploaded
- **ref** – the unique ID of the save to be upgraded

Request Headers

- **Authorization** – optional bearer token to authenticate

Status Codes

- **200 OK** – no error
- **401 Unauthorized** – not logged in or not the owner of the save
- **404 Not Found** – the save does not exist

POST /api/v1/save/(ref)/screenshots

Adds one or more screenshots to an existing save.

Parameters

- **ref** – the unique ID of the save to add screenshots to

Request Headers

- **Authorization** – optional bearer token to authenticate

Status Codes

- **200 OK** – no error

- 401 **Unauthorized** – not logged in or not the owner of the save
- 404 **Not Found** – the save does not exist

DELETE /api/v1/save/(ref)

Delete an existing save.

Parameters

- **ref** – the unique ID of the save to delete

Request Headers

- **Authorization** – optional bearer token to authenticate

Status Codes

- 200 **OK** – no error
- 401 **Unauthorized** – not logged in or not the owner of the save
- 404 **Not Found** – the save does not exist

1.5 Mod Management

GET /api/v1/modification

A paginated list of mods.

Query Parameters

- **page** – an optional page number, defaulting to 1
- **per_page** – an optional number of ships per page to return, defaulting to 15 and limited to 100 or less

Status Codes

- 200 **OK** – no error

GET /api/v1/modification/(ref)

Information about a specific mod, identified by **ref**.

Parameters

- **ref** – the unique ID of the mod to query

Status Codes

- 200 **OK** – no error
- 404 **Not Found** – the mod does not exist

POST /api/v1/modification

Upload a new mod file.

Parameters

- **file** – the mod file being uploaded

Request Headers

- **Authorization** – optional bearer token to authenticate

Status Codes

- 200 OK – no error
- 401 Unauthorized – not logged in or banned from uploading

GET /api/v1/modification/(ref)/download

Download a mod file.

Parameters

- **ref** – the unique ID of the mod to query

Status Codes

- 200 OK – no error
- 404 Not Found – the mod does not exist

GET /api/v1/modification/(ref)/screenshots

A list of screenshots for a mod.

Parameters

- **ref** – the unique ID of the mod to get screenshots for

Status Codes

- 200 OK – no error
- 404 Not Found – the mod does not exist

POST /api/v1/modification/(ref)

Edit an existing mod.

Parameters

- **ref** – the unique ID of the mod to edit

Request Headers

- **Authorization** – optional bearer token to authenticate

Status Codes

- 200 OK – no error
- 404 Not Found – the mod does not exist

POST /api/v1/modification/(ref)/upgrade

Replace an existing mod with a new version.

Older versions will still be accessible. This allows users to upgrade mods to support new features added to The Last Starship. The **ref** will continue to point to the older version, but it's page will display a notice that a newer version is available.

Parameters

- **file** – the mod file being uploaded
- **ref** – the unique ID of the mod to be upgraded

Request Headers

- **Authorization** – optional bearer token to authenticate

Status Codes

- 200 OK – no error

- 401 Unauthorized – not logged in or not the owner of the mod
- 404 Not Found – the mod does not exist

POST /api/v1/modification/(ref)/screenshots

Adds one or more screenshots to an existing modification.

Parameters

- **ref** – the unique ID of the ship to add screenshots to

Request Headers

- **Authorization** – optional bearer token to authenticate

Status Codes

- 200 OK – no error
- 401 Unauthorized – not logged in or not the owner of the mod
- 404 Not Found – the mod does not exist

DELETE /api/v1/modification/(ref)

Delete an existing mod.

Parameters

- **ref** – the unique ID of the mod to delete

Request Headers

- **Authorization** – optional bearer token to authenticate

Status Codes

- 200 OK – no error
- 401 Unauthorized – not logged in or not the owner of the mod
- 404 Not Found – the mod does not exist

1.6 Screenshot Management

GET /api/v1/screenshot/(ref)

Information about a specific screenshot

Parameters

- **ref** – the unique ID of the screenshot to edit

Status Codes

- 200 OK – no error
- 404 Not Found – the screenshot (or the item it belongs to) does not exist

POST /api/v1/screenshot/(ref)

Edit an existing screenshot.

Parameters

- **ref** – the unique ID of the screenshot to edit

Request Headers

- [Authorization](#) – optional bearer token to authenticate

Status Codes

- [200 OK](#) – no error
- [404 Not Found](#) – the screenshot does not exist

DELETE `/api/v1/screenshot/(ref)`

Deletes an existing screenshot.

Parameters

- **ref** – the unique ID of the screenshot to delete

Request Headers

- [Authorization](#) – optional bearer token to authenticate

Status Codes

- [200 OK](#) – no error
- [401 Unauthorized](#) – not logged in or not the owner of the screenshot
- [404 Not Found](#) – the screenshot does not exist

1.7 Administrative

1.7.1 User Management

POST `/api/v1/user/(user_id)`

Edit an existing user.

Parameters

- **ref** – the unique ID of the user to edit

Request Headers

- [Authorization](#) – optional bearer token to authenticate

Status Codes

- [200 OK](#) – no error
- [404 Not Found](#) – the user does not exist

DELETE `/api/v1/user/(user_id)`

Delete an existing user.

Parameters

- **ref** – the unique ID of the user to delete

Request Headers

- [Authorization](#) – optional bearer token to authenticate

Status Codes

- [200 OK](#) – no error
- [401 Unauthorized](#) – not logged in or not the owner of the user

- 404 Not Found – the user does not exist

1.7.2 Tag & Release Management

GET /api/v1/tag

A paginated list of tags.

Query Parameters

- **page** – an optional page number, defaulting to 1
- **per_page** – an optional number of ships per page to return, defaulting to 15 and limited to 100 or less

Request Headers

- **Authorization** – optional bearer token to authenticate

Status Codes

- 200 OK – no error

GET /api/v1/tag/{slug}

Information about a specific tag.

Status Codes

- 200 OK – no error
- 404 Not Found – the tag does not exist

Request Headers

- **Authorization** – optional bearer token to authenticate

POST /api/v1/tag

POST /api/v1/tag/{slug}

DELETE /api/v1/tag/{slug}

Delete an existing tag.

Request Headers

- **Authorization** – optional bearer token to authenticate

GET /api/v1/release

A paginated list of releases.

Query Parameters

- **page** – an optional page number, defaulting to 1
- **per_page** – an optional number of ships per page to return, defaulting to 15 and limited to 100 or less

Request Headers

- **Authorization** – optional bearer token to authenticate

Status Codes

- 200 OK – no error

GET /api/v1/release/(slug)

Information about a specific release.

Status Codes

- 200 OK – no error
- 404 Not Found – the release does not exist

Request Headers

- Authorization – optional bearer token to authenticate

POST /api/v1/release

POST /api/v1/release/(slug)

DELETE /api/v1/release/(slug)

Delete an existing release.

Request Headers

- Authorization – optional bearer token to authenticate

1.7.3 Permission & Role Management

GET /api/v1/permission

A paginated list of permissions.

Query Parameters

- **page** – an optional page number, defaulting to 1
- **per_page** – an optional number of ships per page to return, defaulting to 15 and limited to 100 or less

Request Headers

- Authorization – optional bearer token to authenticate

Status Codes

- 200 OK – no error

GET /api/v1/permission/(slug)

Information about a specific permission

Status Codes

- 200 OK – no error
- 404 Not Found – the permission does not exist

Request Headers

- Authorization – optional bearer token to authenticate

POST /api/v1/permission

POST /api/v1/permission/(slug)

DELETE /api/v1/permission/(slug)

Delete an existing permission.

Request Headers

- [Authorization](#) – optional bearer token to authenticate

GET /api/v1/role

A paginated list of roles.

Query Parameters

- **page** – an optional page number, defaulting to 1
- **per_page** – an optional number of ships per page to return, defaulting to 15 and limited to 100 or less

Request Headers

- [Authorization](#) – optional bearer token to authenticate

Status Codes

- [200 OK](#) – no error

GET /api/v1/role/(slug)

Information about a specific role

Status Codes

- [200 OK](#) – no error
- [404 Not Found](#) – the role does not exist

Request Headers

- [Authorization](#) – optional bearer token to authenticate

POST /api/v1/role**POST /api/v1/role/(slug)****DELETE /api/v1/role/(slug)**

Delete an existing role.

Request Headers

- [Authorization](#) – optional bearer token to authenticate

Shipyard makes it easy to share saves, ships, and modifications for [Introversion's The Last Starship](#).

Please see the table of contents on the left for the most relevant information.

HTTP ROUTING TABLE

/api

GET /api/v1/me, 4	POST /api/v1/save/(ref)/screenshots, 14
GET /api/v1/modification, 15	POST /api/v1/save/(ref)/upgrade, 14
GET /api/v1/modification/(ref), 15	POST /api/v1/screenshot/(ref), 17
GET /api/v1/modification/(ref)/download, 16	POST /api/v1/ship, 9
GET /api/v1/modification/(ref)/screenshots, 16	POST /api/v1/ship/(ref), 10
GET /api/v1/permission, 20	POST /api/v1/ship/(ref)/screenshots, 11
GET /api/v1/permission/(slug), 20	POST /api/v1/ship/(ref)/upgrade, 11
GET /api/v1/release, 19	POST /api/v1/tag, 19
GET /api/v1/release/(slug), 19	POST /api/v1/tag/(slug), 19
GET /api/v1/role, 21	POST /api/v1/user/(user_id), 18
GET /api/v1/role/(slug), 21	DELETE /api/v1/modification/(ref), 17
GET /api/v1/save, 12	DELETE /api/v1/permission/(slug), 20
GET /api/v1/save/(ref), 12	DELETE /api/v1/release/(slug), 20
GET /api/v1/save/(ref)/download, 13	DELETE /api/v1/role/(slug), 21
GET /api/v1/save/(ref)/screenshots, 13	DELETE /api/v1/save/(ref), 15
GET /api/v1/screenshot/(ref), 17	DELETE /api/v1/screenshot/(ref), 18
GET /api/v1/ship, 5	DELETE /api/v1/ship/(ref), 12
GET /api/v1/ship/(ref), 8	DELETE /api/v1/tag/(slug), 19
GET /api/v1/ship/(ref)/download, 9	DELETE /api/v1/user/(user_id), 18
GET /api/v1/ship/(ref)/screenshots, 10	
GET /api/v1/tag, 19	
GET /api/v1/tag/(slug), 19	
GET /api/v1/version, 1	
POST /api/v1/activate/(token), 3	
POST /api/v1/login, 3	
POST /api/v1/logout, 4	
POST /api/v1/modification, 15	
POST /api/v1/modification/(ref), 16	
POST /api/v1/modification/(ref)/screenshots, 17	
POST /api/v1/modification/(ref)/upgrade, 16	
POST /api/v1/permission, 20	
POST /api/v1/permission/(slug), 20	
POST /api/v1/register, 2	
POST /api/v1/release, 20	
POST /api/v1/release/(slug), 20	
POST /api/v1/role, 21	
POST /api/v1/role/(slug), 21	
POST /api/v1/save, 13	
POST /api/v1/save/(ref), 14	