Swagger Petstore - OpenAPI 3.0

API Reference Documentation

Contents

Section	
- o `GET /store/order/{orderId}` (getOrderById)	
- o `DELETE /store/order/{orderId}` (deleteOrder)	
- user	
- o `POST /user` (createUser)	
- o `POST /user/createWithList` (createUsersWithListInput)	
- o `GET /user/login` (loginUser)	
- o `GET /user/logout` (logoutUser)	
- o `GET /user/{username}` (getUserByName)	
- o `PUT /user/{username}` (updateUser)	
- o `DELETE /user/{username}` (deleteUser)	

General Information

Swagger Petstore - OpenAPI 3.0 v1.0.11

This is a sample Pet Store Server based on the OpenAPI 3.0 specification. You can find out more about Swagger at https://swagger.io. In the third iteration of the pet store, we've switched to the design first approach!

Terms of Service

Usage of this API is subject to its Terms of Service which you may access at http://swagger.io/terms/.

Contact

- Email: apiteam@swagger.io

License

This API is under an Apache 2.0 license.

Read the full license here: http://www.apache.org/licenses/LICENSE-2.0.html

Servers

The following servers may be utilized for this API:

URL	Description	Variables
https://petstore3.swagger.io/api/v3	-	-

Security

The following security schemes are used by this API for authentication and authorization. These must be taken into account in every request unless otherwise stated.

API Key Authentication

This API is secured through API Keys which you must obtain separately.

Api Key

To authenticate requests, include `api_key` as a request header.

```
GET /example
Host: example.com
api_key: SAMPLE-API-KEY-a29@102&3djSF9
```

OAuth 2.0 Authorization

This API is secured using the OAuth 2.0 authorization framework.

Implicit Flow

This API supports the implicit flow.

- Authorization URL: https://petstore3.swagger.io/oauth/authorize
- Scopes:

- `write:pets` : modify pets in your account
- `read:pets` : read your pets

Operations

- [pet](#)
- [store](#)
- [user](#)

pet

Everything about your Pets

Read more about this category here: http://swagger.io

- [PUT /pet (updatePet)](#)
- [POST /pet (addPet)](#)
- [GET /pet/findByStatus (findPetsByStatus)](#)
- [GET /pet/findByTags (findPetsByTags)](#)
- [GET /pet/{petId} (getPetById)](#)
- [POST /pet/{petId} (updatePetWithForm)](#)
- [DELETE /pet/{petId} (deletePet)](#)
- [POST /pet/{petId}/uploadImage (uploadFile)](#)

`PUT /pet` (updatePet)

Update an existing pet

Update an existing pet by Id

Required Scopes

This endpoint requires the following scopes:

- `write:pets`
- `read:pets`

Path Parameters

No Path Parameters.

Query Parameters

No Query Parameters.

Request Headers

No Specific Request Headers

Cookies

No Cookies.

Request Body

Required.

Update an existent pet in the store

Media Type: `application/json`

Key	Type	Required	Example	Notes
id	integer	No	10	- Format: int64
name	string	Yes	doggie	-

Key	Type	Required	Example	Notes
category	object	No	-	-
category.id	integer	No	1	• Format: int64
category.name	string	No	Dogs	-
photoUrls	array	Yes	-	-
photoUrls[*]	string	No	-	-
tags	array	No	-	-
tags[*]	object	No	-	-
tags[*].id	integer	No	-	• Format: int64
tags[*].name	string	No	-	-
status	string	No	-	pet status in the store • Allowed Values: available, pending, sold

Media Type: `application/xml`

Key	Type	Required	Example	Notes
id	integer	No	10	• Format: int64
name	string	Yes	doggie	-
category	object	No	-	-
category.id	integer	No	1	• Format: int64
category.name	string	No	Dogs	-
photoUrls	array	Yes	-	-
photoUrls[*]	string	No	-	-
tags	array	No	-	-
tags[*]	object	No	-	-
tags[*].id	integer	No	-	• Format: int64
tags[*].name	string	No	-	-
status	string	No	-	pet status in the store • Allowed Values: available, pending, sold

Media Type: `application/x-www-form-urlencoded`

Key	Type	Required	Example	Notes
id	integer	No	10	• Format: int64
name	string	Yes	doggie	-
category	object	No	-	-

Key	Type	Required	Example	Notes
category.id	integer	No	1	• Format: int64
category.name	string	No	Dogs	-
photoUrls	array	Yes	-	-
photoUrls[*]	string	No	-	-
tags	array	No	-	-
tags[*]	object	No	-	-
tags[*].id	integer	No	-	• Format: int64
tags[*].name	string	No	-	-
status	string	No	-	pet status in the store • Allowed Values: available, pending, sold

Responses

- [200 OK](#)
- [400 Bad Request](#)
- [404 Not Found](#)
- [422 Unprocessable Content](#)

`200 OK`

Successful operation

RESPONSE HEADERS

No Specific Response Headers.

RESPONSE BODY

Media Type: `application/json`

Key	Type	Required	Example	Notes
id	integer	No	10	• Format: int64
name	string	Yes	doggie	-
category	object	No	-	-
category.id	integer	No	1	• Format: int64
category.name	string	No	Dogs	-
photoUrls	array	Yes	-	-
photoUrls[*]	string	No	-	-
tags	array	No	-	-
tags[*]	object	No	-	-
tags[*].id	integer	No	-	• Format: int64
tags[*].name	string	No	-	-

Key	Type	Required	Example	Notes
status	string	No	-	pet status in the store • Allowed Values: available, pending, sold

Media Type: `application/xml`

Key	Type	Required	Example	Notes
id	integer	No	10	• Format: int64
name	string	Yes	doggie	-
category	object	No	-	-
category.id	integer	No	1	• Format: int64
category.name	string	No	Dogs	-
photoUrls	array	Yes	-	-
photoUrls[*]	string	No	-	-
tags	array	No	-	-
tags[*]	object	No	-	-
tags[*].id	integer	No	-	• Format: int64
tags[*].name	string	No	-	-
status	string	No	-	pet status in the store • Allowed Values: available, pending, sold

`400 Bad Request`

Invalid ID supplied

RESPONSE HEADERS

No Specific Response Headers.

RESPONSE BODY

No Response Body

`404 Not Found`

Pet not found

RESPONSE HEADERS

No Specific Response Headers.

RESPONSE BODY

No Response Body

`422 Unprocessable Content`

Validation exception

RESPONSE HEADERS

No Specific Response Headers.

RESPONSE BODY

No Response Body

`POST /pet` (addPet)

Add a new pet to the store

Add a new pet to the store

Required Scopes

This endpoint requires the following scopes:

- `write:pets`
- `read:pets`

Path Parameters

No Path Parameters.

Query Parameters

No Query Parameters.

Request Headers

No Specific Request Headers

Cookies

No Cookies.

Request Body

Required.

Create a new pet in the store

Media Type: `application/json`

Key	Type	Required	Example	Notes
id	integer	No	10	- Format: int64
name	string	Yes	doggie	-
category	object	No	-	-
category.id	integer	No	1	- Format: int64
category.name	string	No	Dogs	-
photoUrls	array	Yes	-	-
photoUrls[*]	string	No	-	-
tags	array	No	-	-
tags[*]	object	No	-	-
tags[*].id	integer	No	-	- Format: int64
tags[*].name	string	No	-	-
status	string	No	-	pet status in the store - Allowed Values: available, pending, sold

Media Type: `application/xml`

Key	Type	Required	Example	Notes
id	integer	No	10	• Format: int64
name	string	Yes	doggie	-
category	object	No	-	-
category.id	integer	No	1	• Format: int64
category.name	string	No	Dogs	-
photoUrls	array	Yes	-	-
photoUrls[*]	string	No	-	-
tags	array	No	-	-
tags[*]	object	No	-	-
tags[*].id	integer	No	-	• Format: int64
tags[*].name	string	No	-	-
status	string	No	-	pet status in the store • Allowed Values: available, pending, sold

Media Type: `application/x-www-form-urlencoded`

Key	Type	Required	Example	Notes
id	integer	No	10	• Format: int64
name	string	Yes	doggie	-
category	object	No	-	-
category.id	integer	No	1	• Format: int64
category.name	string	No	Dogs	-
photoUrls	array	Yes	-	-
photoUrls[*]	string	No	-	-
tags	array	No	-	-
tags[*]	object	No	-	-
tags[*].id	integer	No	-	• Format: int64
tags[*].name	string	No	-	-
status	string	No	-	pet status in the store • Allowed Values: available, pending, sold

Responses

- [200 OK](#)
- [400 Bad Request](#)
- [422 Unprocessable Content](#)

200 OK

Successful operation

RESPONSE BODY

Media Type: `application/json`

Key	Type	Required	Example	Notes
id	integer	No	10	• Format: int64
name	string	Yes	doggie	-
category	object	No	-	-
category.id	integer	No	1	• Format: int64
category.name	string	No	Dogs	-
photoUrls	array	Yes	-	-
photoUrls[*]	string	No	-	-
tags	array	No	-	-
tags[*]	object	No	-	-
tags[*].id	integer	No	-	• Format: int64
tags[*].name	string	No	-	-
status	string	No	-	pet status in the store • Allowed Values: available, pending, sold

Media Type: `application/xml`

Key	Type	Required	Example	Notes
id	integer	No	10	• Format: int64
name	string	Yes	doggie	-
category	object	No	-	-
category.id	integer	No	1	• Format: int64
category.name	string	No	Dogs	-
photoUrls	array	Yes	-	-
photoUrls[*]	string	No	-	-
tags	array	No	-	-
tags[*]	object	No	-	-
tags[*].id	integer	No	-	• Format: int64
tags[*].name	string	No	-	-

Key	Type	Required	Example	Notes
status	string	No	-	pet status in the store • Allowed Values: available, pending, sold

`400 Bad Request`

Invalid input

RESPONSE HEADERS

No Specific Response Headers.

RESPONSE BODY

No Response Body

`422 Unprocessable Content`

Validation exception

RESPONSE HEADERS

No Specific Response Headers.

RESPONSE BODY

No Response Body

`GET /pet/findByStatus` (findPetsByStatus)

Finds Pets by status

Multiple status values can be provided with comma separated strings

Required Scopes

This endpoint requires the following scopes:

- `write:pets`
- `read:pets`

Path Parameters

No Path Parameters.

Query Parameters

Key	Type	Required	Example	Notes
status	string	No	-	• Default: available • Allowed Values: available, pending, sold

Request Headers

No Specific Request Headers

Cookies

No Cookies.

Request Body

No Request Body.

Responses

- [200 OK](#)
- [400 Bad Request](#)

`200 OK`

successful operation

No Specific Response Headers.

RESPONSE BODY

Media Type: `application/json`

Key	Type	Required	Example	Notes
	array			The root schema is an array.
[*]	object	No	-	-
[*].id	integer	No	10	• Format: int64
[*].name	string	Yes	doggie	-
[*].category	object	No	-	-
[*].category.id	integer	No	1	• Format: int64
[*].category.name	string	No	Dogs	-
[*].photoUrls	array	Yes	-	-
[].photoUrls[]	string	No	-	-
[*].tags	array	No	-	-
[].tags[]	object	No	-	-
[].tags[].id	integer	No	-	• Format: int64
[].tags[].name	string	No	-	-
[*].status	string	No	-	pet status in the store • Allowed Values: available, pending, sold

Media Type: `application/xml`

Key	Type	Required	Example	Notes
	array			The root schema is an array.
[*]	object	No	-	-
[*].id	integer	No	10	• Format: int64
[*].name	string	Yes	doggie	-
[*].category	object	No	-	-
[*].category.id	integer	No	1	• Format: int64
[*].category.name	string	No	Dogs	-
[*].photoUrls	array	Yes	-	-
[].photoUrls[]	string	No	-	-
[*].tags	array	No	-	-

Key	Type	Required	Example	Notes
[].*tags*[]	object	No	-	-
[].*tags*[].id	integer	No	-	• Format: int64
[].*tags*[].name	string	No	-	-
[*].status	string	No	-	pet status in the store • Allowed Values: available, pending, sold

`400 Bad Request`

Invalid status value

RESPONSE HEADERS

No Specific Response Headers.

RESPONSE BODY

No Response Body

`GET /pet/findByTags` (findPetsByTags)

Finds Pets by tags

Multiple tags can be provided with comma separated strings. Use tag1, tag2, tag3 for testing.

Required Scopes

This endpoint requires the following scopes:

- `write:pets`
- `read:pets`

Path Parameters

No Path Parameters.

Query Parameters

Key	Type	Required	Example	Notes
tags	array	No	-	-
tags[*]	string	No	-	-

Request Headers

No Specific Request Headers

Cookies

No Cookies.

Request Body

No Request Body.

Responses

- 200 OK
- 400 Bad Request

`200 OK`

successful operation

RESPONSE HEADERS

No Specific Response Headers.

RESPONSE BODY

Media Type: `application/json`

Key	Type	Required	Example	Notes
	array			The root schema is an array.
[*]	object	No	-	-
[*].id	integer	No	10	• Format: int64
[*].name	string	Yes	doggie	-
[*].category	object	No	-	-
[*].category.id	integer	No	1	• Format: int64
[*].category.name	string	No	Dogs	-
[*].photoUrls	array	Yes	-	-
[].photoUrls[]	string	No	-	-
[*].tags	array	No	-	-
[].tags[]	object	No	-	-
[].tags[].id	integer	No	-	• Format: int64
[].tags[].name	string	No	-	-
[*].status	string	No	-	pet status in the store • Allowed Values: available, pending, sold

Media Type: `application/xml`

Key	Type	Required	Example	Notes
	array			The root schema is an array.
[*]	object	No	-	-
[*].id	integer	No	10	• Format: int64
[*].name	string	Yes	doggie	-
[*].category	object	No	-	-
[*].category.id	integer	No	1	• Format: int64
[*].category.name	string	No	Dogs	-
[*].photoUrls	array	Yes	-	-
[].photoUrls[]	string	No	-	-
[*].tags	array	No	-	-
[].tags[]	object	No	-	-
[].tags[].id	integer	No	-	• Format: int64

Key	Type	Required	Example	Notes
[*].*tags*[].name	string	No	-	-
[*].status	string	No	-	pet status in the store • Allowed Values: available, pending, sold

`400 Bad Request`

Invalid tag value

RESPONSE HEADERS

No Specific Response Headers.

RESPONSE BODY

No Response Body

`GET /pet/{petId}` (getPetById)

Find pet by ID

Returns a single pet

Required Scopes

This endpoint requires the following scopes:

- `write:pets`
- `read:pets`

Path Parameters

Key	Type	Required	Example	Notes
petId	integer	Yes	–	• Format: int64

Query Parameters

No Query Parameters.

Request Headers

No Specific Request Headers

Cookies

No Cookies.

Request Body

No Request Body.

Responses

- 200 OK
- 400 Bad Request
- 404 Not Found

`200 OK`

successful operation

RESPONSE HEADERS

No Specific Response Headers.

RESPONSE BODY

Media Type: `application/json`

Key	Type	Required	Example	Notes
id	integer	No	10	• Format: int64
name	string	Yes	doggie	-
category	object	No	-	-
category.id	integer	No	1	• Format: int64
category.name	string	No	Dogs	-
photoUrls	array	Yes	-	-
photoUrls[*]	string	No	-	-
tags	array	No	-	-
tags[*]	object	No	-	-
tags[*].id	integer	No	-	• Format: int64
tags[*].name	string	No	-	-
status	string	No	-	pet status in the store • Allowed Values: available, pending, sold

Media Type: `application/xml`

Key	Type	Required	Example	Notes
id	integer	No	10	• Format: int64
name	string	Yes	doggie	-
category	object	No	-	-
category.id	integer	No	1	• Format: int64
category.name	string	No	Dogs	-
photoUrls	array	Yes	-	-
photoUrls[*]	string	No	-	-
tags	array	No	-	-
tags[*]	object	No	-	-
tags[*].id	integer	No	-	• Format: int64
tags[*].name	string	No	-	-
status	string	No	-	pet status in the store • Allowed Values: available, pending, sold

`400 Bad Request`

Invalid ID supplied

RESPONSE HEADERS

No Specific Response Headers.

RESPONSE BODY

No Response Body

`404 Not Found`

Pet not found

RESPONSE HEADERS

No Specific Response Headers.

RESPONSE BODY

No Response Body

`POST /pet/{petId}` (updatePetWithForm)

Updates a pet in the store with form data

Required Scopes

This endpoint requires the following scopes:

- `write:pets`
- `read:pets`

Path Parameters

Key	Type	Required	Example	Notes
petId	integer	Yes	–	• Format: int64

Query Parameters

Key	Type	Required	Example	Notes
name	string	No	-	-
status	string	No	-	-

Request Headers

No Specific Request Headers

Cookies

No Cookies.

Request Body

No Request Body.

Responses

- [400 Bad Request](#)

`400 Bad Request`

Invalid input

RESPONSE HEADERS

No Specific Response Headers.

RESPONSE BODY

No Response Body

`DELETE /pet/{petId}` (deletePet)

Deletes a pet

delete a pet

Required Scopes

This endpoint requires the following scopes:

- `write:pets`
- `read:pets`

Path Parameters

Key	Type	Required	Example	Notes
petId	integer	Yes	–	• Format: int64

Query Parameters

No Query Parameters.

Request Headers

Key	Type	Required	Example	Notes
api_key	string	No	–	–

Cookies

No Cookies.

Request Body

No Request Body.

Responses

- `400 Bad Request`

`400 Bad Request`

Invalid pet value

RESPONSE HEADERS

No Specific Response Headers.

RESPONSE BODY

No Response Body

POST /pet/{petId}/uploadImage (uploadFile)

uploads an image

Required Scopes

This endpoint requires the following scopes:

- `write:pets`
- `read:pets`

Path Parameters

Key	Type	Required	Example	Notes
petId	integer	Yes	–	• Format: int64

Query Parameters

Key	Type	Required	Example	Notes
additionalMetadata	string	No	-	-

Request Headers

No Specific Request Headers

Cookies

No Cookies.

Request Body

Media Type: `application/octet-stream`

Type	Example	Notes
string	-	• Format: binary

Responses

- [200 OK](#)

`200 OK`

successful operation

RESPONSE HEADERS

No Specific Response Headers.

RESPONSE BODY

Media Type: `application/json`

Key	Type	Required	Example	Notes
code	integer	No	-	• Format: int32
type	string	No	-	-
message	string	No	-	-

store

Access to Petstore orders

Read more about this category here: http://swagger.io

- [GET /store/inventory (getInventory)](#)
- [POST /store/order (placeOrder)](#)
- [GET /store/order/{orderId} (getOrderById)](#)
- [DELETE /store/order/{orderId} (deleteOrder)](#)

`GET /store/inventory (getInventory)`

Returns pet inventories by status

Returns a map of status codes to quantities

Required Scopes

This endpoint requires the following scopes:

Path Parameters

No Path Parameters.

Query Parameters

No Query Parameters.

Request Headers

No Specific Request Headers

Cookies

No Cookies.

Request Body

No Request Body.

Responses

- [200 OK](#)

`200 OK`

successful operation

RESPONSE HEADERS

No Specific Response Headers.

RESPONSE BODY

Media Type: `application/json`

Key	Type	Required	Example	Notes
{attribute}	integer	No	–	• Format: int32

`POST /store/order` (placeOrder)

Place an order for a pet

Place a new order in the store

Path Parameters

No Path Parameters.

Query Parameters

No Query Parameters.

Request Headers

No Specific Request Headers

Cookies

No Cookies.

Request Body

Media Type: `application/json`

Key	Type	Required	Example	Notes
id	integer	No	10	• Format: int64
petId	integer	No	198772	• Format: int64

Key	Type	Required	Example	Notes
quantity	integer	No	7	• Format: int32
shipDate	string	No	-	• Format: date-time
status	string	No	approved	Order Status • Allowed Values: placed, approved, delivered
complete	boolean	No	-	-

Media Type: `application/xml`

Key	Type	Required	Example	Notes
id	integer	No	10	• Format: int64
petId	integer	No	198772	• Format: int64
quantity	integer	No	7	• Format: int32
shipDate	string	No	-	• Format: date-time
status	string	No	approved	Order Status • Allowed Values: placed, approved, delivered
complete	boolean	No	-	-

Media Type: `application/x-www-form-urlencoded`

Key	Type	Required	Example	Notes
id	integer	No	10	• Format: int64
petId	integer	No	198772	• Format: int64
quantity	integer	No	7	• Format: int32
shipDate	string	No	-	• Format: date-time
status	string	No	approved	Order Status • Allowed Values: placed, approved, delivered
complete	boolean	No	-	-

Responses

- [200 OK](#)
- [400 Bad Request](#)
- [422 Unprocessable Content](#)

`200 OK`

successful operation

RESPONSE HEADERS

No Specific Response Headers.

RESPONSE BODY

Media Type: `application/json`

Key	Type	Required	Example	Notes
id	integer	No	10	• Format: int64
petId	integer	No	198772	• Format: int64
quantity	integer	No	7	• Format: int32
shipDate	string	No	–	• Format: date-time
status	string	No	approved	Order Status • Allowed Values: placed, approved, delivered
complete	boolean	No	–	–

`400 Bad Request`

Invalid input

RESPONSE HEADERS

No Specific Response Headers.

RESPONSE BODY

No Response Body

`422 Unprocessable Content`

Validation exception

RESPONSE HEADERS

No Specific Response Headers.

RESPONSE BODY

No Response Body

`GET /store/order/{orderId}` (getOrderById)

Find purchase order by ID

For valid response try integer IDs with value <= 5 or > 10. Other values will generate exceptions.

Path Parameters

Key	Type	Required	Example	Notes
orderId	integer	Yes	–	• Format: int64

Query Parameters

No Query Parameters.

Request Headers

No Specific Request Headers

Cookies

No Cookies.

Request Body

No Request Body.

Responses

- [200 OK](#)
- [400 Bad Request](#)
- [404 Not Found](#)

`200 OK`

successful operation

RESPONSE HEADERS

No Specific Response Headers.

RESPONSE BODY

Media Type: `application/json`

Key	Type	Required	Example	Notes
id	integer	No	10	• Format: int64
petId	integer	No	198772	• Format: int64
quantity	integer	No	7	• Format: int32
shipDate	string	No	-	• Format: date-time
status	string	No	approved	Order Status • Allowed Values: placed, approved, delivered
complete	boolean	No	-	-

Media Type: `application/xml`

Key	Type	Required	Example	Notes
id	integer	No	10	• Format: int64
petId	integer	No	198772	• Format: int64
quantity	integer	No	7	• Format: int32
shipDate	string	No	-	• Format: date-time
status	string	No	approved	Order Status • Allowed Values: placed, approved, delivered
complete	boolean	No	-	-

`400 Bad Request`

Invalid ID supplied

RESPONSE HEADERS

No Specific Response Headers.

RESPONSE BODY

No Response Body

`404 Not Found`

Order not found

RESPONSE HEADERS

No Specific Response Headers.

`DELETE /store/order/{orderId}` (deleteOrder)

Delete purchase order by ID

For valid response try integer IDs with value < 1000. Anything above 1000 or nonintegers will generate API errors

Path Parameters

Key	Type	Required	Example	Notes
orderId	integer	Yes	–	• Format: int64

Query Parameters

No Query Parameters.

Request Headers

No Specific Request Headers

Cookies

No Cookies.

Request Body

No Request Body.

Responses

- [400 Bad Request](#)
- [404 Not Found](#)

`400 Bad Request`

Invalid ID supplied

RESPONSE HEADERS

No Specific Response Headers.

RESPONSE BODY

No Response Body

`404 Not Found`

Order not found

RESPONSE HEADERS

No Specific Response Headers.

RESPONSE BODY

No Response Body

user

Operations about user

- [POST /user](#) [(createUser)](#)
- [POST /user/createWithList](#) [(createUsersWithListInput)](#)
- [GET /user/login](#) [(loginUser)](#)
- [GET /user/logout](#) [(logoutUser)](#)
- [GET /user/{username}](#) [(getUserByName)](#)
- [PUT /user/{username}](#) [(updateUser)](#)

- [DELETE /user/{username} (deleteUser)](#)

`POST /user` (createUser)

Create user

This can only be done by the logged in user.

Path Parameters

No Path Parameters.

Query Parameters

No Query Parameters.

Request Headers

No Specific Request Headers

Cookies

No Cookies.

Request Body

Created user object

Media Type: `application/json`

Key	Type	Required	Example	Notes
id	integer	No	10	• Format: int64
username	string	No	theUser	-
firstName	string	No	John	-
lastName	string	No	James	-
email	string	No	john@email.com	-
password	string	No	12345	-
phone	string	No	12345	-
userStatus	integer	No	1	User Status • Format: int32

Media Type: `application/xml`

Key	Type	Required	Example	Notes
id	integer	No	10	• Format: int64
username	string	No	theUser	-
firstName	string	No	John	-
lastName	string	No	James	-
email	string	No	john@email.com	-
password	string	No	12345	-
phone	string	No	12345	-

Key	Type	Required	Example	Notes
userStatus	integer	No	1	User Status • Format: int32

Media Type: `application/x-www-form-urlencoded`

Key	Type	Required	Example	Notes
id	integer	No	10	• Format: int64
username	string	No	theUser	-
firstName	string	No	John	-
lastName	string	No	James	-
email	string	No	john@email.com	-
password	string	No	12345	-
phone	string	No	12345	-
userStatus	integer	No	1	User Status • Format: int32

Responses

• [default Unknown Status Code](default_Unknown_Status_Code)

`default Unknown Status Code`

successful operation

RESPONSE HEADERS

No Specific Response Headers.

RESPONSE BODY

Media Type: `application/json`

Key	Type	Required	Example	Notes
id	integer	No	10	• Format: int64
username	string	No	theUser	-
firstName	string	No	John	-
lastName	string	No	James	-
email	string	No	john@email.com	-
password	string	No	12345	-
phone	string	No	12345	-
userStatus	integer	No	1	User Status • Format: int32

Media Type: `application/xml`

Key	Type	Required	Example	Notes
id	integer	No	10	• Format: int64
username	string	No	theUser	-
firstName	string	No	John	-
lastName	string	No	James	-
email	string	No	john@email.com	-
password	string	No	12345	-
phone	string	No	12345	-
userStatus	integer	No	1	User Status • Format: int32

`POST /user/createWithList` (createUsersWithListInput)

Creates list of users with given input array

Creates list of users with given input array

Path Parameters

No Path Parameters.

Query Parameters

No Query Parameters.

Request Headers

No Specific Request Headers

Cookies

No Cookies.

Request Body

Media Type: `application/json`

Key	Type	Required	Example	Notes
	array			The root schema is an array.
[*]	object	No	-	-
[*].id	integer	No	10	• Format: int64
[*].username	string	No	theUser	-
[*].firstName	string	No	John	-
[*].lastName	string	No	James	-
[*].email	string	No	john@email.com	-
[*].password	string	No	12345	-
[*].phone	string	No	12345	-

Key	Type	Required	Example	Notes
[*].userStatus	integer	No	1	User Status • Format: int32

Responses

- [200 OK](#)
- [default Unknown Status Code](#)

`200 OK`

Successful operation

RESPONSE HEADERS

No Specific Response Headers.

RESPONSE BODY

Media Type: `application/json`

Key	Type	Required	Example	Notes
id	integer	No	10	• Format: int64
username	string	No	theUser	-
firstName	string	No	John	-
lastName	string	No	James	-
email	string	No	john@email.com	-
password	string	No	12345	-
phone	string	No	12345	-
userStatus	integer	No	1	User Status • Format: int32

Media Type: `application/xml`

Key	Type	Required	Example	Notes
id	integer	No	10	• Format: int64
username	string	No	theUser	-
firstName	string	No	John	-
lastName	string	No	James	-
email	string	No	john@email.com	-
password	string	No	12345	-
phone	string	No	12345	-
userStatus	integer	No	1	User Status • Format: int32

`default Unknown Status Code`

successful operation

`GET /user/login` (loginUser)

Logs user into the system

Path Parameters

No Path Parameters.

Query Parameters

Key	Type	Required	Example	Notes
username	string	No	-	-
password	string	No	-	-

Request Headers

No Specific Request Headers

Cookies

No Cookies.

Request Body

No Request Body.

Responses

- [200 OK](#)
- [400 Bad Request](#)

`200 OK`

successful operation

RESPONSE HEADERS

Key	Type	Required	Example
X-Rate-Limit	integer	No	-
X-Expires-After	string	No	-

RESPONSE BODY

Media Type: `application/xml`

Type	Example	Notes
string	-	-

Media Type: `application/json`

Type	Example	Notes
string	-	-

`400 Bad Request`

Invalid username/password supplied

No Specific Response Headers.

No Response Body

`GET /user/logout` (logoutUser)

Logs out current logged in user session

Path Parameters

No Path Parameters.

Query Parameters

No Query Parameters.

Request Headers

No Specific Request Headers

Cookies

No Cookies.

Request Body

No Request Body.

Responses

- default Unknown Status Code

`default Unknown Status Code`

successful operation

RESPONSE HEADERS

No Specific Response Headers.

RESPONSE BODY

No Response Body

`GET /user/{username}` (getUserByName)

Get user by user name

Path Parameters

Key	Type	Required	Example	Notes
username	string	Yes	–	–

Query Parameters

No Query Parameters.

Request Headers

No Specific Request Headers

Cookies

No Cookies.

Request Body

No Request Body.

Responses

- [200 OK](#)
- [400 Bad Request](#)
- [404 Not Found](#)

`200 OK`

successful operation

RESPONSE HEADERS

No Specific Response Headers.

RESPONSE BODY

Media Type: `application/json`

Key	Type	Required	Example	Notes
id	integer	No	10	• Format: int64
username	string	No	theUser	-
firstName	string	No	John	-
lastName	string	No	James	-
email	string	No	john@email.com	-
password	string	No	12345	-
phone	string	No	12345	-
userStatus	integer	No	1	User Status • Format: int32

Media Type: `application/xml`

Key	Type	Required	Example	Notes
id	integer	No	10	• Format: int64
username	string	No	theUser	-
firstName	string	No	John	-
lastName	string	No	James	-
email	string	No	john@email.com	-
password	string	No	12345	-
phone	string	No	12345	-
userStatus	integer	No	1	User Status • Format: int32

`400 Bad Request`

Invalid username supplied

RESPONSE HEADERS

No Specific Response Headers.

RESPONSE BODY

No Response Body

`404 Not Found`

User not found

RESPONSE HEADERS

No Specific Response Headers.

RESPONSE BODY

No Response Body

`PUT /user/{username}` (updateUser)

Update user

This can only be done by the logged in user.

Path Parameters

Key	Type	Required	Example	Notes
username	string	Yes	-	-

Query Parameters

No Query Parameters.

Request Headers

No Specific Request Headers

Cookies

No Cookies.

Request Body

Update an existent user in the store

Media Type: `application/json`

Key	Type	Required	Example	Notes
id	integer	No	10	• Format: int64
username	string	No	theUser	-
firstName	string	No	John	-
lastName	string	No	James	-
email	string	No	john@email.com	-
password	string	No	12345	-
phone	string	No	12345	-
userStatus	integer	No	1	User Status • Format: int32

Media Type: `application/xml`

Key	Type	Required	Example	Notes
id	integer	No	10	• Format: int64

Key	Type	Required	Example	Notes
username	string	No	theUser	-
firstName	string	No	John	-
lastName	string	No	James	-
email	string	No	john@email.com	-
password	string	No	12345	-
phone	string	No	12345	-
userStatus	integer	No	1	User Status • Format: int32

Media Type: `application/x-www-form-urlencoded`

Key	Type	Required	Example	Notes
id	integer	No	10	• Format: int64
username	string	No	theUser	-
firstName	string	No	John	-
lastName	string	No	James	-
email	string	No	john@email.com	-
password	string	No	12345	-
phone	string	No	12345	-
userStatus	integer	No	1	User Status • Format: int32

Responses

• [default Unknown Status Code](#)

`default Unknown Status Code`

successful operation

RESPONSE HEADERS

No Specific Response Headers.

RESPONSE BODY

No Response Body

`DELETE /user/{username}` (deleteUser)

Delete user

This can only be done by the logged in user.

Path Parameters

Key	Type	Required	Example	Notes
username	string	Yes	-	-

Query Parameters

No Query Parameters.

Request Headers

No Specific Request Headers

Cookies

No Cookies.

Request Body

No Request Body.

Responses

- [400 Bad Request](#)
- [404 Not Found](#)

`400 Bad Request`

Invalid username supplied

RESPONSE HEADERS

No Specific Response Headers.

RESPONSE BODY

No Response Body

`404 Not Found`

User not found

RESPONSE HEADERS

No Specific Response Headers.

RESPONSE BODY

No Response Body