

Wishify API Specification

Endpoints - Basic.....	2
Auth.....	2
Categories.....	2
Contributions.....	3
Events.....	4
Ideas.....	5
Items.....	6
notifications.....	6
Payments.....	7
Status.....	7
Users.....	8
Wishlists.....	9

Endpoints - Basic

- Endpoints that contain '**token**' require an authorization bearer token in the header
- All **Bolded** fields are **required**

Auth

Request	Description	Input	Output
Post /register	Create a user account.	email, password, displayName, bio, notifications	id, email, displayName, notifications
Post /login	Login to account	email, password	token
Post /logout	Logout of account	token	
Get /me	return user with ID.	token	id, bio, setup, email, oauth, displayName, picture
Post /forgot-password	Send an email with recovery code	email	200 Success 400 otherwise
Post /reset-password	Reset password with one time code	email, etc, newPassword	200 Success 400 otherwise
Get /google	Used for Oauth by google		
Get /google/callback	Used for Oauth by google		

Categories

Request	Description	Input	Output
Get /	Create a user account.		For[Id, name, description, created]
Get /:id	Login to account		id, name, description, created
Post /	create a category must be the admin account.	Token, name, description	id,name, Description, created

Contributions

Request	Description	Input	Output
Get /	Get all contributions from logged in user.	token	For[id, item_id, quantity, purchased, note, dateUpdated, dateCreated, wishlists_name]
Get /wishlists/:id	Get all contributions from wishlist	token	For[id, item_id, quantity, purchased, note, dateUpdated, dateCreated, user_displayName, user_id, pro, picture]
GET /items/:id	Get contributions for an item	token	For[id, item_id, quantity, purchased, note, dateUpdated, dateCreated, user_displayName, user_id, pro, picture]
POST /	Add a contribution (given item_id)	token item_id , quantity, purchased, note	id, item_id, quantity, purchased, note, dateUpdated, dateCreated, user_displayName, user_id, pro, picture
PUT /:id	Update a contribution (mark as purchased, etc.)	token quantity, purchased, note	id, item_id, quantity, purchased, note, dateUpdated, dateCreated, user_displayName, user_id, pro, picture
DELETE /:id	Remove a contribution	token	200 - success 400 error otherwise

Events

Request	Description	Input	Output
Get /	Get all events.	token	For[id, name, eventId, description, image, addr, city, dateUpdated, dateCreated, owner, creator_displayName]
Get /:id	get event, wish lists, and members	token	{event, members wishlists}
Post /	Make an event	token, name, description, image, addr, city, deadline	id, name, eventId, city, description, image, addr, dateUpdated, dateCreated
Put /id	edit an event	token, name, description, image, addr, city, deadline	id, name, eventId, city, description, image, addr, dateUpdated, dateCreated
Delete /id	Delete event	token	200 - success 400 - error
Get /:id/wishlists	get list of wishlists	token	For[id, eventId, name, description, image, deadline, creator_id, dateUpdated, dateCreated, owner]
Get /:id/members	get list of members	token	For[id, displayName, email, picture, pro, owner]
Post /:id/members	add a member (must be the owner)	token, userId, owner	200 - success 400 - otherwise
Delete /:id/members	remove a member (must be the owner)	token userId	200 - success 400 - otherwise
Put /:id/members	Update a membership (must be the owner)	token, owner, notifications	id, user_id, event_id, owner, notifications dateUpdated, dateCreated
Post /members	add a member with share token	token, share_token	id
Post /share	Add user to event with email	event_id, email	200 success 400 otherwise

Ideas

Request	Description	Input	Output
Get /	return ideas based on user categories	token	id, name, description url, image, price, sponsored, uses, created, categories[{ Id, name}] score
Get /trending	Get the top 10 most popular items.	N/A	id, name, description url, image, price, sponsored, uses, created, categories[{ Id, name}]
Post /	Create an idea (must be an admin) (Formdata file for picture)	token, name, description, url, sponsored, price, categories[id, name], picture	id, name, description url, image, price, sponsored, uses, created
Post /id/categories	Assign categories to an idea	token, ideald, or categories[id]	200 success 400 otherwise
Post /upload/id	Upload an image. (Formdata file for picture)	token, picture	200 success 400 otherwise

Items

Request	Description	Input	Output
Get /:id	Get item details	token	id, wishlists_id name, quantity, description, url, image, price
PUT /:id	Update an item	token , name, price, quantity, description, url, image	wishlists_id name, quantity, description, url, image, price
PUT /	Update a array of items	token For[name, price, description, url, image, quantity]	For[wishlists_id name, quantity, description, url, image, price]
DELETE /:id	Remove an item	token	200 - success 400 - error otherwise
POST /	create an item	token, name, wishlists_id quantity, description, url, image, price	id, wishlists_id, name, quantity, description, url, image, price
POST / (Adding from Idea page)	Adds an item from idea page to a wishlist	token wishlists_id idea_id quantity	id, wishlists_id, name, quantity, description, url, image, price
POST /upload.id	upload an item picture	token, FormData file: 'picture'	200 - success 400 error otherwise

notifications

Request	Description	Input	Output
Get /	Get all notifications	token	id, title, body, url, is_read, created
PUT /:id	Update a notifications	token , is_read	id, title, body, url, is_read, created
DELETE /:id	Remove an item	token	200 - success 400 - error otherwise

Payments

Request	Description	Input	Output
Get /subscription	Get current users subscription status	token	status, cancelAt, currentPeriodEnd, since, plan, price: { amount, currency, interval: price}
Post /create-subscription-session	Get link to launch a checkout page	token	url, sessionId
Post /create-portal-session	Get link to launch a manage subscription page	token	url
Post /reactivate-subscription	Reactivate subscription renewal	token	200 Success 400 failure
Post /cancel-subscription	Deactivate subscription renewal	token	200 Success 400 failure
Post /webhook	Performs actions depending on what stripe sends	N/A called by Stripe	N/A

Status

Request	Description	Input	Output
Get /	Get the API status	N/A	Status information

Users

Request	Description	Input	Output
Get /	return current user profile and categories	token	user: id, email, displayName, bio, picture, pro, setup, notifications, datecreated, dateupdated,oauth For categories:[id, name, description created, love]
Put /	edit your account	token , email, displayName, bio, password, newPassword, setup	id, email, displayName, bio, picture, pro, setup, notifications, datecreated, dateupdated
Delete /	delete your account	token , password	200 - success 400 error otherwise
Get /userId	Get user information by id	token	user: id, bio, picture, displayName, pro, notifications, datecreated, dateupdated,oauth For categories:[id, name, description created, love]
Post /categories/:id	Add category to user	token love	200 - success 400 error otherwise
Delete /categories/:id	Remove category from user	token	200 - success 400 error otherwise
Put /categories/:id	Update a category love	token love	200 - success 400 error otherwise
POST /upload	upload a profile picture	token Formdata file: 'picture'	200 - success 400 error otherwise

Wishlists

Request	Description	Input	Output
Get /	Get all your wishlists.	token	For[id, eventid, name, description, image, deadline, dateUpdated, dateCreated, creator_displayName , blind, owner]
Post /	Make a wishlist	token, name, event_id, description, image, deadline, notifications	id, eventid, name, description, image, deadline, dateUpdated, dateCreated, owner
Get /id	Composite get for a specific wishlist.	token	wishlist: { id, eventid, name, description, image, deadline, dateUpdated, dateCreated, creator_displayName , blind, owner, notifications, is_wishlist_member, is_event_member} For items: [] For contributions []
Put /:id	Update wishlist details	token, id, eventid, name, image, description, image, deadline	id, eventid, name, description, image, deadline, dateUpdated, dateCreated
Delete /:id	Delete wishlist	token	200 - success 400 - error otherwise
Post /id/duplicate	Duplicate the wishlist	token	200 - success 400 - error otherwise
Delete /:id/members	remove a member (must be the owner)	token userId	200 - success 400 - error otherwise
Get /:id/members	get list of members	token	For members[id, user_id, event_id,

			blind, owner, dateUpdated, dateCreated]
Post /:id/members	add a membership (must be the owner)	token, userId owner, blind	200 - success 400 - error otherwise
Put /:id/members	Update a membership (must be the owner)	token , owner, blind	id, user_id, event_id, blind, owner, dateUpdated, dateCreated
Get /:id/items	Get all items in a wishlist	token	For[id, name, description, url, image, quantity, price, priority, dateUpdated, dateCreated, user_displayName, user_id]
GET /share/:token	get the shared wishlist		id, eventId, name, description, image, deadline, dateUpdated, dateCreated For items: []
Post /members	add a membership with share token	token, share_token owner, blind	id, 200 - success 400 - error otherwise
Post /share	Add user to wishlist with email	wishlist_id, email	200 - success 400 - otherwise