Routes

/user					
Action	**Route**	**Methods**	**Requires**	**Returns**	**Info**
Register a new user and receive an auth token	/signup	PUT	email, password	token, error	• validates user input • adds user to database • returns token if successful
Sign in a user and get an auth token	/signin	GET	email, password	token, error	• validates user input • verify user credentials • returns token if successful
Sign out user and cancel the token	/signout	GET	email, password, token	success message	• verify user identity and token • expires the token
Get details of a specific user	/details	GET	token	User object	• verify token • return object containing user profile details
Update details of a specific user (except password)	/details/	POST	token	User object, success message	• verify userid and token • return object containing updated user details
Delete current user account	/delete/	POST	email, password, token	User object, success message	• verify credentials and token

					• deletes the user • returns user details

/items					
Action	**Route**	**Methods**	**Requires**	**Returns**	**Info**
Get list of all items of a user	/all	GET	token	Items object	• authenticate • return all items for the user
Get a specific item	/item/:itemId	GET	token, itemid	Item object	• auth • return details of a specific item
Add a new item	/item	PUT	token, item object	Item object, success message	• auth • validation • adds item to user's items • returns the newly added item back with success message
Update an existing item	/item	POST	token, item object	Item object, success message	• auth • validation • check if item exists • updates the item with new info • returns the new item object
Delete an existing	/item/:itemId	DELETE	token, itemid	success message	• auth

item					• check if itemid valid • delete the item
Get list of all category names	/categories	GET	token	string containing comma separated names of all categories	• auth • get category name for all categories • return comma separated string with names
Get items for a specific category	/category/:categoryId	GET	token, categoryId	Items object	• auth • return Items object containing all items for a specific category
Add a new category	/category	PUT	token, Category object	Category object, success message	• auth • validate input • add the new category to DB • return added Category object
Update an existing category	/category	POST	token, Category object	Category object, success message	• auth • validation • get existing category • update category info • return Category obj
Delete an existing category	/category/:categoryId	DELETE	token, categoryId	success message	• auth • delete category

					- success message