Shared Spaces App API Reference

This document functions as a technical reference to the Shared Spaces API. The Shared Spaces API is a standard JSON-based RESTful API. Any non-JSON requests will return an error, as will any requests to endpoints not listed here. Unless specified otherwise, the API will return a simple 200 OK status code and an empty response on success, and an expressive error code with an explanatory message on failure. All requests can return error 400 if not all parameters are found or valid, all requests can return error 401 if the requester does not have authority to perform the desired operation, and all requests but POST can return 404 if the requested ID number does not exist on the server. Unhandled errors return error 500. Unless specified, these are the only errors that should be received from a request.

All locations should be prepended with 18.216.3.210/api/. Some parameters are also passed in the URI accessed. For example, groups/$GROUPID requires a group ID number in place of $GROUPID. JSON is denoted as {required_variable:type1, [array_variable:type2], (optional_variable:type3), ...}.

Login Requests

Login requests are special. A login request must be performed before almost any other requests are submitted. The token received from this login request must be included with all other requests.

Location: login
Request type: POST
Request format: {email:string, password:string, stayLoggedIn:boolean}
Request success returns: {token:string}

stayLoggedIn determines how long the authentication token is valid for. If it is ‘false’, the login token is valid for 24 hours. If it is ‘true’, the login token is valid for 30 days.

Logouts are also special. The following API will log out a specific authentication token.

Location: logout
Request type: POST
Request format: {token:string}

POST Requests

POST requests are used for creating things. The following endpoints are available:

Purpose: Create group
Location: groups
Request format: {token:string}
Response success returns: {id:number}
Request failure returns: error 409 if the user is already in a group

Purpose: Create user
Location: users
Request parameters: {first:string, last:string, email:string, password:string, phoneNumber:string, (facebook:string, twitter:string, instagram:string, groupID:number)}
Response success returns: {token:string}
Request failure returns: error 409 if the email is already in use

Purpose: Create group debt
Location: groupDebts/$groupId
Request parameters: {debtType:string, amount:number}

Purpose: Create personal debt
Location: personalDebts
Request parameters: {amount:number, lender:number, borrower:number} (lender and borrower are user IDs)

Purpose: Create grocery item
Location: groceries/$groupId
Request parameters: {amount:number, userID:number}

Purpose: Create chore
Location: chores/$groupId
Request parameters: {chore:string, due_date:date, userID:number}

Purpose: Create rent
Location: rent/$groupId
Request parameters: {amount:number, userID:number}

GET Requests

GET requests are used for reading things. GET requests do not require any additional parameters.

Purpose: Get group information
Location: groups/$groupId
Request success returns: {group_info:{group_name:string, id:number]}, users:{[id:number, first_name:string, last_name:string, email:string, phone_number:string, facebook_profile:string, twitter_handle:string, instagram:string, group_id:number]}, debts:{[id:number, debt_type:string, amount:number, group_id:number]}}

Purpose: Get user information
Location: users/$userId
Request success returns: {id:number, first_name:string, last_name:string, email:string, phone_number:string, facebook_profile:string, twitter_handle:string, instagram:string, group_id:number}

Purpose: Get group debt list
Location: groupDebts/$groupId
Request success returns: {[id:number, debt_type:string, amount:number, group_id:number]}

Purpose: Get group debt info (for an individual debt)
Location: groupDebts/$groupId/$debtId
Request success returns: {id:number, debt_type:string, amount:number, group_id:number}

Purpose: Get personal debt list
Location: personalDebts/$userId
Request success returns: {[id:number, amount:number, lender_id:number, borrower_id:number]}

Purpose: Get personal debt info (for a single debt)
Location: personalDebts/$userId/$debtId
Request success returns: {id:number, amount:number, lender_id:number, borrower_id:number}

Purpose: Get grocery list for group
Location: groceries/$groupId
Request success returns: {[id:number, amount_due:number, paid_status:boolean, user_id:number, group_id:number]}

Purpose: Get information about a single grocery item
Location: groceries/$groupId/$groceryId
Request success returns: {id:number, amount_due:number, paid_status:boolean, user_id:number, group_id:number}

Purpose: Get chores list
Location: chores/$groupId
Request success returns: {[id:number, chore:string, due_date:date, chore_complete:boolean, user_id:number, group_id:number]}

Purpose: Get chores info (for a single chore)
Location: chores/$groupId/$choreId
Request success returns: {id:number, chore:string, due_date:date, chore_complete:boolean, user_id:number, group_id:number}

Purpose: Get rent info
Location: rent/$groupId
Request success returns: {id:number, rent_amount:number, rent_paid:boolean, user_id:number, group_id:number}

PUT Requests

PUT requests are used for updating things.

Purpose: Update group information
Location: groups/$groupId
Request parameters: {groupName:string, groundRules:string, users:[number]}

Purpose: Update user information
Location: users/$userId
Request parameters: {first:string, last:string, email:string, password:string, phoneNumber:string, facebook:string, twitter:string, instagram:string, groupID:number}

Purpose: Update group debt
Location: groupDebts/$groupId/$debtId
Request parameters: {debtType:string, amount:number}

Purpose: Update personal debt
Location: personalDebts/$userId/$debtId
Request parameters: {amount:number, lenderID:number, borrowerID:number}

Purpose: Update grocery item
Location: groceries/$groupId/$groceryId
Request parameters: {amount:number, paid:boolean, userID:number}

Purpose: Update chore item
Location: chores/$groupId/$choreId
Request parameters: {chore:string, duedate:date, complete:boolean, userID:number}

Purpose: Update rent information
Location: rent/$groupId
Request parameters: {amount:number, paid:boolean, userID:number}

DELETE Requests

DELETE requests are for deleting things. DELETE requests have no additional parameters.

Purpose: Delete group
Location: groups/$groupId
Note: This will remove all information associated with this group, and set the group ID of all users in the group to null.

Purpose: Delete user
Location: users/$userId
Note: Since this deletes your account, it will also sign out all active authentication tokens. It will also delete any personal debts associated with the user and set the user ID of any group-related items associated with the user to null.

Purpose: Delete group debt
Location: groupDebts/$debtId

Purpose: Delete personal debt
Location: personalDebts/$debtId

Purpose: Delete grocery list
Location: groceries/$groupId

Purpose: Delete grocery item
Location: groceries/$groupId/$groceryId

Purpose: Delete chores list
Location: chores/$groupId

Purpose: Delete chore item
Location: chores/$groupId/$choreId