

CollegeChef API

Endpoint	Method	Token	Description	Input JSON [] means optional	Output JSON [] means array	Possible Errors	Scope
/login	POST		Log in	userName, password	userId, userName, isAdmin, token	400: missing required input 403: incorrect userName or password, login failed	All
/logout	GET	Required	Logout	Nothing	An HTTP OK	401: authentication failed (token has expired or token is missing)	Any logged-in users or admin
/user	POST		Create user	userName, password, [email]	An HTTP OK and created user object	400: not validated 403: Username is already in use	All
	DELETE	Required	Delete a user	userId	An HTTP OK	401: authentication failed 400: not validated or current user is not admin	Admin only
/user/:userId	GET	Required	Get user info	Nothing	userId, userName, email, isAdmin, description, profilePhoto	401: authentication failed 403: no data in database	Any logged-in users or admin
	PUT	Required	Edit a user profile by given userId	userId, [email], [description], [profilePhoto]	An HTTP OK	401: authentication failed or Lacking proper credentials 403: no user in database	Admin only
/user/:userId/password	PUT	Required	Change password	userId, userName, password, newPassword	An HTTP OK	401: authentication failed or Lacking proper credentials 400: missing password(s) 403: invalid password or user does not exist	Any logged-in users or admin
/users	GET	Required	Get a list of all users	Nothing	Users (array of id, userName, email)	401: authentication failed or lacking proper credentials(not admin) 404: no user found	Admin only
/feedback	POST		Submits feedback to database	feedback, [name], [email]	An HTTP OK and created feedback object	400: missing feedback	
/feedback	GET	Required	Gets a list of all submitted feedback	Nothing	Array of feedback: [{feedbackId, feedback, name, email}]	401: authentication failed 404: no feedback found	Admin only

/ingredients	GET		Get a list of all ingredients	Nothing	Array of Ingredients: [{_id, name, imgUrl}]	403: no data in database	All								
/search?searchtype=(Query: <table><tr><td>Key</td><td>Values</td></tr><tr><td>searchtype</td><td>include</td></tr><tr><td></td><td>equal</td></tr><tr><td></td><td>exclude</td></tr></table>)	Key	Values	searchtype	include		equal		exclude	POST		Search for recipes by selected ingredients	Array of ingredientId: { ingredients: [ids]}	Array of Recipes: [{recipeId, recipeName, description, imgUrl}]	400: Bad Request (no ingredients passed or searchtype is missing or invalid)	All
Key	Values														
searchtype	include														
	equal														
	exclude														
/recipes/hot	GET		Get a list of hot (mostly commented) recipes	Nothing	Array of Recipes: [{_id, commentCount, recipeName, description, imgUrl}]		All								
/recipes/remarkable	GET		Get a list of remarkable (highest average rated) recipes	Nothing	Array of Recipes: [{_id, avgScore, recipeName, description, imgUrl}]		All								
/recipes/new	GET		Get a list of new recipes (ModifiedDate)	Nothing	Array of Recipes: [{_id, ModifiedDate, recipeName, description, imgUrl}]		All								
/recipes/favorite	GET	Required	Get a list of favorite (starred by current user) recipes	Nothing	Array of Recipes: [{id, recipeName, description, imgUrl}]	401: Unauthorized (missing token or token expired)	Any logged-in users or admin								
/recipes/uploaded	POST	Required	Get list of recipes uploaded by specified user	[userName] (if userName is not passed, default to current user)	Array of Recipes: [{_id, recipeName, description, imgUrl}]	401: Unauthorized (missing token or token expired)	Uploader or admin								
/recipes (Optional query: username, category)	GET		Get a list of all recipes	Nothing	Array of Recipes: [{_id, recipeName, description, imgUrl, uploaderName, categoryName}]		All								
/recipe	POST	Required	Add a recipe	recipeName, categoryId, description, instruction, imgUrl,	Created recipe: {recipeId, recipeName, description, instruction, imgUrl, numServings,	401: authentication failed	Any logged-in users or admin								

				numServings, [notes], ingredients: [{id, amount}]	ModifiedDate, uploaderId, uploaderName, uploaderImg, modifierId, modifierName, categoryId, categoryName, avgRating, ingredients: [{ingredientId, name, amount, imgUrl}] }		
/recipe/:recipeId	DELETE	Required	Delete a recipe (record still in database with isDeleted fields set to be true)	{ } (empty JSON)	An HTTP OK	401: Unauthorized (missing token or token expired) 403: Forbidden (only Admin or recipe owner can delete this recipe) 404: Not Found (recipe to delete not found)	Any logged-in users or admin
	PUT	Required	Modify a recipe	At least one of (recipeName, categoryId, description, instruction, imgUrl, numServings, notes, ingredients: [{id, amount}])	Updated recipe: {recipeId, recipeName, description, instruction, imgUrl, numServings, ModifiedDate, uploaderId, uploaderName, uploaderImg, modifierId, modifierName, categoryId, categoryName, avgRating, ingredients: [{ingredientId, name, amount, imgUrl}] }	400: Bad Request (no modifiable field passed) 401: Unauthorized (missing token or token expired) 404: Not Found (recipe to update not found)	Any logged-in users or admin
	GET		Get a recipe	Noting	{recipeId, recipeName, description, instruction, imgUrl, numServings, ModifiedDate, uploaderId, uploaderName, uploaderImg, modifierId, modifierName, categoryId, categoryName, avgRating, ingredients: [{ingredientId, name, amount, imgUrl}] }	404: Not Found (recipe not found)	All

/recipe/:recipeId/comments/text	GET		Get a list of text comments of the current recipe	Nothing	Array of Comments: [{profilePhoto, createdAt, userName, message}]		All
/recipe/:recipeId/comments/image	GET		Get a list of image comments of the current recipe	Nothing	Array of Comments: [{createdAt, userName, message}]		All
/recipe/:recipeId/comments	POST	Required	Comment a recipe	isImage (boolean), message(string)	An HTTP OK	400: Bad Request (isImage or message not passed) 401: Unauthorized (missing token or token expired) 404: Not Found (recipe to comment not found)	Any logged-in users or admin
/recipe/:recipeId/rate	POST	Required	Rate a recipe or Update rate	scores(number)	An HTTP OK	400: Bad Request (scores not passed) 401: Unauthorized (missing token or token expired) 404: Not Found (recipe to rate not found)	Any logged-in users or admin
	GET	Required	Get current user's rate of the recipe (0 if not rated)	Nothing	scores	401: Unauthorized (missing token or token expired)	All
/recipe/:recipeId/ratings	GET		Get average rating of the recipe (0 is never rated)	Nothing	avgScore		All
/recipe/:recipeId/rating list	GET	Optional	Get list of ratings of the recipe If logged in, exclude current user's rating	Nothing	[[scores, userName]]		All
/recipe/:recipeId/favorite	POST	Required	Favorite a recipe	{ } (empty JSON)	An HTTP OK	401: Unauthorized (missing token or token expired) 403: Forbidden (recipe already favorited by current user) 404: Not Found (recipe to favorite not found)	Any logged-in users or admin
	DELETE	Required	Unfavorite a recipe	{ } (empty JSON)	An HTTP OK	401: Unauthorized (missing token or token expired)	Any logged-in users or admin

						403: Forbidden (recipe is not favorited by current user) 404: Not Found (recipe to unfavorite not found)							
	GET	Required	Check if it's favorited	Nothing	{isFavorited}	401: Unauthorized (missing token or token expired)	Any logged-in users or admin						
/categories	GET		Get all built-in categories	Nothing	Array of categories	403: No Categories in database	Any logged-in users or admin						
/notification (Optional queries: <table><tr><td>Key</td><td>Values</td></tr><tr><td>recipetype</td><td>uploaded favorite</td></tr><tr><td>actiontype</td><td>rate comment favorite update delete</td></tr></table>)	Key	Values	recipetype	uploaded favorite	actiontype	rate comment favorite update delete	GET	Required	Get list of notification histories of current user	Nothing	Array of notifications: [{recipeOwnerId, recipeOwnerName, operatorId, operatorName, recipId, recipeName, actionTypeId, actionTypeName, actionTypeMsg, actionDate}]	400: Bad Request (Invalid query key or value) 401: Unauthorized (missing token or token expired)	Any logged-in users or admin
Key	Values												
recipetype	uploaded favorite												
actiontype	rate comment favorite update delete												

- If Token is required, pass token via bearer authorization HTTP header in request.
- Get current userId, userName and status(Admin or user) from request.userId, request.userName and request.isAdmin

Collections:

- User: userName(String), [email(String)], password(String), isAdmin(Boolean), [description(String)], [profilePhoto(String)]
- Recipe: personId(User.Id), recipeName(String), ModifiedDate(default: Date.now), ModifiedById(User.Id), categoryId(Category.Id), description(String), instruction(String), imgUrl(String), numServings(Number), [notes(String)]
- Ingredient