Error Codes

queryFailed Query failed (server problem)
forbiddenField Field not allowed in the body
missingField Field missing from the request
badValue Field has bad value
notFound Entity not found in the database
badLogin Email/password combination invalid, for errors logging.
dupEmail Email duplicates an existing email
noTerms Acceptance of terms is required
noOldPwd Change of password requires an old password
oldPwdMismatch Old password that was provided is incorrect.
dupIngredient Ingredient already exists in the database

Resource for Recipes

Ssns

Login sessions (Ssns) establish an AU and are required for most service access. A user obtains one via POST to Ssns.

GET Returns a list of current session

prsId ID of Person logged in
loginTime Date and time of login

POST A successful POST generates a browser-session cookie that will permit continued access for 2 hours. Indicated Person becomes the AU. An unsuccessful POST results in a 400 with a badLogin tag and no further information.

email Email of user requesting login
password Password of user

DELETE Logout the current Session.

Prss/

Requires AU for all requests.

GET

Returns the person that is logged in. No data for other than the AU is returned in any event. Data per person:

email principal string identifier, unique across all Persons
id id of person with said email, so that URI would be Prss/{id}

POST

Adds a new Person. User must not be logged in.

email unique Email for new person

firstName
lastName
Password
termsAccepted boolean--were site terms and conditions accepted?

Email and lastName required and must be nonempty. Error if email is nonunique. Error if terms were not accepted. Nonempty password required.

PUT

Updates currently logged in user with body giving an object with one or more of *firstName*, *lastName*, *password.*Attempt to change other fields in Person such as *termsAccepted* or *whenRegistered* results inBAD_REQUEST and forbiddenField error(s). An additional field *oldPassword* is required for changing *password*. Password, if supplied, must be nonempty.

DELETE

Deletes the current logged in user

Fdg

Requires an AU.

GET

Returns an array of AU's available ingredients

name name of the ingredient

POST *name={ingredient name}*

Adds a new ingredient to the AU's fridge

name name of the ingredient (must be unique and not already exist in the fridge)

DELETE *name={ingredient name}*

Deletes an ingredient from the AU's fridge

Rcp

Requires an AU.

GET

Returns an array of AU's planned recipes.

id Id of the planned recipe
recipeId Recipe id corresponding to the food2fork API
date Planned date for the recipe to be created

POST

Adds a planned recipe.

recipeId Recipe id corresponding to the food2fork API
date Planned date for the recipe to be created

Rcp/{id}

Requires an AU.

DELETE

Remove a planned recipe. AU must be the owner of that planned recipe.

Rat/{recipeId}

GET

Returns all messages, likes, and dislikes tied to a recipe.

likes Amount of likes for the recipe
dislikes Amount of dislikes for the recipe
comment User submitted comments

Rat/{recipeId}/Cmts

POST

Adds a comment to specified recipe.

comment The comment content

Rat/{recipeId}/Lkes

POST

Adds a like to specified recipe.

DELETE

Deletes a like from a specified recipe.

Rat/{recipeId}/Dlks

POST

Adds a dislike to specified recipe.

DELETE

Deletes a dislike from a specified recipe.