

Test Plan

Module	Function	Endpoint	Test data	Expected return	Test result
user_controller.py	print all users	@user.get("/")	none	list of users	Pass
	print users of primary key	@user.get("/<int:id>")	valid id	user	Pass
			incorrect or no id	400, missing key error	Pass
	Register new user	@user.post("/")	Valid data set	created user	Pass
			none or missing	400, missing key error	Pass
	Delete User (admin or user can delete himself)	@user.delete("/<string:email>")	valid email	deleted user id and email	Pass
			none	the method is not allowed error	Pass
	Amend User (only by the user itself)	@user.put("/update")	email does not exist	abort, email does not exist	Pass
			Valid data set	updated user details	Pass
			none or missing	400, missing key error	Pass
	login user	@user.post("/login")	Valid data set	email, token, admin status, user_id	Pass
			none or missing	400, missing key error	
workout_controller.py	print all workouts of logged in user	@workout.get("/")	none	all workout data sets	Pass
	print workout of a speific id	@workout.get("/<int:id>")	valid id	workout record	Pass
			incorrect id	abort, workout with id does not exist	Pass
	Create new workout with the user id of logged in user	@workout.post("/")	Valid data set	workout record	Pass
			incomplete	400, missing key error	Pass
	Delete workout (only admin or user who created it can delete)	@workout.delete("/<string:workout_name>")	valid workout	user id and name of deleted workout	Pass
			none or not existing	abort, workout with the workout name does not exist	Pass
	Amend workout (only user who created it)	@workout.put("/update/<string:workout_name>")	valid workout and data set	set of updated workout	Pass
			none or not existing	abort, workout with the workout name does not exist	Pass
progres_controller.py	print all progres records of logged in user	@progres.get("/")	none	all progres record data sets	Pass
	print progres record of a speific id	@progres.get("/<int:id>")	valid id	progres record	Pass
			incorrect id	abort, progres with id does not exist	Pass
	Create new progres record with the user id of logged in user	@progres.post("/")	Valid data set	progres record	Pass
			incomplete	400, missing key error	Pass
	Delete progres record (only admin or user who created it can delete)	@progres.delete("/<string:progres_name>")	valid progres	user id and name of deleted progres	Pass
			none or not existing	abort, progres with the progres name does not exist	Pass
	Amend progres record (only user who created it)	@progres.put("/update/<string:progres_name>")	valid progres and data set	set of updated progres	Pass
			none or not existing	abort, progres with the progres name does not exist	Pass
workout_exe_controller.py	Display all exercises belonging to a workout	@workout_exercise.get("/<string:workout_name>")	valid workout	workout data set and nested exercises	Pass
			none or not existing	400, missing key error	Pass
	Create 4 new workout_exercise entries, exercise random for an existing workout	@workout_exercise.post("/<int:workout_id>")	valid workout_id	4 workout_exercise records	Pass
	valid workout id required as secondary key		none or not existing	abort, workout with id does not exist	Pass
	Manually add exercises to workout	@workout_exercise.post("/<string:workout_name>")	valid workout name and exercise id	name of added exercise and workout	Pass
	Display workout exercise record	@workout_exercise.get("/<int:id>")	valid id	exercise data set	Pass
			incorrect id	abort, workout with id does not exist	Pass

Module	Function	Endpoint	Test data	Expected return	Test result
exercise_controller.py	print all exercises	@exercise.get("/")	none	all exercise data sets	Pass
	print all exercises for a specific body region	@exercise.get("/<string:body_region>")	valid body region	exercise data sets for body region	Pass
			incorrect / no body region	abort, no exercises for body region created yet	Pass
	Create new exercise (admin only)	@exercise.post("/")	valid data set	data set of created exercise	Pass
			none or missing	400, missing key error	Pass
	Delete exercise (only admin)	@exercise.delete("/<int:id>")	valid id	name and id of deleted exercise	Pass
			incorrect id	abort, exercise with id does not exist	Pass
	Amend exercise (only admin)	@exercise.put("/update")	valid data set	name, id and level of updated exercise	Pass
			none or missing	400, missing key error	Pass