

Test Charter 2

Time	Reporter	Type	Content	Screenshot
5:02 PM 10/23/2020	Arneet Singh	Session Reporter	Kevin Li - 260862327 (kevin.li2@mail.mcgill.ca) Arneet Singh - 260865569 (arneet.kalra@mail.mcgill.ca)	
5:02 PM 10/23/2020	Arneet Singh	Session Tester	Kevin Li	
5:03 PM 10/23/2020	Arneet Singh	Session Charter	For each capability create a script or small program to demonstrate the capability.	
5:03 PM 10/23/2020	Arneet Singh	Setup	Windows 10 64-bit Postman v7.34.0 Python 3.8.2	
5:03 PM 10/23/2020	Arneet Singh	Note	Start rest api todo list manager	
5:04 PM 10/23/2020	Arneet Singh	Command	Create multiple projects with different titles. With POST /projects	<pre>for i in range(len(someProjects)): projectData = {"title": someProjects[i]} r = requests.post(url=URL + "/projects", json=projectData) print(r.json())</pre>
5:04 PM 10/23/2020	Arneet Singh	Result	Success Printed project json have correct titles assigned to them	[{"id": "7", "title": "FACC 200", "completed": "false", "active": "false", "description": ""}, {"id": "8", "title": "FACC 205", "completed": "false", "active": "false", "description": ""}, {"id": "9", "title": "ECSE 308", "completed": "false", "active": "false", "description": ""}, {"id": "10", "title": "ECSE 309", "completed": "false", "active": "false", "description": ""}, {"id": "11", "title": "ECSE 310", "completed": "false", "active": "false", "description": ""}]

5:05 PM 10/23/2020	Arneet Singh	Command	Create multiple categories with different titles. With POST /categories	<pre>for i in range(len(someCategories)): categoryData = {"title": someCategories[i]} r = requests.post(url=URL + "/categories", json=categoryData) print(r.json())</pre>
5:05 PM 10/23/2020	Arneet Singh	Result	Success Printed categories json have correct titles assigned to them	<pre>{"id": "11", "title": "Work From Home", "description": ""} {"id": "12", "title": "Groceries", "description": ""} {"id": "13", "title": "House Keeping", "description": ""} {"id": "14", "title": "Free Time", "description": ""} {"id": "15", "title": "Extra", "description": ""} {"id": "16", "title": "Homework", "description": ""} {"id": "17", "title": "Assignments", "description": ""} {"id": "18", "title": "Work From Garden", "description": ""}</pre>
5:07 PM 10/23/2020	Arneet Singh	Command	Create multiple tasks with different titles. With POST /todos	<pre>for i in range(len(someTodos)): todosData = {"title": someTodos[i]} r = requests.post(url=URL + "/todos", json=todosData) print(r.json())</pre>
5:07 PM 10/23/2020	Arneet Singh	Result	Success Printed title json have correct titles assigned to them	<pre>{"id": "7", "title": "readwrite pt1", "doneStatus": "false", "description": ""} {"id": "8", "title": "write pt2", "doneStatus": "false", "description": ""} {"id": "9", "title": "test pt1", "doneStatus": "false", "description": ""} {"id": "10", "title": "test pt2", "doneStatus": "false", "description": ""}</pre>

5:09 PM 10/23/2020	Arneet Singh	Command	Randomly create relationships between projects and tasks and projects with categories.	<pre> for i in range(len(someProjects)): for j in range(random.randint(0, len(someCategories))): categoryData = {"id": str(random.randint(2, len(someCategories)))} r = requests.post(url=URL + "/projects/" + str(i+1) + "/categories", json=categoryData) for i in range(len(someProjects)): for j in range(random.randint(0, random.randint(0, len(someTodos)))): todosData = {"id": str(random.randint(2, len(someTodos)))} r = requests.post(url=URL + "/projects/" + str(i+1) + "/tasks", json=todosData) print("=====NEW PROJECTS=====") r = requests.get(url=URL + "/projects") print(r.json()) </pre>
5:09 PM 10/23/2020	Arneet Singh	Result	Success Printed all new projects with relationships	<pre> =====NEW PROJECTS===== {"projects": [{"id": "2", "title": "FACC 200", "completed": "false", "active": "false", "description": "", "tasks": [{"id": "3"}]}, {"categories": [{"id": "7"}, {"id": "8"}]}, {"id": "11", "title": "ECSE 310", "completed": "false", "active": "false", "description": ""}, {"id": "4", "title": "ECSE 308", "completed": "false", "active": "false", "description": ""}, {"id": "10", "title": "ECSE 309", "completed": "false", "active": "false", "description": ""}], "tasks": [{"id": "3"}], "categories": [{"id": "4"}, {"id": "3"}, {"id": "8"}]}, {"id": "8", "title": "FACC 205", "completed": "false", "active": "false", "description": ""}, {"tasks": [{"id": "3"}]}, {"id": "1", "title": "Office Work", "completed": "false", "active": "false", "description": ""}, {"tasks": [{"id": "4"}]}, {"id": "5", "title": "FACC 205", "completed": "false", "active": "false", "description": ""}, {"tasks": [{"id": "4"}]}, {"id": "2", "title": "ECSE 310", "completed": "false", "active": "false", "description": ""}, {"tasks": [{"id": "1"}]}, {"id": "9", "title": "ECSE 308", "completed": "false", "active": "false", "description": ""}, {"tasks": [{"id": "7"}]}] </pre>
5:10 PM 10/23/2020	Arneet Singh	Command	Using POST, amend description of newly created todos.	<pre> todosJson = todos.json().get("todos") for i in range(sizeTodos): todosID = todosJson[i].get("id") todosData = {"description": "POST ammend " + todosJson[i].get("title") + ". ID#:" + todosID} r = requests.post(url=URL + "/todos/" + todosID, json=todosData) todos = requests.get(url=URL + "/todos") print(todos.json()) </pre>
5:10 PM 10/23/2020	Arneet Singh	Result	Success Newly todos have updated description	<pre> =====POST /todos/:id===== ["todos": [{"id": "2", "title": "file paperwork", "doneStatus": "false", "description": "POST ammend file paperwork. ID# 2"}, {"id": "4", "title": "write pt2", "doneStatus": "false", "description": "POST ammend write pt2. ID# 4"}, {"id": "9", "title": "test pt1", "doneStatus": "false", "description": "POST ammend test pt1. ID# 9"}, {"id": "8", "title": "write pt2", "doneStatus": "false", "description": "POST ammend write pt2. ID# 8"}, {"id": "1", "title": "scan paperwork", "doneStatus": "false", "description": "POST ammend scan paperwork. ID# 1"}, {"id": "1", "title": "test pt2", "doneStatus": "false", "description": "POST ammend test pt2. ID# 6"}, {"id": "3", "title": "read/write pt1", "doneStatus": "false", "description": "POST ammend read/write pt1. ID# 3"}, {"id": "7", "title": "read/write pt1", "doneStatus": "false", "description": "POST ammend read/write pt1. ID# 7"}, {"id": "10", "title": "test pt2", "doneStatus": "false", "description": "POST ammend test pt2. ID# 5"}, {"id": "5", "title": "test pt1", "doneStatus": "false", "description": "POST ammend test pt1. ID# 5"}]} </pre>

5:10 PM 10/23/2020	Arneet Singh	Command	Using PUT, amend description of newly created todos.	<pre> todos = requests.get(url=URL + "/todos") todosJson = todos.json().get("todos") # PUT /todos/:id sizeTodos = len(todos.json().get("todos")) for i in range(sizeTodos): todosID = todosJson[i].get("id") todosData = {"description": "PUT ammend " + todosJson[i].get("title") + ". ID#: " + todosID} r = requests.put(url=URL + "/todos/" + todosID, json=todosData) todos = requests.get(url=URL + "/todos") print(todos.json()) </pre>
5:11 PM 10/23/2020	Arneet Singh	Result	Failure New todos have updated descriptions. However they have been reset.	<pre> =====PUT /todos/:id===== {'todos': [{"id": "2", "title": "file paperwork", "doneStatus": "false", "description": "POST ammend file paperwork. ID#: 2"}, {"id": "4", "title": "write pt2", "doneStatus": "false", "description": "POST ammend write pt2. ID#: 4"}, {"id": "9", "title": "test pt1", "doneStatus": "false", "description": "POST ammend test pt1. ID#: 9"}, {"id": "8", "title": "write pt2", "doneStatus": "false", "description": "POST ammend write pt2. ID#: 8"}, {"id": "1", "title": "scan paperwork", "doneStatus": "false", "description": "POST ammend scan paperwork. ID#: 1"}, {"id": "6", "title": "test pt2", "doneStatus": "false", "description": "POST ammend test pt2. ID#: 6"}, {"id": "3", "title": "readwrite pt1", "doneStatus": "false", "description": "POST ammend readwrite pt1. ID#: 3"}, {"id": "7", "title": "readwrite pt1", "doneStatus": "false", "description": "POST ammend readwrite pt1. ID#: 7"}, {"id": "10", "title": "test pt2", "doneStatus": "false", "description": "POST ammend test pt2. ID#: 10"}, {"id": "5", "title": "test pt1", "doneStatus": "false", "description": "POST ammend test pt1. ID#: 5"}]} </pre>
5:12 PM 10/23/2020	Arneet Singh	Command	Using POST, create a relationship between task and the first category	<pre> categories = requests.get(url=URL + "/categories") categoriesJson = categories.json().get("categories") # POST /todos/:id/categories todosJson = todos.json().get("todos") sizeTodos = len(todos.json().get("todos")) for i in range(sizeTodos): todosID = todosJson[i].get("id") try: todosData = {"id": categoriesJson[0].get("id")} r = requests.post(url=URL + "/todos/" + todosID + "/categories", json=todosData) except: print("NO CATEGORIES") todos = requests.get(url=URL + "/todos") print(todos.json()) </pre>
5:12 PM 10/23/2020	Arneet Singh	Result	Success GET todos returns all todos with the same first category	<pre> =====POST /todos/:id/categories===== {'todos': [{"id": "2", "title": "file paperwork", "doneStatus": "false", "description": "POST ammend file paperwork. ID#: 2", "categories": [{"id": "17"}]}, {"id": "4", "title": "write pt2", "doneStatus": "false", "description": "POST ammend write pt2. ID#: 4", "categories": [{"id": "17"}]}, {"id": "9", "title": "test pt1", "doneStatus": "false", "description": "POST ammend test pt1. ID#: 9", "categories": [{"id": "17"}]}, {"id": "8", "title": "write pt2", "doneStatus": "false", "description": "POST ammend write pt2. ID#: 8", "categories": [{"id": "17"}]}, {"id": "1", "title": "scan paperwork", "doneStatus": "false", "description": "POST ammend scan paperwork. ID#: 1", "categories": [{"id": "17"}]}, {"id": "6", "title": "test pt2", "doneStatus": "false", "description": "POST ammend test pt2. ID#: 6", "categories": [{"id": "17"}]}, {"id": "3", "title": "readwrite pt1", "doneStatus": "false", "description": "POST ammend readwrite pt1. ID#: 3", "categories": [{"id": "17"}]}, {"id": "7", "title": "readwrite pt1", "doneStatus": "false", "description": "POST ammend readwrite pt1. ID#: 7", "categories": [{"id": "17"}]}, {"id": "10", "title": "test pt2", "doneStatus": "false", "description": "POST ammend test pt2. ID#: 10", "categories": [{"id": "17"}]}, {"id": "5", "title": "test pt1", "doneStatus": "false", "description": "POST ammend test pt1. ID#: 5", "categories": [{"id": "17"}]}]} </pre>

5:13 PM 10/23/2020	Arneet Singh	Command	Using DELETE, to remove relationship between the first category and all tasks	<pre> todos = requests.get(url=URL + "/todos") todosJson = todos.json().get("todos") sizeTodos = len(todos.json().get("todos")) for i in range(sizeTodos): todosID = todosJson[i].get("id") try: r = requests.delete(url=URL + "/todos/" + todosID + "/categories/" + requests.get(url=URL + "/todos").json().get("todos")[i].get("categories")[-1].get("id")) except: print("TODOS " + todosID + " has no categories") todos = requests.get(url=URL + "/todos") print(todos.json()) </pre>
5:13 PM 10/23/2020	Arneet Singh	Result	Success GET todos returns all todos without the first category	<pre> =====DELETE /todos/:id/categories/:id===== [{"todos": [{"id": "2", "title": "file paperwork", "doneStatus": "false", "description": "POST ammend file paperwork. ID#: 2"}, {"id": "4", "title": "write pt2", "doneStatus": "false", "description": "POST ammend write pt2. ID#: 4"}, {"id": "9", "title": "test pt1", "doneStatus": "false", "description": "POST ammend test pt1. ID#: 9"}, {"id": "8", "title": "write pt2", "doneStatus": "false", "description": "POST ammend write pt2. ID#: 8"}, {"id": "1", "title": "scan paperwork", "doneStatus": "false", "description": "POST ammend scan paperwork. ID#: 1"}, {"id": "6", "title": "test pt2", "doneStatus": "false", "description": "POST ammend test pt2. ID#: 6"}, {"id": "3", "title": "read/write pt1", "doneStatus": "false", "description": "POST ammend read/write pt1. ID#: 3"}, {"id": "7", "title": "read/write pt1", "doneStatus": "false", "description": "POST ammend read/write pt1. ID#: 7"}, {"id": "10", "title": "test pt2", "doneStatus": "false", "description": "POST ammend test pt2. ID#: 10"}, {"id": "5", "title": "test pt1", "doneStatus": "false", "description": "POST ammend test pt1. ID#: 5"}]} </pre>
5:16 PM 10/23/2020	Arneet Singh	Command	Using POST, create a relationship between task and the first project	<pre> projects = requests.get(url=URL + "/projects") projectsJson = projects.json().get("projects") # POST /todos/:id/projects todos = requests.get(url=URL + "/todos") todosJson = todos.json().get("todos") sizeTodos = len(todos.json().get("todos")) for i in range(sizeTodos): todosID = todosJson[i].get("id") try: todosData = {"id": projectsJson[0].get("id")} r = requests.post(url=URL + "/todos/" + todosID + "/tasksof", json=todosData) except: print("NO PROJECTS") todos = requests.get(url=URL + "/todos") print(todos.json()) </pre>
5:16 PM 10/23/2020	Arneet Singh	Result	Success GET todos returns all todos with the same first project	<pre> =====POST /todos/:id/taskof===== [{"todos": [{"id": "2", "title": "file paperwork", "doneStatus": "false", "description": "POST ammend file paperwork. ID#: 2", "tasksof": [{"id": "1"}]}, {"id": "4", "title": "write pt2", "doneStatus": "false", "description": "POST ammend write pt2. ID#: 4", "tasksof": [{"id": "1"}]}, {"id": "9", "title": "test pt1", "doneStatus": "false", "description": "POST ammend test pt1. ID#: 9", "tasksof": [{"id": "1"}]}, {"id": "8", "title": "write pt2", "doneStatus": "false", "description": "POST ammend write pt2. ID#: 8", "tasksof": [{"id": "1"}]}, {"id": "1", "title": "scan paperwork", "doneStatus": "false", "description": "POST ammend scan paperwork. ID#: 1", "categories": [{"id": "1"}], "tasksof": [{"id": "1"}]}, {"id": "6", "title": "test pt2", "doneStatus": "false", "description": "POST ammend test pt2. ID#: 6", "tasksof": [{"id": "1"}]}, {"id": "3", "title": "read/write pt1", "doneStatus": "false", "description": "POST ammend read/write pt1. ID#: 3", "tasksof": [{"id": "1"}]}, {"id": "7", "title": "read/write pt1", "doneStatus": "false", "description": "POST ammend read/write pt1. ID#: 7", "tasksof": [{"id": "1"}]}, {"id": "10", "title": "test pt2", "doneStatus": "false", "description": "POST ammend test pt2. ID#: 10", "tasksof": [{"id": "1"}]}, {"id": "5", "title": "test pt1", "doneStatus": "false", "description": "POST ammend test pt1. ID#: 5", "tasksof": [{"id": "1"}]}]} </pre>

5:16 PM 10/23/2020	Arneet Singh	Command	Using DELETE, to remove relationship between the first project and all tasks	<pre>projects = requests.get(url=URL + "/projects") projectsJson = projects.json().get("projects") todos = requests.get(url=URL + "/todos") todosJson = todos.json().get("todos") for i in range(sizeTodos): todosID = todosJson[i].get("id") try: r = requests.delete(url=URL + "/todos/" + todosID + "/projects/" + requests.get(url=URL + "/todos").json().get("todos")[i].get("taskof")[0].get("id")) except: print("TODOS " + todosID + " has no projects") todos = requests.get(url=URL + "/todos") print(todos.json())</pre>
5:17 PM 10/23/2020	Arneet Singh	Result	Success GET todos returns all todos without the first project	<pre>=====DELETE /todos/:id/projects/:id===== {'todos': [{'id': '2', 'title': 'file paperwork', 'doneStatus': 'false', 'description': 'POST ammend file paperwork. ID#: 2', 'tasksof': [{'id': '1'}]}, {'id': '4', 'title': 'write pt2', 'doneStatus': 'false', 'description': 'POST ammend write pt2. ID#: 4', 'tasksof': [{'id': '1'}]}, {'id': '9', 'title': 'test pt1', 'doneStatus': 'false', 'description': 'POST ammend test pt1. ID#: 9', 'tasksof': [{'id': '1'}]}, {'id': '8', 'title': 'write pt2', 'doneStatus': 'false', 'description': 'POST ammend write pt2. ID#: 8', 'tasksof': [{'id': '1'}]}, {'id': '1', 'title': 'scan paperwork', 'doneStatus': 'false', 'description': 'POST ammend scan paperwork. ID#: 1', 'categories': [{"id": "1"}], 'tasksof': [{"id": "1"}]}, {"id": "6", "title": "test pt2", "doneStatus": "false", "description": "POST ammend test pt2. ID#: 6", "tasksof": [{"id": "1"}]}, {"id": "3", "title": "read/write pt1", "doneStatus": "false", "description": "POST ammend read/write pt1. ID#: 3", "tasksof": [{"id": "1"}]}, {"id": "7", "title": "read/write pt1", "doneStatus": "false", "description": "POST ammend read/write pt1. ID#: 7", "tasksof": [{"id": "1"}]}, {"id": "10", "title": "test pt2", "doneStatus": "false", "description": "POST ammend test pt2. ID#: 10", "tasksof": [{"id": "1"}]}, {"id": "5", "title": "test pt1", "doneStatus": "false", "description": "POST ammend test pt1. ID#: 5", "tasksof": [{"id": "1"}]}}</pre>
5:17 PM 10/23/2020	Arneet Singh	Command	Using POST, amend description of newly created projects.	<pre>projects = requests.get(url=URL + "/projects") projectsJson = projects.json().get("projects") sizeProjects = len(projects.json().get("projects")) todos = requests.get(url=URL + "/todos") for i in range(sizeProjects): projectsID = projectsJson[i].get("id") projectData = {"description": "POST ammend " + projectsJson[i].get("title") + ". ID#: " + projectsID} r = requests.post(url=URL + "/projects/" + projectsID, json=projectData) projects = requests.get(url=URL + "/projects") print(projects.json())</pre>
5:17 PM 10/23/2020	Arneet Singh	Result	Success Newly projects have updated description	<pre>=====POST /projects/:id===== {'projects': [{'id': '1', 'title': '', 'completed': 'false', 'active': 'false', 'description': 'POST ammend . ID#: 1', 'tasks': [{'id': '7'}, {"id": "8"}, {"id": "10"}, {"id": "1"}, {"id": "6"}, {"id": "2"}, {"id": "5"}, {"id": "4"}, {"id": "9"}, {"id": "3"}]}, {"id": "5", "title": "", "completed": "false", "active": "false", "description": "POST ammend . ID#: 5"}, {"id": "9", "title": "", "completed": "false", "active": "false", "description": "POST ammend . ID#: 9"}, {"id": "3", "title": "", "completed": "false", "active": "false", "description": "POST ammend . ID#: 3"}, {"id": "8", "title": "", "completed": "false", "active": "false", "description": "POST ammend . ID#: 8"}, {"id": "11", "title": "", "completed": "false", "active": "false", "description": "POST ammend . ID#: 11"}, {"id": "4", "title": "", "completed": "false", "active": "false", "description": "POST ammend . ID#: 4"}, {"id": "10", "title": "", "completed": "false", "active": "false", "description": "POST ammend . ID#: 10"}, {"id": "6", "title": "", "completed": "false", "active": "false", "description": "POST ammend . ID#: 6"}, {"id": "7", "title": "", "completed": "false", "active": "false", "description": "POST ammend . ID#: 7"}, {"id": "2", "title": "", "completed": "false", "active": "false", "description": "POST ammend . ID#: 2"}]</pre>

5:19 PM 10/23/2020	Arneet Singh	Command	Using PUT, amend description of newly created projects.	<pre>projects = requests.get(url=URL + "/projects") projectsJson = projects.json().get("projects") sizeProjects = len(projects.json().get("projects")) for i in range(sizeProjects): projectsID = projectsJson[i].get("id") projectData = {"description": "PUT ammend " + projectsJson[i].get("title") + ". ID#: " + projectsID} r = requests.put(url=URL + "/projects/" + projectsID, json=projectData) projects = requests.get(url=URL + "/projects") print(projects.json())</pre>
5:19 PM 10/23/2020	Arneet Singh	Result	Failure New projects have updated descriptions. However they have been reset.	<pre>=====PUT /projects:id===== [{"projects": [{"id": "1", "title": "", "completed": "false", "active": "false", "description": "PUT ammend . ID#: 1"}, {"id": "5", "title": "", "completed": "false", "active": "false", "description": "PUT ammend . ID#: 5"}, {"id": "9", "title": "", "completed": "false", "active": "false", "description": "PUT ammend . ID#: 9"}, {"id": "3", "title": "", "completed": "false", "active": "false", "description": "PUT ammend . ID#: 3"}, {"id": "8", "title": "", "completed": "false", "active": "false", "description": "PUT ammend . ID#: 8"}, {"id": "11", "title": "", "completed": "false", "active": "false", "description": "PUT ammend . ID#: 11"}, {"id": "4", "title": "", "completed": "false", "active": "false", "description": "PUT ammend . ID#: 4"}, {"id": "10", "title": "", "completed": "false", "active": "false", "description": "PUT ammend . ID#: 10"}, {"id": "6", "title": "", "completed": "false", "active": "false", "description": "PUT ammend . ID#: 6"}, {"id": "7", "title": "", "completed": "false", "active": "false", "description": "PUT ammend . ID#: 7"}, {"id": "2", "title": "", "completed": "false", "active": "false", "description": "PUT ammend . ID#: 2"}]}</pre>
5:19 PM 10/23/2020	Arneet Singh	Command	Using POST, create a relationship between project and the first task	<pre>todos = requests.get(url=URL + "/todos") todosJson = todos.json().get("todos") projects = requests.get(url=URL + "/projects") projectsJson = projects.json().get("projects") sizeProjects = len(projects.json().get("projects")) for i in range(sizeProjects): projectsID = projectsJson[i].get("id") try: projectData = {"id": todosJson[0].get("id")} r = requests.post(url=URL + "/projects/" + projectsID + "/tasks", json=projectData) except: print("NO TASKS") todos = requests.get(url=URL + "/projects") print(todos.json())</pre>
5:20 PM 10/23/2020	Arneet Singh	Result	Success GET todos returns all projects with the same first task	<pre>=====POST /projects:id/tasks===== [{"projects": [{"id": "1", "title": "", "completed": "false", "active": "false", "description": "PUT ammend . ID#: 1"}, {"tasks": [{"id": "2"}]}, {"id": "5", "title": "", "completed": "false", "active": "false", "description": "PUT ammend . ID#: 5", "tasks": [{"id": "2"}]}, {"id": "9", "title": "", "completed": "false", "active": "false", "description": "PUT ammend . ID#: 9", "tasks": [{"id": "2"}]}, {"id": "3", "title": "", "completed": "false", "active": "false", "description": "PUT ammend . ID#: 3", "tasks": [{"id": "2"}]}, {"id": "8", "title": "", "completed": "false", "active": "false", "description": "PUT ammend . ID#: 8", "tasks": [{"id": "2"}]}, {"id": "11", "title": "", "completed": "false", "active": "false", "description": "PUT ammend . ID#: 11", "tasks": [{"id": "2"}]}, {"id": "4", "title": "", "completed": "false", "active": "false", "description": "PUT ammend . ID#: 4", "tasks": [{"id": "2"}]}, {"id": "10", "title": "", "completed": "false", "active": "false", "description": "PUT ammend . ID#: 10", "tasks": [{"id": "2"}]}, {"id": "6", "title": "", "completed": "false", "active": "false", "description": "PUT ammend . ID#: 6", "tasks": [{"id": "2"}]}, {"id": "7", "title": "", "completed": "false", "active": "false", "description": "PUT ammend . ID#: 7", "tasks": [{"id": "2"}]}, {"id": "2", "title": "", "completed": "false", "active": "false", "description": "PUT ammend . ID#: 2", "tasks": [{"id": "2"}]}]}</pre>

5:20 PM 10/23/2020	Arneet Singh	Command	Using DELETE, to remove relationship between the first task and all projects	<pre> todos = requests.get(url=URL + "/todos") todosJson = todos.json().get("todos") projects = requests.get(url=URL + "/projects") projectsJson = projects.json().get("projects") sizeProjects = len(projects.json().get("projects")) for i in range(sizeProjects): projectsID = projectsJson[i].get("id") try: r = requests.delete(url=URL + "/projects/" + projectsID + "/tasks/" + requests.get(url=URL + "/projects").json().get("projects")[i].get("tasks")[0].get("id")) except: print("PROJECTS " + projectsID + " has no tasks") projects = requests.get(url=URL + "/projects") print(projects.json()) </pre>
5:20 PM 10/23/2020	Arneet Singh	Result	Success GET todos returns all todos without the first task	<pre> ===== DELETE /projects/:id/tasks/:id===== [{"projects": [{"id": "1", "title": "", "completed": "false", "active": "false", "description": "PUT ammend . ID#: 1"}, {"id": "5", "title": "", "completed": "false", "active": "false", "description": "PUT ammend . ID#: 5"}, {"id": "9", "title": "", "completed": "false", "active": "false", "description": "PUT ammend . ID#: 9"}, {"id": "3", "title": "", "completed": "false", "active": "false", "description": "PUT ammend . ID#: 3"}, {"id": "8", "title": "", "completed": "false", "active": "false", "description": "PUT ammend . ID#: 8"}, {"id": "11", "title": "", "completed": "false", "active": "false", "description": "PUT ammend . ID#: 11"}, {"id": "4", "title": "", "completed": "false", "active": "false", "description": "PUT ammend . ID#: 4"}, {"id": "10", "title": "", "completed": "false", "active": "false", "description": "PUT ammend . ID#: 10"}, {"id": "6", "title": "", "completed": "false", "active": "false", "description": "PUT ammend . ID#: 6"}, {"id": "7", "title": "", "completed": "false", "active": "false", "description": "PUT ammend . ID#: 7"}, {"id": "2", "title": "", "completed": "false", "active": "false", "description": "PUT ammend . ID#: 2"}]} </pre>
5:20 PM 10/23/2020	Arneet Singh	Command	Using POST, create a relationship between project and the first category	<pre> categories = requests.get(url=URL + "/categories") categoriesJson = categories.json().get("categories") projects = requests.get(url=URL + "/projects") projectsJson = projects.json().get("projects") sizeProjects = len(projects.json().get("projects")) for i in range(sizeProjects): projectsID = projectsJson[i].get("id") try: projectData = {"id": categoriesJson[0].get("id")} r = requests.post(url=URL + "/projects/" + projectsID + "/categories", json=projectData) except: print("NO CATEGORIES") projects = requests.get(url=URL + "/projects") print(projects.json()) </pre>
5:20 PM 10/23/2020	Arneet Singh	Result	Success GET todos returns all projects with the same first category	<pre> ===== POST /projects/:id/categories===== [{"projects": [{"id": "1", "title": "", "completed": "false", "active": "false", "description": "PUT ammend . ID#: 1", "categories": [{"id": "17"}]}, {"id": "5", "title": "", "completed": "false", "active": "false", "description": "PUT ammend . ID#: 5", "categories": [{"id": "17"}]}, {"id": "9", "title": "", "completed": "false", "active": "false", "description": "PUT ammend . ID#: 9", "categories": [{"id": "17"}]}, {"id": "3", "title": "", "completed": "false", "active": "false", "description": "PUT ammend . ID#: 3", "categories": [{"id": "17"}]}, {"id": "8", "title": "", "completed": "false", "active": "false", "description": "PUT ammend . ID#: 8", "categories": [{"id": "17"}]}, {"id": "11", "title": "", "completed": "false", "active": "false", "description": "PUT ammend . ID#: 11", "categories": [{"id": "17"}]}, {"id": "4", "title": "", "completed": "false", "active": "false", "description": "PUT ammend . ID#: 4", "categories": [{"id": "17"}]}, {"id": "10", "title": "", "completed": "false", "active": "false", "description": "PUT ammend . ID#: 10", "categories": [{"id": "17"}]}, {"id": "6", "title": "", "completed": "false", "active": "false", "description": "PUT ammend . ID#: 6", "categories": [{"id": "17"}]}, {"id": "7", "title": "", "completed": "false", "active": "false", "description": "PUT ammend . ID#: 7", "categories": [{"id": "17"}]}, {"id": "2", "title": "", "completed": "false", "active": "false", "description": "PUT ammend . ID#: 2", "categories": [{"id": "17"}]}]} </pre>

5:21 PM 10/23/2020	Arneet Singh	Command	Using DELETE, to remove relationship between the first category and all projects	<pre>categories = requests.get(url=URL + "/categories") categoriesJson = categories.json().get("categories") projects = requests.get(url=URL + "/projects") projectsJson = projects.json().get("projects") sizeProjects = len(projects.json().get("projects")) for i in range(sizeProjects): projectsID = projectsJson[i].get("id") try: r = requests.delete(url=URL + "/projects/" + projectsID + "/categories/" + requests.get(url=URL + "/projects").json().get("projects")[i].get("categories")[0].get("id")) except: print("PROJECTS " + projectsID + " has no categories") projects = requests.get(url=URL + "/projects") print(projects.json())</pre>
5:21 PM 10/23/2020	Arneet Singh	Result	Success GET todos returns all todos without the first category	<pre>=====DELETE /projects/:id/categories/:id===== {'projects': [{'id': '1', 'title': '', 'completed': 'false', 'active': 'false', 'description': 'PUT ammend . ID#: 1'}, {'id': '5', 'title': '', 'completed': 'false', 'active': 'false', 'description': 'PUT ammend . ID#: 5'}, {'id': '9', 'title': '', 'completed': 'false', 'active': 'false', 'description': 'PUT ammend . ID#: 9'}, {'id': '3', 'title': '', 'completed': 'false', 'active': 'false', 'description': 'PUT ammend . ID#: 3'}, {'id': '8', 'title': '', 'completed': 'false', 'active': 'false', 'description': 'PUT ammend . ID#: 8'}, {'id': '11', 'title': '', 'completed': 'false', 'active': 'false', 'description': 'PUT ammend . ID#: 11'}, {'id': '4', 'title': '', 'completed': 'false', 'active': 'false', 'description': 'PUT ammend . ID#: 4'}, {'id': '10', 'title': '', 'completed': 'false', 'active': 'false', 'description': 'PUT ammend . ID#: 10'}, {'id': '6', 'title': '', 'completed': 'false', 'active': 'false', 'description': 'PUT ammend . ID#: 6'}, {'id': '7', 'title': '', 'completed': 'false', 'active': 'false', 'description': 'PUT ammend . ID#: 7'}, {'id': '2', 'title': '', 'completed': 'false', 'active': 'false', 'description': 'PUT ammend . ID#: 2'}]}</pre>
5:21 PM 10/23/2020	Arneet Singh	Command	Using PUT, amend description of newly created categories.	<pre>categories = requests.get(url=URL + "/categories") categoriesJson = categories.json().get("categories") sizeCategories = len(categories.json().get("categories")) for i in range(sizeCategories): categoriesID = categoriesJson[i].get("id") categoryData = {"description": "PUT ammend " + categoriesJson[i].get("title") + ". ID#: " + categoriesID} r = requests.put(url=URL + "/categories/" + categoriesID, json=categoryData) categories = requests.get(url=URL + "/categories") print(categories.json())</pre>
5:21 PM 10/23/2020	Arneet Singh	Result	Failure New categories have updated descriptions. However they have been reset.	<pre>=====PUT /categories/:id===== {'categories': [{'id': '17', 'title': 'Assignments', 'description': 'POST ammend Assignments. ID#: 17'}, {'id': '6', 'title': 'Free Time', 'description': 'POST ammend Free Time. ID#: 6'}, {'id': '1', 'title': 'Office', 'description': 'POST ammend Office. ID#: 1'}, {'id': '5', 'title': 'House Keeping', 'description': 'POST ammend House Keeping. ID#: 5'}, {'id': '3', 'title': 'Work From Home', 'description': 'POST ammend Work From Home. ID#: 3'}, {'id': '11', 'title': 'Work From Home', 'description': 'POST ammend Work From Home. ID#: 11'}, {'id': '4', 'title': 'Groceries', 'description': 'POST ammend Groceries. ID#: 4'}, {'id': '13', 'title': 'House Keeping', 'description': 'POST ammend House Keeping. ID#: 13'}, {'id': '12', 'title': 'Groceries', 'description': 'POST ammend Groceries. ID#: 12'}, {'id': '15', 'title': 'Extra', 'description': 'POST ammend Extra. ID#: 15'}, {'id': '14', 'title': 'Free Time', 'description': 'POST ammend Free Time. ID#: 14'}, {'id': '16', 'title': 'Homework', 'description': 'POST ammend Homework. ID#: 16'}, {'id': '18', 'title': 'Work From Garden', 'description': 'POST ammend Work From Garden. ID#: 18'}, {'id': '2', 'title': 'Home', 'description': 'POST ammend Home. ID#: 2'}, {'id': '8', 'title': 'Homework', 'description': 'POST ammend Home work. ID#: 8'}, {'id': '10', 'title': 'Work From Garden', 'description': 'POST ammend Work From Garden. ID#: 10'}, {'id': '9', 'title': 'Assignments', 'description': 'POST ammend Assignments. ID#: 9'}, {'id': '7', 'title': 'Extra', 'description': 'POST ammend Extra. ID#: 7'}]}</pre>

5:22 PM 10/23/2020	Arneet Singh	Command	Using POST, amend description of newly created projects.	<pre>categories = requests.get(url=URL + "/categories") categoriesJson = categories.json().get("categories") sizeCategories = len(categories.json().get("categories")) for i in range(sizeCategories): categoriesID = categoriesJson[i].get("id") categoryData = {"description": "POST ammend " + categoriesJson[i].get("title") + ". ID#: " + categoriesID} r = requests.post(url=URL + "/categories/" + categoriesID, json=categoryData) categories = requests.get(url=URL + "/categories") print(categories.json())</pre>
5:22 PM 10/23/2020	Arneet Singh	Result	Success Newly categories have updated description	<pre>=====POST /categories:id===== [{"categories": [{"id": "17", "title": "Assignments", "description": "POST ammend Assignments. ID#: 17"}, {"id": "6", "title": "Free Time", "description": "POST ammend Free Time. ID#: 6"}, {"id": "1", "title": "Office", "description": "POST ammend Office. ID#: 1"}, {"id": "5", "title": "House Keeping", "description": "POST ammend House Keeping. ID#: 5"}, {"id": "3", "title": "Work From Home", "description": "POST ammend Work From Home. ID#: 3"}, {"id": "11", "title": "Work From Home", "description": "POST ammend Work From Home. ID#: 11"}, {"id": "4", "title": "Groceries", "description": "POST ammend Groceries. ID#: 4"}, {"id": "13", "title": "House Keeping", "description": "POST ammend House Keeping. ID#: 13"}, {"id": "12", "title": "Groceries", "description": "POST ammend Groceries. ID#: 12"}, {"id": "15", "title": "Extra", "description": "POST ammend Extra. ID#: 15"}, {"id": "14", "title": "Free Time", "description": "POST ammend Free Time. ID#: 14"}, {"id": "16", "title": "Homework", "description": "POST ammend Homework. ID#: 16"}, {"id": "18", "title": "Work From Garden", "description": "POST ammend Work From Garden. ID#: 18"}, {"id": "2", "title": "Home", "description": "POST ammend Home. ID#: 2"}, {"id": "8", "title": "Homework", "description": "POST ammend Homework. ID#: 8"}, {"id": "10", "title": "Work From Garden", "description": "POST ammend Work From Garden. ID#: 10"}, {"id": "9", "title": "Assignments", "description": "POST ammend Assignments. ID#: 9"}, {"id": "7", "title": "Extra", "description": "POST ammend Extra. ID#: 7"}]}</pre>
5:23 PM 10/23/2020	Arneet Singh	Command	Using POST, create a relationship between category and the first task	<pre>todos = requests.get(url=URL + "/todos") todosJson = todos.json().get("todos") categories = requests.get(url=URL + "/categories") categoriesJson = categories.json().get("categories") sizeCategories = len(categories.json().get("categories")) for i in range(sizeCategories): categoriesID = categoriesJson[i].get("id") try: categoryData = {"id": todosJson[0].get("id")} r = requests.post(url=URL + "/categories/" + categoriesID + "/todos", json=projectData) except: print("NO TODOS") categories = requests.get(url=URL + "/categories") print(categories.json())</pre>
5:23 PM 10/23/2020	Arneet Singh	Result	Success GET /categories returns all categories with same first task	<pre>=====POST /categories:id/todos===== [{"categories": [{"id": "17", "title": "Assignments", "description": "POST ammend Assignments. ID#: 17"}, {"id": "6", "title": "Free Time", "description": "POST ammend Free Time. ID#: 6"}, {"id": "1", "title": "Office", "description": "POST ammend Office. ID#: 1"}, {"id": "5", "title": "House Keeping", "description": "POST ammend House Keeping. ID#: 5"}, {"id": "3", "title": "Work From Home", "description": "POST ammend Work From Home. ID#: 3"}, {"id": "11", "title": "Work From Home", "description": "POST ammend Work From Home. ID#: 11"}, {"id": "4", "title": "Groceries", "description": "POST ammend Groceries. ID#: 4"}, {"id": "13", "title": "House Keeping", "description": "POST ammend House Keeping. ID#: 13"}, {"id": "12", "title": "Groceries", "description": "POST ammend Groceries. ID#: 12"}, {"id": "15", "title": "Extra", "description": "POST ammend Extra. ID#: 15"}, {"id": "14", "title": "Free Time", "description": "POST ammend Free Time. ID#: 14"}, {"id": "16", "title": "Homework", "description": "POST ammend Homework. ID#: 16"}, {"id": "18", "title": "Work From Garden", "description": "POST ammend Work From Garden. ID#: 18"}, {"id": "2", "title": "Home", "description": "POST ammend Home. ID#: 2"}, {"id": "8", "title": "Homework", "description": "POST ammend Homework. ID#: 8"}, {"id": "10", "title": "Work From Garden", "description": "POST ammend Work From Garden. ID#: 10"}, {"id": "9", "title": "Assignments", "description": "POST ammend Assignments. ID#: 9"}, {"id": "7", "title": "Extra", "description": "POST ammend Extra. ID#: 7"}]}</pre>

5:23 PM 10/23/2020	Arneet Singh	Command	Using DELETE, to remove relationship between the first task and all categories	<pre> todos = requests.get(url=URL + "/todos") todosJson = todos.json().get("todos") sizeTodos = len(todos.json().get("todos")) categories = requests.get(url=URL + "/categories") categoriesJson = categories.json().get("categories") sizeCategories = len(categories.json().get("categories")) for i in range(sizeTodos): try: categoriesID = requests.get(url=URL + "/categories").json().get("categories")[0].get("id") r = requests.delete(url=URL + "/categories/" + categoriesID + "/todos/" + requests.get(url=URL + "/todos").json()[i].get("id")) except: print("CATEGORY " + categoriesID + " has no todos") categories = requests.get(url=URL + "/categories") print(categories.json()) </pre>
5:23 PM 10/23/2020	Arneet Singh	Result	Success GET /categories Returns all categories without the first task	<pre> ===== {'categories': [{'id': '17', 'title': 'Assignments', 'description': 'POST ammend Assignments. ID#: 17'}, {'id': '6', 'title': 'Free Time', 'description': 'POST ammend Free Time. ID#: 6'}, {'id': '1', 'title': 'Office', 'description': 'POST ammend Office. ID#: 1'}, {'id': '5', 'title': 'House Keeping', 'description': 'POST ammend House Keeping. ID#: 5'}, {'id': '3', 'title': 'Work From Home', 'description': 'POST ammend Work From Home. ID#: 3'}, {'id': '11', 'title': 'Work From Home', 'description': 'POST ammend Work From Home. ID#: 11'}, {'id': '4', 'title': 'Groceries', 'description': 'POST ammend Groceries. ID#: 4'}, {'id': '13', 'title': 'House Keeping', 'description': 'POST ammend House Keeping. ID#: 13'}, {'id': '12', 'title': 'Groceries', 'description': 'POST ammend Groceries. ID#: 12'}, {'id': '15', 'title': 'Extra', 'description': 'POST ammend Extra. ID#: 15'}, {'id': '14', 'title': 'Free Time', 'description': 'POST ammend Free Time. ID#: 14'}, {'id': '16', 'title': 'Homework', 'description': 'POST ammend Homework. ID#: 16'}, {'id': '18', 'title': 'Work From Garden', 'description': 'POST ammend Work From Garden. ID#: 18'}, {'id': '2', 'title': 'Home', 'description': 'POST ammend Home. ID#: 2'}, {'id': '8', 'title': 'Homework', 'description': 'POST ammend Home. ID#: 8'}, {'id': '10', 'title': 'Work From Garden', 'description': 'POST ammend Work From Garden. ID#: 10'}, {'id': '9', 'title': 'Assignments', 'description': 'POST ammend Assignments. ID#: 9'}, {'id': '7', 'title': 'Extra', 'description': 'POST ammend Extra. ID#: 7'}]} </pre>
5:23 PM 10/23/2020	Arneet Singh	Command	Using POST, create a relationship between categories and the first project	<pre> projects = requests.get(url=URL + "/projects") projectsJson = projects.json().get("projects") categories = requests.get(url=URL + "/categories") categoriesJson = categories.json().get("categories") sizeCategories = len(categories.json().get("categories")) for i in range(sizeCategories): categoriesID = categoriesJson[i].get("id") try: categoryData = {"id": projectsJson[0].get("id")} r = requests.post(url=URL + "/categories/" + categoriesID + "/projects", json=projectData) except: print("NO PROJECTS") categories = requests.get(url=URL + "/categories") print(categories.json()) </pre>

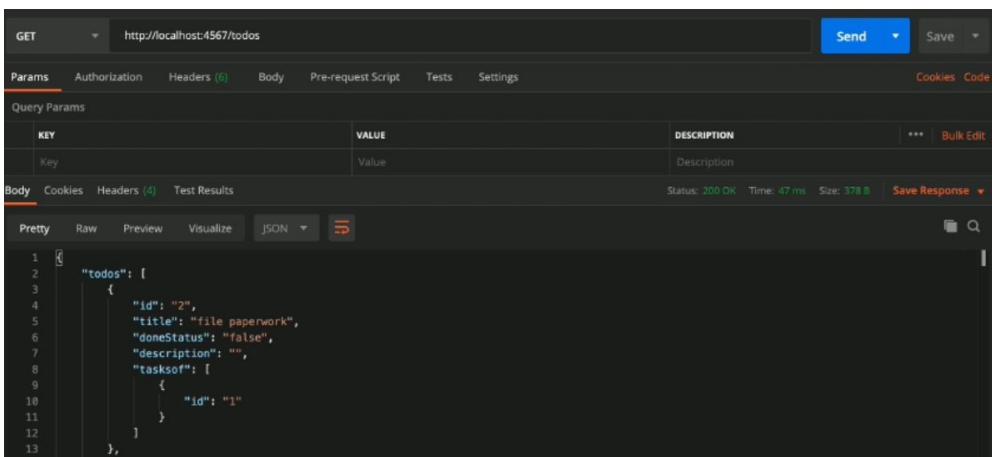
5:24 PM 10/23/2020	Arneet Singh	Result	Success GET /categories Returns all categories with the same first project	<pre>=====POST /categories/:id/projects===== [{"categories": [{"id": "17", "title": "Assignments", "description": "POST ammend Assignments. ID#: 17"}, {"id": "6", "title": "Free Time", "description": "POST ammend Free Time. ID#: 6"}, {"id": "1", "title": "Office", "description": "POST ammend Office. ID#: 1"}, {"id": "5", "title": "House Keeping", "description": "POST ammend House Keeping. ID#: 5"}, {"id": "3", "title": "Work From Home", "description": "POST ammend Work From Home. ID#: 3"}, {"id": "11", "title": "Work From Home", "description": "POST ammend Work From Home. ID#: 11"}, {"id": "4", "title": "Groceries", "description": "POST ammend Groceries. ID#: 4"}, {"id": "13", "title": "House Keeping", "description": "POST ammend House Keeping. ID#: 13"}, {"id": "12", "title": "Groceries", "description": "POST ammend Groceries. ID#: 12"}, {"id": "15", "title": "Extra", "description": "POST ammend Extra. ID#: 15"}, {"id": "14", "title": "Free Time", "description": "POST ammend Free Time. ID#: 14"}, {"id": "16", "title": "Homework", "description": "POST ammend Homework. ID#: 16"}, {"id": "18", "title": "Work From Garden", "description": "POST ammend Work From Garden. ID#: 18"}, {"id": "2", "title": "Home", "description": "POST ammend Home. ID#: 2"}, {"id": "8", "title": "Homework", "description": "POST ammend Homework. ID#: 8"}, {"id": "10", "title": "Work From Garden", "description": "POST ammend Work From Garden. ID#: 10"}, {"id": "9", "title": "Assignments", "description": "POST ammend Assignments. ID#: 9"}, {"id": "7", "title": "Extra", "description": "POST ammend Extra. ID#: 7"}]}</pre>
5:24 PM 10/23/2020	Arneet Singh	Command	Using DELETE, to remove relationship between the first project and all categories	<pre>projects = requests.get(url=URL + "/projects") projectsJson = projects.json().get("projects") sizeProjects = len(projects.json().get("projects")) categories = requests.get(url=URL + "/categories") categoriesJson = categories.json().get("categories") sizeCategories = len(categories.json().get("categories")) for i in range(sizeProjects): try: categoriesID = requests.get(url=URL + "/categories").json()[0].get("id") r = requests.delete(url=URL + "/categories/" + categoriesID + "/projects/" + requests.get(url=URL + "/projects").json().get("projects")[i].get("id")) except: print("CATEGORY " + categoriesID + " has no projects") categories = requests.get(url=URL + "/categories") print(categories.json())</pre>
5:24 PM 10/23/2020	Arneet Singh	Result	Success GET /categories Returns categories without first project	<pre>=====DELETE /categories/:id/projects/:id===== [{"categories": [{"id": "17", "title": "Assignments", "description": "POST ammend Assignments. ID#: 17"}, {"id": "6", "title": "Free Time", "description": "POST ammend Free Time. ID#: 6"}, {"id": "1", "title": "Office", "description": "POST ammend Office. ID#: 1"}, {"id": "5", "title": "House Keeping", "description": "POST ammend House Keeping. ID#: 5"}, {"id": "3", "title": "Work From Home", "description": "POST ammend Work From Home. ID#: 3"}, {"id": "11", "title": "Work From Home", "description": "POST ammend Work From Home. ID#: 11"}, {"id": "4", "title": "Groceries", "description": "POST ammend Groceries. ID#: 4"}, {"id": "13", "title": "House Keeping", "description": "POST ammend House Keeping. ID#: 13"}, {"id": "12", "title": "Groceries", "description": "POST ammend Groceries. ID#: 12"}, {"id": "15", "title": "Extra", "description": "POST ammend Extra. ID#: 15"}, {"id": "14", "title": "Free Time", "description": "POST ammend Free Time. ID#: 14"}, {"id": "16", "title": "Homework", "description": "POST ammend Homework. ID#: 16"}, {"id": "18", "title": "Work From Garden", "description": "POST ammend Work From Garden. ID#: 18"}, {"id": "2", "title": "Home", "description": "POST ammend Home. ID#: 2"}, {"id": "8", "title": "Homework", "description": "POST ammend Homework. ID#: 8"}, {"id": "10", "title": "Work From Garden", "description": "POST ammend Work From Garden. ID#: 10"}, {"id": "9", "title": "Assignments", "description": "POST ammend Assignments. ID#: 9"}, {"id": "7", "title": "Extra", "description": "POST ammend Extra. ID#: 7"}]}</pre>

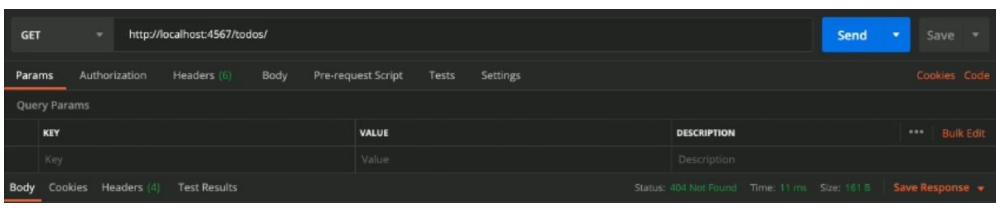
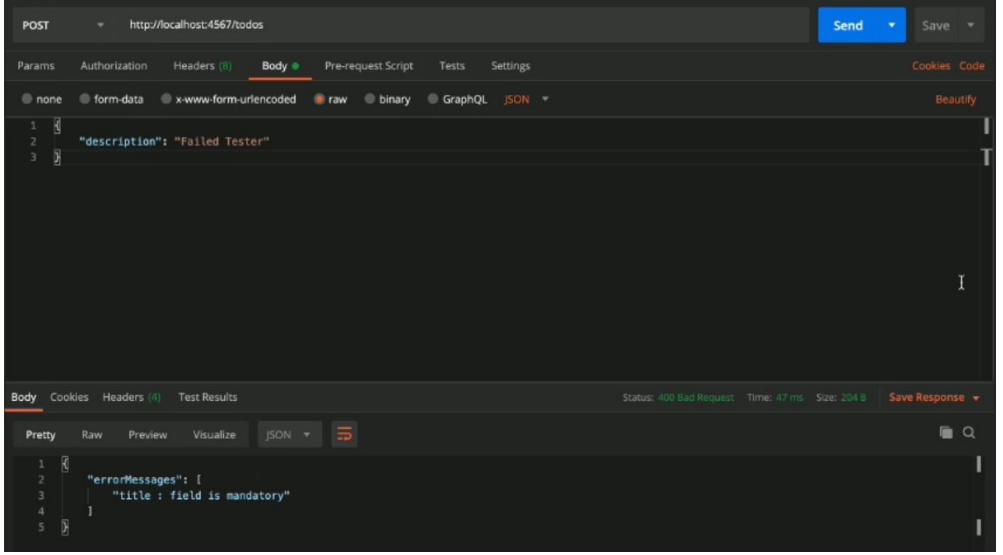
5:25 PM 10/23/2020	Arneet Singh	Command	Verify the current dataset	<pre> todos = requests.get(url=URL + "/todos") projects = requests.get(url=URL + "/projects") categories = requests.get(url=URL + "/categories") # print(len(r.json().get("todos"))) sizeTodos = len(todos.json().get("todos")) sizeProjects = len(projects.json().get("projects")) sizeCategories = len(categories.json().get("categories")) print(todos.json()) print() print(projects.json()) print() print(categories.json()) print() print("OLD len todos: " + str(sizeTodos)) print("OLD len projects: " + str(sizeProjects)) print("OLD len categories: " + str(sizeCategories)) </pre>
5:25 PM 10/23/2020	Arneet Singh	Result	Success Print out all todos, projects and categories and their length	<pre> {'todos': [{"id": "2", "title": "file paperwork", "doneStatus": "false", "description": "POST ammend file paperwork. ID#: 2"}, {"id": "6", "title": "test pt2", "doneStatus": "false", "description": "POST ammend test pt2. ID#: 6"}, {"id": "4", "title": "write pt2", "doneStatus": "false", "description": "POST ammend write pt2. ID#: 4"}, {"id": "3", "title": "read/write pt1", "doneStatus": "false", "description": "POST ammend read/write pt1. ID#: 3"}, {"id": "5", "title": "test pt1", "doneStatus": "false", "description": "POST ammend test pt1. ID#: 5"}, {"id": "1", "title": "scan paperwork", "doneStatus": "false", "description": "POST ammend scan paperwork. ID#: 1"}, {"categories": [{"id": "4", "title": "Groceries", "description": "POST ammend Groceries. ID#: 4"}, {"id": "2", "title": "Home", "description": "POST ammend Home. ID#: 2"}, {"projects": [{"id": "4"}]}, {"id": "9", "title": "Assignments", "description": "POST ammend Assignments. ID#: 9"}, {"projects": [{"id": "4"}]}, {"todos": [{"id": "4"}]}, {"id": "1", "title": "Office", "description": "POST ammend Office. ID#: 1"}, {"projects": [{"id": "4"}]}, {"todos": [{"id": "4"}]}, {"id": "3", "title": "Work From Home", "description": "POST ammend Work From Home. ID#: 3"}, {"projects": [{"id": "4"}]}, {"todos": [{"id": "4"}]}, {"id": "10", "title": "Work From Garden", "description": "POST ammend Work From Garden. ID#: 10"}, {"projects": [{"id": "4"}]}, {"todos": [{"id": "4"}]}, {"id": "6", "title": "Free Time", "description": "POST ammend Free Time. ID#: 6"}, {"projects": [{"id": "4"}]}, {"todos": [{"id": "4"}]}, {"id": "7", "title": "Extra", "description": "POST ammend Extra. ID#: 7"}, {"projects": [{"id": "4"}]}, {"todos": [{"id": "4"}]}, {"id": "5", "title": "House Keeping", "description": "POST ammend House Keeping. ID#: 5"}, {"projects": [{"id": "4"}]}, {"todos": [{"id": "4"}]}, {"id": "8", "title": "Homework", "description": "POST ammend Home work. ID#: 8"}, {"projects": [{"id": "4"}]}, {"todos": [{"id": "4"}]}]} OLD len todos: 6 OLD len projects: 6 OLD len categories: 10 </pre>

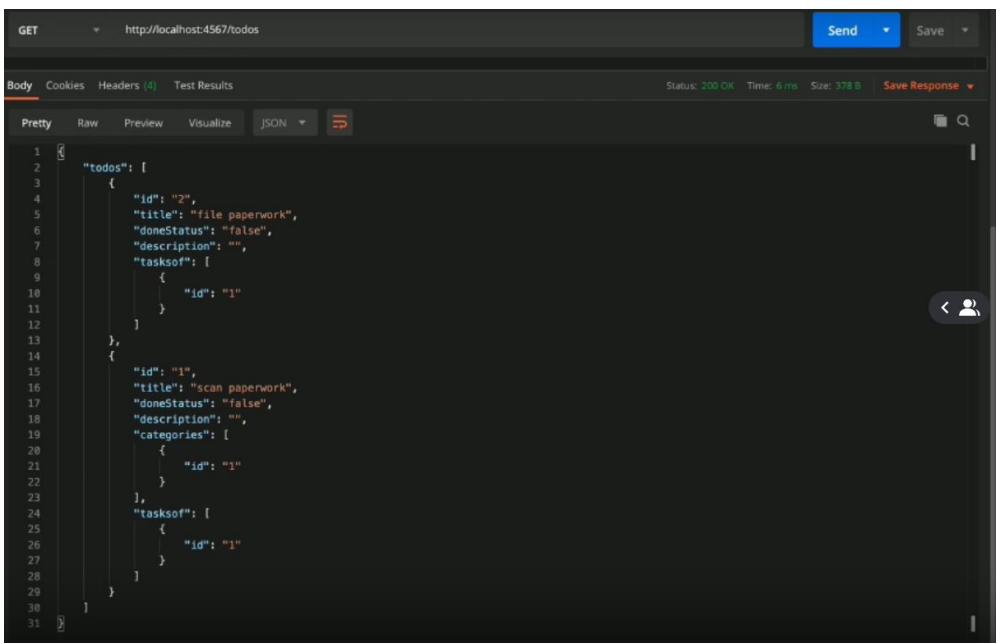
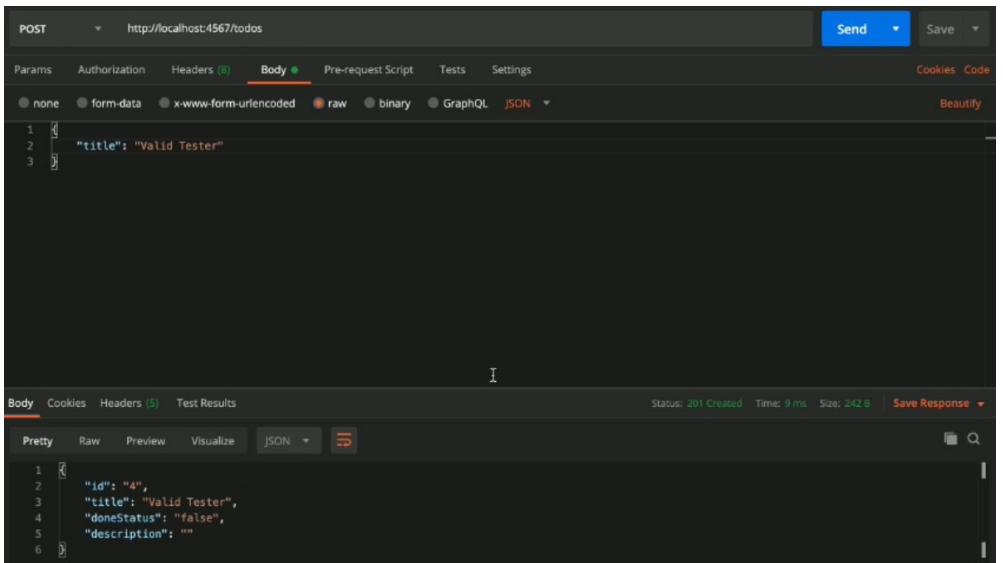
5:26 PM 10/23/2020	Arneet Singh	Command	Delete half the dataset	<pre> for i in range(1, sizeTodos//2 + 1): requests.delete(url=URL + "/todos/" + str(i)) todos = requests.get(url=URL + "/todos") print(todos.json()) print() for i in range(1, sizeProjects//2 + 1): requests.delete(url=URL + "/projects/" + str(i)) projects = requests.get(url=URL + "/projects") print(projects.json()) print() for i in range(1, sizeCategories//2 + 1): requests.delete(url=URL + "/categories/" + str(i)) categories = requests.get(url=URL + "/categories") print(categories.json()) print() sizeTodos = len(todos.json().get("todos")) sizeProjects = len(projects.json().get("projects")) sizeCategories = len(categories.json().get("categories")) print("NEW len todos: " + str(sizeTodos)) print("NEW len projects: " + str(sizeProjects)) print("NEW len categories: " + str(sizeCategories)) </pre>
5:26 PM 10/23/2020	Arneet Singh	Result	Success The printed dataset is halved	<pre> {'todos': [{'id': '6', 'title': 'test pt2', 'doneStatus': 'false', 'description': 'POST ammend test pt2. ID#: 6'}, {'id': '4', 'title': 'write pt2', 'doneStatus': 'false', 'description': 'POST ammend write pt2. ID# 4'}, {'id': '5', 'title': 'test pt 1', 'doneStatus': 'false', 'description': 'POST ammend test pt1. ID# 5'}]} {'projects': [{'id': '4', 'title': '', 'completed': 'false', 'active': 'false', 'description': 'PUT ammend ECSE 308. ID# 4'}, {'id': '6', 'title': '', 'completed': 'false', 'active': 'false', 'description': 'PUT ammend ECSE 310. ID# 6'}, {'id': '5', 'title': '', 'completed': 'false', 'active': 'false', 'description': 'PUT ammend ECSE 309. ID# 5'}]} {'categories': [{'id': '9', 'title': 'Assignments', 'description': 'POST ammend Assignments. ID# 9', 'projects': [{"id": "4"}]}, {"todos": [{"id": "4"}]}, {"id": "10", "title": "Work From Garden", "description": "POST ammend Work From Garden. ID# 10", "projects": [{"id": "4"}]}, {"todos": [{"id": "4"}]}, {"id": "6", "title": "Free Time", "description": "POST ammend Free Time . ID# 6", "projects": [{"id": "4"}]}, {"todos": [{"id": "4"}]}, {"id": "7", "title": "Extra", "description": "POST ammend Extra. ID# 7", "projects": [{"id": "4"}]}, {"todos": [{"id": "4"}]}, {"id": "8", "title": "Homework", "description": "POST ammend Homework. ID# 8", "projects": [{"id": "4"}]}, {"todos": [{"id": "4"}]}]} NEW len todos: 3 NEW len projects: 3 NEW len categories: 5 </pre>

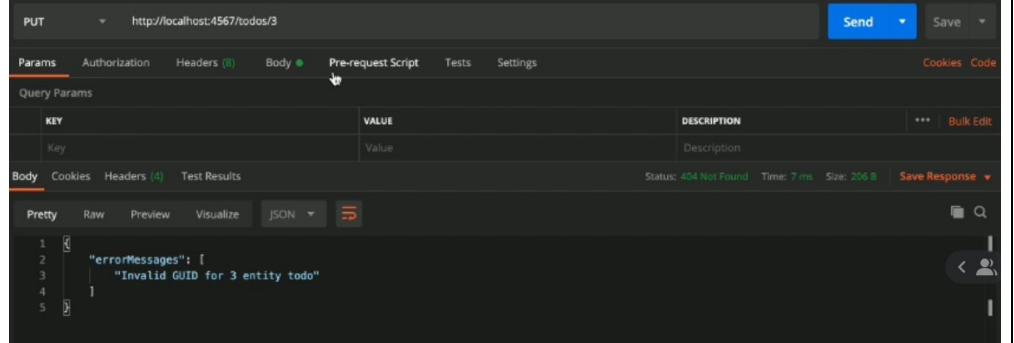
5:27 PM 10/23/2020	Arneet Singh	Command	Verify custom query By printing every single todos by querying it's title	<pre> todos = requests.get(url=URL + "/todos") projects = requests.get(url=URL + "/projects") categories = requests.get(url=URL + "/categories") sizeTodos = len(todos.json().get("todos")) sizeProjects = len(projects.json().get("projects")) sizeCategories = len(categories.json().get("categories")) for i in range(sizeTodos): todosTitle = todos.json()[i].get("title") r = requests.get(url=URL + "/todos?title=" + todosTitle.replace(" ", "%20")) print(r.json()) </pre>
5:27 PM 10/23/2020	Arneet Singh	Result	Success	<pre> { "todos": [{"id": "4", "title": "write pt2", "doneStatus": "false", "description": "POST ammend write pt2. ID#: 4"}, {"id": "6", "title": "test pt2", "doneStatus": "false", "description": "POST ammend test pt2. ID#: 6"}], "todos": [{"id": "5", "title": "test pt1", "doneStatus": "false", "description": "POST ammend test pt1. ID#: 5"}] } </pre>
5:28 PM 10/23/2020	Arneet Singh	Command	Verify custom query By printing every single project by querying it's title	<pre> for i in range(sizeProjects): projectsTitle = projects.json()[i].get("title") r = requests.get(url=URL + "/projects?title=" + projectsTitle.replace(" ", "%20")) print(r.json()) </pre>
5:28 PM 10/23/2020	Arneet Singh	Result	Success	<pre> { "projects": [{"id": "4", "title": "", "completed": "false", "active": "false", "description": "PUT ammend ECSE 308. ID#: 4"}, {"id": "6", "title": "", "completed": "false", "active": "false", "description": "PUT ammend ECSE 310. ID#: 6"}, {"id": "5", "title": "", "completed": "false", "active": "false", "description": "PUT ammend ECSE 309. ID#: 5"}], "projects": [{"id": "4", "title": "", "completed": "false", "active": "false", "description": "PUT ammend ECSE 308. ID#: 4"}, {"id": "6", "title": "", "completed": "false", "active": "false", "description": "PUT ammend ECSE 310. ID#: 6"}, {"id": "5", "title": "", "completed": "false", "active": "false", "description": "PUT ammend ECSE 309. ID#: 5"}] } </pre>

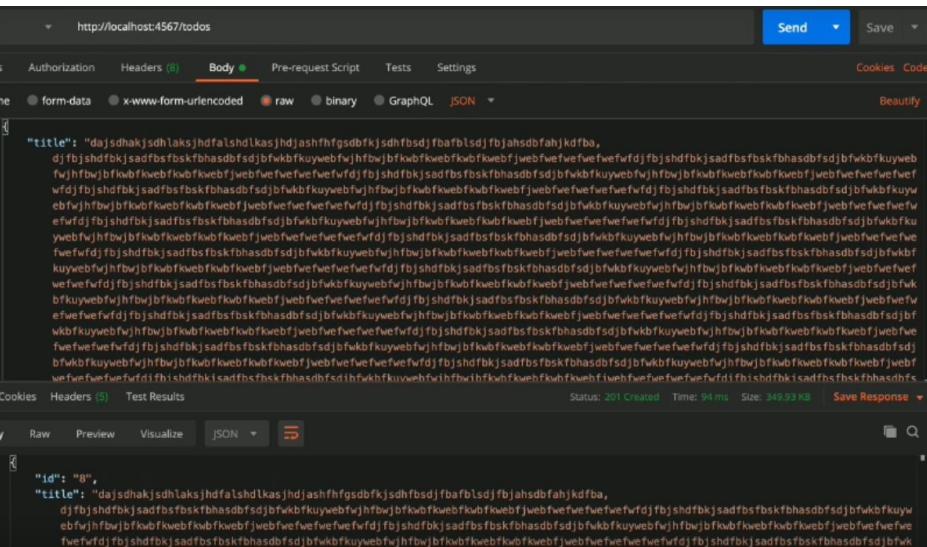
5:28 PM 10/23/2020	Arneet Singh	Command	Verify custom query By printing every single category by querying it's title	<pre>for i in range(sizeCategories): categoriesTitle = categories.json().get("categories")[i].get("title") r = requests.get(url=URL + "/categories?title=" + categoriesTitle.replace(" ", "%20")) print(r.json())</pre>
5:28 PM 10/23/2020	Arneet Singh	Result	Success	<pre>{"categories": [{"id": "10", "title": "Work From Garden", "description": "POST ammend Work From Garden. ID#: 10"}] {"categories": [{"id": "8", "title": "Homework", "description": "POST ammend Homework. ID#: 8"}]} {"categories": [{"id": "7", "title": "Extra", "description": "POST ammend Extra. ID#: 7"}]} {"categories": [{"id": "9", "title": "Assignments", "description": "POST ammend Assignments. ID#: 9"}]} {"categories": [{"id": "6", "title": "Free Time", "description": "POST ammend Free Time. ID#: 6"}]}</pre>

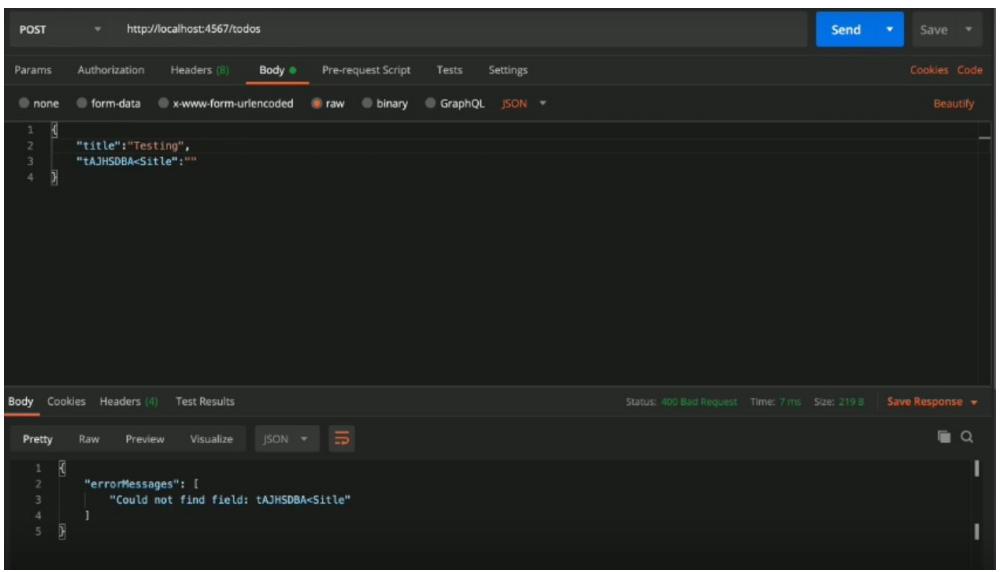
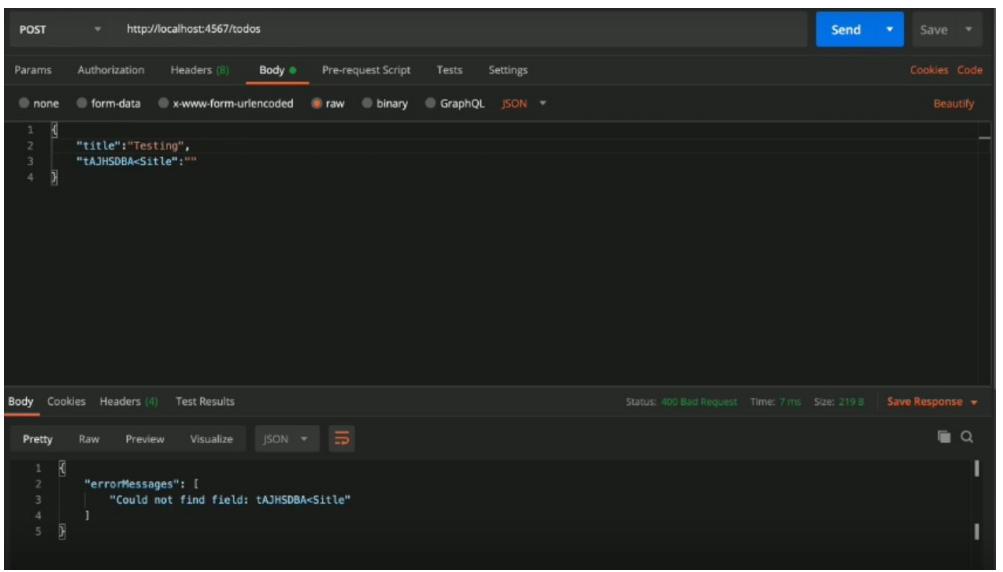
Time	Reporter	Type	Content	Screenshot
5:31 PM 10/23/2020	Kevin Li	Session Reporter	Kevin Li - 260862327 (kevin.li2@mail.mcgill.ca) Arneet Singh - 260865569 (arneet.kalra@mail.mcgill.ca)	
5:31 PM 10/23/2020	Kevin Li	Session Tester	Arneet Singh	
5:31 PM 10/23/2020	Kevin Li	Session Charter	Identify capabilities and areas of potential instability of the “rest api todo list manager”	
5:31 PM 10/23/2020	Kevin Li	Setup	Windows 10 64-bit Postman v7.34.0 Python 3.8.2	
5:32 PM 10/23/2020	Kevin Li	Note	Start rest api todo list manager	
5:33 PM 10/23/2020	Kevin Li	Command	Send a GET request with no extra forward slash at the end	http://localhost:4567/todos
5:33 PM 10/23/2020	Kevin Li	Result	SUCCESS	
5:34 PM 10/23/2020	Kevin Li	Command	Send a GET request with an extra forward slash at the end	http://localhost:4567/todos/

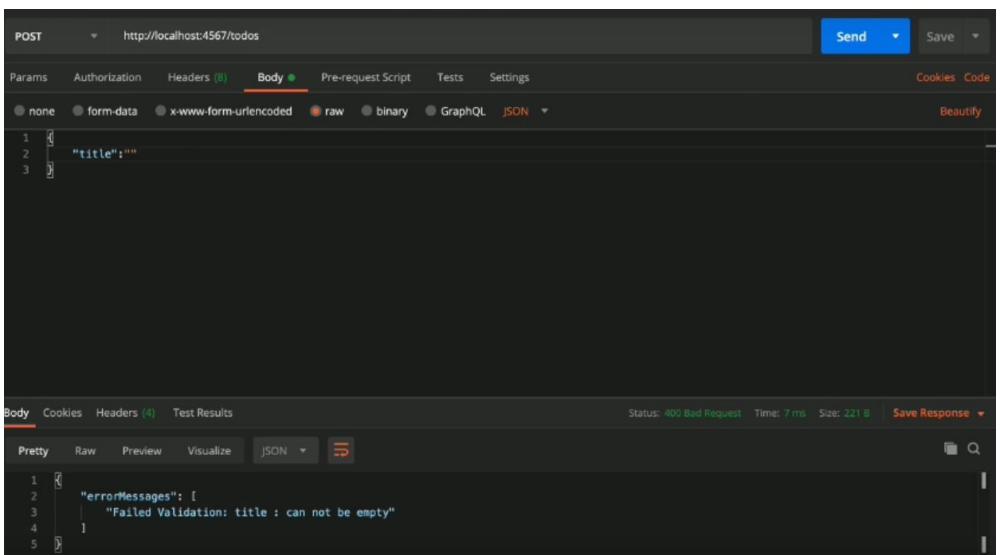
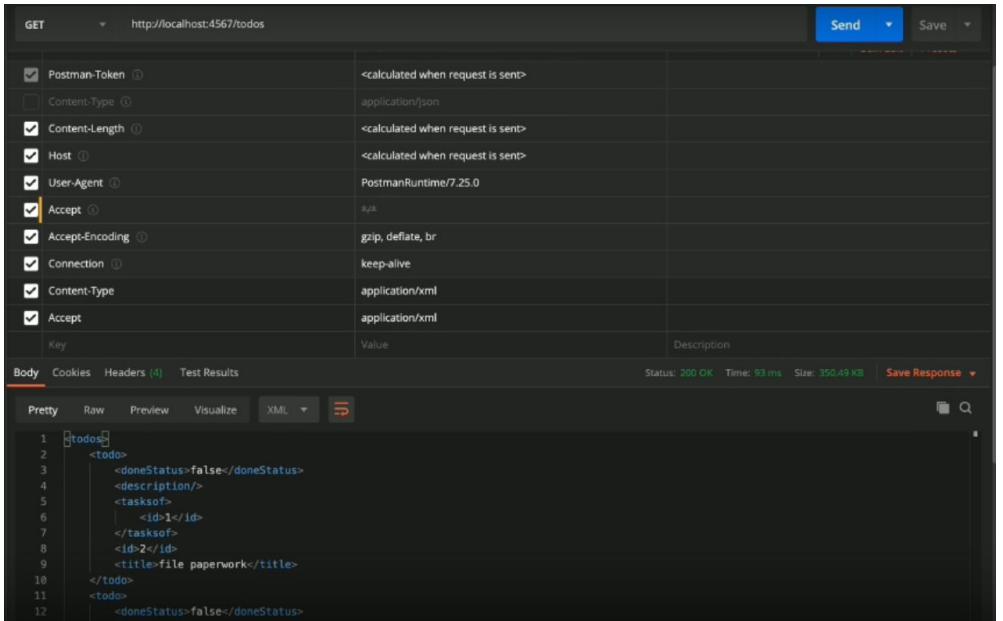
5:34 PM 10/23/2020	Kevin Li	Result	INSTABILITY (FAIL) Received 404 not found after adding an extra forward slash	
5:34 PM 10/23/2020	Kevin Li	Command	Send a POST request with only a description and no title to get a failed request.	http://localhost:4567/todos
5:34 PM 10/23/2020	Kevin Li	Result	Post failed as intended	
5:35 PM 10/23/2020	Kevin Li	Command	Send a GET request to see if it was indeed not created	http://localhost:4567/todos

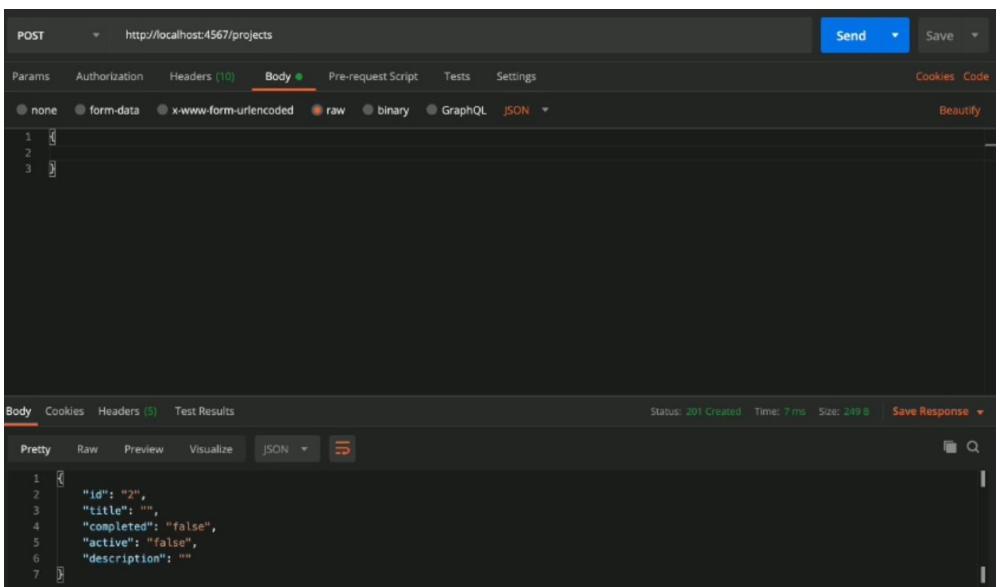
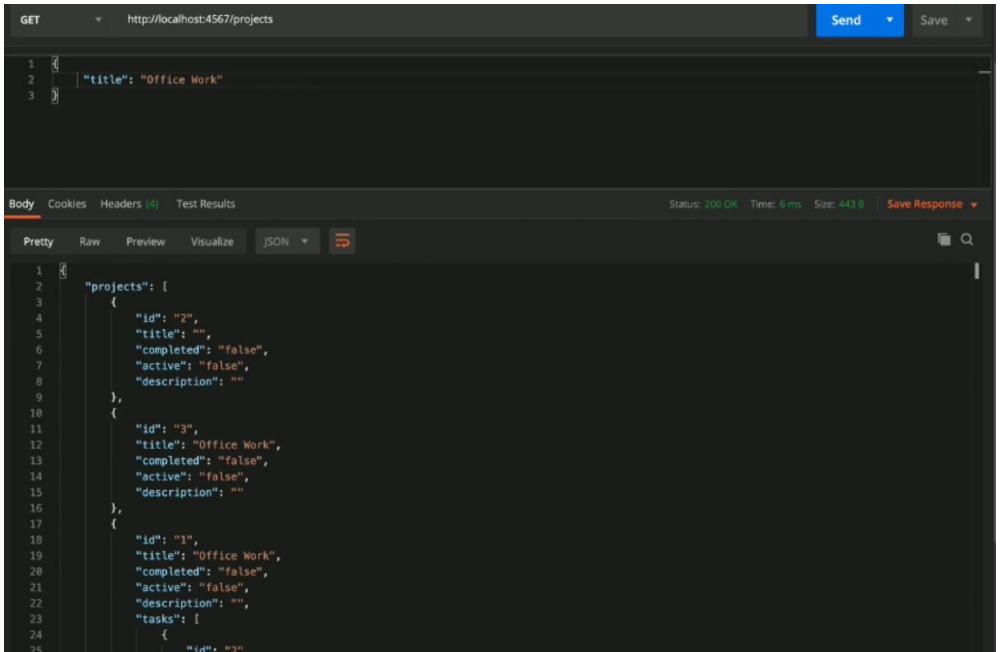
5:35 PM 10/23/2020	Kevin Li	Result	SUCCESS	 <pre> GET http://localhost:4567/todos Status: 200 OK Time: 6 ms Size: 378 B Save Response Body Cookies Headers (4) Test Results Pretty Raw Preview Visualize JSON ↻ 1 [2 "todos": [3 { 4 "id": "2", 5 "title": "file paperwork", 6 "doneStatus": "false", 7 "description": "", 8 "tasksof": [9 { 10 "id": "1" 11 } 12], 13 }, 14 { 15 "id": "1", 16 "title": "scan paperwork", 17 "doneStatus": "false", 18 "description": "", 19 "categories": [20 { 21 "id": "1" 22 } 23], 24 "tasksof": [25 { 26 "id": "1" 27 } 28] 29 } 30] 31] </pre>
5:36 PM 10/23/2020	Kevin Li	Command	Send a POST request with a valid response body	http://localhost:4567/todos
5:36 PM 10/23/2020	Kevin Li	Result	SUCCESS But note how it uses id 4 instead of id 3	 <pre> POST http://localhost:4567/todos Status: 201 Created Time: 9 ms Size: 242 B Save Response Params Authorization Headers (8) Body ↻ Pre-request Script Tests Settings Cookies Code Beautify 1 { 2 "title": "Valid Tester" 3 } Body Cookies Headers (5) Test Results Pretty Raw Preview Visualize JSON ↻ 1 [2 "id": "4", 3 "title": "Valid Tester", 4 "doneStatus": "false", 5 "description": "" 6] </pre>

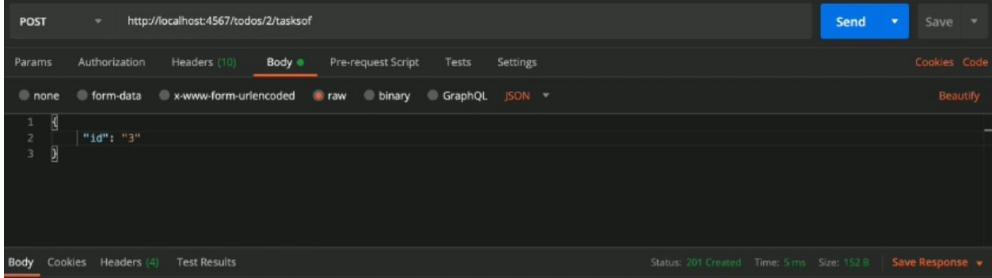
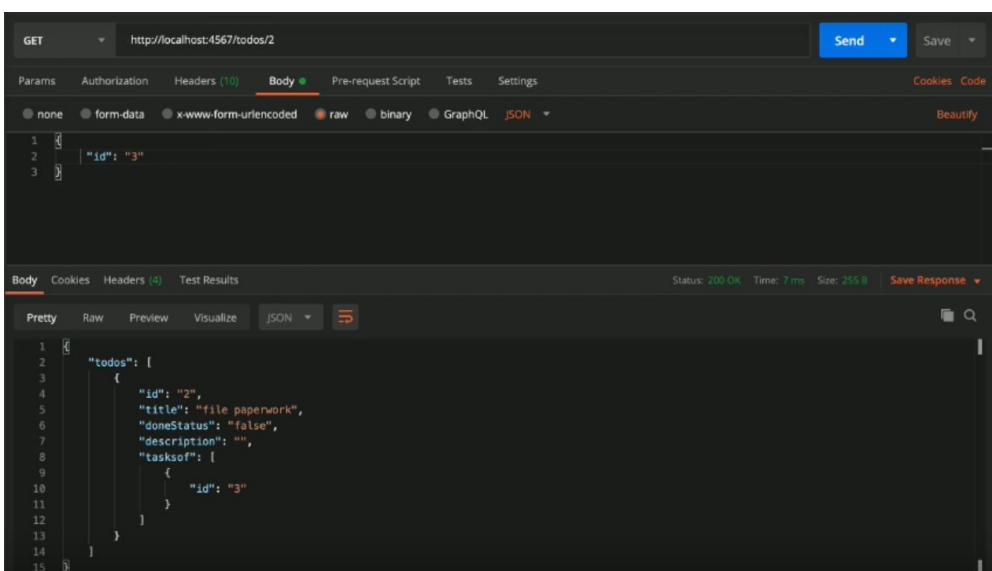
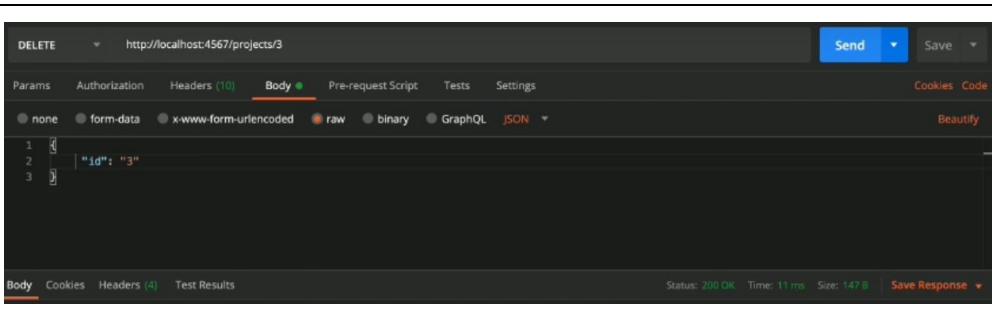
5:36 PM 10/23/2020	Kevin Li	Result	(Instability)	<p>The newly created task uses id 4 instead of id 3.</p> <pre> 1 [2 "todos": [3 { 4 "id": "2", 5 "title": "file paperwork", 6 "doneStatus": "false", 7 "description": "", 8 "tasksof": [9 { 10 "id": "1" 11 } 12], 13 }, 14 { 15 "id": "4", 16 "title": "Valid Tester", 17 "doneStatus": "false", 18 "description": "" 19 }, 20 { 21 "id": "1", 22 "title": "scan paperwork", 23 "doneStatus": "false", 24 "description": "", 25 "categories": [26 { 27 "id": "1" 28 } 29], 30 "tasksof": [31 { 32 "id": "1" 33 } 34] 35 } 36] 37 </pre>
5:37 PM 10/23/2020	Kevin Li	Command	Send a PUT request to attempt to create a todos for id 3.	http://localhost:4567/todos/3
5:37 PM 10/23/2020	Kevin Li	Result	(Instability)	<p>Since you cannot specify the id when creating a todo. The ID 3 is forever never used.</p> 
5:38 PM 10/23/2020	Kevin Li	Command	Send a POST request with an extremely long title.	http://localhost:4567/todos

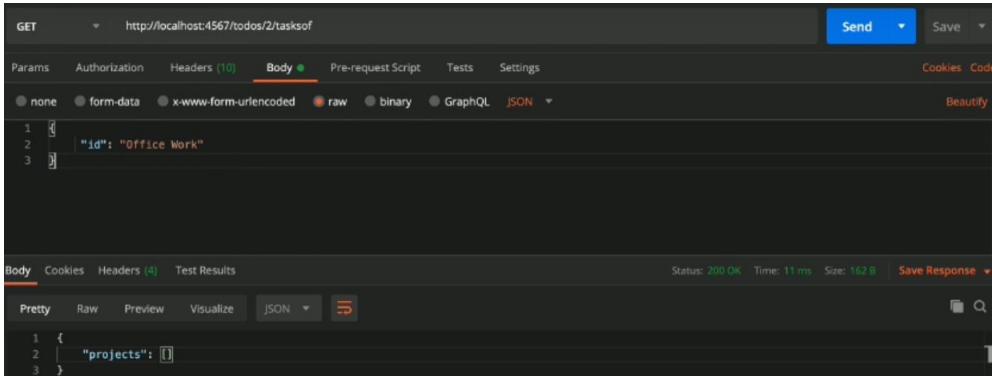
5:38 PM 10/23/2020	Kevin Li	Result	SUCCESS Capability: It created a todo with a long title	
5:41 PM 10/23/2020	Kevin Li	Command	Look at the GUI to see if it successfully created the todo with the long title	http://localhost:4567/docs
5:41 PM 10/23/2020	Kevin Li	Result	SUCCESS Capability: The title is fully displayed	
5:42 PM 10/23/2020	Kevin Li	Command	Send a POST request with a response body including id to see if it allows a creation	http://localhost:4567/todos {“id”: “3”, “title”: “Testing”}

5:42 PM 10/23/2020	Kevin Li	Result	FAILED as intended.	 <pre>POST http://localhost:4567/todos Body Params Authorization Headers (8) Body Pre-request Script Tests Settings none form-data x-www-form-urlencoded raw binary GraphQL JSON 1 2 "title": "Testing", 3 "tAJHSDBA-Sitle": "" 4 Body Cookies Headers (4) Test Results Pretty Raw Preview Visualize JSON 1 2 "errorMessages": [3 "Could not find field: tAJHSDBA-Sitle" 4] 5</pre> <p>Status: 400 Bad Request Time: 7 ms Size: 219 B Save Response</p>
5:43 PM 10/23/2020	Kevin Li	Command	Send a POST request with a invalid field	http://localhost:4567/todos {“title”: “Testing”, “tAJHSDBA<Sitle”: “”}
5:43 PM 10/23/2020	Kevin Li	Result	FAILED as intended	 <pre>POST http://localhost:4567/todos Body Params Authorization Headers (8) Body Pre-request Script Tests Settings none form-data x-www-form-urlencoded raw binary GraphQL JSON 1 2 "title": "Testing", 3 "tAJHSDBA-Sitle": "" 4 Body Cookies Headers (4) Test Results Pretty Raw Preview Visualize JSON 1 2 "errorMessages": [3 "Could not find field: tAJHSDBA-Sitle" 4] 5</pre> <p>Status: 400 Bad Request Time: 7 ms Size: 219 B Save Response</p>
5:43 PM 10/23/2020	Kevin Li	Command	Send a POST request with a response body of an empty title	http://localhost:4567/todos {“title”: “”}

5:44 PM 10/23/2020	Kevin Li	Result	FAILED as intended	 <pre>POST http://localhost:4567/todos Params Authorization Headers (8) Body Pre-request Script Tests Settings Body (none) form-data x-www-form-urlencoded raw binary GraphQL JSON 1 "title": "" 2 3 4 5 Body Cookies Headers (4) Test Results Pretty Raw Preview Visualize JSON 1 2 "errorMessages": [3 "Failed Validation: title : can not be empty" 4] 5 Status: 400 Bad Request Time: 7 ms Size: 221 B Save Response</pre>
5:44 PM 10/23/2020	Kevin Li	Command	Send a GET request with the headers having: Accept = application/xml Content-Type = application/xml	http://localhost:4567/todos
5:44 PM 10/23/2020	Kevin Li	Result	SUCCESS Receives xml file	 <pre>GET http://localhost:4567/todos Params Authorization Headers (4) Test Results Postman-Token <calculated when request is sent> Content-Type application/json Content-Length <calculated when request is sent> Host <calculated when request is sent> User-Agent PostmanRuntime/7.25.0 Accept application/xml Accept-Encoding gzip, deflate, br Connection keep-alive Content-Type application/xml Accept application/xml Body Cookies Headers (4) Test Results Pretty Raw Preview Visualize XML 1 <todos> 2 <todo> 3 <doneStatus>false</doneStatus> 4 <description/> 5 <tasksOf> 6 <id>1</id> 7 </tasksOf> 8 <id>2</id> 9 <title>file paperwork</title> 10 </todo> 11 <todo> 12 <doneStatus>false</doneStatus></pre>
5:44 PM 10/23/2020	Kevin Li	Command	Send a POST request with an empty response body	https://localhost:4567/projects {}

5:44 PM 10/23/2020	Kevin Li	Result	SUCCESS Created a project	
5:44 PM 10/23/2020	Kevin Li	Command	Send a Post request with the response body of an existing title	http://localhost:4567/projects {“title”: “Office Work”}
5:44 PM 10/23/2020	Kevin Li	Result	SUCCESS Capability: Able to create a project with duplicated titles	
5:45 PM	Kevin Li	Command	Send a POST request with the	http://localhost:4567/todos/2/tasks/of

10/23/2020			response body of id: 3	{ "id": "3"}
5:45 PM 10/23/2020	Kevin Li	Result	SUCCESS linked	
5:46 PM 10/23/2020	Kevin Li	Command	Send a GET request to validate the link	http://localhost:4567/todos/2
5:46 PM 10/23/2020	Kevin Li	Result	SUCCESS	
5:47 PM 10/23/2020	Kevin Li	Command	Send a DELETE request to delete the newly created link	http://localhost:4567/projects/3
5:47 PM 10/23/2020	Kevin Li	Result	SUCCESS	

5:47 PM 10/23/2020	Kevin Li	Command	Send a GET request to see if the link between task is also removed after projects deleted	http://localhost:4567/todos/2/tasks/of
5:47 PM 10/23/2020	Kevin Li	Result	<p>SUCCESS</p> <p>Capability: if you delete the projects, the link also disappears</p>	 <pre> GET http://localhost:4567/todos/2/tasks/of Params Authorization Headers (10) Body Pre-request Script Tests Settings none form-data x-www-form-urlencoded raw binary GraphQL JSON 1 { 2 "id": "Office Work" 3 } Body Cookies Headers (4) Test Results Pretty Raw Preview Visualize JSON 1 { 2 "projects": [] 3 } Status: 200 OK Time: 11 ms Size: 162 B Save Response </pre>

Summary of session findings

- All the request methods are working properly and are demonstrated using a python script
- Found a few capabilities and instabilities that are identified above
- Using PUT to amend a specific field in an instance does not behave as described in documentation. It amends all fields and body. But reset all other non mandatory fields.

List of concerns identified in session

- Instability: Adding a forward slash at the end of the url request will cause a 404 error
- Instability: When a creation fails, it will waste an ID number

List of new testing ideas identified in session

- Test the forward slash at the end of the url request instability
- Test the ID creation instability using unit tests