

# **Assignment 2 Upper Layers of the OSI-Model**

John Dawood

Version: October 22, 2024

## **Prerequisites**

1. Module 2 readings on Canvas
2. Module 2 lecture videos on Canvas
3. Run/Understand the following examples from the GitHub repository:
  - Network/SimpleGrabHttpURL
  - Network/SimpleGrabURL
  - Network/HTTP-JSON
4. Setup of a second device (second computer, AWS EC2) – see Canvas for details
5. Videos/Tutorials about Wireshark
6. Understand the lower layers of the OSI model (Module 1)

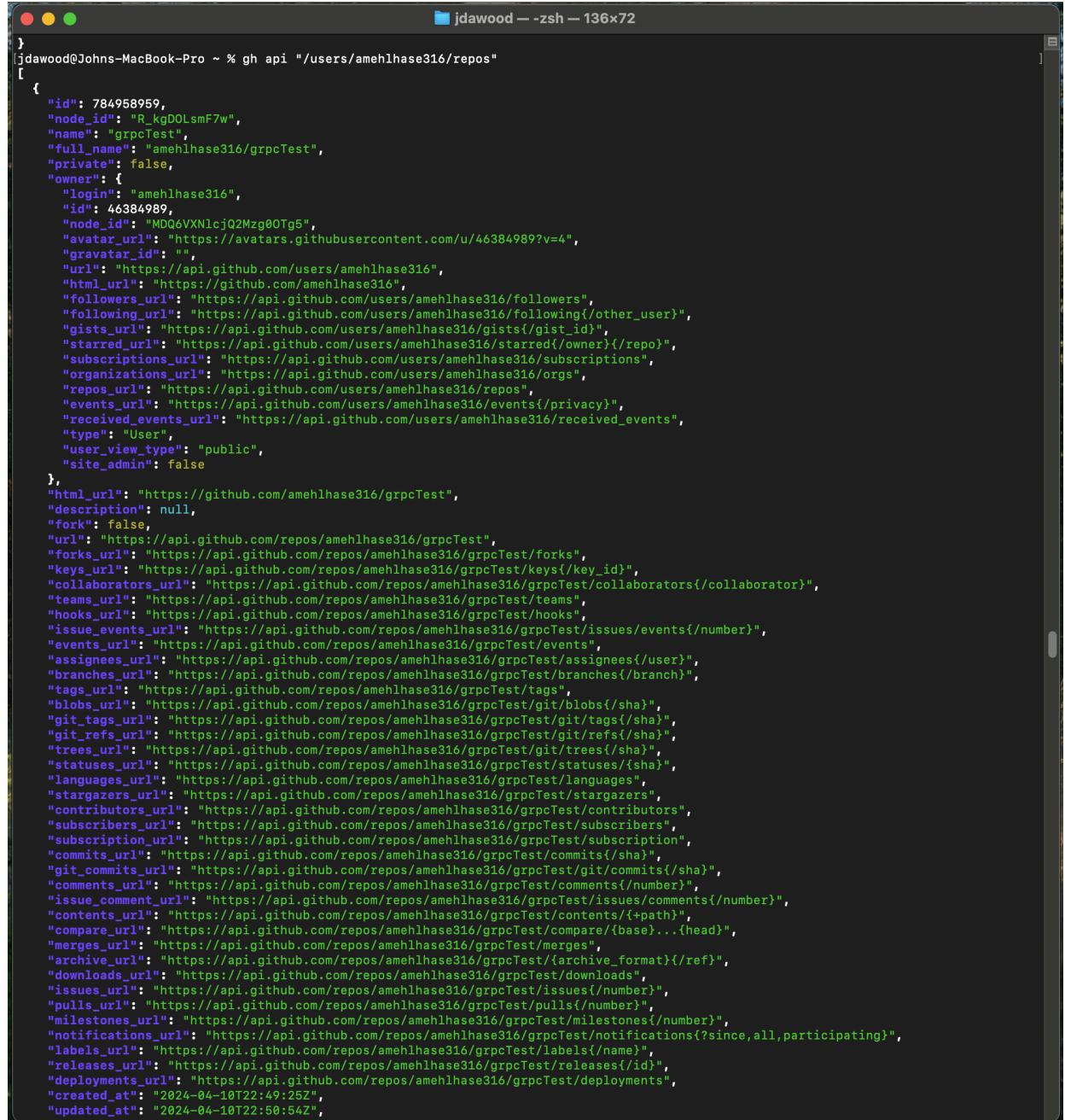
## **The learning outcomes of this assignment are:**

- Understanding the purpose and function of the upper layer network protocols
- Understanding how these protocols differ (HTTP vs. HTTPS)
  - In particular, the differences in each protocols message traffic
- And as always, familiarization of terminal / command line usage – One of the most important skills to develop!

## 1 Understanding HTTP

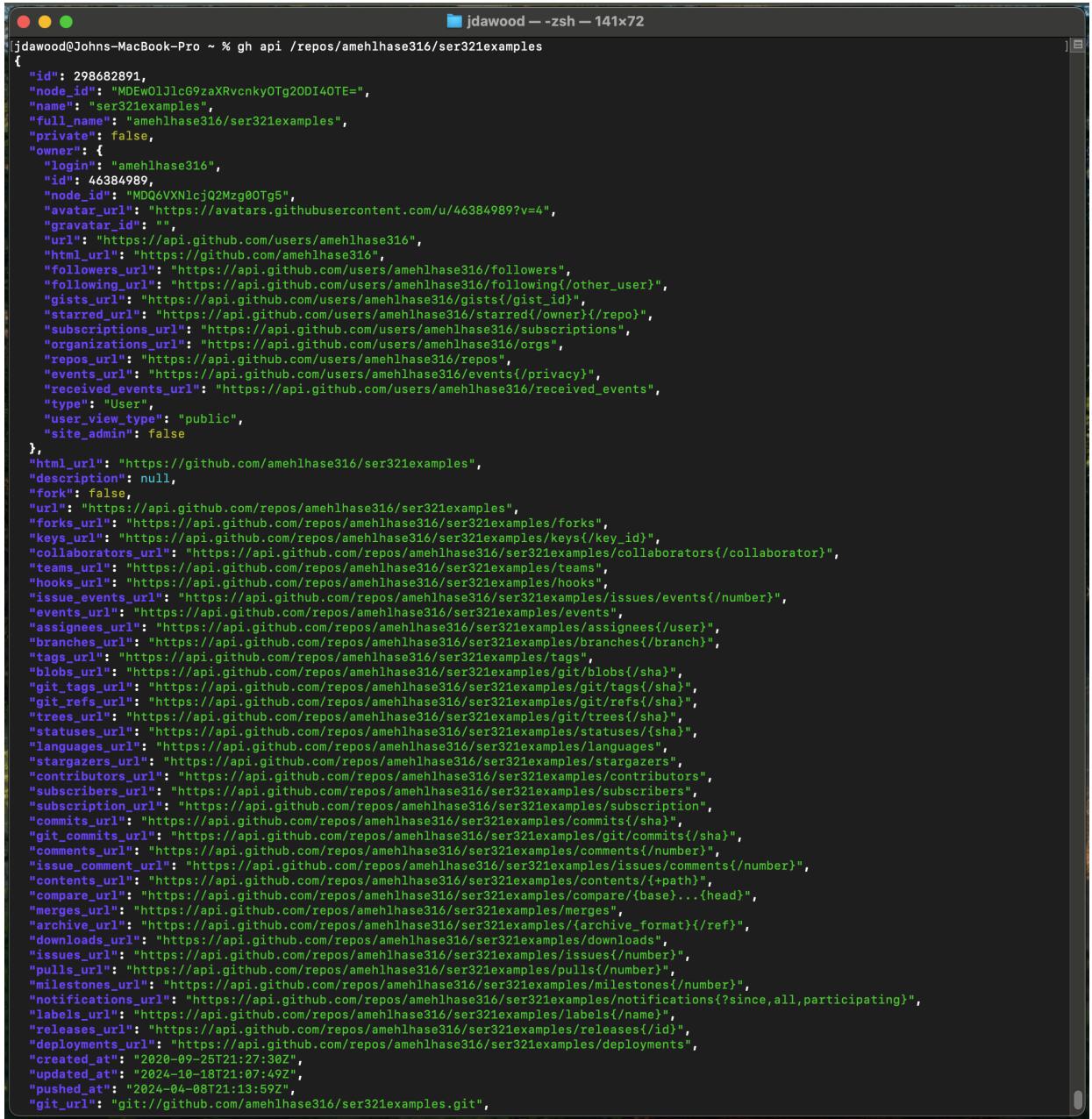
For this section, you will only need your web browser and Wireshark. (Wireshark only for taking a look at things yourself.) To understand GET HTTP requests a little better, we want to do a couple of GET requests through an API. (For this assignment, it is for the GitHub API.) The API works by nesting topics - where you will need to do a base call to get some data, use that data to perform the next call, and so forth.

<https://api.github.com/users/amehlhase316/repos>



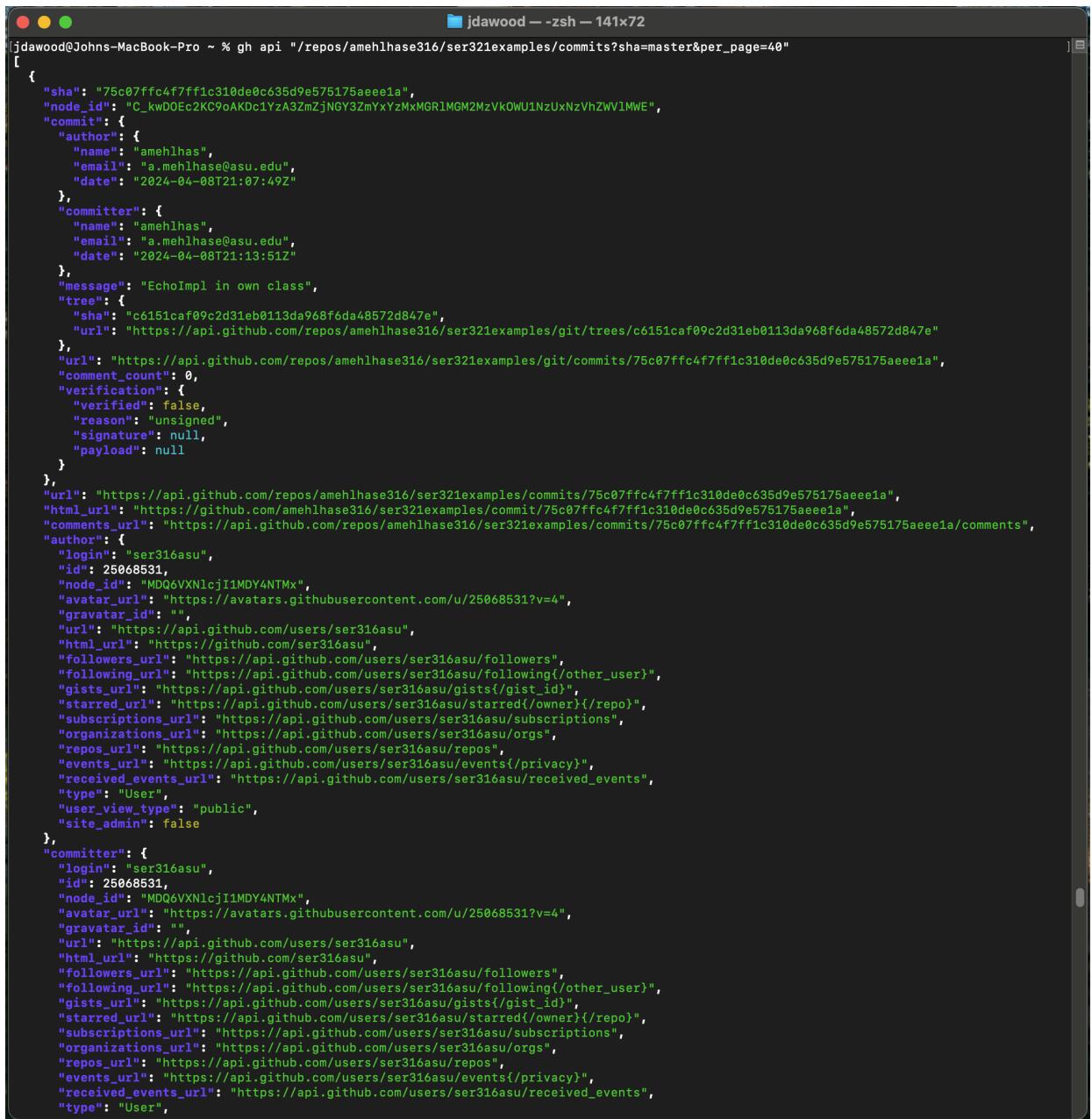
```
jdawood@Johns-MacBook-Pro ~ % gh api "/users/amehlhase316/repos"
[
  {
    "id": 784958959,
    "node_id": "R_kgDOLsmF7w",
    "name": "grpcTest",
    "full_name": "amehlhase316/grpcTest",
    "private": false,
    "owner": {
      "login": "amehlhase316",
      "id": 46384989,
      "node_id": "MDQ6VXNlcjQ2Mzg0OTg5",
      "avatar_url": "https://avatars.githubusercontent.com/u/46384989?v=4",
      "gravatar_id": "",
      "url": "https://api.github.com/users/amehlhase316",
      "html_url": "https://github.com/amehlhase316",
      "followers_url": "https://api.github.com/users/amehlhase316/followers",
      "following_url": "https://api.github.com/users/amehlhase316/following{/other_user}",
      "gists_url": "https://api.github.com/users/amehlhase316/gists{/gist_id}",
      "starred_url": "https://api.github.com/users/amehlhase316/starred{/owner}{/repo}",
      "subscriptions_url": "https://api.github.com/users/amehlhase316/subscriptions",
      "organizations_url": "https://api.github.com/users/amehlhase316/orgs",
      "repos_url": "https://api.github.com/users/amehlhase316/repos",
      "events_url": "https://api.github.com/users/amehlhase316/events{/privacy}",
      "received_events_url": "https://api.github.com/users/amehlhase316/received_events",
      "type": "User",
      "user_view_type": "public",
      "site_admin": false
    },
    "html_url": "https://github.com/amehlhase316/grpcTest",
    "description": null,
    "fork": false,
    "url": "https://api.github.com/repos/amehlhase316/grpcTest",
    "forks_url": "https://api.github.com/repos/amehlhase316/grpcTest/forks",
    "keys_url": "https://api.github.com/repos/amehlhase316/grpcTest/keys{/key_id}",
    "collaborators_url": "https://api.github.com/repos/amehlhase316/grpcTest/collaborators{/collaborator}",
    "teams_url": "https://api.github.com/repos/amehlhase316/grpcTest/teams",
    "hooks_url": "https://api.github.com/repos/amehlhase316/grpcTest/hooks",
    "issue_events_url": "https://api.github.com/repos/amehlhase316/grpcTest/issues/events{/number}",
    "events_url": "https://api.github.com/repos/amehlhase316/grpcTest/events",
    "assignees_url": "https://api.github.com/repos/amehlhase316/grpcTest/assignees{/user}",
    "branches_url": "https://api.github.com/repos/amehlhase316/grpcTest/branches{/branch}",
    "tags_url": "https://api.github.com/repos/amehlhase316/grpcTest/tags",
    "blobs_url": "https://api.github.com/repos/amehlhase316/grpcTest/git/blobs{/sha}",
    "git_tags_url": "https://api.github.com/repos/amehlhase316/grpcTest/git/tags{/sha}",
    "git_refs_url": "https://api.github.com/repos/amehlhase316/grpcTest/git/refs{/sha}",
    "trees_url": "https://api.github.com/repos/amehlhase316/grpcTest/git/trees{/sha}",
    "statuses_url": "https://api.github.com/repos/amehlhase316/grpcTest/statuses{/sha}",
    "languages_url": "https://api.github.com/repos/amehlhase316/grpcTest/languages",
    "stargazers_url": "https://api.github.com/repos/amehlhase316/grpcTest/stargazers",
    "contributors_url": "https://api.github.com/repos/amehlhase316/grpcTest/contributors",
    "subscribers_url": "https://api.github.com/repos/amehlhase316/grpcTest/subscribers",
    "subscription_url": "https://api.github.com/repos/amehlhase316/grpcTest/subscription",
    "commits_url": "https://api.github.com/repos/amehlhase316/grpcTest/commits{/sha}",
    "git_commits_url": "https://api.github.com/repos/amehlhase316/grpcTest/git/commits{/sha}",
    "comments_url": "https://api.github.com/repos/amehlhase316/grpcTest/comments{/number}",
    "issue_comment_url": "https://api.github.com/repos/amehlhase316/grpcTest/issues/comments{/number}",
    "contents_url": "https://api.github.com/repos/amehlhase316/grpcTest/contents/{+path}",
    "compare_url": "https://api.github.com/repos/amehlhase316/grpcTest/compare/{base}...{head}",
    "merges_url": "https://api.github.com/repos/amehlhase316/grpcTest/merges",
    "archive_url": "https://api.github.com/repos/amehlhase316/grpcTest/{archive_format}{/ref}",
    "downloads_url": "https://api.github.com/repos/amehlhase316/grpcTest/downloads",
    "issues_url": "https://api.github.com/repos/amehlhase316/grpcTest/issues{/number}",
    "pulls_url": "https://api.github.com/repos/amehlhase316/grpcTest/pulls{/number}",
    "milestones_url": "https://api.github.com/repos/amehlhase316/grpcTest/milestones{/number}",
    "notifications_url": "https://api.github.com/repos/amehlhase316/grpcTest/notifications{/since,all,participating}",
    "labels_url": "https://api.github.com/repos/amehlhase316/grpcTest/labels{/name}",
    "releases_url": "https://api.github.com/repos/amehlhase316/grpcTest/releases{/id}",
    "deployments_url": "https://api.github.com/repos/amehlhase316/grpcTest/deployments",
    "created_at": "2024-04-10T22:49:25Z",
    "updated_at": "2024-04-10T22:50:54Z"
  }
]
```

<https://github.com/amehlhase316/ser321examples>



```
jdawood@Johns-MacBook-Pro ~ % gh api /repos/amehlhase316/ser321examples
{
  "id": 298682891,
  "node_id": "MDExOlJlcG9zaXRvcnkyOTg2ODI4OTE=",
  "name": "ser321examples",
  "full_name": "amehlhase316/ser321examples",
  "private": false,
  "owner": {
    "login": "amehlhase316",
    "id": 46384989,
    "node_id": "MDQ6VXNlcjQ2Mzg0OTg5",
    "avatar_url": "https://avatars.githubusercontent.com/u/46384989?v=4",
    "gravatar_id": "",
    "url": "https://api.github.com/users/amehlhase316",
    "html_url": "https://github.com/amehlhase316",
    "followers_url": "https://api.github.com/users/amehlhase316/followers",
    "following_url": "https://api.github.com/users/amehlhase316/following{/other_user}",
    "gists_url": "https://api.github.com/users/amehlhase316/gists{/gist_id}",
    "starred_url": "https://api.github.com/users/amehlhase316/starred{/owner}{/repo}",
    "subscriptions_url": "https://api.github.com/users/amehlhase316/subscriptions",
    "organizations_url": "https://api.github.com/users/amehlhase316/orgs",
    "repos_url": "https://api.github.com/users/amehlhase316/repos",
    "events_url": "https://api.github.com/users/amehlhase316/events{/privacy}",
    "received_events_url": "https://api.github.com/users/amehlhase316/received_events",
    "type": "User",
    "user_view_type": "public",
    "site_admin": false
  },
  "html_url": "https://github.com/amehlhase316/ser321examples",
  "description": null,
  "fork": false,
  "url": "https://api.github.com/repos/amehlhase316/ser321examples",
  "forks_url": "https://api.github.com/repos/amehlhase316/ser321examples/forks",
  "keys_url": "https://api.github.com/repos/amehlhase316/ser321examples/keys{/key_id}",
  "collaborators_url": "https://api.github.com/repos/amehlhase316/ser321examples/collaborators{/collaborator}",
  "teams_url": "https://api.github.com/repos/amehlhase316/ser321examples/teams",
  "hooks_url": "https://api.github.com/repos/amehlhase316/ser321examples/hooks",
  "issue_events_url": "https://api.github.com/repos/amehlhase316/ser321examples/issues/events{/number}",
  "events_url": "https://api.github.com/repos/amehlhase316/ser321examples/events",
  "assignees_url": "https://api.github.com/repos/amehlhase316/ser321examples/assignees{/user}",
  "branches_url": "https://api.github.com/repos/amehlhase316/ser321examples/branches{/branch}",
  "tags_url": "https://api.github.com/repos/amehlhase316/ser321examples/tags",
  "blobs_url": "https://api.github.com/repos/amehlhase316/ser321examples/git/blobs{/sha}",
  "git_tags_url": "https://api.github.com/repos/amehlhase316/ser321examples/git/tags{/sha}",
  "git_refs_url": "https://api.github.com/repos/amehlhase316/ser321examples/git/refs{/sha}",
  "trees_url": "https://api.github.com/repos/amehlhase316/ser321examples/git/trees{/sha}",
  "statuses_url": "https://api.github.com/repos/amehlhase316/ser321examples/statuses{/sha}",
  "languages_url": "https://api.github.com/repos/amehlhase316/ser321examples/languages",
  "stargazers_url": "https://api.github.com/repos/amehlhase316/ser321examples/stargazers",
  "contributors_url": "https://api.github.com/repos/amehlhase316/ser321examples/contributors",
  "subscribers_url": "https://api.github.com/repos/amehlhase316/ser321examples/subscribers",
  "subscription_url": "https://api.github.com/repos/amehlhase316/ser321examples/subscription",
  "commits_url": "https://api.github.com/repos/amehlhase316/ser321examples/commits{/sha}",
  "git_commits_url": "https://api.github.com/repos/amehlhase316/ser321examples/git/commits{/sha}",
  "comments_url": "https://api.github.com/repos/amehlhase316/ser321examples/comments{/number}",
  "issue_comment_url": "https://api.github.com/repos/amehlhase316/ser321examples/issues/comments{/number}",
  "contents_url": "https://api.github.com/repos/amehlhase316/ser321examples/contents/{path}",
  "compare_url": "https://api.github.com/repos/amehlhase316/ser321examples/compare/{base}...{head}",
  "merges_url": "https://api.github.com/repos/amehlhase316/ser321examples/merges",
  "archive_url": "https://api.github.com/repos/amehlhase316/ser321examples/archive_format{/ref}",
  "downloads_url": "https://api.github.com/repos/amehlhase316/ser321examples/downloads",
  "issues_url": "https://api.github.com/repos/amehlhase316/ser321examples/issues{/number}",
  "pulls_url": "https://api.github.com/repos/amehlhase316/ser321examples/pulls{/number}",
  "milestones_url": "https://api.github.com/repos/amehlhase316/ser321examples/milestones{/number}",
  "notifications_url": "https://api.github.com/repos/amehlhase316/ser321examples/notifications{?since,all,participating}",
  "labels_url": "https://api.github.com/repos/amehlhase316/ser321examples/labels{/name}",
  "releases_url": "https://api.github.com/repos/amehlhase316/ser321examples/releases{/id}",
  "deployments_url": "https://api.github.com/repos/amehlhase316/ser321examples/deployments",
  "created_at": "2020-09-25T21:27:30Z",
  "updated_at": "2024-10-18T21:07:49Z",
  "pushed_at": "2024-04-08T21:13:59Z",
  "git_url": "git://github.com/amehlhase316/ser321examples.git"
}
```

[https://api.github.com/repos/amehlhase316/ser321examples/commits  
?sha=master&per\\_page=40](https://api.github.com/repos/amehlhase316/ser321examples/commits?sha=master&per_page=40)



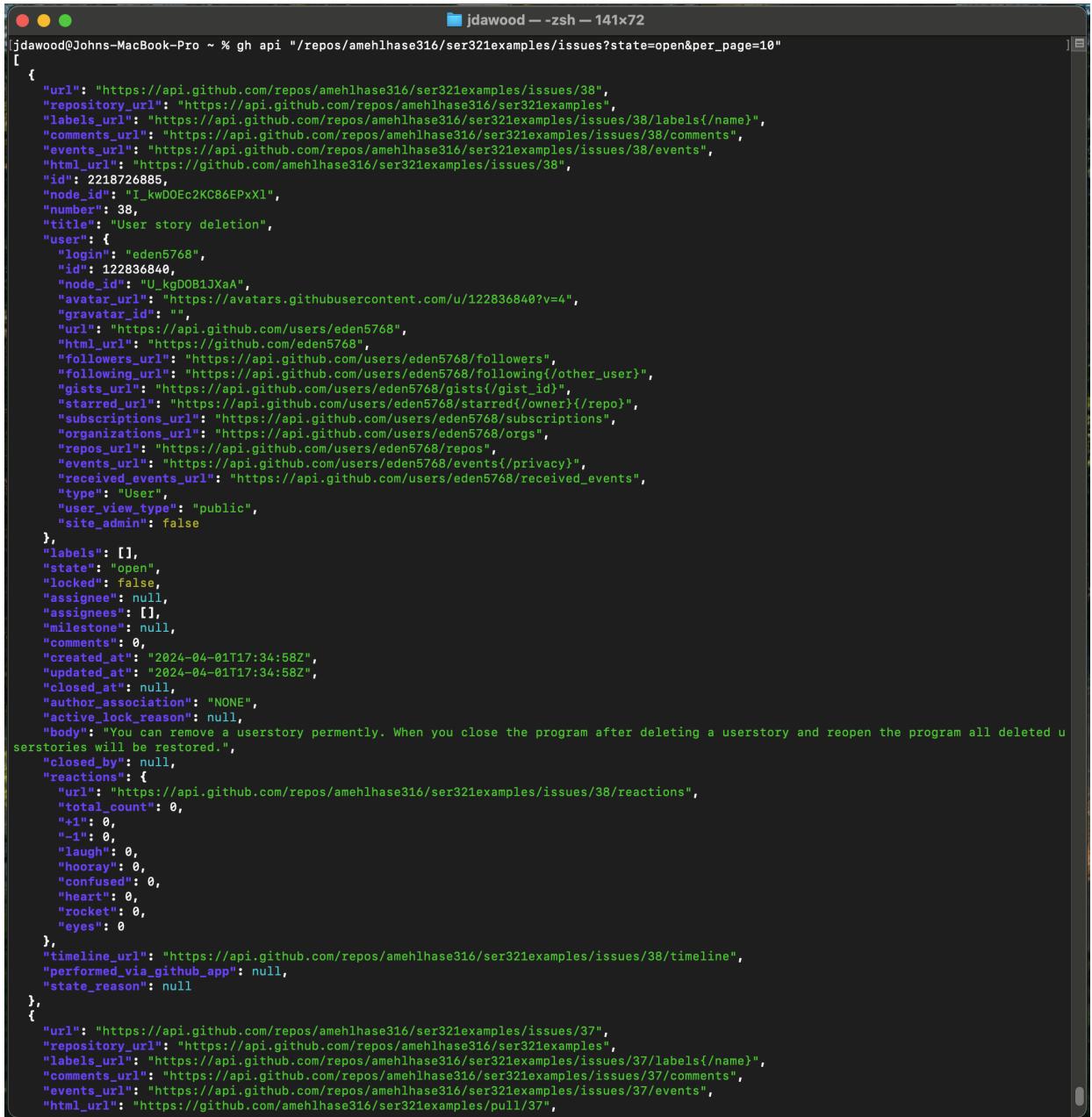
A screenshot of a terminal window titled "jdawood -- zsh -- 141x72". The command entered is "gh api "/repos/amehlhase316/ser321examples/commits?sha=master&per\_page=40". The output is a JSON object representing a commit. The commit has a sha of "75c07fffc4f7ff1c310de0c635d9e575175aeee1a", a node\_id of "C\_kwDOEc2KC9oAKDc1YzA3ZmZjNGY3ZmYxMxMGR1MGM2MzVkJOWU1NzUXNzVhZWV1MwE", and was made by the author "amehlhas" (email: a.mehlhase@asu.edu) at 2024-04-08T21:07:49Z. It was committed by the same person at 2024-04-08T21:13:51Z. The message is "EchoImpl in own class". The tree has a sha of "c6151caf09c2d31eb0113da968f6da48572d847e" and a url of "https://api.github.com/repos/amehlhase316/ser321examples/git/trees/c6151caf09c2d31eb0113da968f6da48572d847e". The commit has a comment\_count of 0, a verification status of false (unsigned), and no payload. The commit URL is "https://api.github.com/repos/amehlhase316/ser321examples/commits/75c07fffc4f7ff1c310de0c635d9e575175aeee1a", the HTML URL is "https://github.com/amehlhase316/ser321examples/commit/75c07fffc4f7ff1c310de0c635d9e575175aeee1a", and the comments URL is "https://api.github.com/repos/amehlhase316/ser321examples/commits/75c07fffc4f7ff1c310de0c635d9e575175aeee1a/comments". The author and committer are both "ser316asu" (id: 25068531). The JSON object contains many URLs for user and repository details.

```
[{"sha": "75c07fffc4f7ff1c310de0c635d9e575175aeee1a", "node_id": "C_kwDOEc2KC9oAKDc1YzA3ZmZjNGY3ZmYxMxMGR1MGM2MzVkJOWU1NzUXNzVhZWV1MwE", "commit": { "author": { "name": "amehlhas", "email": "a.mehlhase@asu.edu", "date": "2024-04-08T21:07:49Z" }, "committer": { "name": "amehlhas", "email": "a.mehlhase@asu.edu", "date": "2024-04-08T21:13:51Z" }, "message": "EchoImpl in own class", "tree": { "sha": "c6151caf09c2d31eb0113da968f6da48572d847e", "url": "https://api.github.com/repos/amehlhase316/ser321examples/git/trees/c6151caf09c2d31eb0113da968f6da48572d847e" }, "url": "https://api.github.com/repos/amehlhase316/ser321examples/commits/75c07fffc4f7ff1c310de0c635d9e575175aeee1a", "html_url": "https://github.com/amehlhase316/ser321examples/commit/75c07fffc4f7ff1c310de0c635d9e575175aeee1a", "comments_url": "https://api.github.com/repos/amehlhase316/ser321examples/commits/75c07fffc4f7ff1c310de0c635d9e575175aeee1a/comments", "author": { "login": "ser316asu", "id": 25068531, "node_id": "MDQ6VXNlcjI1MDY4NTMx", "avatar_url": "https://avatars.githubusercontent.com/u/25068531?v=4", "gravatar_id": "", "url": "https://api.github.com/users/ser316asu", "html_url": "https://github.com/ser316asu", "followers_url": "https://api.github.com/users/ser316asu/followers", "following_url": "https://api.github.com/users/ser316asu/following{/other_user}", "gists_url": "https://api.github.com/users/ser316asu/gists{/gist_id}", "starred_url": "https://api.github.com/users/ser316asu/starred{/owner}{/repo}", "subscriptions_url": "https://api.github.com/users/ser316asu/subscriptions", "organizations_url": "https://api.github.com/users/ser316asu/orgs", "repos_url": "https://api.github.com/users/ser316asu/repos", "events_url": "https://api.github.com/users/ser316asu/events{/privacy}", "received_events_url": "https://api.github.com/users/ser316asu/received_events", "type": "User", "user_view_type": "public", "site_admin": false }, "committer": { "login": "ser316asu", "id": 25068531, "node_id": "MDQ6VXNlcjI1MDY4NTMx", "avatar_url": "https://avatars.githubusercontent.com/u/25068531?v=4", "gravatar_id": "", "url": "https://api.github.com/users/ser316asu", "html_url": "https://github.com/ser316asu", "followers_url": "https://api.github.com/users/ser316asu/followers", "following_url": "https://api.github.com/users/ser316asu/following{/other_user}", "gists_url": "https://api.github.com/users/ser316asu/gists{/gist_id}", "starred_url": "https://api.github.com/users/ser316asu/starred{/owner}{/repo}", "subscriptions_url": "https://api.github.com/users/ser316asu/subscriptions", "organizations_url": "https://api.github.com/users/ser316asu/orgs", "repos_url": "https://api.github.com/users/ser316asu/repos", "events_url": "https://api.github.com/users/ser316asu/events{/privacy}", "received_events_url": "https://api.github.com/users/ser316asu/received_events", "type": "User" }}
```

[https://api.github.com/repos/amehlhase316/ser321examples/contributors?per\\_page=10](https://api.github.com/repos/amehlhase316/ser321examples/contributors?per_page=10)

```
[jdawood@Johns-MacBook-Pro ~ % gh api "/repos/amehlhase316/ser321examples/contributors?per_page=10"
[
  {
    "login": "amehlhase316",
    "id": 46384989,
    "node_id": "MDQ6VXNlcjQ2Mzg0OTg5",
    "avatar_url": "https://avatars.githubusercontent.com/u/46384989?v=4",
    "gravatar_id": "",
    "url": "https://api.github.com/users/amehlhase316",
    "html_url": "https://github.com/amehlhase316",
    "followers_url": "https://api.github.com/users/amehlhase316/followers",
    "following_url": "https://api.github.com/users/amehlhase316/following{/other_user}",
    "gists_url": "https://api.github.com/users/amehlhase316/gists{/gist_id}",
    "starred_url": "https://api.github.com/users/amehlhase316/starred{/owner}{/repo}",
    "subscriptions_url": "https://api.github.com/users/amehlhase316/subscriptions",
    "organizations_url": "https://api.github.com/users/amehlhase316/orgs",
    "repos_url": "https://api.github.com/users/amehlhase316/repos",
    "events_url": "https://api.github.com/users/amehlhase316/events{/privacy}",
    "received_events_url": "https://api.github.com/users/amehlhase316/received_events",
    "type": "User",
    "user_view_type": "public",
    "site_admin": false,
    "contributions": 53
  },
  {
    "login": "kgary",
    "id": 1686725,
    "node_id": "MDQ6VXNlcjE2ODY3MjU=",
    "avatar_url": "https://avatars.githubusercontent.com/u/1686725?v=4",
    "gravatar_id": "",
    "url": "https://api.github.com/users/kgary",
    "html_url": "https://github.com/kgary",
    "followers_url": "https://api.github.com/users/kgary/followers",
    "following_url": "https://api.github.com/users/kgary/following{/other_user}",
    "gists_url": "https://api.github.com/users/kgary/gists{/gist_id}",
    "starred_url": "https://api.github.com/users/kgary/starred{/owner}{/repo}",
    "subscriptions_url": "https://api.github.com/users/kgary/subscriptions",
    "organizations_url": "https://api.github.com/users/kgary/orgs",
    "repos_url": "https://api.github.com/users/kgary/repos",
    "events_url": "https://api.github.com/users/kgary/events{/privacy}",
    "received_events_url": "https://api.github.com/users/kgary/received_events",
    "type": "User",
    "user_view_type": "public",
    "site_admin": false,
    "contributions": 5
  },
  {
    "login": "klmoor21",
    "id": 56544899,
    "node_id": "MDQ6VXNlcjU2NTQ0ODk5",
    "avatar_url": "https://avatars.githubusercontent.com/u/56544899?v=4",
    "gravatar_id": "",
    "url": "https://api.github.com/users/klmoor21",
    "html_url": "https://github.com/klmoor21",
    "followers_url": "https://api.github.com/users/klmoor21/followers",
    "following_url": "https://api.github.com/users/klmoor21/following{/other_user}",
    "gists_url": "https://api.github.com/users/klmoor21/gists{/gist_id}",
    "starred_url": "https://api.github.com/users/klmoor21/starred{/owner}{/repo}",
    "subscriptions_url": "https://api.github.com/users/klmoor21/subscriptions",
    "organizations_url": "https://api.github.com/users/klmoor21/orgs",
    "repos_url": "https://api.github.com/users/klmoor21/repos",
    "events_url": "https://api.github.com/users/klmoor21/events{/privacy}",
    "received_events_url": "https://api.github.com/users/klmoor21/received_events",
    "type": "User",
    "user_view_type": "public",
    "site_admin": false,
    "contributions": 2
  },
  {
    "login": "ser316asu",
    "id": 25068531,
    "node_id": "MDQ6VXNlcjI1MDY4NTMx",
  }
]
```

[https://api.github.com/repos/amehlhase316/ser321examples/issues?  
state=open&per\\_page=10](https://api.github.com/repos/amehlhase316/ser321examples/issues?state=open&per_page=10)



A screenshot of a terminal window titled "jdawood -- zsh -- 141x72". The command entered is "gh api "/repos/amehlhase316/ser321examples/issues?state=open&per\_page=10". The output is a JSON object representing an issue. The issue has the following properties:

- url**: https://api.github.com/repos/amehlhase316/ser321examples/issues/38
- repository\_url**: https://api.github.com/repos/amehlhase316/ser321examples
- labels\_url**: https://api.github.com/repos/amehlhase316/ser321examples/issues/38/labels{/name}
- comments\_url**: https://api.github.com/repos/amehlhase316/ser321examples/issues/38/comments
- events\_url**: https://api.github.com/repos/amehlhase316/ser321examples/issues/38/events
- html\_url**: https://github.com/amehlhase316/ser321examples/issues/38
- id**: 2218726885
- node\_id**: `I_kwDOEc2KC86EPxXl`
- number**: 38
- title**: "User story deletion"
- user**:
  - login**: eden5768
  - id**: 122836840
  - node\_id**: U\_kgDOB1JXaA
  - avatar\_url**: https://avatars.githubusercontent.com/u/122836840?v=4
  - gravatar\_id**: ""
  - url**: https://api.github.com/users/eden5768
  - html\_url**: https://github.com/eden5768
  - followers\_url**: https://api.github.com/users/eden5768/followers
  - following\_url**: https://api.github.com/users/eden5768/following{/other\_user}
  - gists\_url**: https://api.github.com/users/eden5768/gists{/gist\_id}
  - starred\_url**: https://api.github.com/users/eden5768/starred{/owner}{/repo}
  - subscriptions\_url**: https://api.github.com/users/eden5768/subscriptions
  - organizations\_url**: https://api.github.com/users/eden5768/orgs
  - repos\_url**: https://api.github.com/users/eden5768/repos
  - events\_url**: https://api.github.com/users/eden5768/events{/privacy}
  - received\_events\_url**: https://api.github.com/users/eden5768/received\_events
  - type**: "User"
  - user\_view\_type**: "public"
  - site\_admin**: false
- labels**: []
- state**: "open"
- locked**: false
- assignee**: null
- assignees**: []
- milestone**: null
- comments**: 0
- created\_at**: "2024-04-01T17:34:58Z"
- updated\_at**: "2024-04-01T17:34:58Z"
- closed\_at**: null
- author\_association**: "NONE"
- active\_lock\_reason**: null
- body**: "You can remove a userstory permently. When you close the program after deleting a userstory and reopen the program all deleted userstories will be restored."
- closed\_by**: null
- reactions**:
  - url**: https://api.github.com/repos/amehlhase316/ser321examples/issues/38/reactions
  - total\_count**: 0
  - +1**: 0
  - 1**: 0
  - laugh**: 0
  - hooray**: 0
  - confused**: 0
  - heart**: 0
  - rocket**: 0
  - eyes**: 0
- timeline\_url**: https://api.github.com/repos/amehlhase316/ser321examples/issues/38/timeline
- performed\_via\_github\_app**: null
- state\_reason**: null

## API Call 1: List Commits for the master Branch

- Command:

```
gh api "/repos/amehlhase316/ser321examples/commits?sha=master&per_page=40"
```

- Purpose: This API call fetches the list of up to 40 commits from the master branch of the repository ser321examples owned by the user amehlhase316. The parameter sha=master specifies that we want commits from the master branch, and per\_page=40 increases the default limit of results from 30 to 40.
- Link to  
Documentation: <https://docs.github.com/en/rest/commits/commits?apiVersion=2022-11-28#list-commits>

## API Call 2: List Contributors to the Repository

- Command:

```
gh api "/repos/amehlhase316/ser321examples/contributors?per_page=10"
```

- Purpose: This API call retrieves a list of up to 10 contributors to the ser321examples repository. The query parameter per\_page=10 limits the number of contributors returned in the response to 10.
- Link to  
Documentation: <https://docs.github.com/en/rest/commits/commits?apiVersion=2022-11-28#list-repository-contributors>

## API Call 3: List Open Issues in the Repository

- Command:

```
gh api "/repos/amehlhase316/ser321examples/issues?state=open&per_page=10"
```

- Purpose: This API call returns up to 10 open issues from the ser321examples repository. The state=open query parameter ensures only open issues are returned, and per\_page=10 limits the number of results to 10 issues.
- Link to  
Documentation: <https://docs.github.com/en/rest/issues/issues?apiVersion=2022-11-28#list-repository-issues>

## **Stateless Communication:**

- Stateless communication means that each request from the client to the server is independent. The server does not retain any information about previous requests. In this model, the client must include all the information needed to process the request in each individual request.
- **Example:** HTTP (the protocol used for web communication) is stateless. Each API call you make to GitHub (like fetching commits or contributors) does not rely on previous API calls. GitHub does not remember the state of previous requests unless a token or session data is explicitly included in each request.

## **Stateful Communication:**

- Stateful communication, on the other hand, means that the server retains some information (or "state") about the client across multiple requests. This allows the server to maintain session information, such as login status, between different requests.
- **Example:** FTP (File Transfer Protocol) is an example of a stateful protocol. In FTP, a connection is established, and the server retains information about the session, such as current working directories, until the connection is explicitly closed.

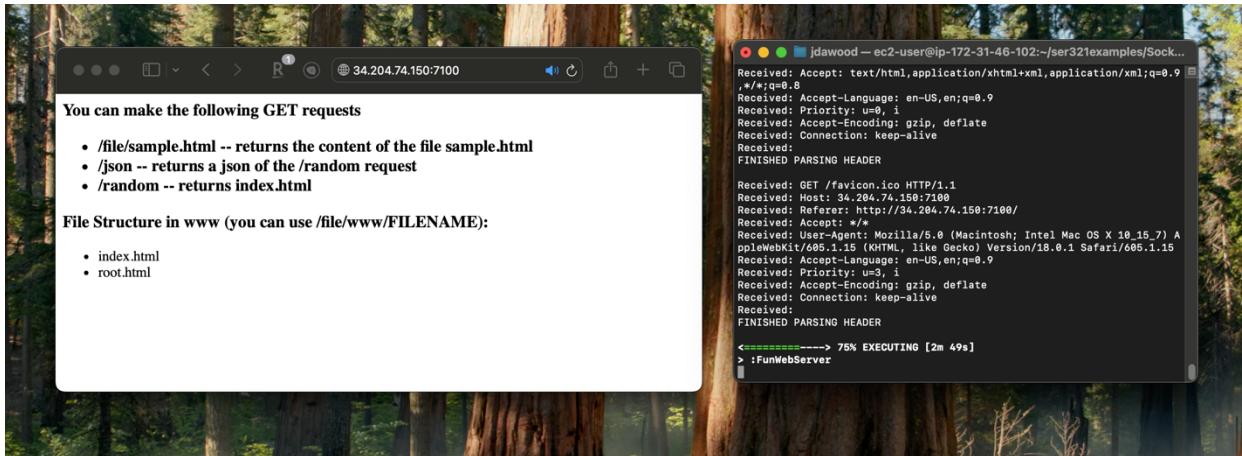
## 2 Set up your second system and run servers on it

For this part, you will need to set up your second machine (AWS, which you should have already done). See the setup page on Canvas. We will call your local machine "first machine" and your second one (AWS) "second machine" in this document.

### 2.1 Getting sample code onto your systems (should be done already)

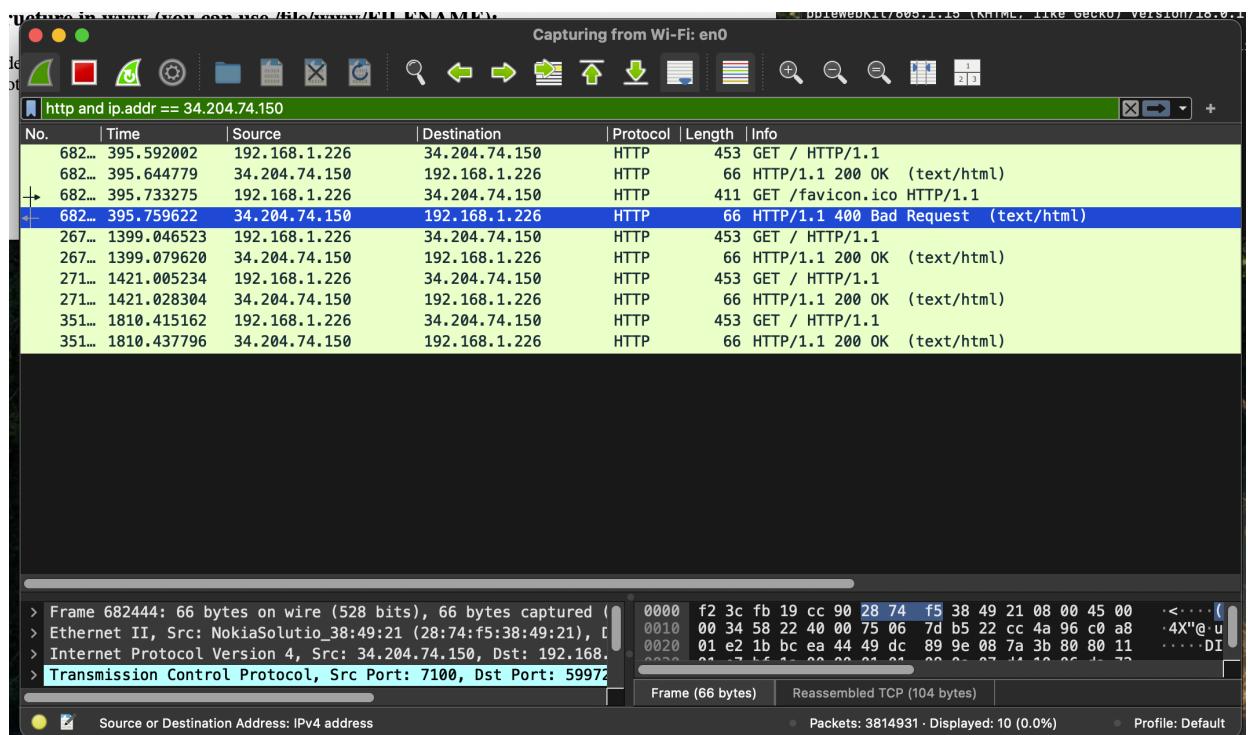
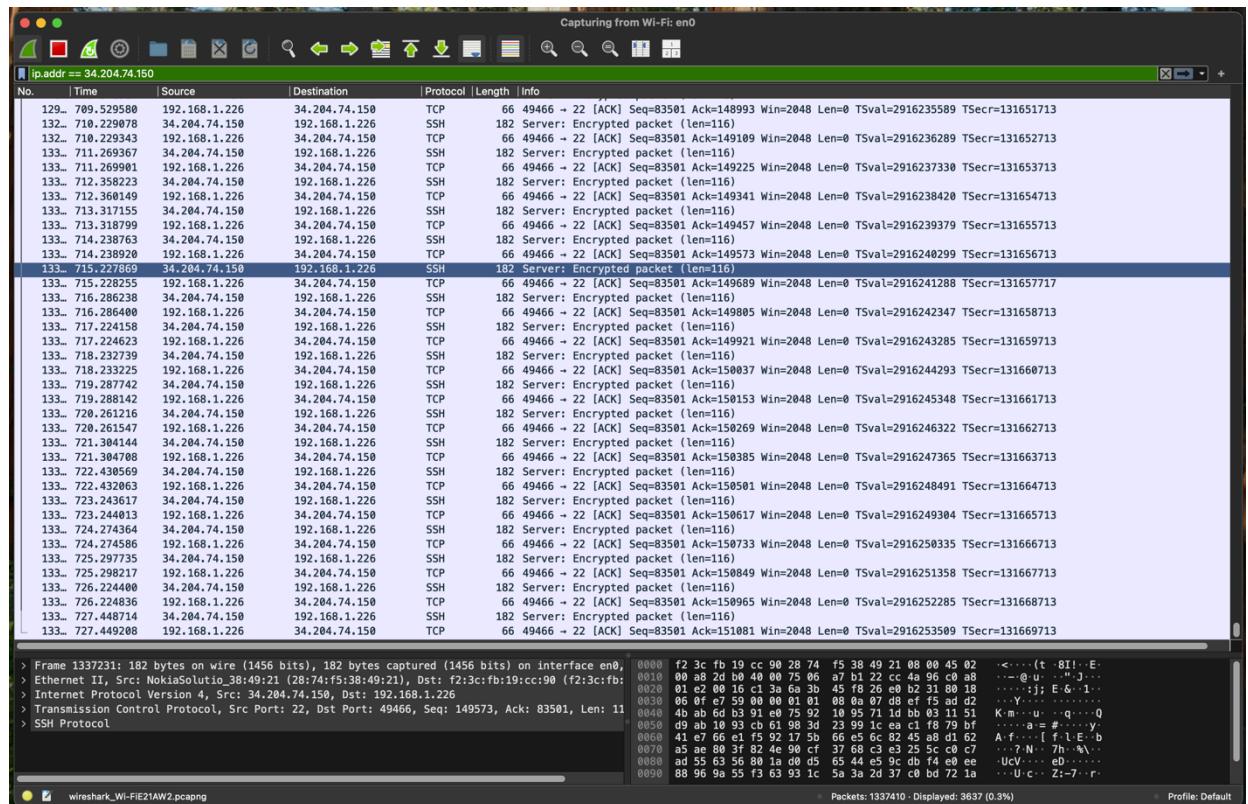
Now that you have your second machine set up, you should make sure the example GitHub repo is available on both machines. I would advise you to fork the given repository and clone it on all the machines you want to work on - so you can still make changes to commit, push, and pull. This fork will be public - thus it is not for your assignment changes, but just for "playing" with the examples.

### 2.2 Running a Simple Java WebServer



### 2.3 Analyze what happens

Wireshark should still be running in the background. Go to Wireshark, and create a filter that shows network traffic to and from your WebServer. Take a screenshot of your Wireshark capture and add it to your document.



### Questions and Answers:

1. What filter did you use? Explain why you chose that filter.
  - **Filter Used:** ip.addr == 34.204.74.150 or http and ip.addr == 34.204.74.150
  - **Explanation:** This filter is used to capture all traffic between my local machine and the AWS instance hosting the web server. By filtering on the server's public IP address, I can isolate traffic to and from the web server to analyze the HTTP requests and responses effectively.
2. What happens when you are on the /random page and click the "Random" button? Compare this to refreshing your browser. (You can also use the command line output that the WebServer generates to answer this.)
  - When I click the "Random" button on the /random page, it sends an HTTP GET request to the server to fetch the index.html file. The button triggers a predefined behavior of loading the index.html file from the server.
  - **Comparison with Refreshing:** Refreshing the browser resends the last GET request for the current URL, which reloads the same resource (in this case, /random). In contrast, clicking the "Random" button always retrieves the index.html file, regardless of which page I'm on.
3. What types of response codes are you able to receive through different requests to your server?
  - **200 OK:** For successful GET requests, such as retrieving index.html or root.html.
  - **404 Not Found:** When requesting a non-existent file or page (e.g., /file/missing.html).
  - **500 Internal Server Error:** If there's a problem with server-side code execution or misconfigured paths.
4. Explain the response codes you receive and why you get them.
  - **200 OK:** Received when the server successfully processes and returns the requested resource.
  - **400:** bad request from server. (e.g. no favicon.ico)
5. When you do a <publicIPOfYourSecondMachine>:9000, take a look at what Wireshark generates as a server response. Are you able to find the data that the server sends back to you? (This should be the "Data" section of your response.)
  - Yes, in Wireshark, I can see the data section containing the HTML content of the page served by the server. The data is part of the HTTP response packets,

and Wireshark allows me to inspect the raw HTML being returned to the browser (such as the content of index.html or root.html).

6. Based on the previous question, explain why HTTPS is now more common than HTTP.

- **Security:** HTTP sends data in plain text, meaning anyone monitoring the traffic (such as on an open Wi-Fi network) can intercept and read the data. This includes sensitive information like passwords or personal data. HTTPS encrypts the data between the client and server, making it much harder for attackers to intercept or tamper with the communication.
- In the Wireshark capture, since the server uses HTTP, I can see the exact data being transmitted. If this were HTTPS, the data would be encrypted and unreadable in Wireshark.

7. In our case - what port does the server listen to for HTTP requests, and is that the most common port for HTTP?

- The server is listening on **port 7100**.
- **Is it common?:** No, the standard port for HTTP is **port 80**. Ports like 7100 are less common and are typically used when the standard port 80 is unavailable or explicitly chosen for a specific use case.

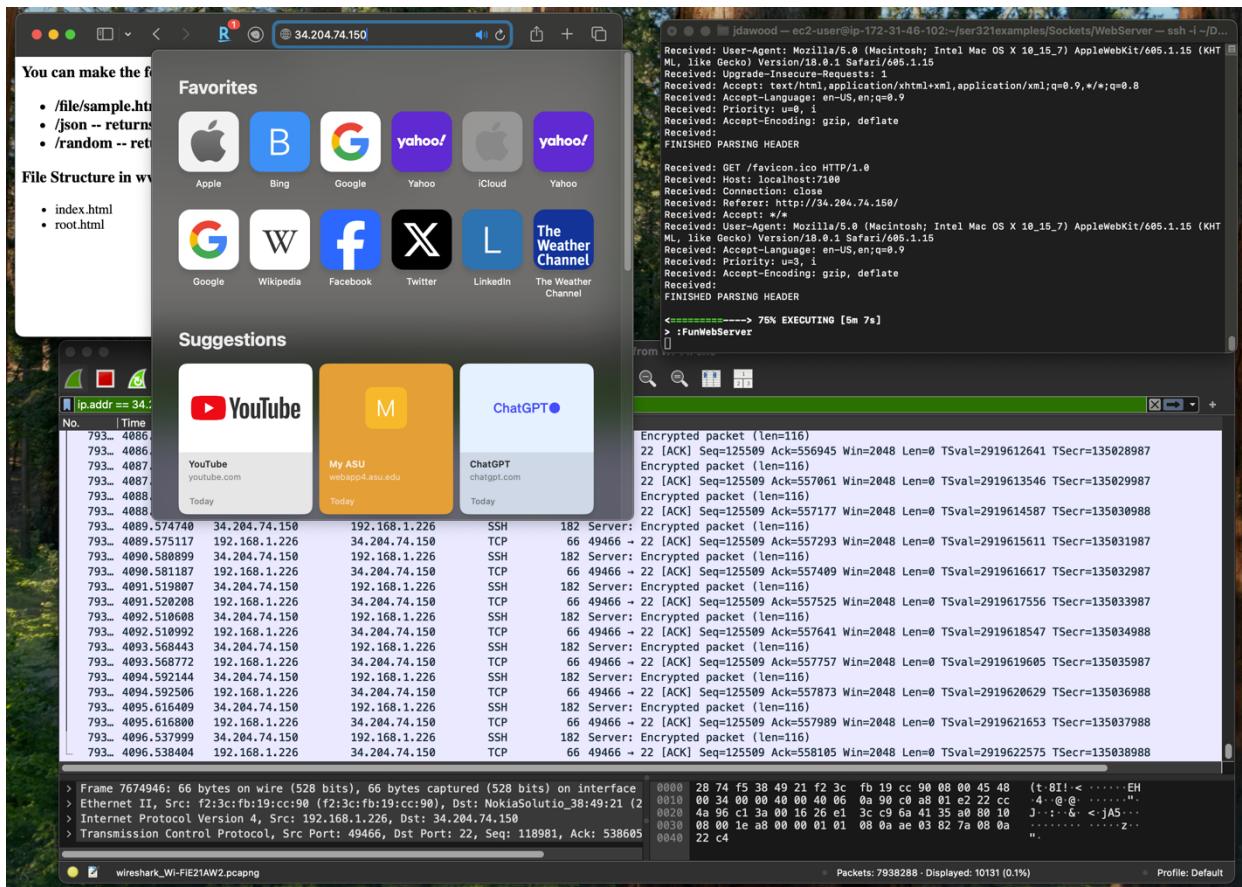
8. Which local port is used when sending different requests to the WebServer?

- The local port used for sending requests to the web server is dynamically assigned by the operating system. For example, in Wireshark, you might see ports like **49152** or higher, which are commonly used by the system as ephemeral (temporary) ports for outbound connections. The local port changes with each new session or connection.

## 2.4 Setting up a "real" WebServer

### Questions and Answers:

1. What is the URL that you can now use to reach the main page?
  - <http://34.204.74.150>: This is the public IP address of your AWS instance, and Nginx is configured to forward requests from port 80 to your Java FunWebServer running on port 7100.
2. Check the traffic to your WebServer. What port is the traffic going to now? Is it the same as before, or is it (and should it) be different?
  - The traffic is now going to **port 80**, as shown in the Wireshark capture and your Nginx configuration.
  - This is different from before, where the traffic was going directly to **port 7100**. It should be different because Nginx is now handling the requests on **port 80** and forwarding them to port 7100.
3. Is it still using HTTP, or is it now using HTTPS? Why?
  - The connection is still using **HTTP**.
  - It hasn't switched to **HTTPS** because we haven't set up SSL/TLS encryption for Nginx. For HTTPS, we would need to configure Nginx with an SSL certificate, and requests would be served over **port 443**.
4. Could you change your security settings on AWS now?
  - Yes, you could update your AWS security group to allow **port 80** (HTTP traffic) and, if you set up SSL in the future, **port 443** (HTTPS traffic). This would ensure that inbound traffic is properly routed through Nginx.
  - Currently, AWS security should be allowing port 80 for HTTP traffic, along with **port 22** for SSH access.
5. Take a screenshot of your web browser, your second machine, and the port number on Wireshark. This should be similar to the screenshot you took before (but also with Wireshark), and add it to your document for this task. Note: If we are unable to see that you reached the WebServer with the "different URL", we will not know if you actually set up the server correctly. Therefore, please make sure that it shows up if you want points for this task.



## 2.5 Setting up HTTPS

### Questions and Answers:

1. What port is your traffic going through now?

- The traffic is now being routed through **port 443**, which is the standard port for HTTPS traffic. After configuring and enabling SSL with Certbot, our Nginx server is handling secure communications over HTTPS.

2. Can you still find the plain text responses that were found with HTTP?

- No, after switching to HTTPS, the traffic is encrypted. This means that any previously visible plain text responses found when using HTTP over port 80 are no longer accessible. HTTPS encrypts the data, so even if tools like Wireshark capture the packets, the contents will not be readable without decrypting the SSL/TLS data. However, if HTTP (port 80) is still active and accessible, then plain text responses can still be found through that protocol, although this is discouraged for security reasons. Once HTTPS is in place, browsers will default to using the secure connection.

You can make the following GET requests

- /file/sample.html -- returns the content of the file sample.html
- /json -- returns a json of the /random request
- /random -- returns index.html

File Structure in www (you can use /file/www/FILENAME):

- index.html
- root.html

```
t/601.2.4 (KHTML, like Gecko) Version/9.0.1 Safari/601.2.4 facebookeexternalhit/1.1 Facebot TwitterBot/1.0 Received: Upgrade-Insecure-Requests: 1 Received: Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.9 Received: Accept-Language: en-US,en;q=0.9 Received: Priority: u=0, i Received: Accept-Encoding: gzip, deflate Received: FINISHED PARING HEADER Received: GET /favicon.ico HTTP/1.0 Received: Host: localhost:7100 Received: Connection: close Received: User-Agent: com.appleWebKit.Networking/20619.1.26.31.7 Network/4277.1 .7 macOS/15.0.1 Received: Accept: */* Received: Accept-Language: en-US,en;q=0.9 Received: Priority: u=3, i Received: Accept-Encoding: gzip, deflate Received: FINISHED PARING HEADER <====> 75% EXECUTING [7m 50s] > FunWebServer
```

Capturing from Wi-Fi: en0

No.	Time	Source	Destination	Protocol	Length	Info
236..	14411.368063	192.168.1.226	34.204.74.150	TCP	66	49499 → 22 [ACK] Seq=5102 Ack=64058 Win=131072 Len=0 TSval=3762943103 TSecr=145353842
236..	14412.390872	34.204.74.150	192.168.1.226	SSHv2	182	Server: Encrypted packet (len=116)
236..	14412.391330	192.168.1.226	34.204.74.150	TCP	66	49499 → 22 [ACK] Seq=5102 Ack=64174 Win=131072 Len=0 TSval=3762944127 TSecr=145354842
236..	14413.414384	34.204.74.150	192.168.1.226	SSHv2	182	Server: Encrypted packet (len=116)
236..	14413.416664	192.168.1.226	34.204.74.150	TCP	66	49499 → 22 [ACK] Seq=5102 Ack=64290 Win=131072 Len=0 TSval=3762945152 TSecr=145355842
236..	14414.337089	34.204.74.150	192.168.1.226	SSHv2	182	Server: Encrypted packet (len=116)
236..	14414.337450	192.168.1.226	34.204.74.150	TCP	66	49499 → 22 [ACK] Seq=5102 Ack=64406 Win=131072 Len=0 TSval=3762946073 TSecr=145356842
236..	14415.462462	34.204.74.150	192.168.1.226	SSHv2	182	Server: Encrypted packet (len=116)
236..	14415.462806	192.168.1.226	34.204.74.150	TCP	66	49499 → 22 [ACK] Seq=5102 Ack=64522 Win=131072 Len=0 TSval=3762947198 TSecr=145357842
236..	14416.384932	34.204.74.150	192.168.1.226	SSHv2	182	Server: Encrypted packet (len=116)
236..	14416.385335	192.168.1.226	34.204.74.150	TCP	66	49499 → 22 [ACK] Seq=5102 Ack=64638 Win=131072 Len=0 TSval=3762948121 TSecr=145358842
236..	14417.405996	34.204.74.150	192.168.1.226	SSHv2	182	Server: Encrypted packet (len=116)
236..	14417.406427	192.168.1.226	34.204.74.150	TCP	66	49499 → 22 [ACK] Seq=5102 Ack=64754 Win=131072 Len=0 TSval=3762949142 TSecr=145359842
236..	14418.338536	34.204.74.150	192.168.1.226	SSHv2	182	Server: Encrypted packet (len=116)
236..	14418.338947	192.168.1.226	34.204.74.150	TCP	66	49499 → 22 [ACK] Seq=5102 Ack=64870 Win=131072 Len=0 TSval=3762950066 TSecr=145360842
236..	14419.456204	34.204.74.150	192.168.1.226	SSHv2	182	Server: Encrypted packet (len=116)
236..	14420.361186	34.204.74.150	192.168.1.226	TCP	66	49499 → 22 [ACK] Seq=5102 Ack=64986 Win=131072 Len=0 TSval=3762951192 TSecr=145361842
236..	14421.402083	34.204.74.150	192.168.1.226	SSHv2	182	Server: Encrypted packet (len=116)
236..	14421.402473	192.168.1.226	34.204.74.150	TCP	66	49499 → 22 [ACK] Seq=5102 Ack=65218 Win=131072 Len=0 TSval=3762953138 TSecr=145363842

```
> Frame 21120536: 278 bytes on wire (2224 bits), 278 bytes captured (2224 bits) on inter 0000 f2 3c fb 19 cc 90 28 74 f5 38 49 21 08 00 45 00 <----(t .81!- E-  
> Ethernet II, Src: NokiaSolutio_38:49:21 (28:74:f5:38:49:21), Dst: f2:3c:fb:19:cc:90 (f 0010 01 08 49 22 40 00 75 06 8b e1 22 cc 4a 96 c8 a8 -I'@ u .."J...  
> Internet Protocol Version 4, Src: 34.204.74.150, Dst: 192.168.1.226 0020 01 e2 00 16 c1 3a 6a 4d 5a dc 2b e2 15 c9 88 18 .....:JM Z &...  
> Transmission Control Protocol, Src Port: 22, Dst Port: 49466, Seq: 1334569, Ack: 17453 0030 06 3c ad d5 00 00 01 01 08 0a 08 78 ef dd ae 6a <----P ..]  
0040 48 24 14 40 69 99 40 c7 53 b1 86 43 ed d2 12 62 Hs M -M S f ...  
0050 7d 53 88 96 38 2b 9a e0 e5 68 96 11 4f c4 0b d9 JS -8+.. h 0...  
Packets: 23677837 - Displayed: 28031 (0.1%) Profile: Default
```

## 2.6 Some programming on your WebServer

This should be done whether you have nginx running or not! This can also be done locally on your working machine, I would even advise doing this locally. I'd advise spending some time on the WebServer code and playing around with it, such that you understand what happens and how things work.

### 2.6.1 Multiply

I modified the WebServer's **Multiply** functionality to include proper error handling. Specifically, I addressed the following scenarios:

- **Missing Parameters:** If either num1 or num2 is missing from the query parameters, the server responds with a 400 Bad Request status and an error message indicating that both parameters are required. We chose 400 Bad Request because the client did not send a valid request.
- **Non-numeric Input:** If the provided inputs are not valid numbers (e.g., if the user sends a string instead of a number), the server responds with 400 Bad Request and an error message indicating that numeric values are required.
- **Unexpected Errors:** If any other unexpected errors occur, the server responds with a **500 Internal Server Error**, ensuring that the server remains stable and does not crash.

By implementing this, the server now handles invalid input gracefully and responds with appropriate HTTP status codes. This ensures better client-side handling and prevents server crashes.

### 2.6.2 GitHub

GitHub repo Assignment2 folder.

### 2.6.3 Make your own requests

URL with example data:

1. Example URL: <http://localhost:9000/convertCurrency?amount=100&from=USD&to=EUR>
2. Converted amount: 85.0 EUR
3. Example URL: <http://localhost:9000/greet?name=John&language=Spanish>
4. Hola, John!

### 2.6.4 WebServer for Everyone

Server, <http://34.204.74.150:9000> #servers on Slack.

## **2.6.5 Test other WebServers**

Comment #1, <comment Link> #comment on Slack.

Comment #2, <comment Link> #comment on Slack.

**# Submitted them to Canvas**