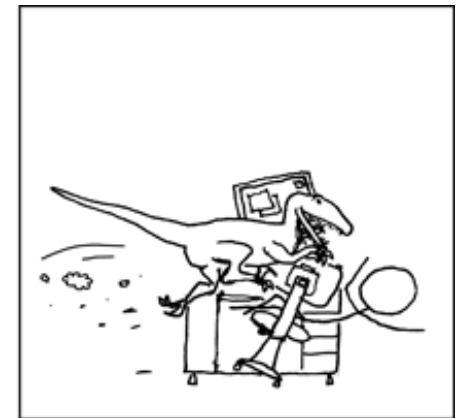
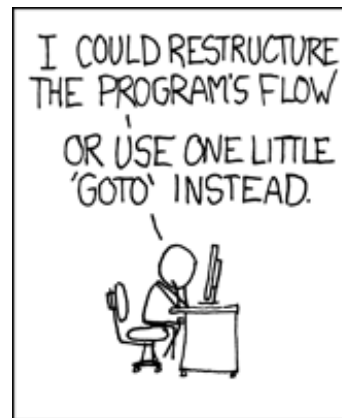


# CSc 337



## LECTURE 17: POST AND PUBLIC APIS

# File Turn In Client

---

Write a page that allows the user to enter their name, email and select an assignment number from a dropdown menu. It should also include a text area where they can paste their code and a submit button.

When the user clicks the submit button the information should be sent to the server as a `POST` request.

# File Turn In Service

---

Write a web service that accepts that data your client page posted. Your service should save the code from the text area into a file named the student's name and section.

The service should send a success message back to the client if it was successful and a failure message back if it was not.

File saving information on the next slide.

# Writing Files

---

appending to a file:

```
fs.appendFile(filename, filecontent, function(err) {  
    if(err) {  
        return console.log(err);  
    }  
    console.log("The file was saved!");  
});
```

writing to a file:

```
fs.writeFile(filename, filecontent, function(err) {  
    if(err) {  
        return console.log(err);  
    }  
    console.log("The file was saved!");  
});
```

# File Turn In Confirmation

---

Have your page display a confirmation message stating whether the request was successful. This message should include the code the user submitted if the request came back successful.

# Final Project

---

- Individual or in pairs
- Creative - can have whatever focus you choose
- Big picture requirements:
  - Must involve server side and client side code
  - A little bigger than a standard 337 project
- Deadlines: proposal (4/5), code (4/25) and presentation (4/25 and 4/30)

# Topics we will cover

---

- Canvas (good for games and animations)
- Databases (a better way to store data than text files)
- Regular expressions (good for input validation)
- Cookies and sessions (good for auto filling out forms with what the user input last time and logins)

You are also welcome to use other things you learn on your own.

# Public APIs

---

There are many publicly available APIs that you can make requests to on the web.

One list: <https://github.com/toddmotto/public-apis>



# Public API Authentication

---

There are three common types of Public API Authentication:

- None
- API Key
  - an extra parameter that you send as an extra parameter whenever you make a request
- OAuth
  - redirect to have the user authenticate through API

# Rate Limiting

---

Some APIs restrict how often you can access them

- Why do they do this?
- What can you do if you need to access them more frequently than they allow?