

Server Side

```
express().listen(80)
```

REST

C	post
R	get
U	put
D	delete

Best Practices

- think about “resources”
- elements & collections
- map out the 4 methods for each
- Prefer Nouns, Plural, Concrete
- Use Parameters for more advanced queries

Express

- routes
- static files
- middleware
- template engine - ejs
- method override

Session Management

HTTP is stateless

Express

- Routes are served by a callback
- Resources can be static or dynamically generated

```
app.post("/courses", (req, res) => {  
  db.add({  
    id: req.body.code,  
    when: new Date(),  
    what: req.body.what,  
    who: req.body.who  
  });  
  res.redirect("/courses")  
});
```

```
app.get("/courses/csc404", (req, res) => {  
  res.send(`<html>  
    <body>  
      <p>This is not a REAL course</p>  
    </body>  
  </html>`)  
});
```

Express

- Express provides a middleware architecture

```
function myLogger(req, res, next) {  
  console.log('LOG:', req.method, req.url, req.body)  
  res.remember = req.url  
  next()  
}  
  
app.use(myLogger)
```

Cookies

- HTTP is stateless
- HTTP cookies provide illusion of state!
 - Managed by server
 - Stored by browser