Resources

Emily Herbert

```
function login(req) {
    function F(resp) {
        let u = req.body.username;
        let p = req.body.username;
        if(resp[u] === p) {
            respond('ok');
        } else {
            respond('error');
        }
    }
    get('passwords.json', F);
}
```

Figure 1. Login

```
let c = require('containerless');

function main(req) {
    function F(resp) {
        let u = req.body.username;
        let p = req.body.password;
        if (resp[u] === p) {
            c.respond('ok');
        } else {
            c.respond('error');
        }
    }
    c.get('passwords.json', F);
}
```

Figure 2. Login

References

```
let c = require('containerless');

function main(req) {
    function F(resp) {
        let u = req.body.username;
        let p = req.body.password;
        if (resp[u] === p) {
            c.respond('ok');
        } else {
            c.respond('error');
        }
    }
    c.get('passwords.json', F);
}

c.listen(main);
```

```
let c = require('containerless');
let t = require('containerless/tracing');

function main(req) {
    let [_req] = t.popArgs();
    function F(resp) {
        let [_resp] = t.popArgs();
        let _clos = t.popClosure();
        t.let('req', t.getClos(_clos, 'req'));
        let u = req.body.username;
        t.let('u', t.get(t.get(t.id('req'), 'body'), 'username'));
        let p = req.body.password;
        t.let('p', t.get(t.get(t.id('req'), 'body'), 'password'));
        t.if(t.eq(t.vget(_resp, t.id('u')), t.id('p')));
        if (resp[u] === p) {
            t.ifTrue();
            t.pushArgs(t.str('ok'));
            c.respond('ok');
            t.popResult();
        } else {
            t.ifFalse();
            t.pushArgs(t.str('error'));
            c.respond('error');
            t.popResult();
        }
        t.exitIf();
        t.exitFunction(t.undefined);
    }
    t.let('F', t.closure({ 'req': _req }));
    t.pushArgs([t.str('passwords.json'), t.id('F')]);
    c.get('passwords.json', F);
    t.popResult();
}

c.listen(main);
```

if (resp[u] === p) {respond('ok'); } **else** { ☠ }

(a) The IR program produced when the username and password combination is correct.

if (resp[u] === p) { ☠ } **else** { respond('error'); }

(b) The IR program produced when the username and password combination is incorrect.

```
if(resp[u] === p) {
    respond('ok');
} else {
    ☠
}
```

```
  if(resp[u] === p) {
      respond('ok');
  } else {
      respond('error');
  }
```

```
  let u = req.body.username;
  let p = req.body.password;
```

```
event('listen', [], closure(), callback (clos, req) {
  event('get', ['passwords.json'], closure(&req),
    callback(clos, resp) {
      let req = *(clos.req);
      // body
  });
});
```

```
event('listen', [], closure(), callback (clos, req) {
  event('get', ['passwords.json'],
    closure(&req), callback(clos, resp) {
      let req = *(clos.req);
      let u = req.body.username;
      let p = req.body.password;
      if (resp[u] === p) {
        respond('ok');
      } else {
        respond('error');
      }
    });
});
```