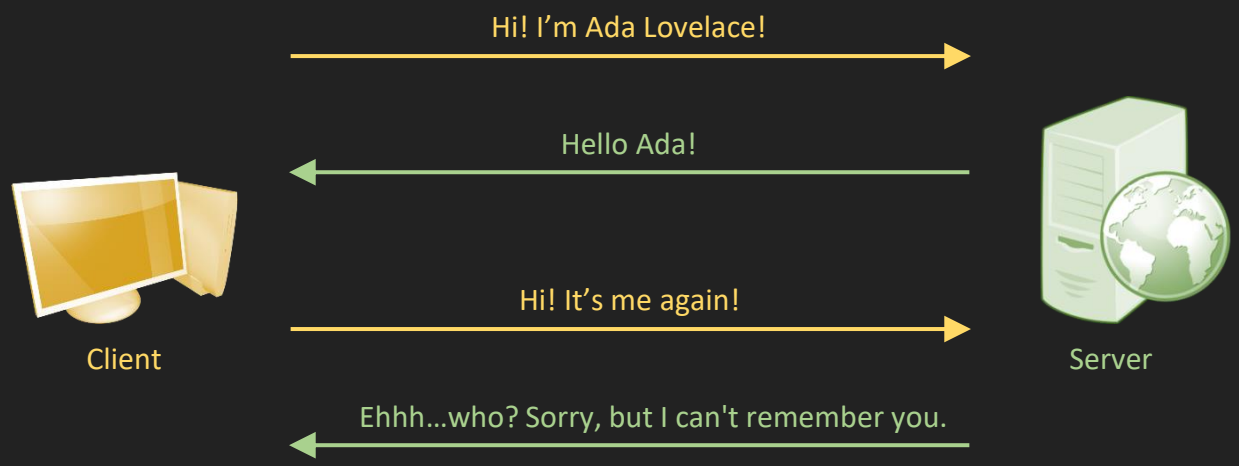
This work is licensed under CC BY 4.0

Sessions

Why web applications need session and how it works in Express

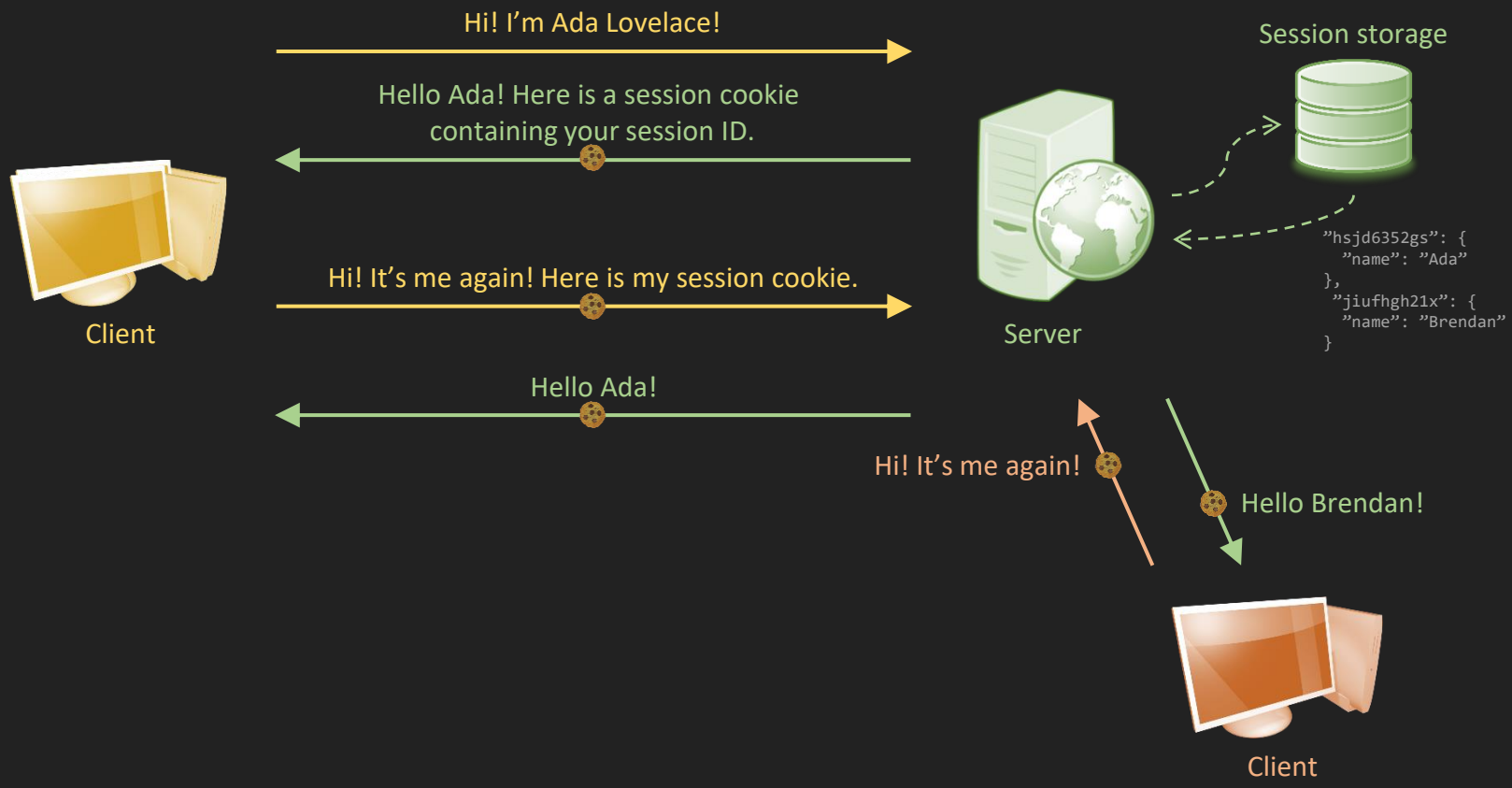


The Web Server Does Not Have a Clue About the State



- ✓ Neither the server nor the browser has an intrinsic way of knowing that it is the same browser that visits the same site.
  - (How is it then possible to log on to something?)

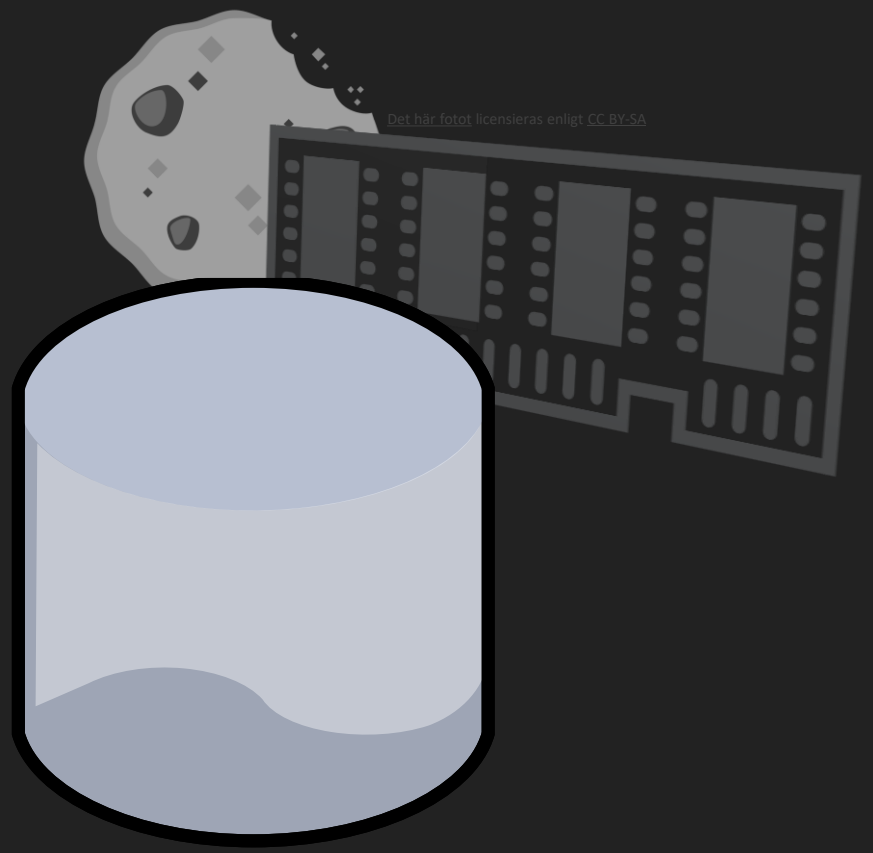
Maintain the State With Sessions



- ✓ A session cookie contains a session ID, a unique identifier used to index the server's session storage.

Storing Session Data

- ✓ In a cookie at the client.
  - Store everything in the cookie.
    - Tampering?
      - JSON Web Token (https://jwt.io)
  - Secret information?
    - Encryption!
- ✓ In the web server's memory.
  - What happens when the server is restarted?
  - Scalability?
- ✓ Session server storage.
  - Redis (in-memory data structure store).
  - MongoDB.



Managing Sessions in Express

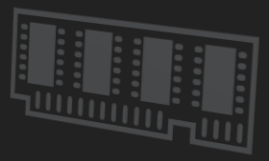
```
import session from 'express-session'

// Setup and use session middleware (https://github.com/expressjs/session)
const sessionOptions = {
  name: process.env.SESSION_NAME, // Don't use default session cookie name.
  secret: process.env.SESSION_SECRET, // Change it!!! The secret is used to hash the session with HMAC.
  resave: false, // Resave even if a request is not changing the session.
  saveUninitialized: false, // Don't save a created but not modified session.
  cookie: {
    maxAge: 1000 * 60 * 60 * 24,  // 1 day
    sameSite: 'strict'
  }
}

app.use(session(sessionOptions))
```

```
# .env
# ------------------------------------------------------------------------
# NOTE! Never push the .env file to a repository. In this solution,
#       it is added to the repository just for clarity.
# ------------------------------------------------------------------------
DB_CONNECTION_STRING="mongodb://localhost:27017/justtaskit"
PORT=8081
BASE_URL="/"
SESSION_NAME="just-task-it.user-session"
SESSION_SECRET="cussed-salute-strangely"
```

- ✓ The default server-side session storage is `MemoryStorage`.
  - Not for production (use Redis).
  - Sufficient for development and testing needs.



- ✓ Just the session ID is stored in the cookie itself.

Using Session Variables

- ✓ Session variables are just properties of the request object's session property.
  - (The response object doesn't have a session property.)

```
req.session.name = 'Ada'       // creates the session variable 'name'

const name = req.session.name // gets the value of the session variable 'name'

delete req.session.name        // removes the session variable 'name'

req.session.destroy()          // destroys this session
req.session.regenerate()       // regenerates the session
```

