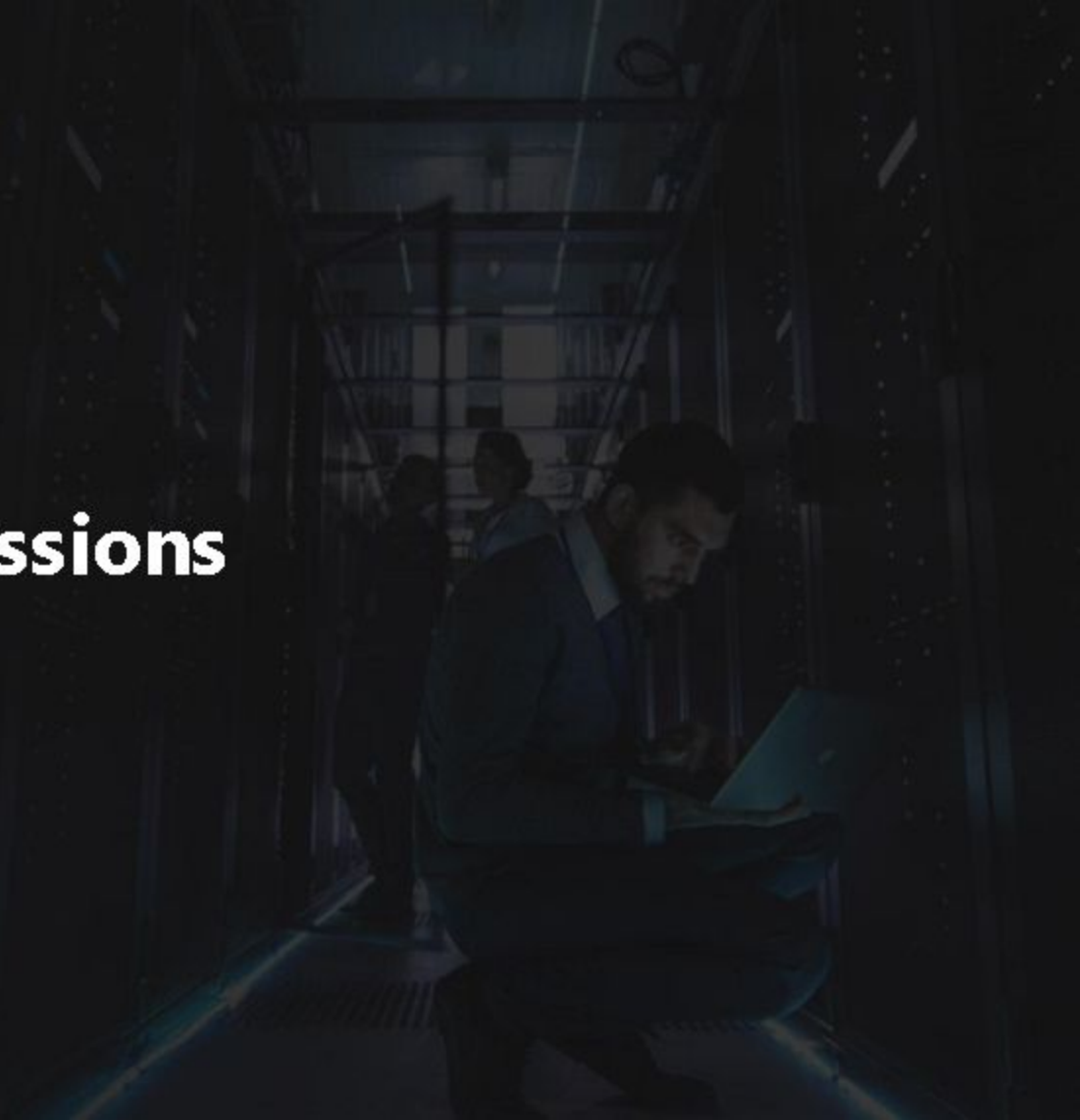




# Sessions



## 3.4 Sessions

---

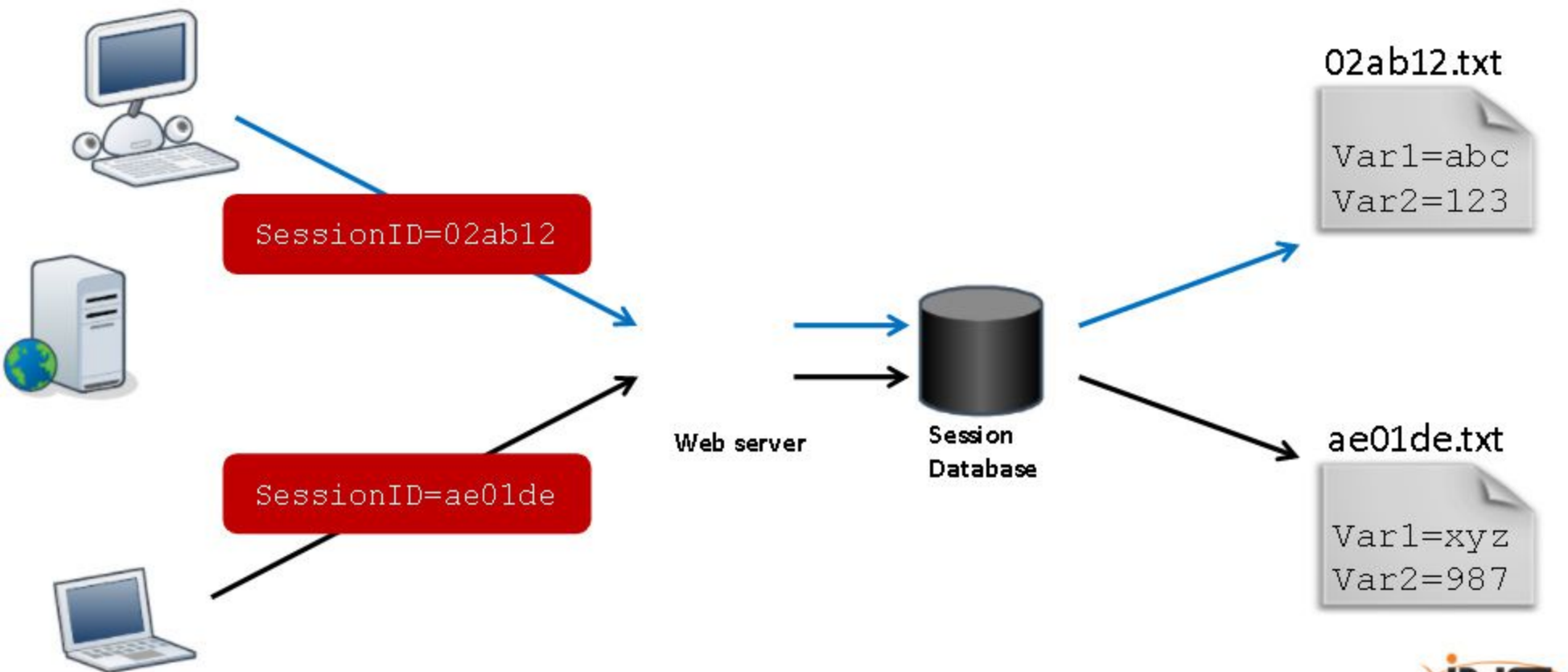
- + Sometimes the web developer prefers to store some information on the **server side** instead of the client side; this happens to hide the application logic or just to avoid the back and forth data transmission typical of cookies.
- + **Sessions** are a mechanism that lets the website store variables specific for a given visit on the **server side**.
- + Each user session is identified by a **session id**, or token, which the server assigns to the client.

## 3.4 Sessions

---

- + The client then presents this ID for each subsequent request, thus being recognized by the server.
- + By means of the session ID, the **server retrieves the state of the client** and all its associated variables. The server stores Session IDs inside text files in its storage. You can find an example in the next slide.

## 3.4.1 Sessions Example



## 3.4.2 Session Cookies

---

Q

*How does a web application install session IDs on a web browser?*

A

*By using session cookies.*

+ It's now time to see how to use **HTTP sessions!**

## 3.4.2 Session Cookies

---

- + Session cookies just contain a single parameter value pair referring to the session.

```
SESSION=0wvCtOBWDH8w  
PHPSESSID=13Kn5Z6Uo4pH  
JSESSIONID=W7DPUBgh7kTM
```

## 3.4.2 Session Cookies

---

- + Websites running PHP install session cookies by using the "PHPSESSID" parameter name, while JSP websites use "JSESSIONID". Each development language has its own default session parameter name.
- + Of course, the web developer can also choose to use a custom parameter name.

## 3.4.2 Session Cookies

---

- + If needed, servers install session cookies after a browser performs some kind of activity, like:
  - Opening a specific page
  - Changing settings in the web application
  - Logging in



## 3.4.2 Session Cookies

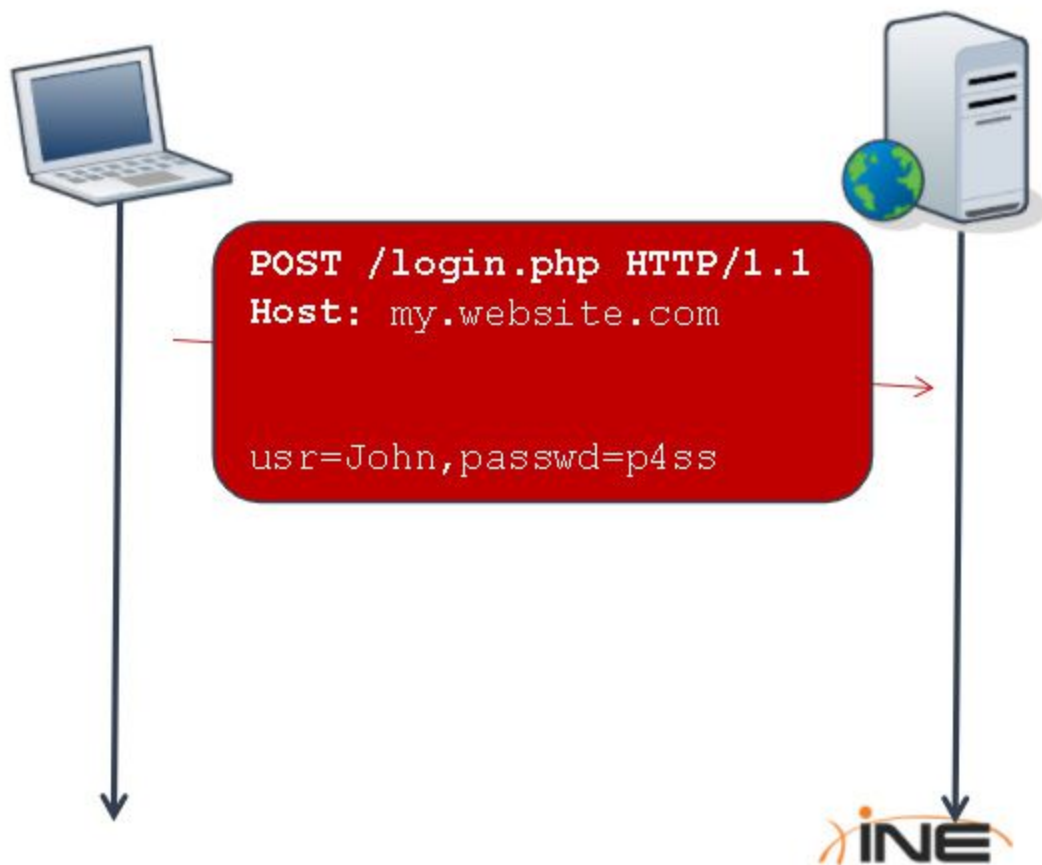
---

- + The browser then uses the cookie in subsequent requests. A session could contain many variables, so sending a small cookie keeps the bandwidth usage low.
- + In the following example, you can see a session cookie in action.

## 3.4.2.1 Session Cookies Example

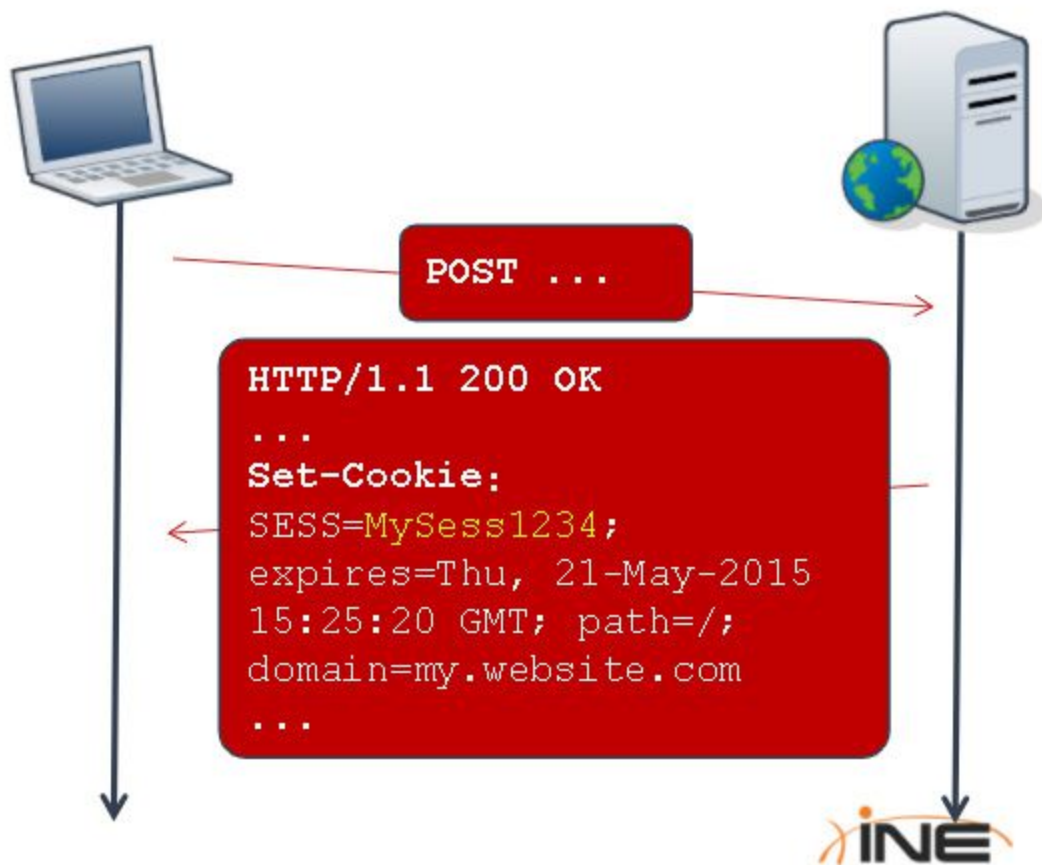
---

- + The client uses a login form to POST the user's credentials.



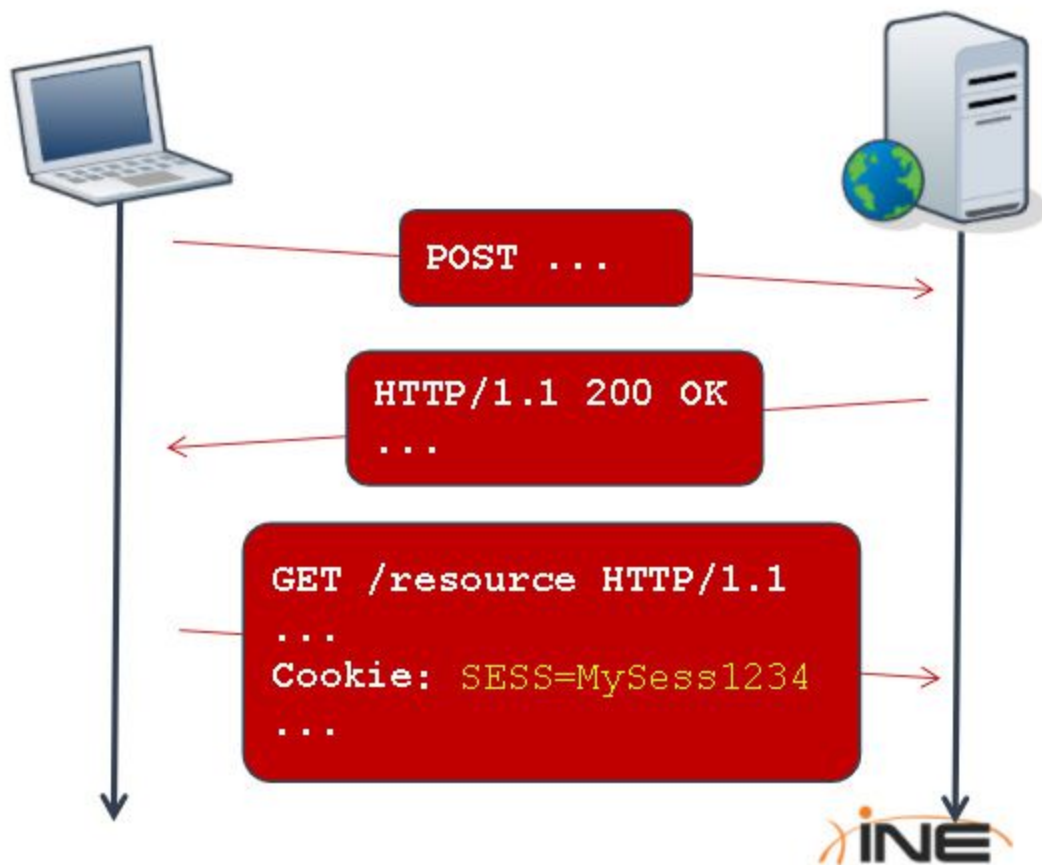
## 3.4.2.1 Session Cookies Example

- + The server sends back a response with a Set-cookie header field.
- + The cookie contains the **session ID**.



## 3.4.2.1 Session Cookies Example

- + The browser will send back the cookie according to the cookie protocol, thus sending the **session ID**.



## 3.4.3 GET Requests

---

### EXAMPLE

- + Session IDs can also be transmitted via **GET requests**.

```
http://example.site/resource.php?sessid=k27rds7h8w
```