

# Web Application Development

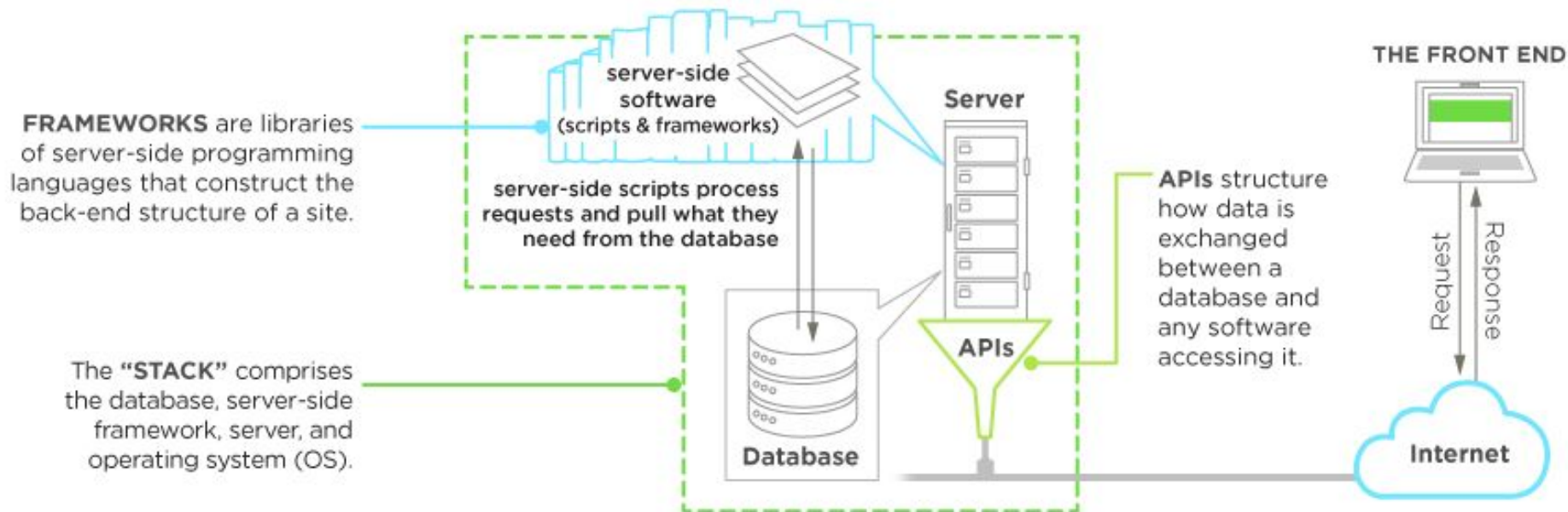
2019



# Backend concepts



# BACK-END DEVELOPMENT & FRAMEWORKS IN SERVER SIDE SOFTWARE



MVC

Model

View

Controller

# Model

- Model represents shape of the data.
- It maintains the data of the application.
- Model objects retrieve and store model state in a database

# View

- View is a user interface.
- View display data using model to the user and also enables them to modify the data.

**NOTE:** Since we are utilizing multi-tiered architecture the “View” abstraction is useless on backend side and UI should be deferred to Front-end application entirely.

# Controller

- Controllers act as an interface between Model and View components to process all the business logic and incoming requests.
- Manipulate data using the Model component.

# Databases



# Relational Databases

- MySQL
- Sybase
- Oracle
- IBM DB2
- MS SQL
- MariaDB
- Postgres
- ...

- Relational databases consist of two or more tables with connected information. each with columns and rows.
- These connected tables are called database objects, and in order to create them and manage them, you need a relational database management system (RDBMS).

# SQL

## Structured Query Language

- Standardized programming language for accessing and manipulating databases.
- SQL writes programming that can manage data and stream data processing:
  - Executing queries, which are “questions” asked of the database
  - Retrieving data
  - Editing data: inserting, updating, deleting
  - Setting permissions
  - Creating new databases

# SQL

Planning is the key

How your database is structured from day one will have major implications for the health of your system down the road.

Questions to consider include:

- What information will you have?  
How should it be stored?
- What data will your system need to retrieve regularly, and how?

# NOSQL

Flexibility is the key

What happens when your data is inconsistent, incomplete, or massive?  
In these cases, you need a more flexible database solution

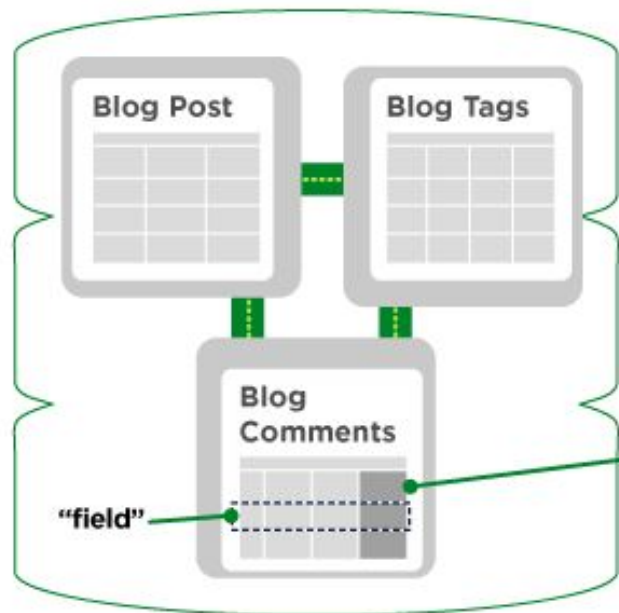
# NOSQL

NoSQL = “Not only SQL.”

Instead of tables, NoSQL databases are document-oriented.

This way, non-structured data (such as articles, photos, social media data, videos, for example, an entire blog post) can be stored in a single document that can be easily found but isn't necessarily categorized into a bunch of pre-set fields.

## RELATIONAL VS. NON-RELATIONAL DATABASES



A non-relational database does not incorporate the table model. Instead, data can be stored in a single document file.

A relational database table organizes structured data fields into defined columns.



# Authentication

# Authentication

- Authentication is a process of ensuring and confirming user's identity
- Authentication process begins when user tries to access a resource



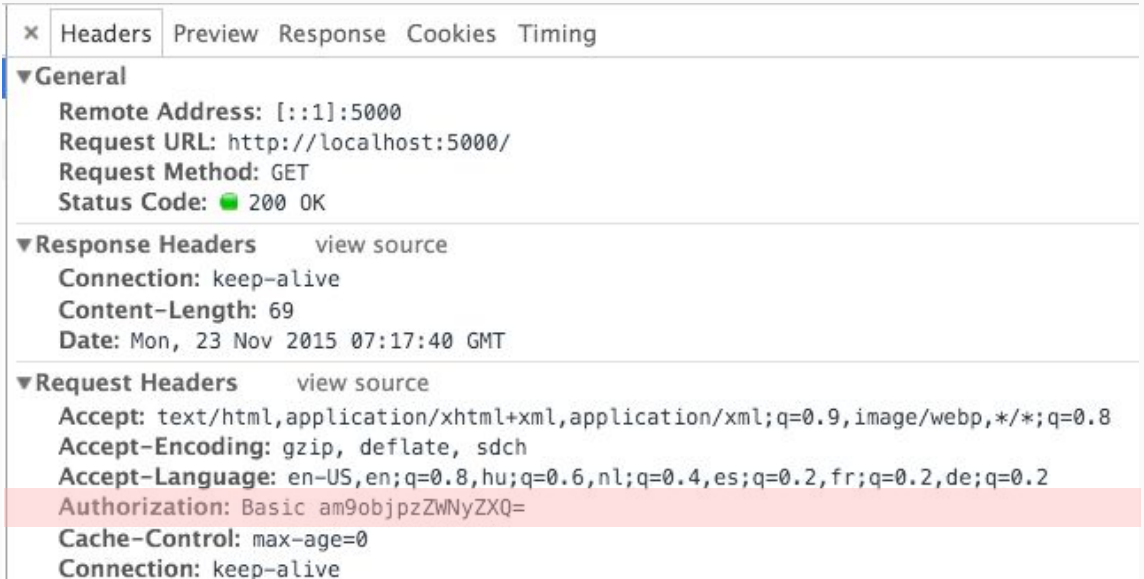
# Authentication

- HTTP Basic Auth
- Session
- JWT

# HTTP Basic Auth

HTTP Basic authentication is a method for the client to provide a username and a password when making a request.

- username and password are concatenated into a single string: `username:password`
- this string is encoded with `Base64`
- the `Basic` keyword is put before this encoded value



× Headers Preview Response Cookies Timing

▼ General

Remote Address: `:::1:5000`  
Request URL: `http://localhost:5000/`  
Request Method: GET  
Status Code: ● 200 OK

▼ Response Headers [view source](#)

Connection: keep-alive  
Content-Length: 69  
Date: Mon, 23 Nov 2015 07:17:40 GMT

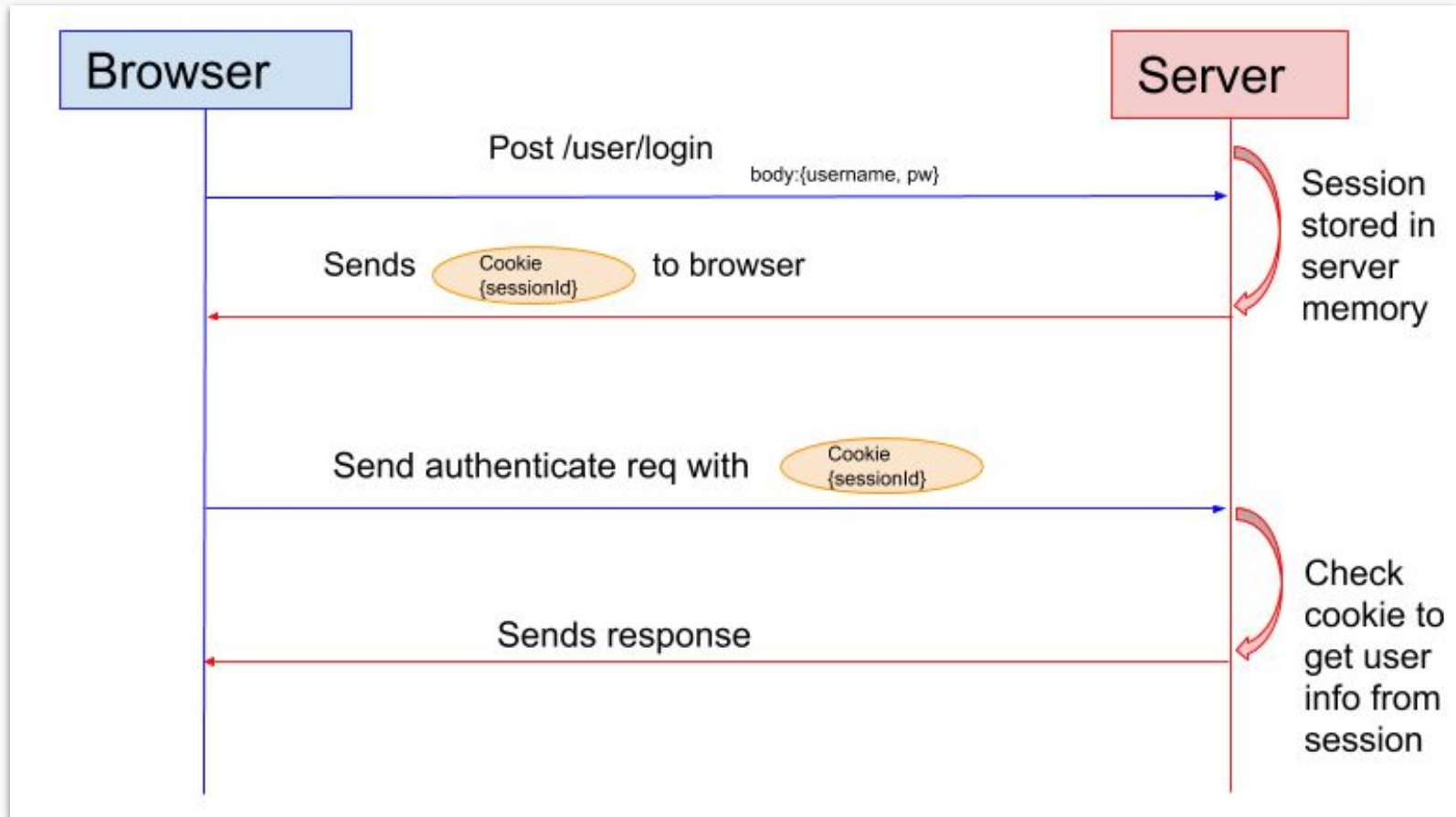
▼ Request Headers [view source](#)

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,\*/\*;q=0.8  
Accept-Encoding: gzip, deflate, sdch  
Accept-Language: en-US,en;q=0.8,hu;q=0.6,nl;q=0.4,es;q=0.2,fr;q=0.2,de;q=0.2  
Authorization: Basic am9objpzZWNYZXQ=  
Cache-Control: max-age=0  
Connection: keep-alive

# Session

- In the session based authentication, the server will create a session for the user after the user logs in.
- The session id is then stored on a cookie on the user's browser.
- While the user stays logged in, the cookie would be sent along with every subsequent request.
- The server can then compare the session id stored on the cookie against the session information stored in the memory to verify user's identity and sends response with the corresponding state!

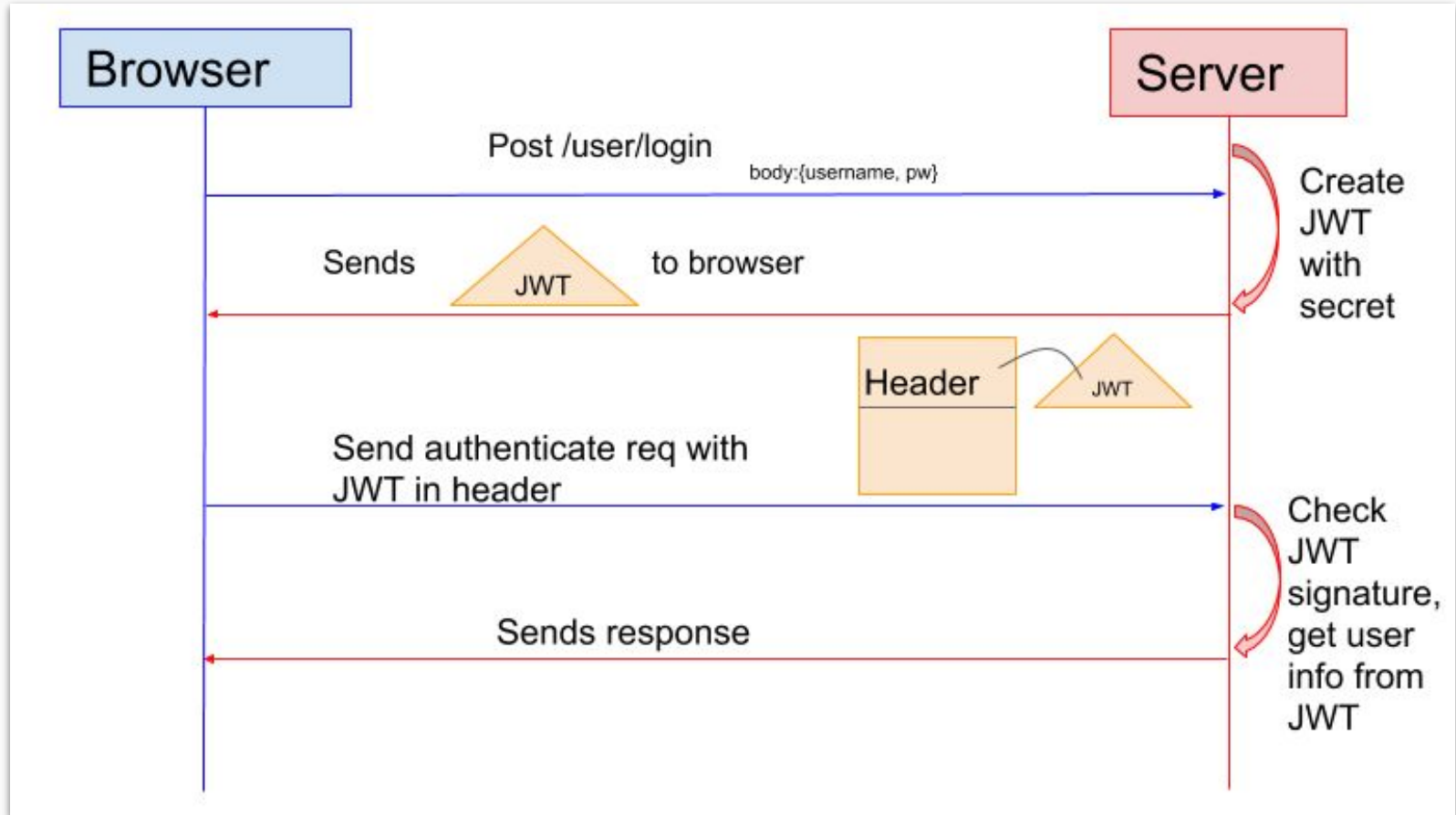
# Session



# JWT

## JSON Web Token

- In the token based application, the server creates JWT with a secret and sends the JWT to the client.
- The client stores the JWT (usually in local storage) and includes JWT in the header with every request.
- The server would then validate the JWT with every request from the client and sends response.



Questions?

