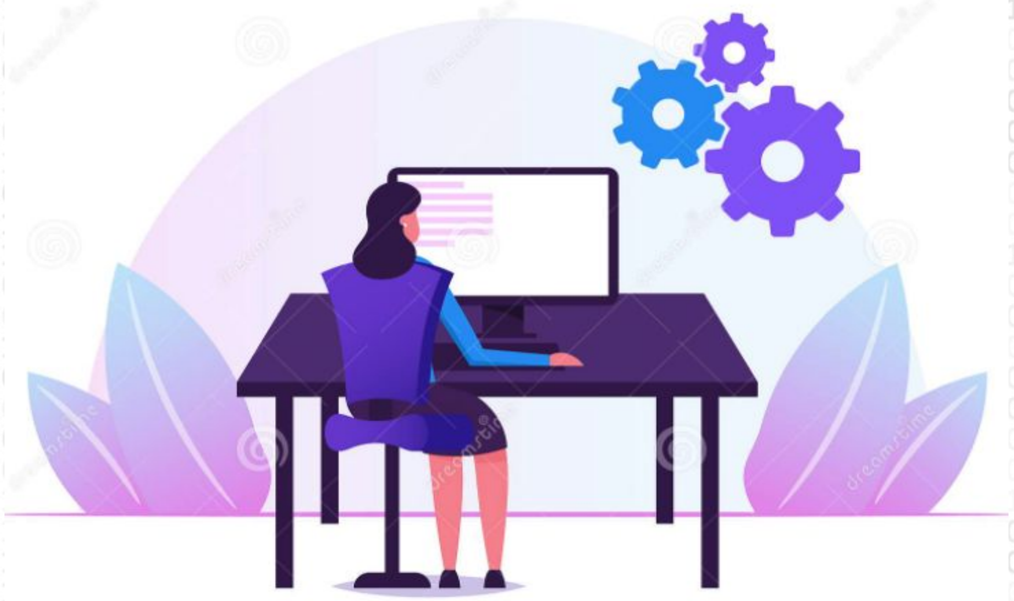
Welcome!



WWCode Digital + **Backend**
Backend Study Group

September 30, 2022

- We'll start in a moment :)
- We are NOT recording tonight's event. We may plan to take screenshots for social media.
- ***If you want to remain anonymous***, change your name & keep video off.
- We'll introduce the hosts & might break in-between for Q/A.
- We will make some time for Q&A at the end of the presentation as well.
- You can come prepared with questions.
- Feel free to take notes.
- Online event best practices:
    - Don't multitask. Distractions reduce your ability to remember concepts.
    - Mute yourself when you aren't talking.
    - We want the session to be interactive.
    - Feel free to unmute & ask questions.
- Turn on your video if you feel comfortable.

- *Disclaimer: Speaker doesn't knows everything!*

WOMEN WHO CODE®

Introduction & Agenda

- Welcome from WWCode!
- Our mission: Inspiring women to excel in technology careers.
- Our vision: A world where women are representative as technical executives, founders,VCs, board members and software engineers.

Prachi Shah
Instructor,
Software Engineer.
Director, WWCode SF

Harini Rajendran
Host,
Software Engineer, Confluent.
Lead, WWCode SF

- API Design:
  - What is an API?
  - What is a RESTful API?
  - SOAP vs. REST.
  - Design Best Practices.
  - Examples and Demo (in Java).
  - Q & A and open discussion.

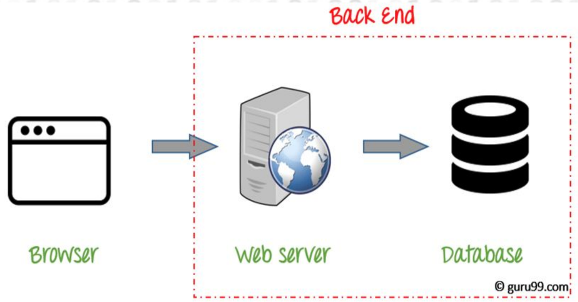
Disclaimer:
- Sessions can be heavy!
- Lots of acronyms.
- Speaker doesn't know everything.

WOMEN WHO CODE

Backend Engineering

- Design, build and maintain server-side web applications.

- Common terms: Client-server architecture, networking, APIs, web frameworks, platform, micro-service, databases, web fundamentals, operating systems, etc.



- Tech Stack: Java, PHP, .NET, C#, Ruby, Python, REST, AWS, Node, SQL, NoSQL.

- Other domains: Front end engineering, full stack engineering, design & user experience, mobile development, devOps engineering, machine learning, etc.

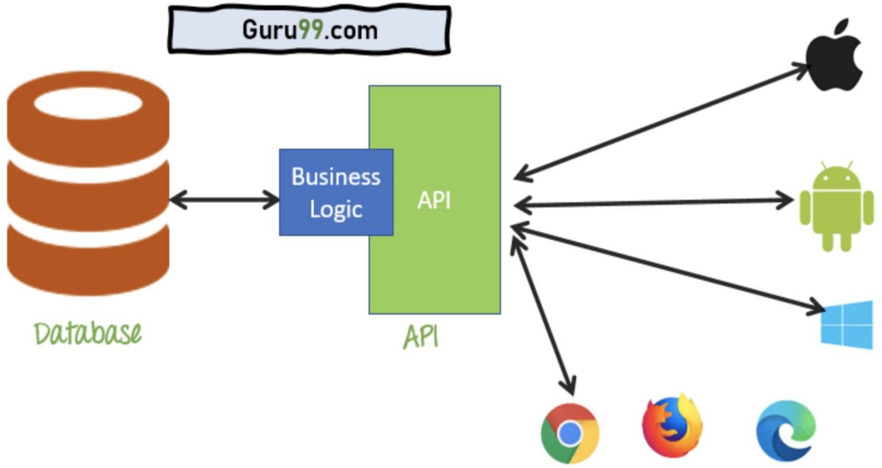
- Examples: Amazon Online Shopping, Instagram, Weather app, etc.

API Design

What is an API?

- Application Programming Interface.
- Contract between frontend and backend.
- Definitions for application communication and integration.

- Why APIs:
  - Backend service can communicate with an Frontend.
  - Frontend and UX are modular, loosely coupled.
  - Easy to customize and add features.
  - Control data flow and security.
  - Support in modern frameworks.
  - Monitor and measure performance.



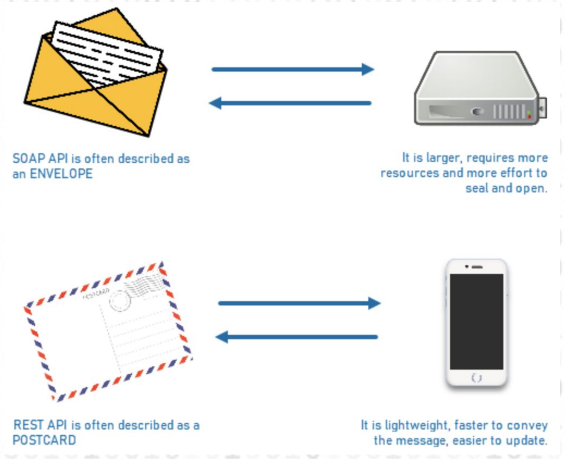
API Design

SOAP:

- Simple Object Access Protocol.
- Protocol specification that uses XML (Extensible Markup Language) for message format.
- Receives requests over HTTP, and does not return human readable response.
- Low performance and difficult to implement.

RESTful:

- REpresentational State Transfer.
- Architectural pattern that uses XML or JSON (JavaScript Object Notation) to send and receive data, and transfers data over HTTP.
- API calls service using URL (Uniform Resource Locator) path.
- Easy to implement, stateless and has better performance.


SOAP API is often described as an ENVELOPE
It is larger, requires more resources and more effort to seal and open.
REST API is often described as a POSTCARD
It is lightweight, faster to convey the message, easier to update.

API Design

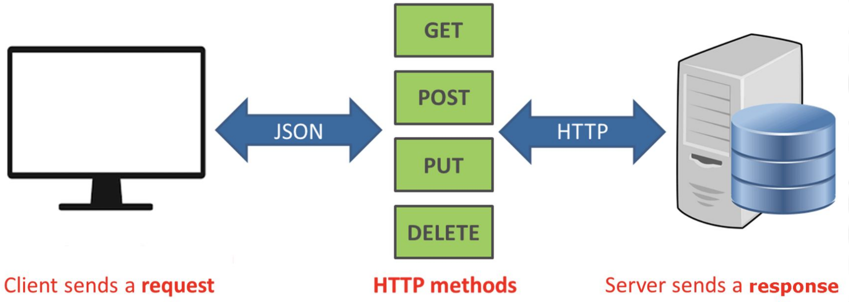
Best Practices:

- URL Path: The path of the API consist of domain name, service path, query, resource name and identifier.

  `Example: GET https://www.amazon.com/amazonshopping/cart/`
- Query parameters: To query for resources.

  `Example: GET https://www.amazon.com/amazonshopping/books?cost=1000`
- Path variables: To get specific resources.

  `Example: GET https://www.instagram.com/profile/1001/description`
- Use HTTP Verbs (CRUD):
  - POST for Create.
  - GET for Read.
  - PUT for Update.
  - DELETE for Delete.
- Public APIs require authentication.
- Idempotent APIs: PUT, GET.
- Limit data exposure.
- Validate data input and response.



WOMEN WHO CODE

API Design

Examples:
For a student management application, we have following functionalities:

- **Add a new student (CREATE)**:
  - API: `POST /studentmanagementservice/student`
  - Request: Student object (name, degree, etc.).
  - Response: `201 CREATED`
- **DELETE a student (by student ID)**:
  - API: `DELETE /studentmanagementservice/student/48`
  - Request: None.
  - Response: `200 OK`
- **UPDATE a student (by student ID)**:
  - API: `PUT /studentmanagementservice/student/50`
  - Request: Student object (name, degree, etc.).
  - Response: `200 OK`
- **Get a student (by student ID) (READ)**:
  - API: `GET /studentmanagementservice/student/52`
  - Request: None.
  - Response: `200 OK`

WOMEN WHO CODE®

API Design

Demo: RESTful APIs

Backend Engineering

References:

- What is an API?
- Securing APIs

Backend Study Group:

- Presentations on GitHub and session recordings are found on WWCode YouTube channel
- Upcoming session:
    - October 27th, 2022 about Database Design
    - November 3rd, 2022 about Improve your code debugging skills
    - December 8th, 2022 about Git and Version Control System

- Technical Tracks and Digital Events for more events.
- Join the Digital mailing list for updates about WWCode.
- Have questions?
    - Contacts us at: contact@womenwhocode.com
    - Join our Slack workspace and join *#backend-study-group*!

You can unmute and talk or use the chat.

WOMEN WHO CODE