How Do You Place an Order in a Fine-Dine Restaurant?

- Do you enter the restaurant’s kitchen and place your order?
- The chef might be Spanish, and you know only English, so how will you communicate?
- Entering the kitchen and directly interacting with the chef is a messy affair.
- Would this allow the kitchen staff to operate efficiently?
- Is it not desirable to have a waiter to collect your order, forward it to kitchen staff, and bring the food to your table when it's ready?

How Do You Get a Blood Test Done?

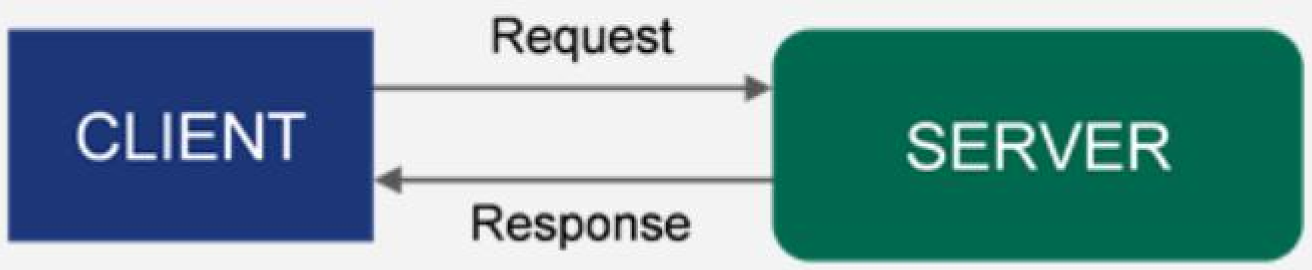
- Do you enter the lab and directly approach the technician performing a blood draw?
- You will need to know and understand how the lab operates if you want to go to the lab to get the blood draws?
- Is this possible for a person who is not medically trained?
- Or is there a lab assistant who collects the blood sample, gives it to the lab and gives you the test reports?

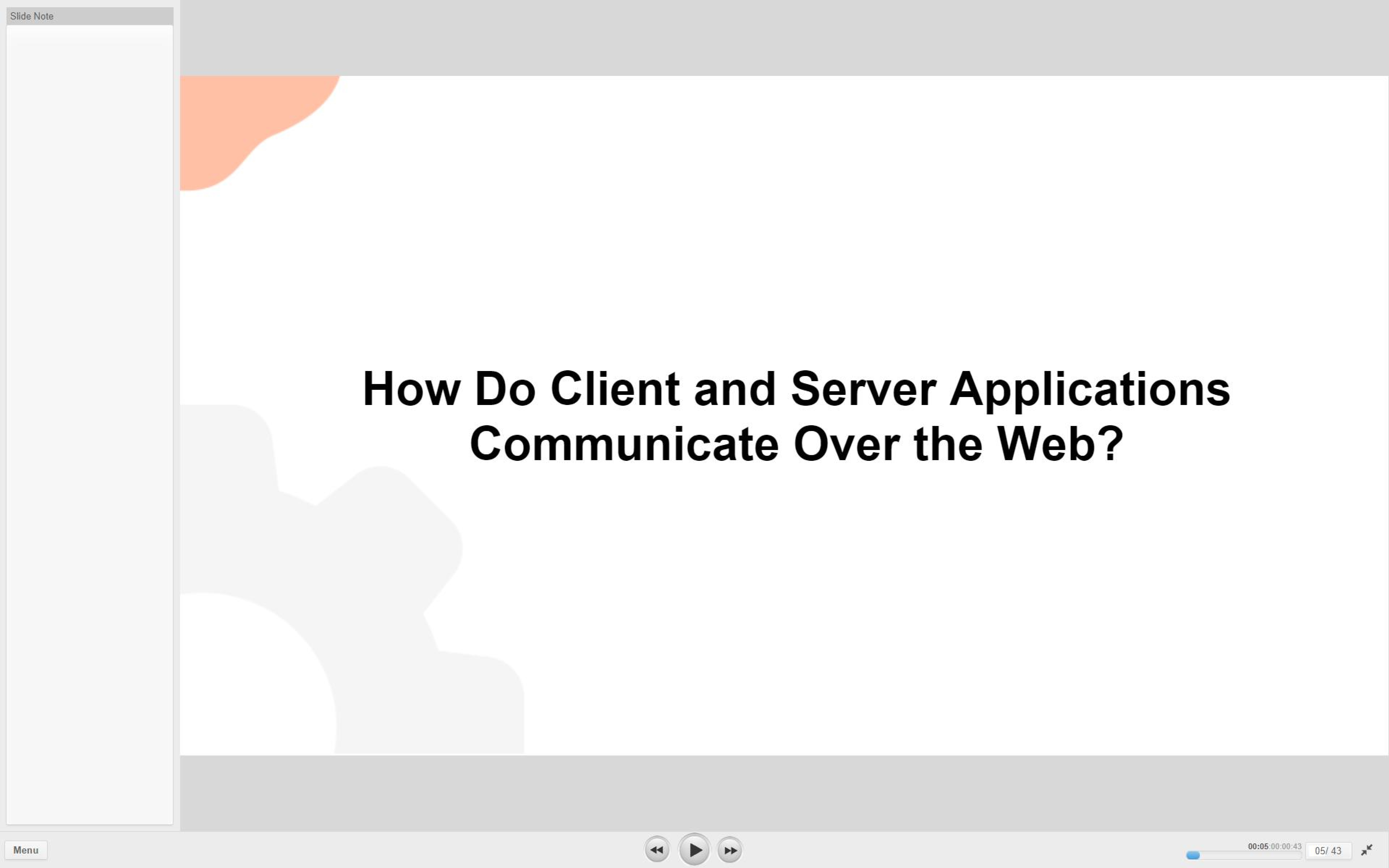
Roles of Waiter and Lab Assistant

- The waiter at a restaurant is an intermediary between a restaurant's kitchen and its customers.
- A lab assistant at a medical lab is an intermediary between a lab and its patients.
- Such interface helps establish communication between two parties irrespective of what languages they speak.
- For communication to flow, both the parties must agree upon a set of rules and procedures.

Web Application Architecture

- Web applications are designed using client-server architecture.
- The application at the user end is the client application.
- The application running remotely is the server application.
- In case of interactive web pages:
  - Client makes request to server for data.
  - Server responds to client with requested data or error in case of failures.
- JavaScript is a popular scripting language for client-side application.
- However, the server-side application might not be developed using JavaScript.



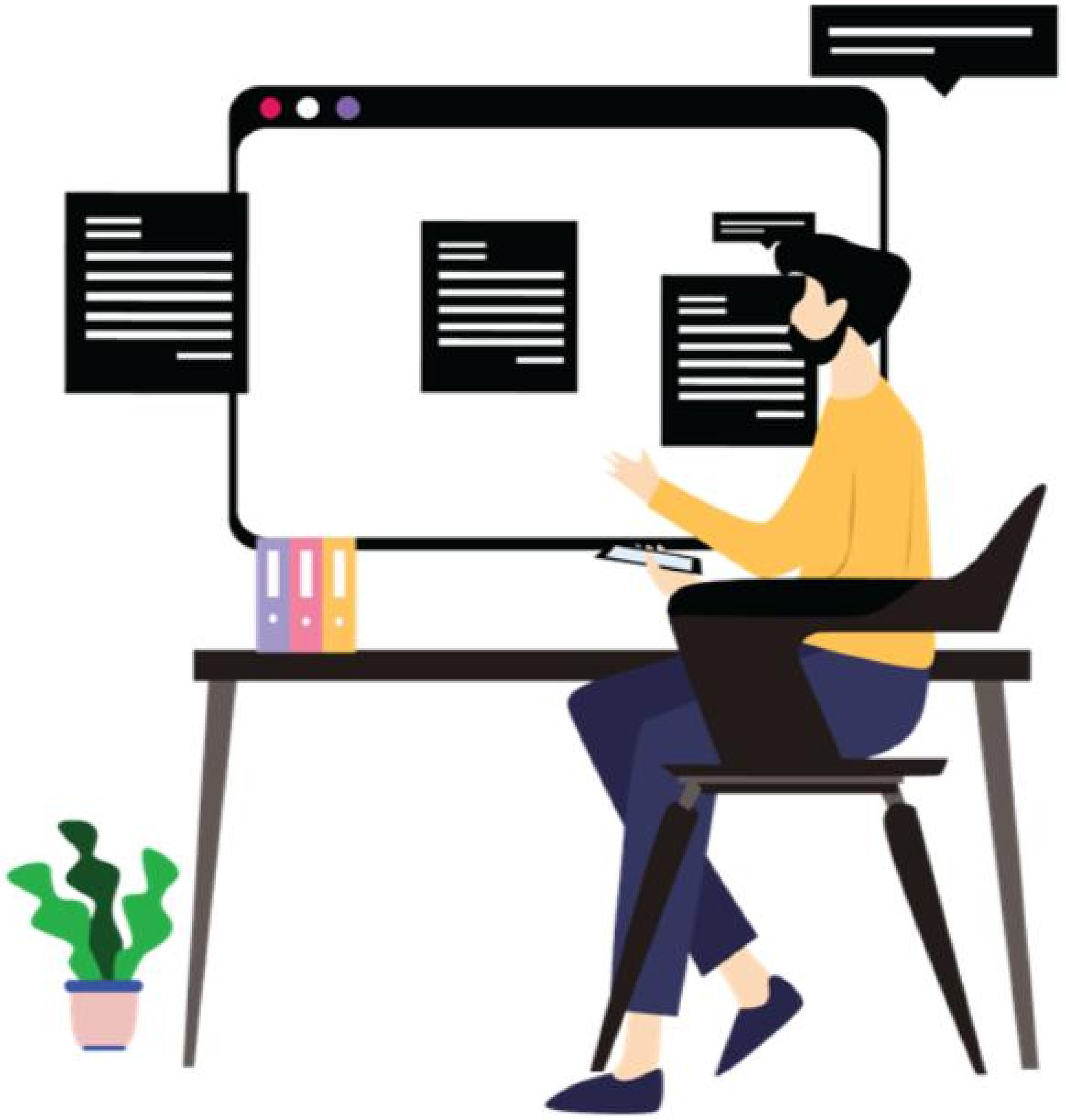


How Do Client and Server Applications Communicate Over the Web?

Think and Tell

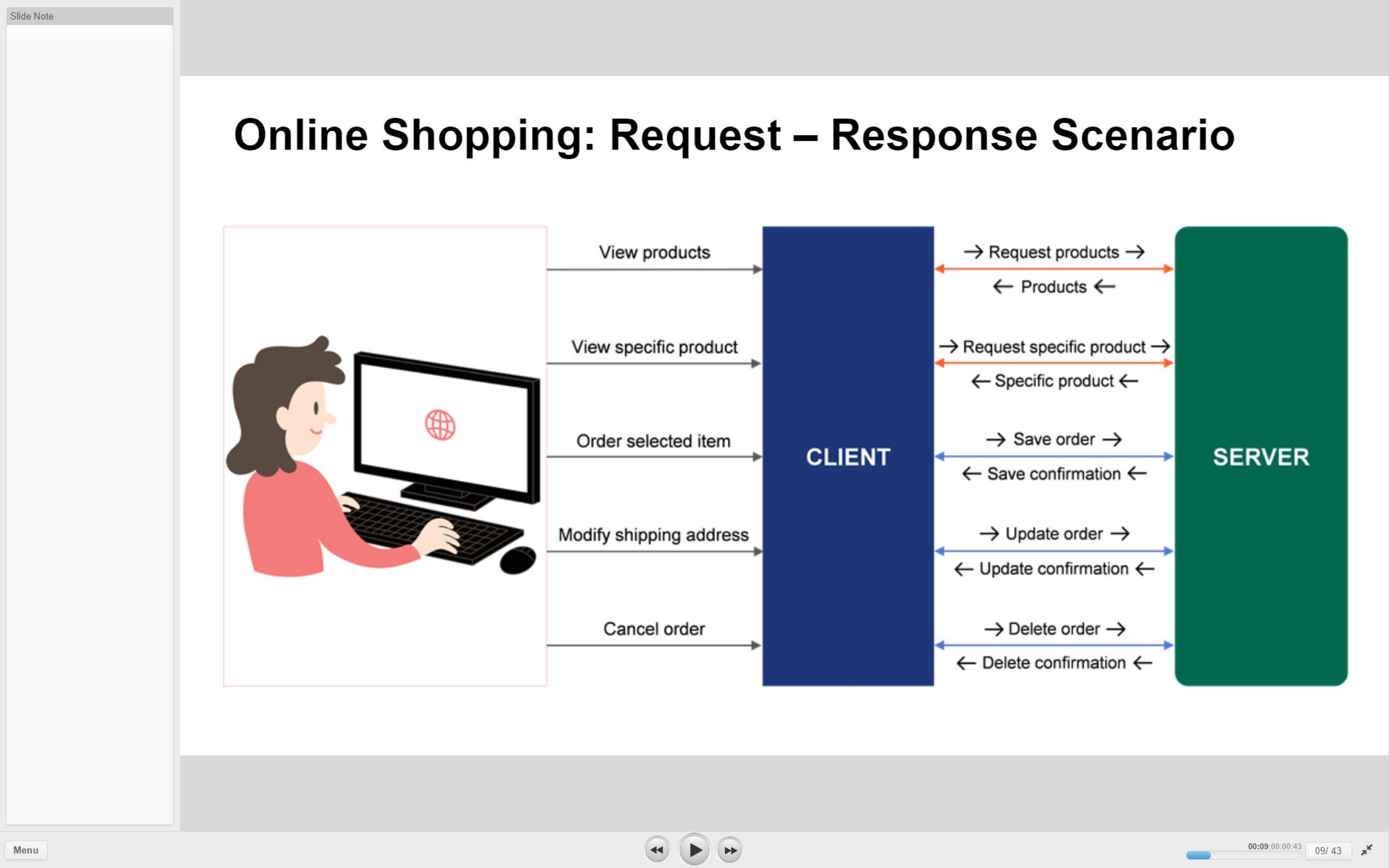
- To handle requests coming from a large number of clients, the number of servers need to be increased.
- Clients are not expected to know which server is serving the request.
- Servers also are not expected to know from which client the request is received.
- In such a complex environment, how efficiently can a client's request be processed?
- Is there a way to design a program that establishes client-server communication without increasing the overheads on the server, and which is also language-agnostic?

Implement Stateless Communication Using REST APIs and JSON



Learning Objectives

- Explain the role of API in client-server communication
- Explore the rules of client-server communication
- Explain REST
- Make HTTP Request
- List HTTP Methods
- Handle HTTP Response
- Explore HTTP Status Codes
- Create fake REST APIs using `json-server`
- Test REST APIs using Postman



Online Shopping: Request – Response Scenario

View products

View specific product

Order selected item

Modify shipping address

Cancel order

CLIENT

→ Request products →

← Products ←

→ Request specific product →

← Specific product ←

→ Save order →

← Save confirmation ←

→ Update order →

← Update confirmation ←

→ Delete order →

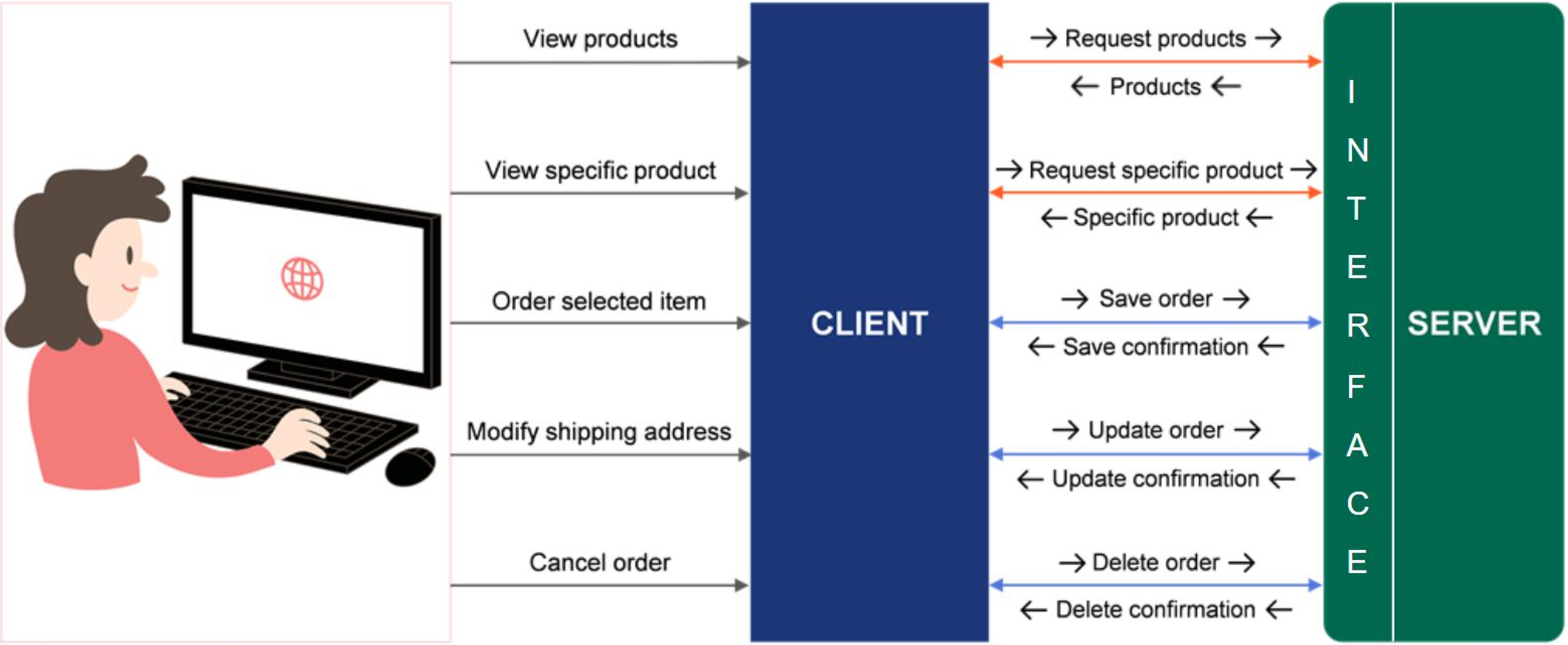
← Delete confirmation ←

SERVER

Think and Tell

- In an online shopping application, the product data is managed by the server.
- Does the client make calls to functions of the server application to fetch, store, update or delete data?
- Remember, client and server applications could be developed using two different programming languages.
- So how can the client application communicate with the server application?
- Is there an interface that establishes a communication link between client and server?

Interface: Communication Link Between Client and Server



View products

View specific product

Order selected item

Modify shipping address

Cancel order

CLIENT

→ Request products →

← Products ←

→ Request specific product →

← Specific product ←

→ Save order →

← Save confirmation ←

→ Update order →

← Update confirmation ←

→ Delete order →

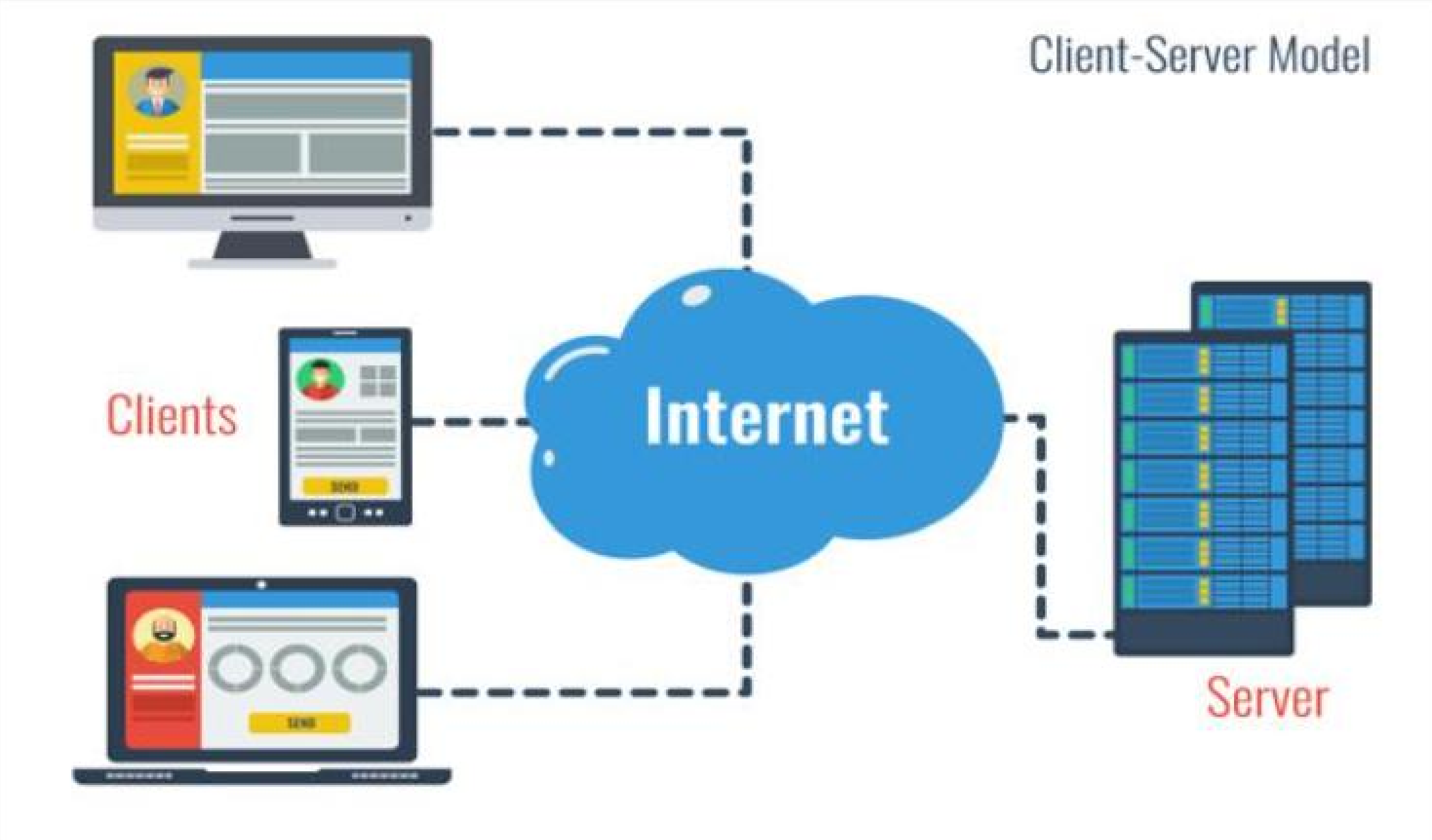
← Delete confirmation ←

INTERFACE

SERVER

Need for Interface

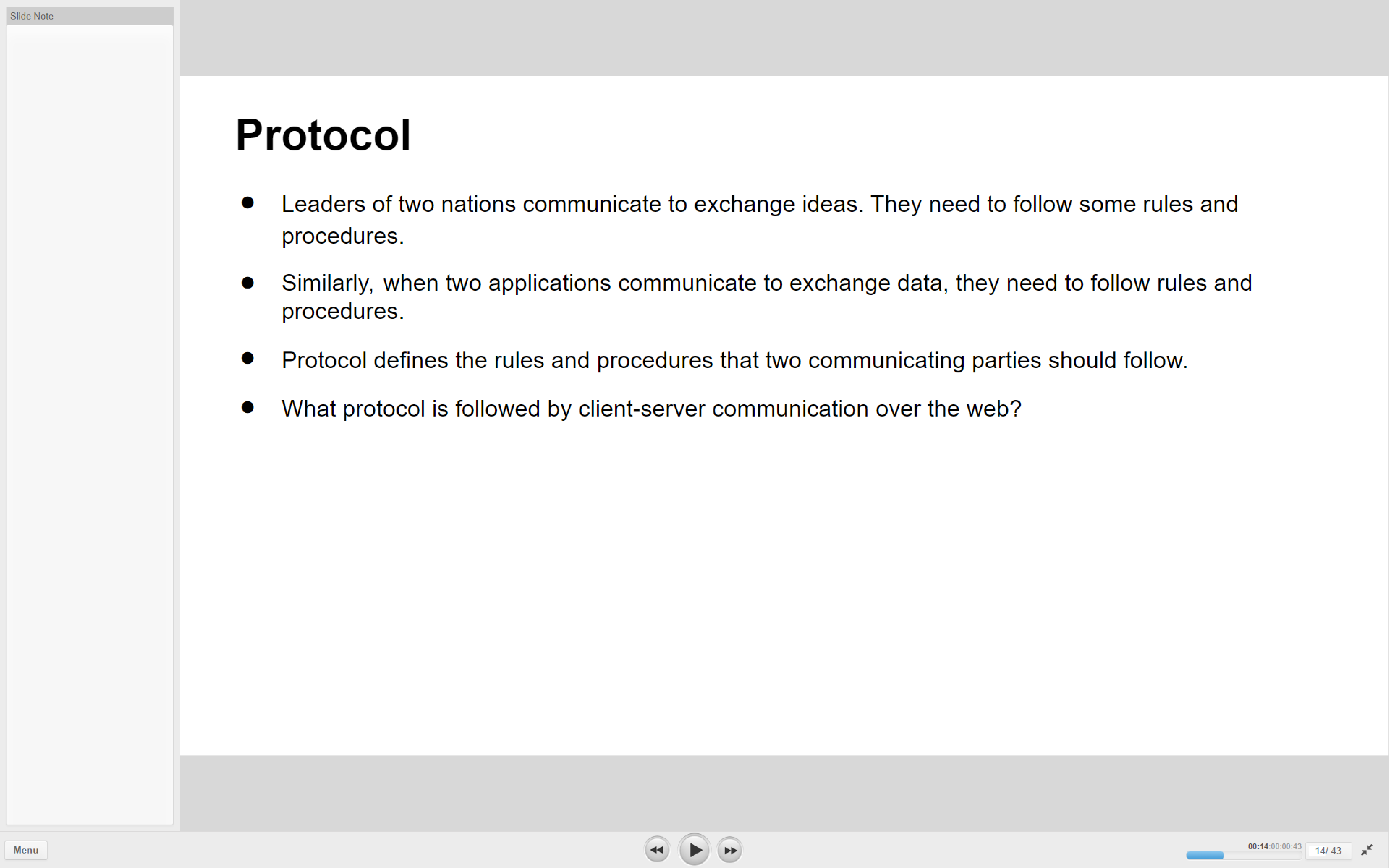
- In a restaurant, the chef is the key person who determines the menu and prepares the orders received.
- Customers order food through the waiter.
- The waiter is the link between the restaurant chef and customers.
- Similarly, in a web application, the server application determines the services it provides.
- The interface is a software program that helps establish a link between client and server.
- The client application, through the interface, requests the server to perform various operations.
- Just as a customer is unaware of the kitchen operations, the client application is unaware of the server operations.



Slide Note

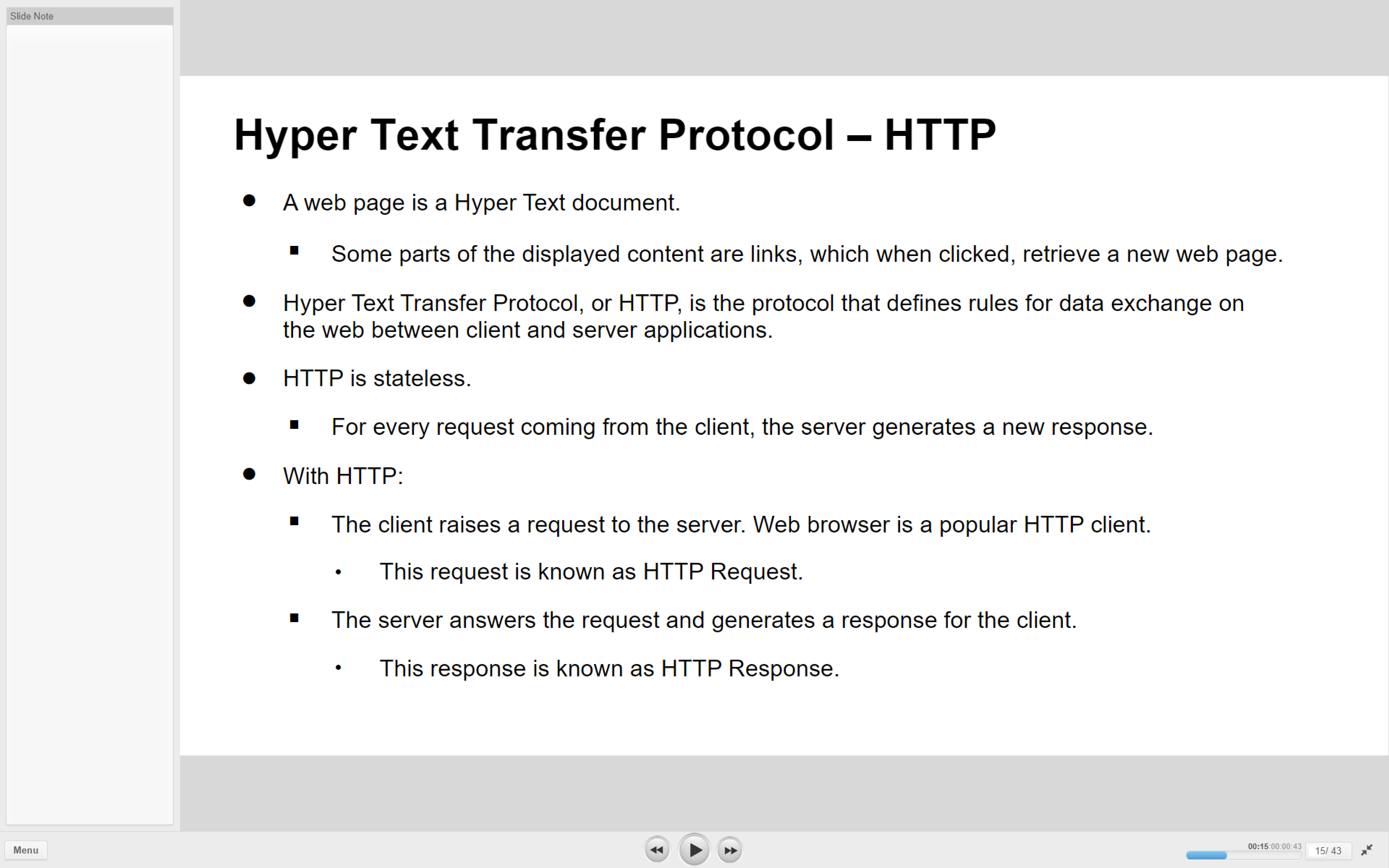
API – Application Programming Interface

- Like a waiter in a restaurant, an interface is required to establish communication between client and server applications.
- This interface is known as **Application Programming Interface – API**.
- API is a software piece residing at the server end that allows a client application to speak to a server application.
- Since this API is consumed (used) by the client over the web, it is also known as Web API.
- The client requests the server through API.
- The server operates based on the incoming request and generates the response.
- The response is sent to the client through API.



Protocol

- Leaders of two nations communicate to exchange ideas. They need to follow some rules and procedures.
- Similarly, when two applications communicate to exchange data, they need to follow rules and procedures.
- Protocol defines the rules and procedures that two communicating parties should follow.
- What protocol is followed by client-server communication over the web?

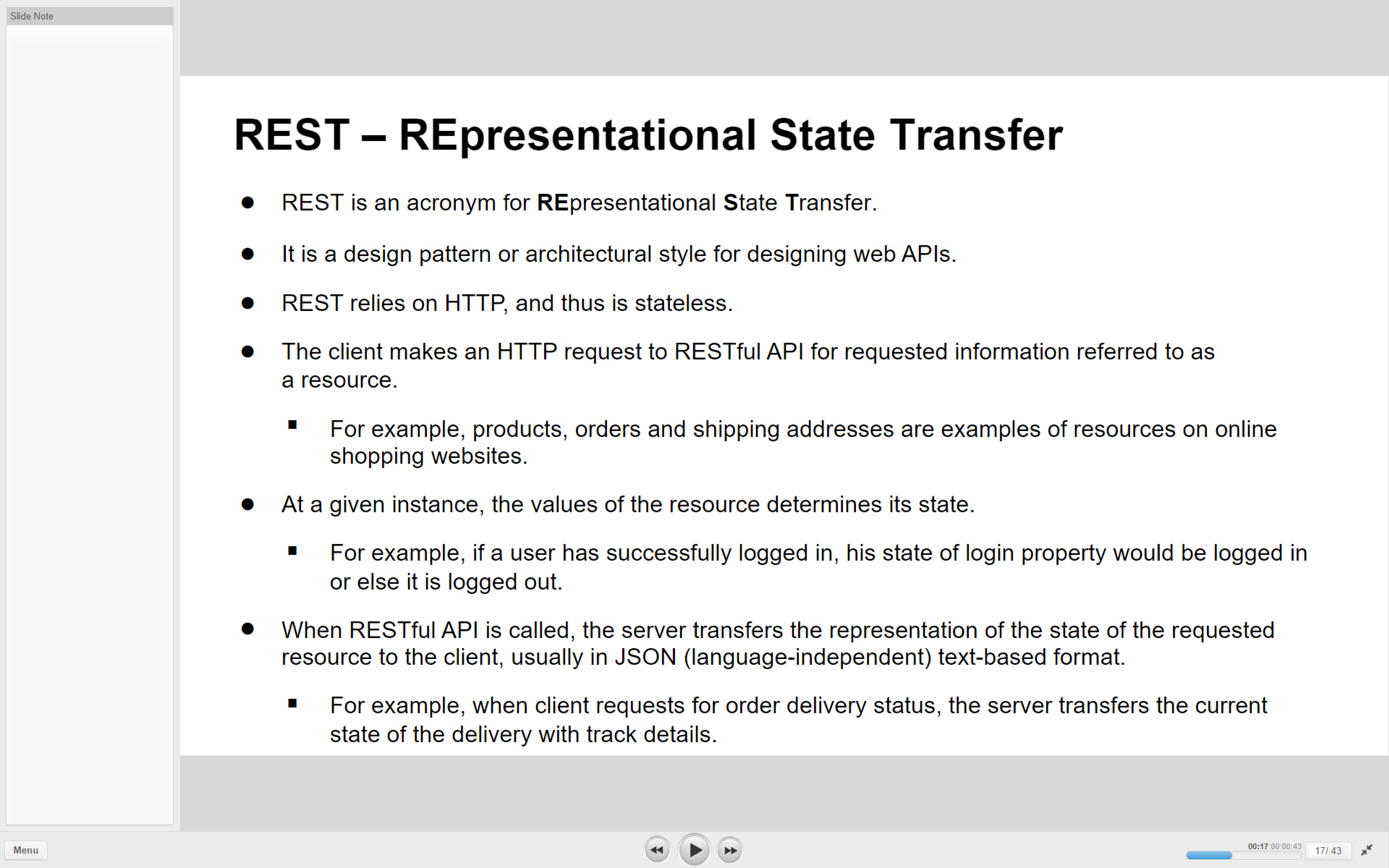


Hyper Text Transfer Protocol – HTTP

- A web page is a Hyper Text document.
  - Some parts of the displayed content are links, which when clicked, retrieve a new web page.
- Hyper Text Transfer Protocol, or HTTP, is the protocol that defines rules for data exchange on the web between client and server applications.
- HTTP is stateless.
  - For every request coming from the client, the server generates a new response.
- With HTTP:
  - The client raises a request to the server. Web browser is a popular HTTP client.
    - This request is known as HTTP Request.
  - The server answers the request and generates a response for the client.
    - This response is known as HTTP Response.

Design Criteria for Web API

- The API designed for client-server applications should:
  - Follow HTTP protocol since it is stateless
  - Accept the request from a client for required information
  - Enable the server to send HTTP responses in a format that is language- independent
- Is there a design pattern that will help design API while meeting these requirements?



REST – REpresentational State Transfer

- REST is an acronym for **RE**presentational **S**tate **T**ransfer.
- It is a design pattern or architectural style for designing web APIs.
- REST relies on HTTP, and thus is stateless.
- The client makes an HTTP request to RESTful API for requested information referred to as a resource.
  - For example, products, orders and shipping addresses are examples of resources on online shopping websites.
- At a given instance, the values of the resource determines its state.
  - For example, if a user has successfully logged in, his state of login property would be logged in or else it is logged out.
- When RESTful API is called, the server transfers the representation of the state of the requested resource to the client, usually in JSON (language-independent) text-based format.
  - For example, when client requests for order delivery status, the server transfers the current state of the delivery with track details.

Request to REST API – HTTP Request

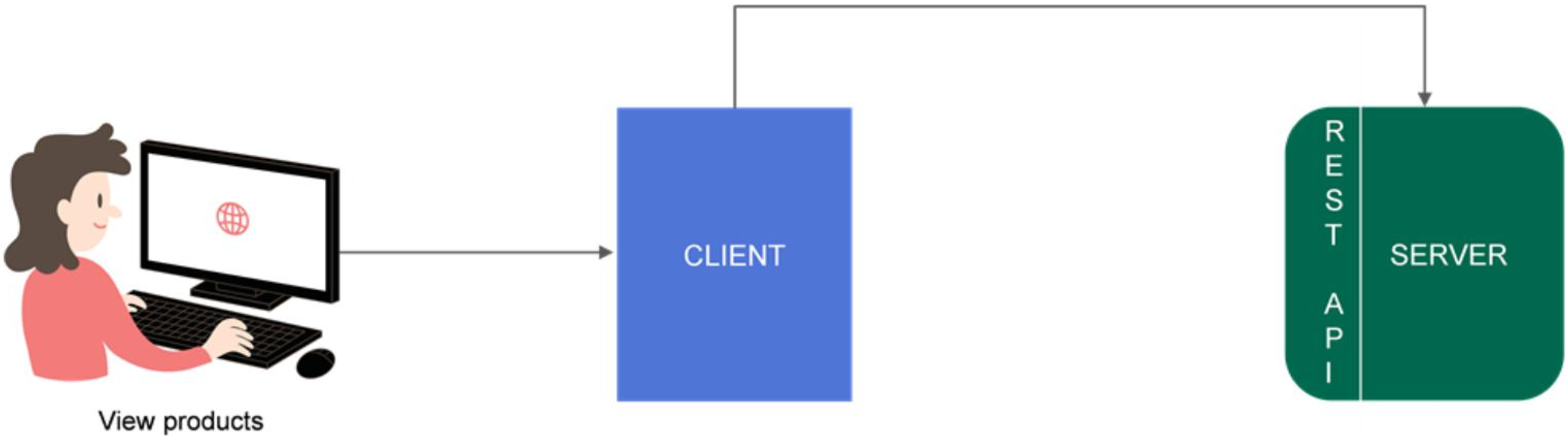
- The HTTP request is made by a client to a server through REST API with the aim of accessing resource on the server.
- The components of an HTTP Request are:
  - **Endpoint**: It’s the URL of the REST server that enables access to the resource on the server.
  - **Method**: This is the type of operation the client wants a server to perform on the resource:
    - GET: Get the resource from the server.
    - POST: Create the resource at the server.
    - PUT: Update the existing resource on the server.
    - DELETE: Delete the existing resource at the server.
  - **Headers**: Additional information provided for communication by the client to the server.
  - **Data**: Information that needs to be sent to the server.

Slide Note

The URL mentioned is a dummy URL, mentioned here to explain the GET Request

Retrieve All Products: Request

To request the resource **Products** from the server, the HTTP Request with **GET** method should be made to http://www.eshop.com

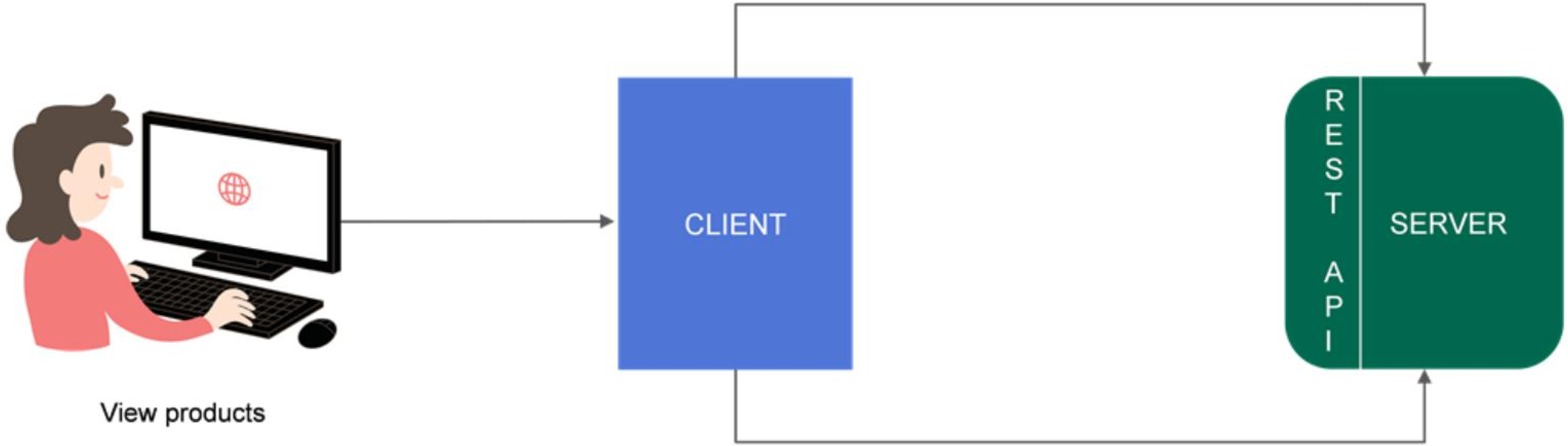


Response From REST API – HTTP Response

- The response from REST API is an HTTP response.
- The HTTP response is an object that contains the response that the server wants to send to the client.
- The response object contains:
  - Status: An HTTP Status code that indicates status of the request.
  - Status Message: A short description of the status code.
  - Headers: An additional information for the response sent by server to client.
  - Body: Optional, and if present, it contains the retrieved response in JSON format.

Retrieve All Products : Request + Response

To request the resource **Products** from the server, the HTTP Request with **GET** method should be made to http://eshop.com/products



The HTTP Response received from the server will contain Products data in the response body with Status Code 200 and the Status Message OK

HTTP Status Code

- HTTP response status codes indicate the status of completion of an HTTP request.
- Responses are grouped in five classes:
  - 100 – 199: Informational responses
    - For example, the status code 102 indicates the server has received and is processing the request. The response is yet not generated.
  - 200 – 299: Successful responses
    - For example, the status code 200 for a Get request indicates the request was successful in retrieving the requested data.

HTTP Status Code (Cont’d)

- 300 – 399: Redirection responses
  - For example, the status code 301 indicates the request has been changed permanently and a new URL is given in the response.
- 400 – 499: Client errors
  - For example, the status code 404 indicates the server could not find the requested resource.
- 500 – 599: Server errors
  - For example, the status code 500 indicates there is an internal server error that the server does not know how to handle. For instance, the resource with a duplicate ID is sent to the server, and the server is not allowed to store a resource with a duplicate ID.

Self-Check

Which of the following HTTP status codes mean NOT FOUND?

1. 400
2. 401
3. 403
4. 404

Self-Check: Solution

Which of the following HTTP status codes mean NOT FOUND?

1. 400
2. 401
3. 403
4. **404**

Think and Tell

- Let’s say JSON data providing details of Hollywood movies is released to date.
- An interactive web application needs to be designed to display and add more movies as per the user’s request.
- The backend server with REST APIs is not yet ready.
- Is there a way to simulate the server and get the APIs created?
    - This will help achieve the main objective which is to write client code.
- Is it possible to create fake REST APIs with zero coding that allows the client application to access movie data?
- Also, is it possible to test these REST APIs before consuming them in the application?

`json-server` – A Development Server

- A development server is a server that runs locally.
- It is designed for local development and testing.
- `json-server` is one such development server.
- It is developed using JavaScript and is packed with functionalities that help in creating a fake REST API.
- To install and run it on the local machine, a JavaScript engine is required.
- Node.js is a JavaScript engine.
- Installing Node.js also installs the tool node package manager – npm.
- This tool helps install packages like `json-server`.

Install `json-server`

Install Node.js to get the npm tool.
Use the npm tool to install `json-server`.

Check the installation using the command `` `json-server –version` ``.

Start `json-server` on default port with existing db.json file.
Start `json-server` on default port with nonexisting db.json file.
Start `json-server` on a different port with existing db.json file.

Click here for the demo solution.

DEMO

Testing APIs Using Postman

- Postman is an API platform for building and using APIs.
- Requests can be sent to API in Postman.
- Postman displays the response received for the request sent from the API, allowing us to examine, visualize, and if necessary, troubleshoot the request.

Slide Note

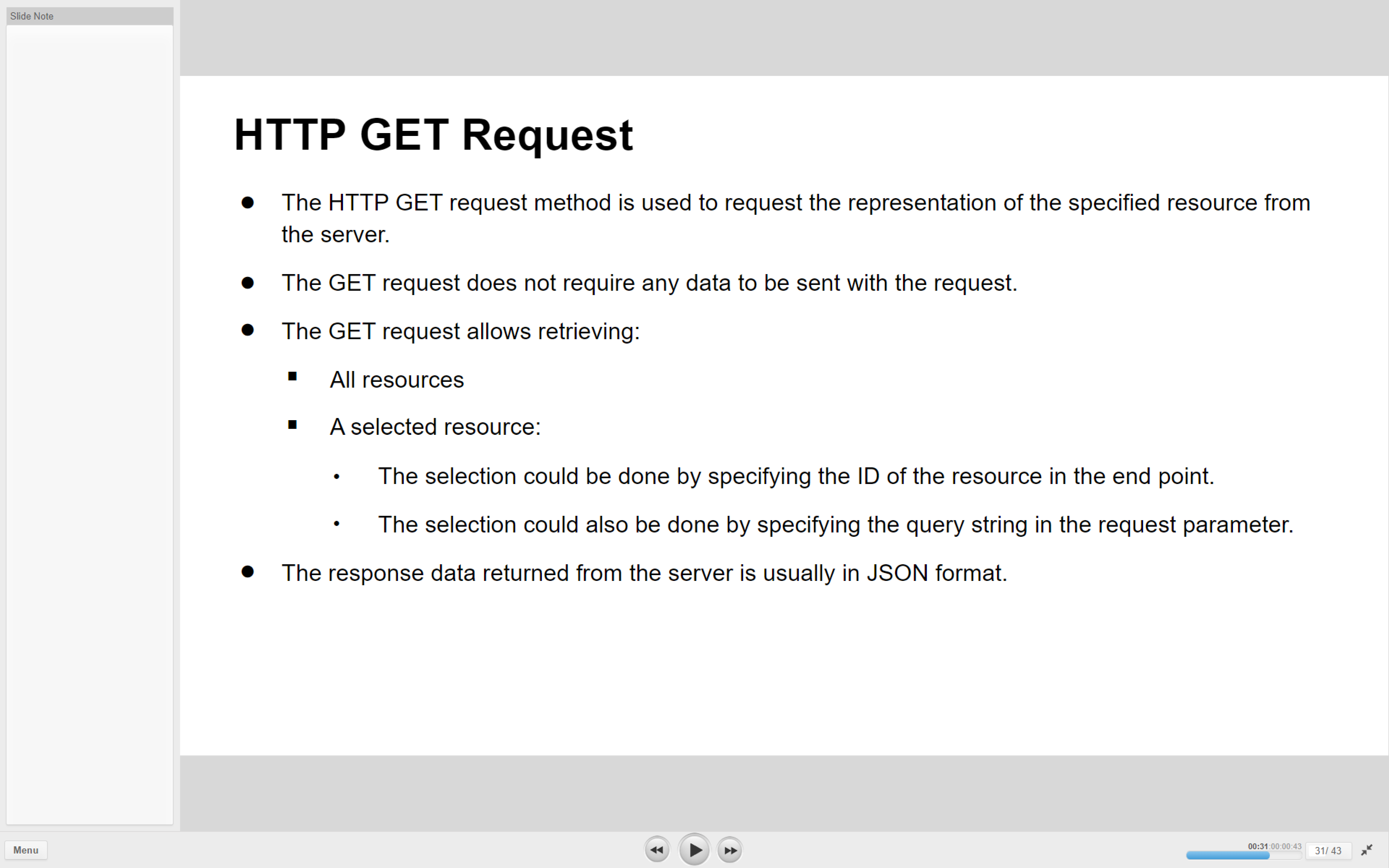
Install Postman

Download the Postman app from this link and install the desktop version of the Postman tool.

Click here for the demo solution.

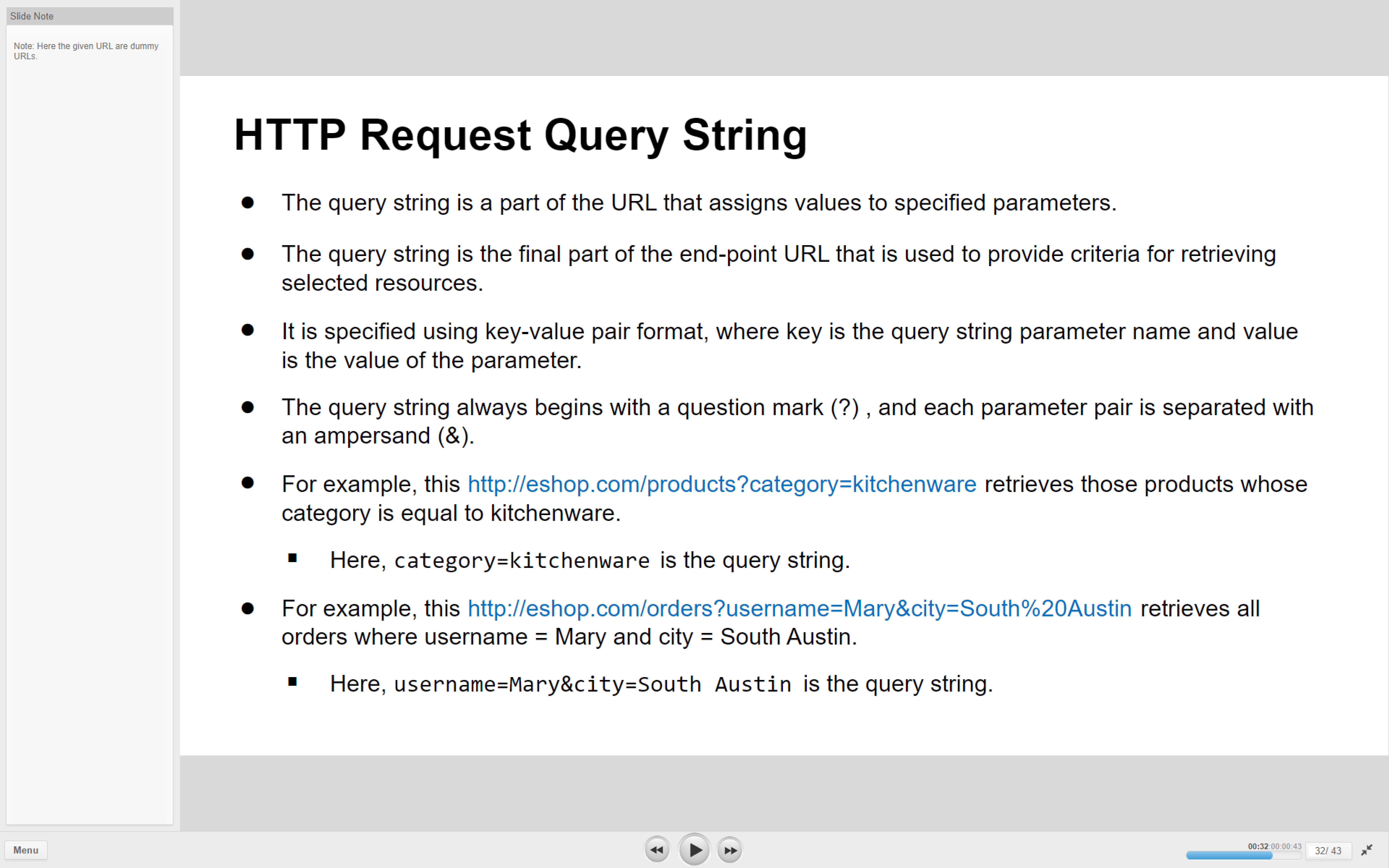
DEMO

Menu



HTTP GET Request

- The HTTP GET request method is used to request the representation of the specified resource from the server.
- The GET request does not require any data to be sent with the request.
- The GET request allows retrieving:
  - All resources
  - A selected resource:
    - The selection could be done by specifying the ID of the resource in the end point.
    - The selection could also be done by specifying the query string in the request parameter.
- The response data returned from the server is usually in JSON format.



Note: Here the given URL are dummy URLs.

HTTP Request Query String

- The query string is a part of the URL that assigns values to specified parameters.
- The query string is the final part of the end-point URL that is used to provide criteria for retrieving selected resources.
- It is specified using key-value pair format, where key is the query string parameter name and value is the value of the parameter.
- The query string always begins with a question mark (?) , and each parameter pair is separated with an ampersand (&).
- For example, this http://eshop.com/products?category=kitchenware retrieves those products whose category is equal to kitchenware.
  - Here, `category=kitchenware` is the query string.
- For example, this http://eshop.com/orders?username=Mary&city=South%20Austin retrieves all orders where username = Mary and city = South Austin.
  - Here, `username=Mary&city=South Austin` is the query string.

Fetch Blog Posts

Using Postman app:

1. Fetch all posts
2. Fetch post with ID 1
3. Fetch post with title "json-server"

Check the status code for success for each of the fetch requests.

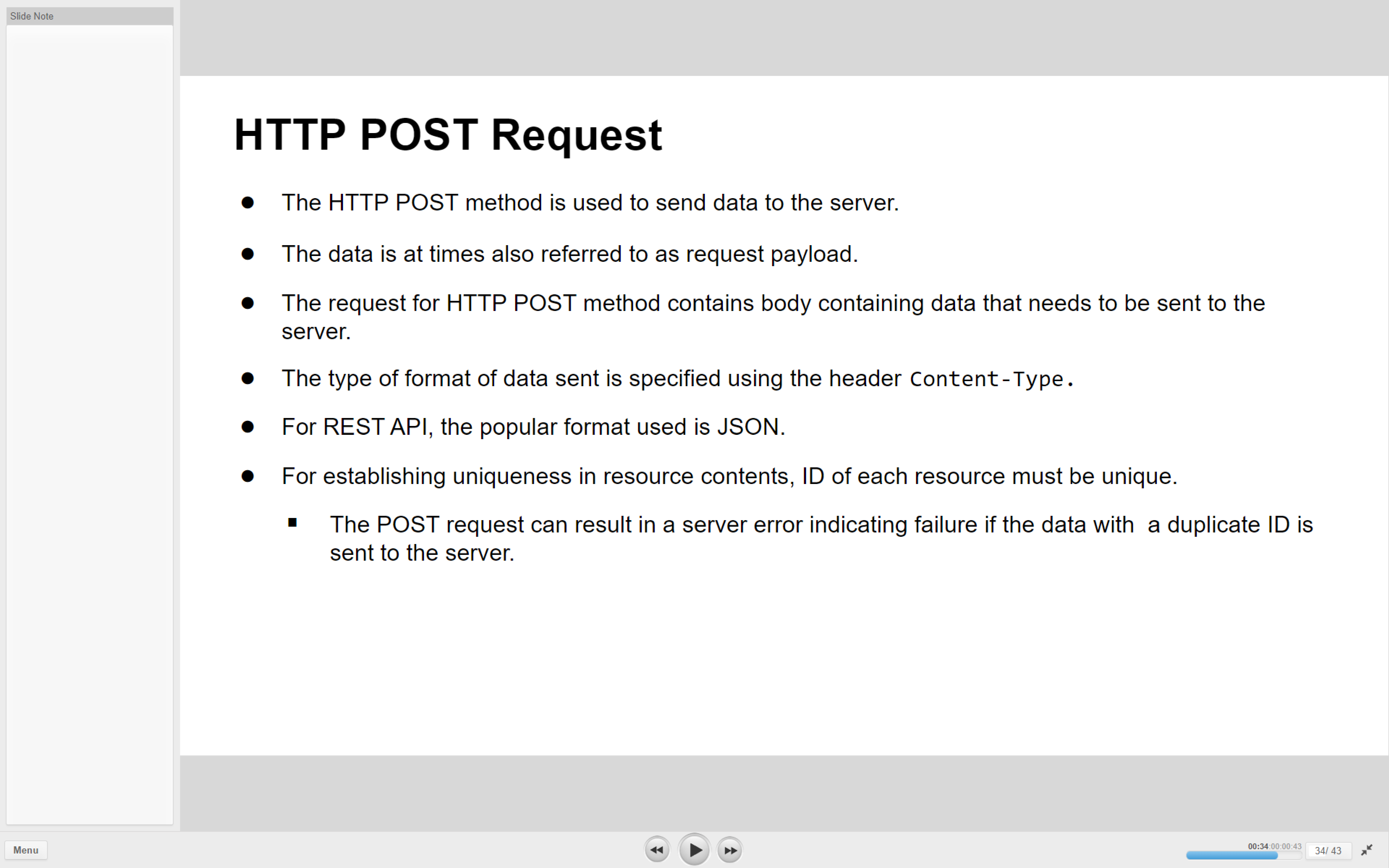
Check the status code if post is fetched with incorrect ID.

Check the status code with the request made with an incorrect port number.

Check the status code with the request made with an incorrect resource name.

Click here for the demo solution.

DEMO



HTTP POST Request

- The HTTP POST method is used to send data to the server.
- The data is at times also referred to as request payload.
- The request for HTTP POST method contains body containing data that needs to be sent to the server.
- The type of format of data sent is specified using the header `Content-Type.`
- For REST API, the popular format used is JSON.
- For establishing uniqueness in resource contents, ID of each resource must be unique.
  - The POST request can result in a server error indicating failure if the data with a duplicate ID is sent to the server.

Slide Note

Save Blog Post

Using Postman, make an HTTP POST request to save the blog post with the following details:

```
id: 3
title: "json"
author: "smith"
isPublic: true
keywords: ["json","javascript","object","json-server"]
```

Check the status code for success of the post request.
Check the status code if the blog post with duplicate ID is posted.

Click here for the demo solution.

DEMO

Menu

PUT Request

- The HTTP PUT request helps perform an update on the target resource.
- The specific resource is usually selected based on the ID of the resource provided with the request URL.
- Like the POST request, the PUT request contains data that is sent for an update to the server.

Slide Note

Update Blog Post

Using Postman, make an HTTP PUT request to update the blog post with id 2 with following details:

```
title: "rest api"
author: "smith rogers"
isPublic: false
keywords: ["rest", "api", "http", "post", "put"]
```

Check the status code when the request is successful.
Check the status code if request is sent with an incorrect ID.

Click here for the demo solution.

DEMO

Menu

00:37 00:00:43

37/ 43

DELETE Request

- The HTTP DELETE method is used to delete the specified resource.
- The specific resource is usually selected based on the ID of the resource provided with the request URL.
- Unlike the GET method, the DELETE method may change the resource state.

Delete Blog Post

Using Postman, make an HTTP DELETE request to update the blog post with id 2.

Check the status code when the request is successful.

Check the status code if the request is made with an incorrect ID.

Click here for Demo Solution.

DEMO

Slide Note

Self-Check

Which HTTP method is used to create a resource?

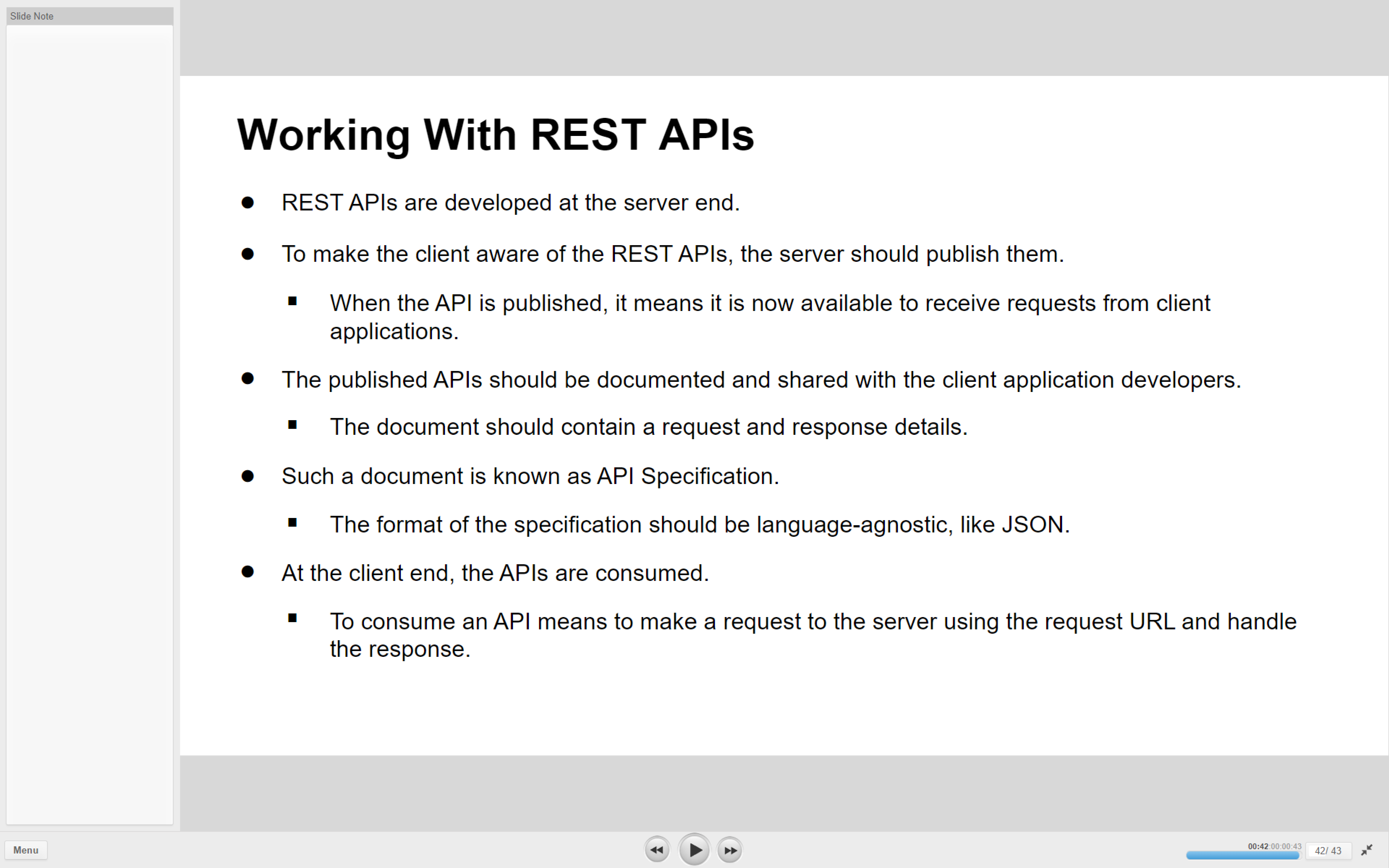
1. CREATE
2. POST
3. GET
4. PUT

Menu

Self-Check: Solution

Which HTTP method is used to create a resource?

1. CREATE
2. **POST**
3. GET
4. PUT



Working With REST APIs

- REST APIs are developed at the server end.
- To make the client aware of the REST APIs, the server should publish them.
  - When the API is published, it means it is now available to receive requests from client applications.
- The published APIs should be documented and shared with the client application developers.
  - The document should contain a request and response details.
- Such a document is known as API Specification.
  - The format of the specification should be language-agnostic, like JSON.
- At the client end, the APIs are consumed.
  - To consume an API means to make a request to the server using the request URL and handle the response.