

SER 321 A Session

SI Session

Sunday, February 2nd 2025

7:00 pm - 8:00 pm MST

Agenda



OSI Review Challenge

HTTP Review

JSON Syntax Review & Practice

Socket Review

Properties & Steps

Port Examination

SI Session Expectations

Thanks for coming to the **SER 321** SI session. We have a packed agenda and we are going to try to get through as many of our planned example problems as possible. This session will be recorded and shared with others.

- If after this you want to see additional examples, please visit the drop-in tutoring center.
- We will post the link in the chat now and at the end of the session.
 - tutoring.asu.edu
- Please keep in mind we are recording this session and it will be made available for you to review 24-48 hours after this session concludes.
- Finally, please be respectful to each other during the session.

Interact with us:

Zoom Features



Zoom Chat

- Use the chat feature to interact with the presenter and respond to presenter's questions.
- Annotations are encouraged

Check out the recording for the solution!

SER 321

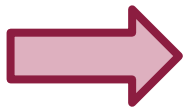
OSI Model

Unit

Layer

What we are *really*
talking about

Data		
Data		
Data		
Segment		
Packet		
Frame		
Bits		



What are the ***FOUR*** request types we reviewed?

1.

2.

Check out the recording for the solution!

3.

4.

What's the difference?

1. GET

2. POST

Check out the recording for the discussion!

3. PUT

4. DELETE

Check out the recording for the solution!

SER 321

HTTP Matching

Match the HTTP response code
with its meaning:

Code:

1XX

2XX

3XX

4XX

5XX

Meaning:

User Error

Server Error

Information

Redirect

Success

SER 321

JSON Structure

Check out the recording for the discussion!

Data is stored in...

Name:Value pairs

AKA

Members

"Katie"

"student" : "Katie"

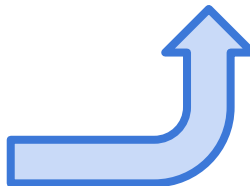
What uses curly braces?

Objects

{ }

What do Objects contain?

Members



SER 321

JSON Structure

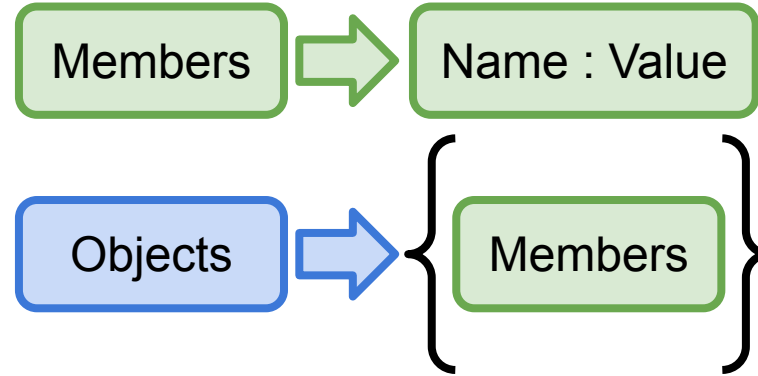
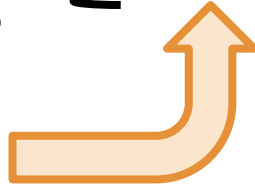
What uses brackets?

Arrays

[]

What do Arrays contain?

Any **Valid** Value



Check out the recording for the discussion!

SER 321

JSON Structure

What is a valid value?

Strings

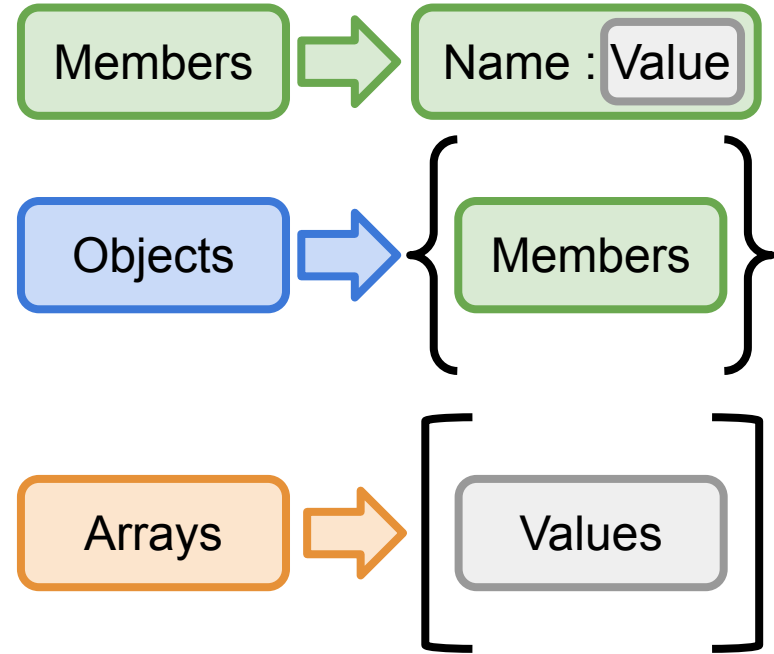
Booleans

Numbers

NULL

Objects

Arrays



Check out the recording for the discussion!

SER 321

JSON Recognition

How many Objects?

How many Arrays?

How many Members?

***Check out
the
recording for
the solution!***

```
{
  "lat": 42.3434,
  "lon": -88.0412,
  "timezone": "America/Chicago",
  "timezone_offset": -21600,
  "current": {
    "dt": 1733070576,
    "sunrise": 1733058144,
    "sunset": 1733091649,
    "temp": 18.57,
    "feels_like": 5.97,
    "pressure": 1025,
    "humidity": 63,
    "dew_point": 9.21,
    "uvi": 0.79,
    "clouds": 0,
    "visibility": 10000,
    "wind_speed": 14.97,
    "wind_deg": 280,
    "wind_gust": 21.85,
    "weather": [
      {
        "id": 800,
        "main": "Clear",
        "description": "clear sky",
        "icon": "01d"
      }
    ]
  }
}
```

SER 321

JSON Practice

JSONObject json =

How would we...

Check for the timezone member?

boolean hasTimezone =

Get the timezone?

String timezone =

```
{
  "lat": 42.3434,
  "lon": -88.0412,
  "timezone": "America/Chicago",
  "timezone_offset": -21600,
  "current": {
    "dt": 1733070576,
    "sunrise": 1733058144,
    "sunset": 1733091649,
    "temp": 18.57,
    "feels_like": 5.97,
    "pressure": 1025,
    "humidity": 63,
    "dew_point": 9.21,
    "uvi": 0.79,
    "clouds": 0,
    "visibility": 10000,
    "wind_speed": 14.97,
    "wind_deg": 280,
    "wind_gust": 21.85,
    "weather": [
      {
        "id": 800,
        "main": "Clear",
        "description": "clear sky",
        "icon": "01d"
      }
    ]
  }
}
```

***Check out
the
recording for
the solution!***

SER 321

JSON Practice

*Check out the
recording for
the
discussion!*

JSONObject json =

How would we...

Recall that
nested
members
require
multiple steps!

```
{  
  "lat": 42.3434,  
  "lon": -88.0412,  
  "timezone": "America/Chicago",  
  "timezone_offset": -21600,  
  "current": {  
    "dt": 1733070576,  
    "sunrise": 1733058144,  
    "sunset": 1733091649,  
    "temp": 18.57,  
    "feels_like": 5.97,  
    "pressure": 1025,  
    "humidity": 63,  
    "dew_point": 9.21,  
    "uvi": 0.79,  
  },  
}
```

Obtain the temp value?

~~String temp = json.getString("temp");~~

Step 1:

Step 2:

Step 3:

Step 4:

SER 321

JSON Practice

**Check out the
recording for
the
discussion!**

JSONObject json =

How would we...

Obtain the temp value?

~~String temp = json.getString("temp");~~

if (json.has("current")) {

}

```
{  
  "lat": 42.3434,  
  "lon": -88.0412,  
  "timezone": "America/Chicago",  
  "timezone_offset": -21600,  
  "current": {  
    "dt": 1733070576,  
    "sunrise": 1733058144,  
    "sunset": 1733091649,  
    "temp": 18.57,  
    "feels_like": 5.97,  
    "pressure": 1025,  
    "humidity": 63,  
    "dew_point": 9.21,  
    "uvi": 0.79,  
    "visibility": 10000,  
    "wind_deg": 280,  
    "wind_speed": 1.6,  
    "clouds": 80  
  }  
}
```

Recall that
nested
members
require
multiple steps!

Step 1: Check for parent object

Step 2:

Step 3:

Step 4:

SER 321

JSON Practice

**Check out the
recording for
the
discussion!**

JSONObject json =

How would we...

Recall that
nested
members
require
multiple steps!

```
{  
  "lat": 42.3434,  
  "lon": -88.0412,  
  "timezone": "America/Chicago",  
  "timezone_offset": -21600,  
  "current": {  
    "dt": 1733070576,  
    "sunrise": 1733058144,  
    "sunset": 1733091649,  
    "temp": 18.57,  
    "feels_like": 5.97,  
    "pressure": 1025,  
    "humidity": 63,  
    "dew_point": 9.21,  
    "uvi": 0.79,  
  },  
}
```

Obtain the temp value?

~~String temp = json.getString("temp");~~

if (json.has("current")) {

JSONObject current =
 json.getJSONObject("current");

Step 1: Check for parent object

Step 2: Obtain parent object

Step 3:

Step 4:

}

SER 321

JSON Practice

**Check out the
recording for
the
discussion!**

JSONObject json =

How would we...

Recall that
nested
members
require
multiple steps!

```
{  
  "lat": 42.3434,  
  "lon": -88.0412,  
  "timezone": "America/Chicago",  
  "timezone_offset": -21600,  
  "current": {  
    "dt": 1733070576,  
    "sunrise": 1733058144,  
    "sunset": 1733091649,  
    "temp": 18.57,  
    "feels_like": 5.97,  
    "pressure": 1025,  
    "humidity": 63,  
    "dew_point": 9.21,  
    "uvi": 0.79,  
  },  
}
```

Step 1: Check for parent object

Step 2: Obtain parent object

Step 3: Check for nested member

Step 4:

Obtain the temp value?

~~String temp = json.getString("temp");~~

if (json.has("current")) {

 JSONObject current =
 json.getJSONObject("current");

 if (current.has("temp")) {
 temp = current.getString("temp");
 }

}

SER 321

JSON Practice

*Check out the
recording for
the
discussion!*

JSONObject json =

How would we create the
“weather” object?

```
{
  "lat": 42.3434,
  "lon": -88.0412,
  "timezone": "America/Chicago",
  "timezone_offset": -21600,
  "current": {
    "dt": 1733070576,
    "sunrise": 1733058144,
    "sunset": 1733091649,
    "temp": 18.57,
    "feels_like": 5.97,
    "pressure": 1025,
    "humidity": 63,
    "dew_point": 9.21,
    "uvi": 0.79,
    "clouds": 0,
    "visibility": 10000,
    "wind_speed": 14.97,
    "wind_deg": 280,
    "wind_gust": 21.85,
    "weather": [
      {
        "id": 800,
        "main": "Clear",
        "description": "clear sky",
        "icon": "01d"
      }
    ]
  }
}
```

SER 321

JSON Practice

**Check out the
recording for
the
discussion!**

JSONObject json =

How would we create the
“weather” object?

```
JSONObject json = new JSONObject();  
JSONObject weather = new JSONArray();  
JSONObject content = new JSONObject();  
content.put("id", 800);  
content.put("main", "Clear");  
content.put("description", "clear sky");  
content.put("icon", "01d");
```

```
weather.put(content.toMap());
```

```
json.put(weather.toMap());
```

```
{  
  "lat": 42.3434,  
  "lon": -88.0412,  
  "timezone": "America/Chicago",  
  "timezone_offset": -21600,  
  "current": {  
    "dt": 1733070576,  
    "sunrise": 1733058144,  
    "sunset": 1733091649,  
    "temp": 18.57,  
    "feels_like": 5.97,  
    "pressure": 1025,  
    "humidity": 63,  
    "dew_point": 9.21,  
    "uvi": 0.79,  
    "clouds": 0,  
    "visibility": 10000,  
    "wind_speed": 14.97,  
    "wind_deg": 280,  
    "wind_gust": 21.85,  
    "weather": [  
      {  
        "id": 800,  
        "main": "Clear",  
        "description": "clear sky",  
        "icon": "01d"  
      }  
    ]  
  }  
}
```

Check out the recording for the discussion!

SER 321

Socket Properties

Sockets allow our client and server to communicate!

Location

Connection
Semantics

Message Format

Need to define **3 properties** before usage

IP or DNS

142.251.46.206

www.google.com

TCP or UDP

Connection
Oriented

Connectionless

Protocol Specs

Synchronous

Asynchronous

Stateless

Stateful

Binary

Text

Headers

No Headers



Check out the recording for the discussion!

SER 321

Socket Properties

Sockets allow our client and server to communicate!

Person

Conversation
Flow

Conversation
Content

Need to define **3 properties** before usage

IP or DNS

142.251.46.206

www.google.com

TCP or UDP

Connection
Oriented

Connectionless

Protocol Specs

Synchronous

Asynchronous

Stateless

Stateful

Binary

Text

Headers

No Headers

Hello!

Welcome!



Check out the recording for the discussion!

SER 321

Socket Protocol Types

Two Main Conversation Models

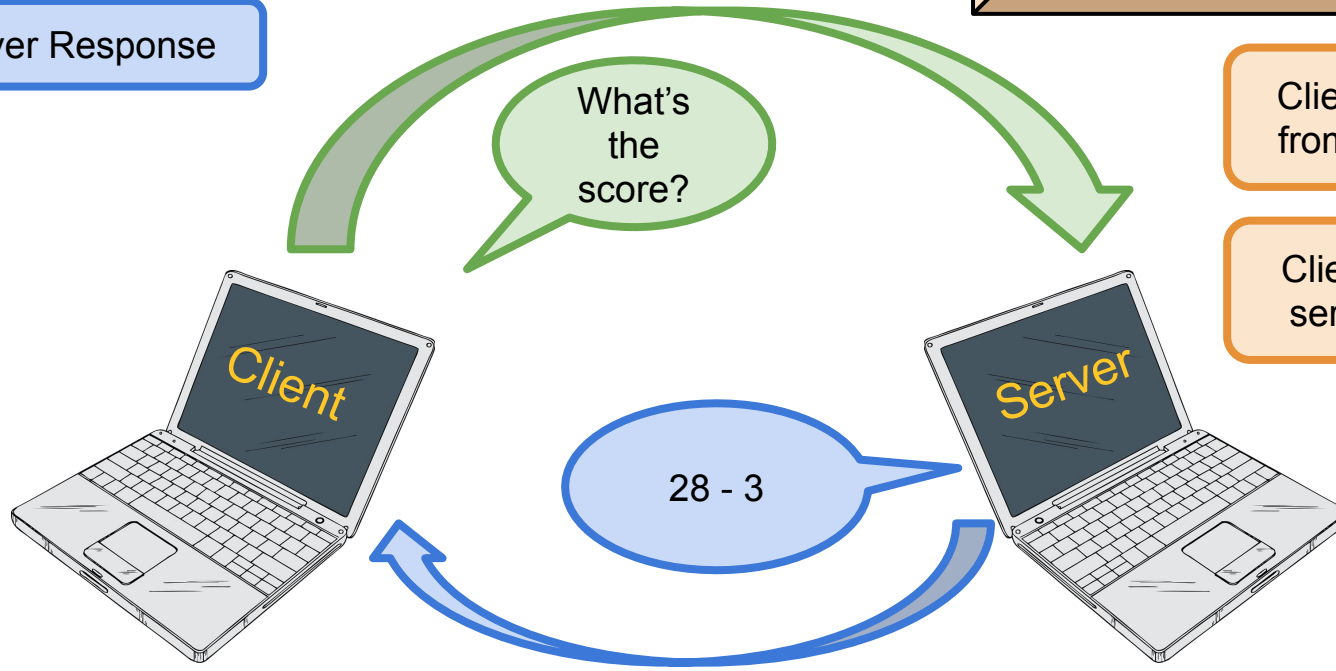
1. Client Request

2. Server Response

Pull/Polling Model

Client *pulls* info from the server

Client *polls* the server for info



Check out the recording for the discussion!

SER 321

Socket Protocol Types

Two Main Conversation Models

1. Server sends update

2. Client acknowledges

Push Model

Server *pushes* info to client

Push notifications



Check out the recording for the discussion!

SER 321

Socket Protocol Types

Two Main Conversation Models

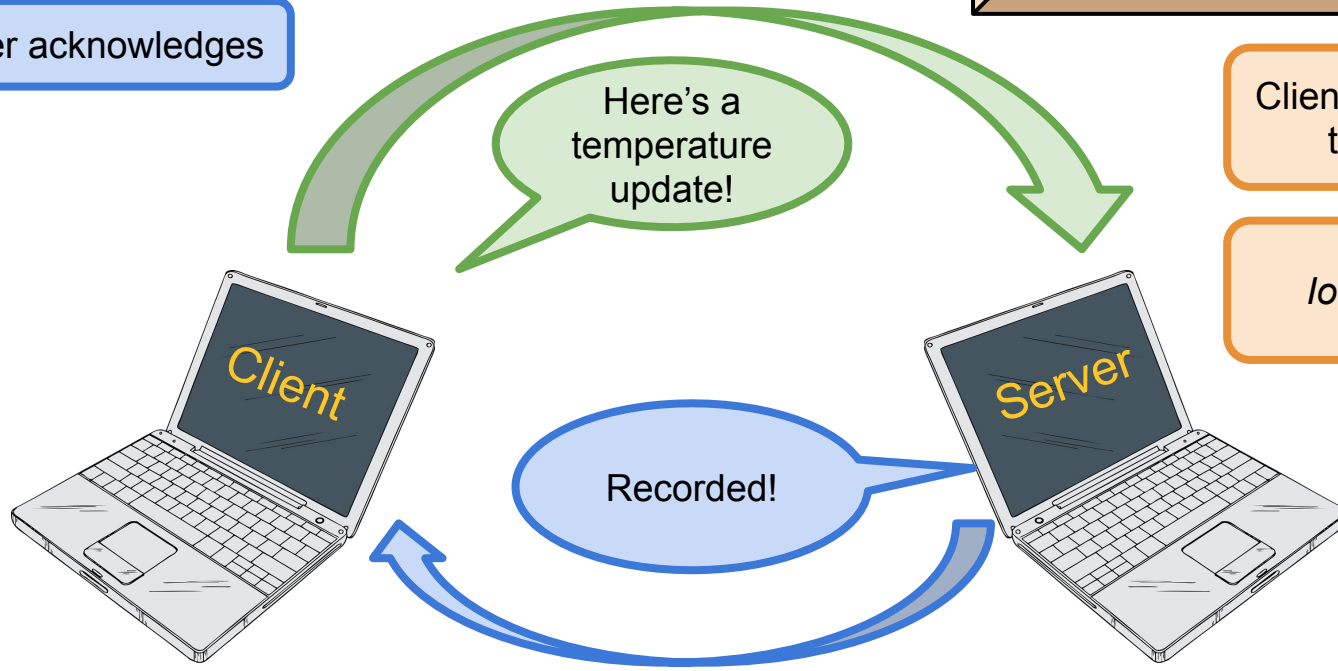
Client Push Model

1. Client sends update

2. Server acknowledges

Client *pushes* info
to Server

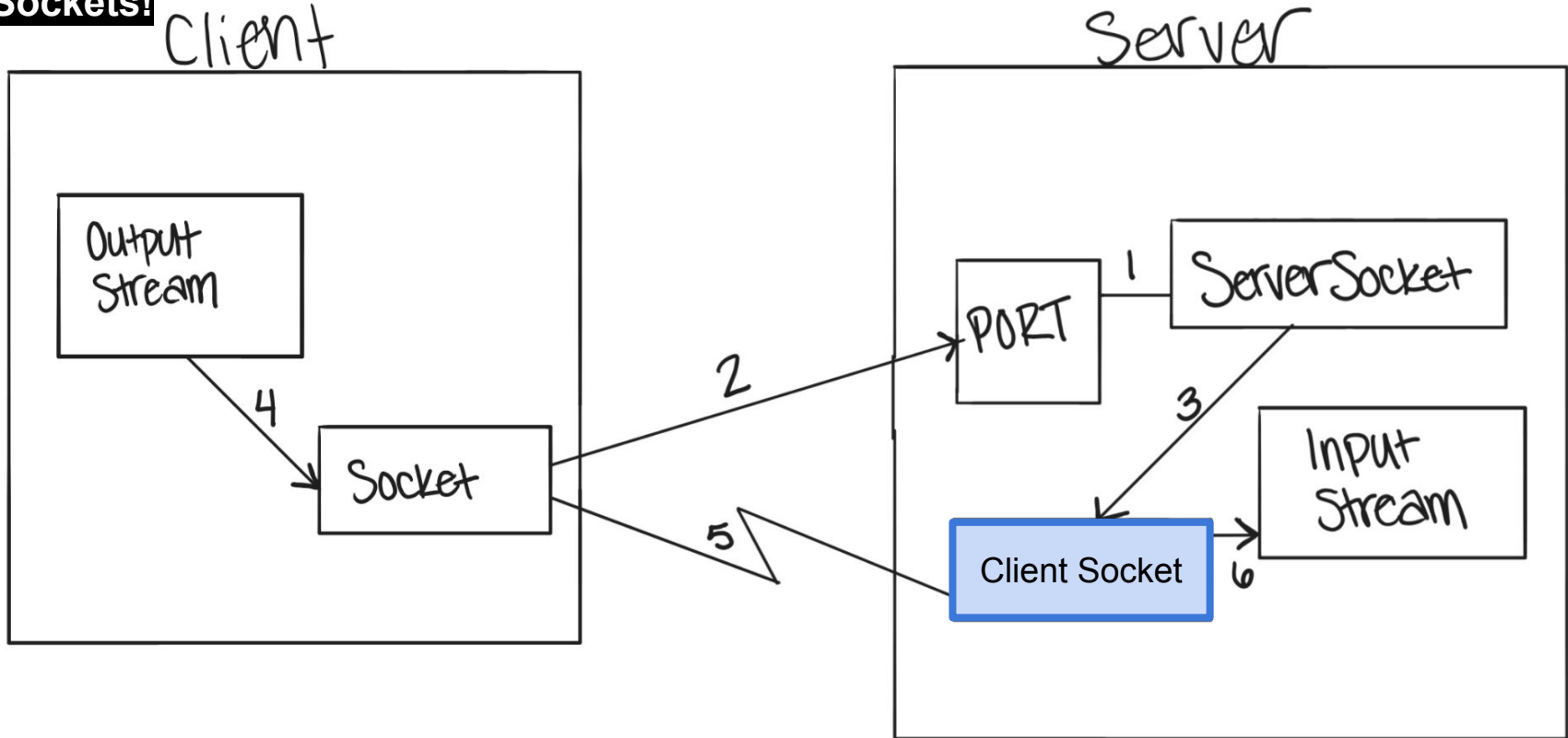
IoT sensors



Check out the recording for the discussion!

SER 321

Sockets!



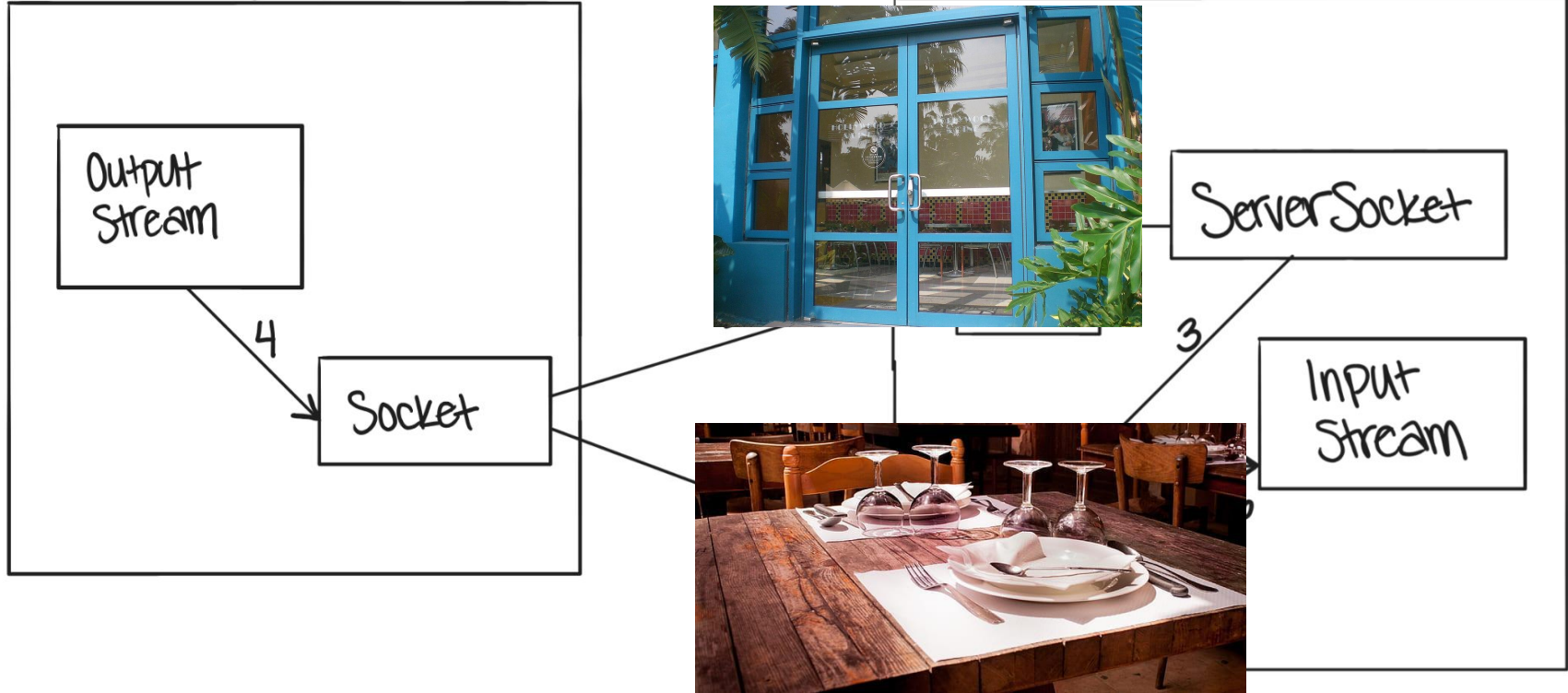
Check out the recording for the discussion!

SER 321

Sockets!

Client

Server

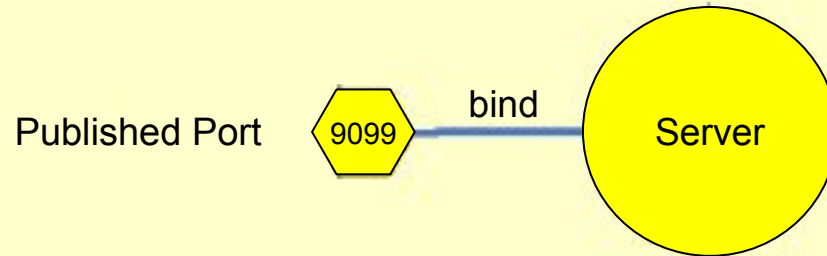


Check out the recording for the discussion!

SER 321

Sockets!

```
> Task :runServer
Server ready for connections
Server is listening on port: 9099
-----
Values of the ServerSocket Object:
Inet Address: 0.0.0.0/0.0.0.0
Local Port: 9099
Server waiting for a connection
Server connected to client
-----
Values of the Client Socket Object after Connection:
    Inet Address: /127.0.0.1
    Local Address: /127.0.0.1
    Local Port: 9099
    Allocated Client Socket (Port): 60296
<=====--> 75% EXECUTING [2m 36s]
> :runServer
```

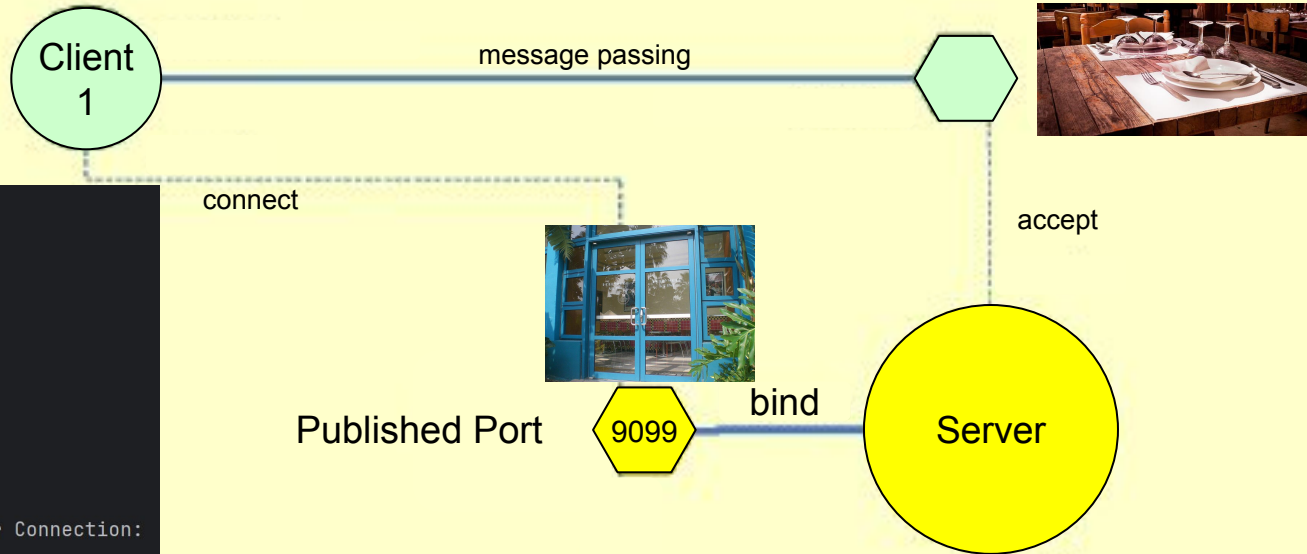


```
> Task :runClient
Connected to server at localhost:9099
Values of the Socket Object for the Server:
    Host: /127.0.0.1
    Port: 9099
    Local Port: 60296
String to send>
<=====--> 75% EXECUTING [2m 18s]s]
> :runClient
```

SER 321

Sockets!

```
> Task :runServer
Server ready for connections
Server is listening on port: 9099
-----
Values of the ServerSocket Object:
Inet Address: 0.0.0.0/0.0.0.0
Local Port: 9099
Server waiting for a connection
Server connected to client
-----
Values of the Client Socket Object after Connection:
Inet Address: /127.0.0.1
Local Address: /127.0.0.1
Local Port: 9099
Allocated Client Socket (Port): 60296
<=====--> 75% EXECUTING [2m 36s]
> :runServer
```



```
> Task :runClient
Connected to server at localhost:9099
Values of the Socket Object for the Server:
Host: /127.0.0.1
Port: 9099
Local Port: 60296
String to send>
<=====--> 75% EXECUTING [2m 18s]s]
> :runClient
```

Check out the recording for the discussion!

SER 321

Scratch Space

Upcoming Events

SI Sessions:

- Tuesday, February 4th at 11:00 am MST
- Thursday, February 6th at 7:00 pm MST
- Sunday, February 9th at 7:00 pm MST

Review Sessions:

- Tuesday, February 25th at 11:00 am MST - **Q&A Session**
- Thursday, February 27th at 7:00 pm MST - **Exam Review Session (2hrs)**

Questions?

Survey:

<https://asuasn.info/ASNSurvey>



More Questions?

Check out our other resources!

tutoring.asu.edu



Academic Support

Academic Support Network (ASN) provides a variety of free services in-person and online to help currently enrolled ASU students succeed academically.

Services



Subject Area Tutoring

Need in-person or online help with math, science, business, or engineering courses? Just hop into our Zoom room or drop into a center for small group tutoring. We'll take it from there.

[Need help using Zoom?](#)

[View the tutoring schedule](#)

[View digital resources](#)

Go to Zoom



Writing Tutoring

Need help with undergraduate or graduate writing assignments? Schedule an in-person or online appointment, access your appointment link, or wait in our drop-in queue.

[Access your appointment link](#)

[Access the drop-in queue](#)

Schedule Appointment



Online Study Hub

Join our online peer communities to connect with your fellow Sun Devils. Engage with our tools to search our bank of resources, videos, and previously asked questions. Or, ask our Tutorbot questions.

Now supporting courses in Math, Science, Business, Engineering, and Writing.

Online Study Hub

1-

Go to Zoom

2-

[Need help using Zoom?](#)

[View the tutoring schedule](#)

[View digital resources](#)



1. Click on 'Go to Zoom' to log onto our Online Tutoring Center.
2. Click on 'View the tutoring schedule' to see when tutors are available for specific courses.

More Questions?

Check out our other resources!

tutoring.asu.edu/online-study-hub

 **Academic Support Network**

Services Faculty and Staff Resources About Us

University College

Online Study Hub

Online peer communities for students and tutors, YouTube channels, and Tutorbots.



What are online peer communities?

Individual courses have an online peer community that allows you to connect with your peers to post and answer questions and to develop study groups.



How can tutoring center videos help?

Videos can help supplement the learning you're doing in and outside of class and include step-by-step methods for how to understand concepts.



How does the Tutorbot work?

You can ask the Tutorbot questions about course concepts and the Tutorbot will recommend additional resources and examples to help address your questions.

Select a subject

- Any -

Apply



Academic Support Network



Services

Faculty and Staff Resources

About Us

University College

Select a subject

- Any -

Apply

Business

ACC 231

Uses of Accounting Info I

Peer Community

ACC 241

Uses of Accounting Info II

Peer Community

CIS 105

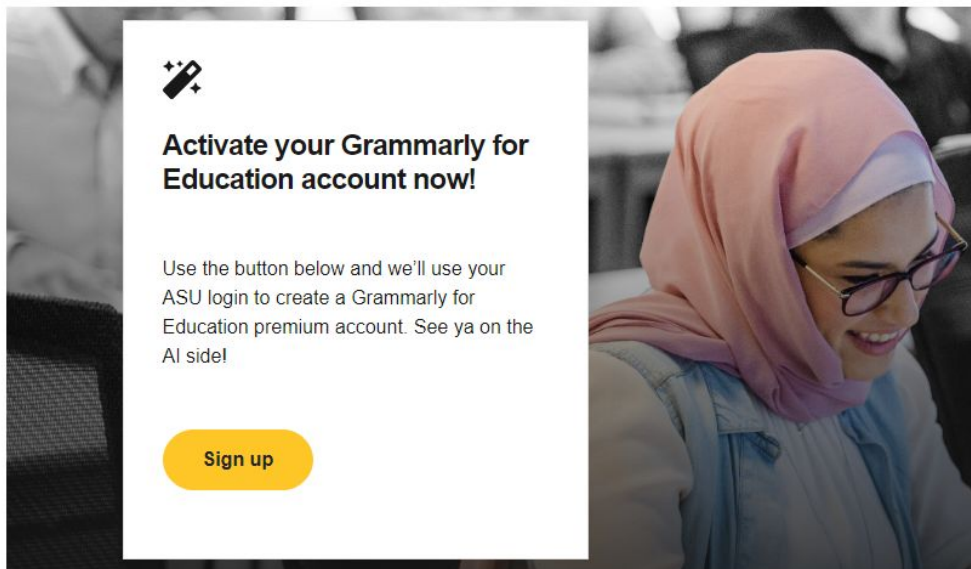
Computer Applications and Information Technology

Peer Community

Don't forget to check out the Online Study Hub for additional resources!

Expanded Writing Support Available

Including Grammarly for Education, at no cost!



tutoring.asu.edu/expanded-writing-support

*Available slots for this pilot are limited

Additional Resources

- [Course Repo](#)
- [Gradle Documentation](#)
- [GitHub SSH Help](#)
- [Linux Man Pages](#)
- [OSI Interactive](#)
- [MDN HTTP Docs](#)
 - [Requests](#)
 - [Responses](#)
- [JSON Guide](#)
- [org.json Docs](#)
- [javax.swing package API](#)
- [Swing Tutorials](#)