

 <p>philipp-matthäus-hahn Technische Schule Nürtingen</p>	Arbeitsblatt	Datum:	pmhs Nürtingen Kanalstraße 29 72622 Nürtingen bar@pmhs.de
	Objekte versenden mit JSON	Name:	

Aufgabe 1: Login auf dem Server

a.) Deklariere in der `server.js` ein neues Array mit dem Namen `"lobby"`, auf dass der gesamte Server Zugriff hat.

b.) Definiere in der Funktion `"connected"` einen neuen Event-Handler, der auf ein Event vom Client reagiert. Du kannst dich an dem Event-Handler für Chat-Nachrichten orientieren. Der neue Event-Handler soll allerdings auf das Event `"login"` reagieren und als Parameter den Socket selbst `"sock"` (also der Absender) und `"data"` (Daten der Nachricht) an die Funktion `"onLogin"` übergeben.

c.) Für das Versenden von Objekten mit JavaScript wird JSON verwendet. Informiere dich im Internet über JSON und notiere dir die wichtigsten Punkte. Für uns sind die Funktionen `"parse"` und `"stringify"` besonders wichtig.

JSON	
parse	
stringify	

d.) Implementiere die Funktion `"onLogin"` im Server. Die Funktion muss die Daten aus der Nachricht parsen (`JSON.parse`) und prüft anschließend, ob der Name des Spielers bereits von einem anderen Spieler in der Lobby verwendet wird. Sollte das der Fall sein wird eine Meldung mit dem Event `"logout"` an den Spieler geschickt und die Funktion mit `"return 0"` beendet. Ist der Name noch nicht vergeben, so wird ein neues Objekt, das den Socket und das Spielerobjekt enthält, in die Lobby eingefügt.



e.) Im Client wird ein `"login"`-Event über den Socket an den Server geschickt sobald der Benutzer einen Namen und eine Mail-Adresse eingegeben hat. Der Inhalt der Nachricht an den Server ist das Objekt `"player"` als JSON-Text (`JSON.stringify`). Du kannst dich am Code zum senden einer Chat-Nachricht orientieren.

f.) Der Client muss auf den Empfang eines `"logout"`-Events reagieren. Implementiere dazu einen Event-Handler, der eine neue Funktion `"onLogout"` aufruft (siehe `"msg"`-Event-Handler). Implementiere außerdem die Funktion `"onLogout"`, die als Parameter eine Nachricht erhält. Sie macht den `"login-wrapper"` wieder sichtbar, gibt die Nachricht als alert aus und setzt das Player-Objekt zurück auf seine Anfangswerte.

Aufgabe 2: Highscore/Lobby beim Client

a.) Definiere im Server eine Funktion "broadcastHighscore". Sie deklariert zunächst ein neues Array "highscore". Anschließend durchläuft eine Schleife die Lobby und pushed für jeden Spieler in der Lobby ein Objekt in den "highscore", dass den "name" und den "score" des Spielers aus der Lobby enthält. Zum Schluss wird das Array "highscore" als JSON-string mit dem neuen Event "highscore" an alle Sockets geschickt (orientiere dich an dem versenden einer Chat-Nachricht an alle Clients).

Die Funktion "broadcastHighscore" wird aufgerufen sobald sich ein neuer Spieler erfolgreich einlogged (onLogin).

b.) Der Client reagiert auf ein "highscore"-Event, indem er die neue Funktion "showHighscore" aufruft (siehe "logout"-Event).

Definiere die Funktion "showHighscore". Sie erhält als Parameter die "data" der Nachricht. "data" wird mit JSON.parse in ein Objekt übersetzt und in einer Variable gespeichert. Da es sich bei den Empfangenen Daten um ein Array handelt muss anschließend das Array schritt für Schritt durchlaufen werden. Für jedes Element des Arrays wird ein neues Listen-Element in der HTML-Liste mit der id "highscore" angelegt, dass als Inhalt den "name" und den "score" eines Elements im Array bekommt.

Orientiere dich am Einfügen einer Nachricht in den Chat.

Zusatzaufgabe: Chat-Nachrichten 2.0 und Styling

a.) Erweitere das Übermitteln von Chat-Nachrichten so, dass statt Text immer JSON-Objekte übertragen werden, die zusätzlich den Absender der Nachricht (Player-Name) enthalten.

b.) Erzeuge Stylesheets für den Client und passe Layout und Gestaltung nach deinen Wünschen an.

Hausaufgabe: Logout

Sobald ein Spieler eingelogged ist soll ein Logout-Button angezeigt werden, über den sich der Spieler wieder ausloggend kann. Außerdem soll ein Spieler ausgelogged werden sobald er die Seite im Browser schließt.

a.) Füge einen Logout-Button auf der HTML-Seite ein, der nur sichtbar wird, wenn ein Spieler eingelogged ist.

b.) Wenn der Button geklickt oder das Browserfenster geschlossen (**beforeunload**) wird sendet der Client ein "logout"-Event mit dem Spieler-Objekt als Inhalt zum Server.

c.) Der Server reagiert auf ein "logout"-Event, indem er den Spieler aus der "lobby" entfernt sowie per "broadcastHighscore" alle Clients informiert. Zum entfernen eines oder mehrerer Elemente aus einem Array wird die Methode "splice" verwendet. (siehe z.B. https://www.w3schools.com/jsref/jsref_splice.asp)

d.) Schreibe eine Funktion, die vor dem Versenden einer Nachricht über den Server überprüft, ob der Benutzer, von dem die Nachricht kommt eingelogged ist und sich in der lobby befindet.