

Einführung in Web Engineering

Übung 3

Aufgabenbeschreibung:

Ziel dieser Übung ist es, den Schiffstracker mit einem Passwort zu schützen, und mit echten live Daten stets aktuell zu halten. Sie werden dazu einen Login Mechanismus bauen und anschließend den Tracker mittels Ajax mit Live Daten versorgen.

Im BSCW finden Sie die Datei resources.zip, welche folgende Dateien enthält:

- login.html: Die Login Seite
- lib/aismessages-1.00-selfcompiled.jar: Bibliothek zum decodieren der Schiffsinformationen
- lib/ShipReceiver.jar: Bibliothek zum Anfrage der von unser bereitgestellten Schiffsinformationen
- lib/servlet-api.jar: Die Implementierung der Servlets.

Sollten Sie etwas nicht verstehen oder Probleme haben, melden Sie sich vorzeitig.

Aufgabe 1: Vorbereitungen, Bibliotheken und Packagestruktur

Öffnen Sie eclipse und erstellen Sie zunächst eine Kopie des Projektes aus der letzten Übung, nennen Sie diese Kopie „Uebung3“ (copy & paste innerhalb von eclipse).

Kopieren Sie die Bibliotheken (JAR Dateien) in das Verzeichnis:

WebContent/WEB-INF/lib

Kopieren Sie die Datei login.html in das Verzeichnis:

WebContent/

Erstellen Sie folgende Packagestruktur:

hbrs.webengineering.u3.filter

hbrs.webengineering.u3.servlet

Aufgabe 2: Login Servlet

Erstellen Sie ein Login Servlet im Package "hbrs.webengineering.u3.servlets". Sie haben freie Namenswahl. Auf der Seite login.html gibt es ein Formular, welches die Eingabe eines Passworts erlaubt. Es soll nur ein gültiges Passwort geben: "secret!". Wenn das Login Servlet dieses Passwort übermittelt bekommt, soll eine Session erstellt werden und mittels eines Attributes mit Namen "authenticated" und Wert Boolean.TRUE gekennzeichnet werden, senden Sie anschließend einen redirect auf die „tracker.html“. Ist das Passwort falsch, soll auf „login.html“ ein redirect erfolgen.

Schritte:

- Erstellen Sie die Servlet Klasse (z.B. „LoginServlet.java“).
- Mappen Sie das Servlet (mittels Annotation oder web.xml).
- Lassen die das Formular in login.html auf ihr Servlet zeigen und geben Sie dem Parameter einen Namen.
- Implementieren Sie das Erstellen der Session sowie das Setzen des "authenticated" Attributes in der Session.

- Führen Sie die Redirects entsprechend durch.

Aufgabe 3: Login Filter

Schreiben Sie einen Filter im Package "hbrs.webengineering.u3.filter", der zunächst überprüft

1. ob eine Session existiert
2. wenn ja, ob die Session auch das Attribute "authenticated" mit Wert Boolean.TRUE besitzt

Sind beide Kriterien erfüllt, soll der Request weitergereicht werden. Wenn nicht, soll ein redirect auf die login.html Seite erfolgen.

Der Filter soll beim Zugriff auf **alle** Ressourcen der Anwendung aktiv sein, **ausgenommen** die login.html Seite und Ihr Login Servlet.

Schritte:

- Erstellen Sie die Filter Klasse.
- Mappen Sie den Filter auf alle Ressourcen der Webanwendung (Annotation oder web.xml)
- Implementieren Sie den Filter entsprechend der Beschreibung. Hinweis: Die Ausnahme bzgl. der "login.html" Seite und Ihres Login Servlets können Sie fest im Filter einprogrammieren.

Aufgabe 4: Live Daten abfragen und im JSON Format bereitstellen

Erstellen Sie ein Servlet, welches auf einen HTTP GET Request die Schiffsinformationen im JSON Format zur Verfügung stellt.

Aktuelle Daten erhalten Sie folgendermaßen:

```
List<Ship> results = ShipReceiver.loadShipsFromServer();
```

Ihre Aufgabe ist es nun, auf dieser Liste eine entsprechende JSON Antwort zu erstellen. Die Antwort sollte immer eine Array von Objekten sein. Jedes Objekt hat die Properties, „mmsi“, „latitude“, „longitude“ und „courseOverGround“.

Informationen zu JSON entnehmen Sie bitte der Vorlesung oder der Seite <http://json.org/>. Hinweis:

Sollten Sie keine Internet Verbindung haben oder der Server vielleicht einmal nicht erreichbar sein, können Sie immer einen Demo Datensatz abfragen:

```
List<Ship> results = ShipReceiver.loadShipsDemoData();
```

Dieser verändert sich nicht, die Schiffe werden sich also nicht bewegen.

Aufgabe 5: Mit AJAX den Tracker aktualisieren

Das aus der letzten Übung bekannte tracker.html soll nun alle 15 Sekunden aktualisiert werden. Fragen Sie dazu in ais.js alle 15 Sekunden per AJAX Ihren Webservice aus Aufgabe 4 ab und rendern Sie das Canvas mit den erhaltenen Daten neu.

Um eine Funktion periodisch aufzurufen, können Sie „setInterval(func, delay)“ verwenden.

<https://developer.mozilla.org/En/Window.setInterval>

Zur Verarbeitung des JSON können Sie den im Firefox eingebauten JSON Parser nutzen: `var obj = JSON.parse(".... hier hier json string");`

https://developer.mozilla.org/En/Using_native_JSON

Sie dürfen die JavaScript Dateien und tracker.html in dieser Übung frei verändern. Der Einsatz von Frameworks wie jQuery ist aber untersagt.

Viel Erfolg!