

Blatt 1

Software Systeme
Webanwendungen

Entwicklungsumgebung - Empfehlung

- Installieren Sie die IDE WebStorm oder IntelliJ. Laden Sie diese herunter:

<https://www.jetbrains.com/webstorm/> .

Registrieren Sie sich als Student bei JetBrains um eine freie Version für das Studium zu erhalten:

<https://www.jetbrains.com/student/> .

Hinweis: Falls Sie lieber mit einer anderen Entwicklungsumgebung arbeiten wollen, dürfen Sie diese frei wählen.

Test der Entwicklungsumgebung

- Erstellen Sie ein neues leeres Projekt in WebStorm und legen Sie eine html Datei mit dem Namen `index.html` an. Füllen Sie die HTML Datei mit dem typischen "Hallo Welt" Beispieltext.
- Verwenden Sie den in WebStorm/IntelliJ integrierten Webserver um die HTML Datei im Browser anzuzeigen.

Abgaben von Aufgaben

Sie müssen bei allen Abgaben, die Programmieraufgaben enthalten, sicherstellen, dass

- die Abgabe den vollständigen Quellcode Ihrer Anwendung beinhaltet.
- Sie gut dokumentiert haben, wie Ihr Projekt zu bauen ist (ich nutze Ubuntu, und es sollten nicht mehr wie ein paar Aufrufe auf der Konsole notwendig sein, um das Projekt zu bauen).
- Sie gut dokumentiert haben, wie Ihre Software gestartet werden kann, d.h. wie der Server zu starten ist, und unter welcher URL dieser dann erreichbar ist.
- Sie dürfen voraussetzen, dass node.js und TypeScript installiert ist. Sollten andere Abhängigkeiten vorhanden sein, führen Sie die Abhängigkeiten klar auf.

Punkte – Note

- Jeder Teil der Veranstaltung Softwaresysteme gibt 30 Punkte.
- Basierend auf der Summe der Punkte beider Teile erhalten Sie Ihre Note für die gesamte Veranstaltung.
- Damit bei den Aufgaben des zweiten Teils keine unangenehmen Rundungsfehler entstehen, erhalten Sie im zweiten Teil 300 Punkte für alle Blätter.
- Die Punktezahl der Blätter wird addiert, durch 10 geteilt und am Ende mathematisch gerundet. Das Ergebnis dieser Rechnung legt die Punkte des zweiten Teils fest.
- Es wird vier Blätter geben. Das letzte Blatt wird 150 Punkte besitzt, und die ersten drei Blätter jeweils 50 Punkte.

A 1.1: Sticky-Sessions **Punkte 10**

Beschreiben Sie, wie Sticky-Sessions funktionieren, und wieso Single Page Webanwendungen bei den Skalierungsproblemen, die mit Sticky-Sessions einhergehen, helfen. Nutzen Sie geeignete Visualisierungen für Ihre Erklärung.

A 1.2: Event Handler – Mouse Click/Move **Punkte 15**

- Erstellen Sie eine HTML Seite auf der ein DIV mit einer grauen Hintergrundfarbe als Zeichenfläche sichtbar ist. Die Zeichenfläche soll eine Höhe und Breite von 500px erhalten.
- Immer wenn Sie auf dieser Fläche Ihre Maus bewegen, zeigen Sie in einem Bereich unterhalb der Fläche die Koordinaten des Mauszeigers relativ zur linken oberen Ecke des Zeichendivs an.
- Wenn Sie auf die Fläche Klicken, fügen Sie einen Eintrag zu einer Punkteliste hinzu, die ebenfalls unterhalb der Zeichenfläche dargestellt wird.

A 1.3: Popup-Menu **Punkte 25**

Erstellen Sie ein Framework zum Erstellen und Nutzen von Popup-Menüs. Sie müssen folgende Features umsetzten:

- Sie können beliebig viele Instanzen des Menüs erstellen, indem Sie eine Funktion oder einen Konstruktor aufrufen. Die einzelnen Menü-Instanzen sollen unabhängig von einander funktionieren.
- Jede Instanz soll einfach durch einen Methodenaufruf (`show` im Beispiel) an einer Stelle innerhalb des Browsers angezeigt werden können.
- Wird ein Kontextmenü angezeigt, wird der Rest der Anwendung deaktiviert, d.h. Click-Events z.B. auf Links, die noch im Hintergrund sichtbar sind, sollen nur das Menü schließen, und nicht dem Link folgen.
- Jedes Menü soll durch eine API einfach um Menüeinträge und Trennlinien ergänzt werden können.
- Jeder Menüeintrag wird durch ein Objekt implementiert, dass eine Methode `render` besitzt. Diese Methode gibt ein DOM Objekt zurück, dass zur Darstellung des Eintrags genutzt wird.
- Für jeden Menüeintrag ist es möglich die Funktionalität, die durch das Klicken auf den Menüeintrag ausgeführt wird, frei zu definieren.

Unten finden Sie ein ausführliches Beispiel, wie so ein Menü zu nutzen ist.

Hinweis: Sie werden das Kontextmenü für das Projekt im 4. Blatt benötigen. Dort wird erwartet, dass Sie die Funktionalität zur Verfügung haben. Sie dürfen das Kontextmenü auch mit TypeScript umsetzen, falls Sie dies wollen, oder mit ES6, z.B. um Klassen zu verwenden.

Menünutzung - Beispiel - „Pseudo-Code“ ähnlich zu JavaScript

```
import menuApi from /* read how to use modules ;) */;

function setupContextMenu(menuApi) {
    const menu = menuApi.createMenu();
    const mItem1 = menuApi.createItem("I 1", (m) => {
        console.log("I 1"); // Imagine we would do something great here
        m.hide(); // hide the menu, so m === menu
    });
    const mItem2 = menuApi.createItem("I 2", () => {
        console.log("I 2"); // Imagine we would do something great here as
                            // well
    });
    const mT1 = menuApi.createSeparator();
    const mItem3 = menuApi.createItem("I 3", (m) => {
        m.hide(); // Here, we just want to hide the menu
    });
    m1.addItems(mItem1, mItem2);
    m1.addItem(mT1);
    m1.addItems(mItem3);

    return menu;
}

const menu = setupContextMenu(menuApi);
domElement.addEventListener('click', (e) => {
    // do something to find the correct x,y values
    menu.show(x,y);
});
domElement2.addEventListener('click', (e) => {
    // do something to find the correct x,y values
    const dynamicItem = menuApi.createItem("II 5", (m) => {
        console.log("wow");
        m.hide();
    });
    // Order after the call: I 1, I 2, dynamicItem, Sep, I 3
    menu.addItemAt(dynamicItem, 2);
    menu.show(x,y);
    menu.removeItem(dynamicItem);
});
```