

Testat

Inhaltsverzeichnis

Zeitplan / Ablauf / Deadlines / Anforderungen	2
Durchführung	2
Vorlage	2
Peer-Review	3
Finale Abgabe	4
Bestehens-Anforderungen	4
Nachkorrektur	4
Aufgabenbeschreibung	5
Schritt 1: Game-Local	5
Schritt 2: Game-Server	7

Frieder Loch <frieder.loch@ost.ch>

Zeitplan / Ablauf / Deadlines / Anforderungen

25.10.	Ausgabe der Testat-Aufgabe
18.11.	Testat für Peer-Review abgegeben
25.11.	Peer-Review abgeschlossen
23.12.	Testat abgegeben

Durchführung

Das Testat ist eine Einzelarbeit und wird selbstständig und in den Übungen bearbeitet.

Vorlage

Verwenden Sie die Vorlagen. Diese ist bereits richtig konfiguriert. Folder-Struktur und die Schnittstelle des Moduls `game-service`.

Getting Started

1. Entpacken Sie die Vorlage.
2. Installieren Sie die Abhängigkeiten (`npm install` im Wurzelverzeichnis des Projekts).
3. Testen Sie, ob alles richtig installiert wurde
 - Console: `npm run all` im Root vom Projekt
 - Erwartete Ausgabe: 4 Warnungen (`unexpected console state no-console` und `npm run all completed`)

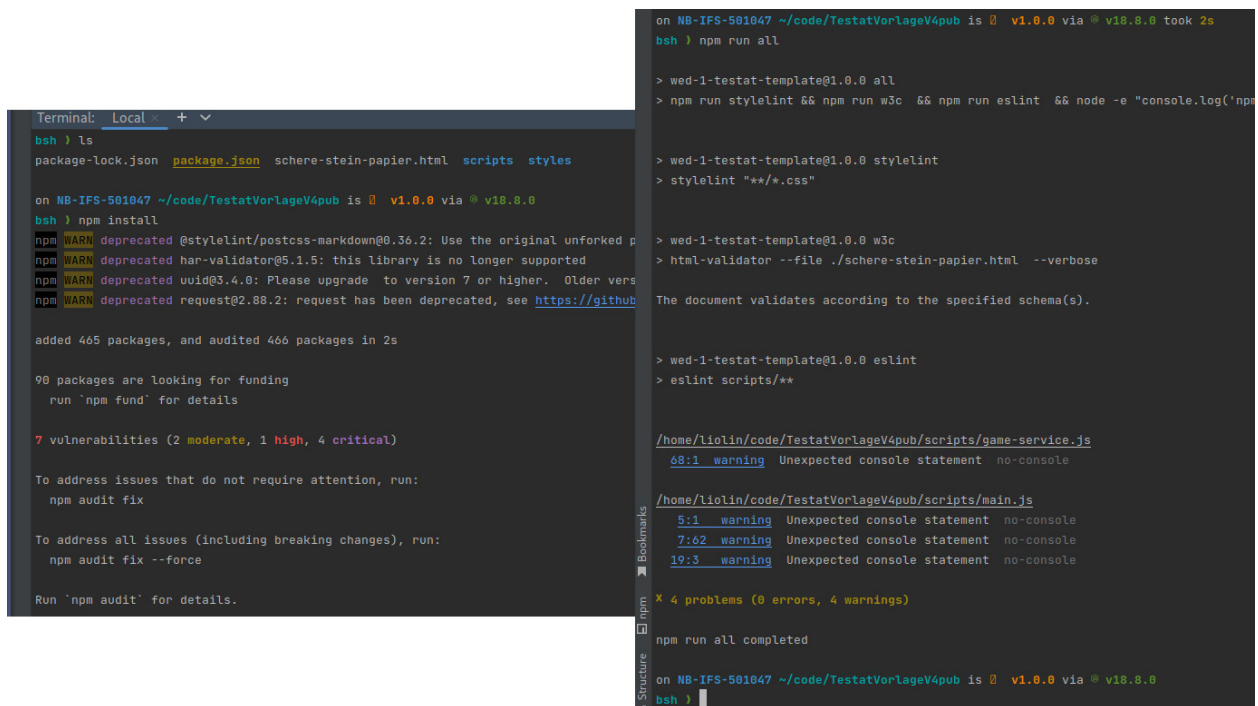


Abbildung 1. Abhängigkeiten installieren & prüfen ob alles installiert ist

Folgende Befehle sind nun möglich:

Befehl	Beschreibung
<code>npm run stylelint</code>	Testet ob CSS-Files in Ordnung sind.
<code>npm run w3c</code>	Testet ob HTML-Files in Ordnung sind.
<code>npm run eslint</code>	Testet ob JS-Files in Ordnung sind.
<code>npm run all</code>	Führt Tests für CSS/HTML/JS aus.

Wichtig: die Abhängigkeiten werden in den Folder `node_modules` abgelegt. Geben Sie diesen Folder **nicht** ab.

Peer-Review

Teil des Testat ist ein Peer-Review. Alle notwendigen Informationen zum Peer-Review werden in Moodle kommuniziert.

- Stellen Sie sicher, dass Ihre Implementation schon vor der Abgabe für das Peer-Review den Anforderungen der Checkliste entspricht.
- Das Review ist nicht anonym. Sie sehen für welche Person Sie den Review machen und in der von uns aktivierten nächsten Phase sehen Sie wer für Sie die Reviews gemacht hat.

- Reviews sollten ehrlich, nach bestem Wissen und Gewissen gemacht werden. Wir überprüfen dies stichprobenweise.
- Beim Peer-Review sollten Sie Ihren zugeteilten «Peers» eine Aussage zur Checkliste machen und Hinweise geben wie die identifizierten Fehler reduziert bzw. eliminiert werden können. Das heisst es ist konstruktive Kritik gefragt.
- Die Durchführung des zugeteilten Peer-Reviews in genügender Qualität ist Pflicht.
- Nicht durchgeführte oder offensichtlich falsche Peer-Reviews führen dazu, dass Ihr Testat als «nicht erfüllt» gewertet wird.

Finale Abgabe

Ihre korrigierte Version können Sie als Abgabe auf Moodle hochladen. Die Ablage wird rechtzeitig ermöglicht und kommuniziert.

Die finale Abgabe muss folgende Kriterien erfüllen:

- Checkliste erfüllt
- Screenshot von `npm run all`
 - Keine (oder nur begründete Fehler/Probleme) im W3C Check
 - Keine (oder nur begründete Fehler/Probleme) im ESLint Check
 - Keine (oder nur begründete Fehler/Probleme) im StyleLint Check

Bestehens-Anforderungen

Für das Bestehen des Testats müssen folgende Bedingungen erfüllt sein:

1. Sie haben das Peer-Review abgegeben
2. Die Reviews sind inhaltlich korrekt (wird in Stichproben überprüft)
3. Ihre finale Abgabe erfüllt die Checkliste

Nachkorrektur

Falls ihre finale Abgabe nicht in Ordnung ist, haben Sie die Möglichkeit einer Nachkorrektur. Die Nachkorrektur wird von uns abschliessend bewertet (pass/fail). Eine zweite Nachkorrektur ist nicht möglich. Die Nachkorrektur wird rechtzeitig

kommuniziert.

Aufgabenbeschreibung

Sie sollen das Spiel Schere **Stein Papier (Brunnen Streichholz)** implementieren. Das Testat ist in zwei Schritte aufgeteilt.

- Schritt 1: Offline Version
- Schritt 2: Online Version

Das auf Moodle publizierte Video gibt weitere Details zu einer möglichen Implementation der Funktionen. Das Aussehen und Layout Ihrer Abgabe muss nicht der im Video gezeigten Version entsprechen.

Wechsle zu Server

Rangliste
1. Rang mit 13 Siegen
Michael
2. Rang mit 10 Siegen
Lisa
3. Rang mit 5 Siegen
Max
4. Rang mit 2 Siegen
Nadine

Starten Sie ein neues Spiel

Name

Spiel Starten

Michael! Wähle deine Hand!

Schere = Stein Papier Brunnen Streichholz

Nächste Runde in 1

Stein

Zurück zum Ranking

History

Resultat	Spieler	Gegner
=	Stein	Stein
✗	Streichholz	Stein
✓	Stein	Schere
✓	Papier	Brunnen

Abbildung 2. Beispiel - Startseite und Spielseite

Schritt 1: Game-Local

Folgende Funktionen sind für Schritt 1 zu implementieren.

Startseite

- Lokale Rangliste darstellen
- Eingabe vom Namen

- Spiel starten, wenn Name eingegeben

Spielseite

- Spieler-Name darstellen
- Mögliche Hände darstellen
- Auswahl der Hand
- Wahl des Computers anzeigen
- Resultat anzeigen
- Zwischen den einzelnen Spielen ist eine Wartezeit einzuhalten (Optional in Schritt 1,
- Verpflichtend bei Schritt 2 und der Endabgabe)
- Die History zeigt alle Spiele der aktuellen Session an.
- Zurück zur Startseite

Table 1. Testat Checkliste: Schritt 1

Kriterium	Evaluation
Kompletter Funktionsumfang ist vorhanden und funktioniert wie im Video: Startseite Rangliste, Eingabe vom Namen (Required), Spiel starten Spielseite Spieler-Name darstellen, Darstellen der möglichen Hände, Auswahl der Hand, Anzeigen der Computer-Wahl, Darstellen vom Resultat, History, Zurück zu Startseite.	Ja / Nein
W3C Check, Stylelint, ESLint ohne Fehler/Warnungen	Ja / Nein
Kein JavaScript im HTML-File	Ja / Nein
Game-Service wird genutzt (keine Netzwerk-Calls/fetch ausserhalb das Game-Service). Der Service ist unabhängig vom DOM, d.h. kein Zugriff auf DOM-Elemente im Service.	Ja / Nein
Folder-Struktur der Vorlage und API des <code>game-service</code> wurde beibehalten.	Ja / Nein
Webseite wird nie unresponsive (hängt) und erzeugt keine Fehler (Dev Konsole).	Ja / Nein

Kriterium	Evaluation
Alle Styles wurden über CSS-Regeln definiert. Es gibt keine Inline-Styles	Ja / Nein
Event-Bubbling wurde sinnvoll eingesetzt. Event-Handler werden nicht mehrfach registriert.	Ja / Nein
HTML-Elemente werden korrekt eingesetzt und nicht missbraucht. z.B. <code>
</code> für Styling; <code><div></code> statt semantischen Tags.	Ja / Nein
Code ist verständlich.	Ja / Nein (Kommentare notwendig)
Clean-Code: Der JS/CSS/HTML-Code beinhaltet keine Probleme: <ul style="list-style-type: none"> • Keine globalen Variablen • Kein duplizierter Code • Sinnvolle und konsistente Namensgebung der Variablen Funktionen / CSS-Klassen • Sinnvolle Wahl von HTML-Elementen (z.B. <code>div</code> statt ...). • Kein <code>!important</code> im CSS. • Kein «dead code» 	Ja / Nein
Es wurden nur im Unterricht eingeführte Libraries eingesetzt. Zusätzlich erlaubt: <code>moment.js</code>	Ja / Nein
Abgabe ist eine Single Page App.	Ja / Nein

Schritt 2: Game-Server

Im Schritt 2 soll Ihre App mit dem Game-Server kommunizieren. Der Game-Server wird folgendermassen angesprochen:

Methode	URL	Beschreibung	Rückgabe
GET	https://stone.sifs0005.infs.ch/ranking	Daten für die Rangliste	Beispiel

Methoden	URL	Beschreibung	Rückgabe
GET	https://stone.sifs0005.infs.ch/play?playerName=DemoUser&playerHand=Stein	Lässt Server ein Spiel auswerten	Beispiele
GET	https://stone.sifs0005.infs.ch/play?playerName=DemoUser&playerHand=Stein&mode=normal	Lässt Server ein Spiel auswerten. System-Wahl ist auf die Werte Schere, Stein, Papier reduziert	Beispiele

Folgende Funktionen sind im Schritt 2 zu implementieren:

- Startseite
 - Wechseln zwischen lokalem und Server-Modus
 - Darstellen vom Ranking vom Server oder Lokal
- Spielseite
 - Das Spiel soll den korrekten Services nutzen (Server oder In-Memory).
 - Zwischen den Spielen ist eine Wartezeit einzuhalten.

Table 2. Testat Checkliste: Schritt 2

Kriterium	Evaluation
<p>Kompletter Funktionsumfang ist vorhanden und funktioniert wie im Video: Startseite</p> <p>Wechseln des Modus («Lokal», «Server»). Im Modus «Server» wird das vom Server gelieferte Ranking angezeigt, im Modus «Lokal» das im Browser verwaltete Ranking.</p> <p>Spielseite</p> <p>Das Spiel kontaktiert im Modus «Server» den Server für die Spielauswertung. Im Modus «Lokal» findet die Auswertung im Browser statt.</p>	Ja / Nein
Für Server-Anfragen benutzt wird fetch() benutzt	Ja / Nein

Kriterium	Evaluation
Zwischen den Spielen muss gewartet werden und es ist keine Eingabe möglich. Dies wird vom System dargestellt.	Ja / Nein