zirpins ...	on 25 Apr
..	
gta_v2/public	6 months ago
README.md	6 months ago

README.md

2. Aufgabe - clientseitige Programmierung

In der zweiten Aufgabe soll die Position (d.h. die Koordinaten) des Geräts, auf dem der Browser läuft, mit JavaScript abgefragt und in das Formular aus Aufgabe 1 eingetragen werden. Es wird dazu die **HTML5 GeoLocationAPI** verwendet. Zusätzlich soll durch Nutzung der **MapQuest API** eine dynamische Karte angezeigt werden.

Die Aufgabe vertieft die Programmierung von **Klassen** und **Callbacks** sowie **DOM Manipulation** mit JavaScript. Zudem wird die Nutzung einer externen **Web API** demonstriert.

2.1. Vorbereitung

Aktualisieren sie zunächst das Github Repository https://github.com/zirpins/vs1lab. Wenn Sie den git-Tipps aus Aufgabe 1 gefolgt sind, gehen sie wie folgt vor (Beispiel für Linux/Mac):

```
cd ~/git/vs1lab # wechsle in das git Verzeichnis
git checkout master # wechsle in den Hauptzweig (eigene Änderungen vorher mit 'commit
git pull # lade Aktualisierungen herunter
git checkout dev # wechsle wieder in den eigenen Branch
git merge master # übernehme mögliche Änderungen aus dem Hauptzweig (Daumen drücken,
```

Nach der Aktualisierung kann die zweite Aufgabe vorbereitet werden.

2.1.1 Vorherige Lösungen übernehmen und IDE vorbereiten

Da die Aufgaben aufeinander aufbauen, aber die Lösungen nicht vermischt werden sollen, muss die Lösung der ersten Aufgabe als Ausgangspunkt für die zweite Aufgabe übernommen werden:

- Kopieren Sie die Datei `Aufgabe1/gta_v1/public/stylesheets/style.css` aus Aufgabe 1 nach `Aufgabe2/gta_v2/public/stylesheets/style.css`
- Kopieren Sie die Datei `Aufgabe1/gta_v1/public/index.html` aus Aufgabe 1 nach `Aufgabe2/gta_v2/public/index.html` und öffnen Sie diese dann im **Editor**.
- Öffnen sie auch das Skript `Aufgabe2/gta_v2/public/javascripts/geotagging.js` in ihrem Editor.

2.1.2 Javascript im HTML aktivieren

Die HTML Datei aus Aufgabe 1 muss nun noch etwas angepasst werden. Fügen sie als letzten Teil im Body des HTML Dokuments folgendes Fragment ein:

```
<!-- Load JavaScripts
    ================================================== -->
    <!-- Placed at the end of the document so the pages load faster -->
    <script src="./javascripts/geotagging.js"></script>
```

2.1.3 Javascript testen

Nun können sie testen, ob die Vorbereitung erfolgreich war. Öffnen Sie die Datei `Aufgabe2/gta_v2/public/index.html` im **Browser**. Es sollte nun ein Alert-Fenster erscheinen. Dadurch sehen sie, dass das zugehörige Skript ausgeführt wird.

2.2. Teilaufgaben

1. Teilaufgabe: Koordinaten in die Formulare eintragen

Das JavaScript enthält zunächst einige Klassen mit Hilfsfunktionen. Die Klasse `LocationHelper` erleichtert die Verwendung der HTML5 Geolocation API zur Bestimmung der Position. Die Funktion `findLocation` nimmt als Parameter eine *Callback Funktion* an, die bei Erfolg mit einem instanziierten LocationHelper Objekt 'zurückgerufen' wird. Das LocationHelper Objekt enthält dann die aktuellen Koordinaten als private Properties, die mit einer 'get'-Methode ausgelesen werden können. Beim Aufruf der Funktion muss die Callback Funktion übergeben werden.

Fügen sie eine Funktion `updateLocation` zum Skript hinzu, die folgendes tut:

- Auslesen der Position mit `findLocation`
- Im Erfolgsfall `latitude` und `longitude` Eingabefelder des Tagging-Formulars *und* des Discovery-Formulars (versteckte Eingabefelder) suchen und in deren `value` -Attribute Koordinaten schreiben.

Rufen sie die neue `updateLocation` -Funktion nach dem Laden des Dokuments automatisch auf.

2. Teilaufgabe: Position auf Karte darstellen

Wir wollen nun die gefundene Position auf einer Karte darstellen. Konkret werden wir zu diesem Zweck *MapQuest* verwenden, ein - für unsere Zwecke - kostenfreier Dienst um statische Karten anzuzeigen. Zunächst benötigen sie einen **API-Schlüssel** für die MapQuestApi (kostenlos). Registrieren sie sich dort, erstellen sie eine App ('Callback URL' kann leer bleiben) und notieren sie sich den Key ('Consumer Key').

Die Klasse `MapManager` enthält einige Hilfsfunktionen zur Verwendung von MapQuest. Um eine Instanz zu erzeugen, wird der Konstruktor mit dem MapQuest API-Key aufgerufen. Dann kann die Methode `getMapUrl` verwendet werden, um die URL einer gewünschten Karte abzufragen. Die Karte selbst erhält man schließlich durch einen HTTP Request.

Ergänzen sie ihre `updateLocation` -Funktion im wie folgt:

- Rufen sie die Funktion `getMapUrl` mit den aktuellen Koordinaten auf. Das Ergebnis ist eine **URL** auf die Karte.
- Suchen sie im DOM das **Image Element** auf der Webseite.
- Ändern sie per DOM Aufruf das `src` -Attribut auf die neue URL.

Checkliste

Zur Übersicht folgen noch mal alle Anforderungen in kompakter Form als Checkliste.

1. Teilaufgabe: Koordinaten bestimmen

- [ ] Funktion `updateLocation` erstellen
  - [ ] Nach dem Laden automatisch aufrufen
  - [ ] Auslesen der Position mit `findLocation`
  - [ ] Koordinaten in die Formulare eintragen
    - [ ] `latitude` und `longitude` Felder
    - [ ] Koordinaten in `value` -Attribute schreiben
    - [ ] Auch versteckte Eingabefelder berücksichtigen

2. Teilaufgabe: Karte darstellen

- [ ] MapQuest API-Schlüssel besorgen und eintragen
- [ ] `updateLocation` -Funktion ergänzen
  - [ ] Funktion `getMapUrl` aufrufen
  - [ ] URL im `src` -Attribut des Image Elements eintragen
