

Erstelle eine Wettervorhersage-App.

Das Ziel ist, eine Wetter-App zu erstellen, die auf verschiedenen Endgeräten (PC, Phone, Tablet) aufgerufen werden kann und Wettervorhersagen zu beliebigen Orten weltweit abgeben kann.

Vorbereitung

- Überlege dir einen Namen für diese App. Sie liefert Wetterdaten in verschiedenen Formaten für verschiedene Orte.
- Überlege dir, welche Möglichkeiten so eine App bieten soll und kann. Welche Daten soll der Benutzer sehen, wo sollen die Daten her kommen?
- Verschaffe dir einen kostenlosen Account bei Github. Über deinen Benutzernamen schaltet dich dann Alex für ein Repository frei, in dem deine App gespeichert wird.

Teil 1: Konsolenprogramm

- Erstelle ein .Net-Konsolenprogramm, das den Benutzer begrüßt und ihn anweist, eine Postleitzahl in Deutschland einzugeben, für die er den Wetterbericht haben will. Eine leere Eingabe (nur Enter) beendet das Programm.
- Nach Eingabe einer Postleitzahl (z.B. 80639) kommt die Meldung:
 - o Für 80639 konnten noch keine Wetterdaten abgerufen werden.

Teil 2: Backend-Funktionalität

- Erstelle eine .Net-Standard Klassenbibliothek (Lib). In dieser Lib wird die komplette Funktionalität der Wetter-App gekapselt.
- Die Lib soll eine Klasse enthalten, die den Wetterdaten-Abruf übernimmt.
- Diese Klasse bietet eine Methode mit dem Namen „GetWeatherForecastForZip“ an.
- Diese Methode nimmt als Parameter die vom Benutzer eingegebene Postleitzahl.
- In der ersten Version gibt sie für jede eingegebene PLZ eine zufällige Temperatur zwischen 10° und 30°C an, und einen zufälligen Wert für die Bewölkung.
- Das Konsolenprogramm wird so abgeändert, dass nach Eingabe der Postleitzahl die Backend-Methode aufgerufen wird und das Ergebnis der Methode in der Konsole angezeigt wird, etwa so:
 - o In 80639 hat es heute 29°C und es ist bewölkt.

Teil 3: Art der Wettervorhersage

- Im Konsolenprogramm kann der Benutzer in einem Menü auswählen, was er als Ergebnis haben will:
 - o 1 - Einfache Voraussage für heute – eine Zeile wie oben
 - o 2 - Stündliche Vorhersage für heute – pro Stunde kommt eine Zeile raus
 - o 3 - 3-Tages-Vorhersage - mit vormittag/nachmittag/nachts
 - o 4 - 14-Tage Vorhersage – pro Tag eine Zeile
- Der Benutzer muss natürlich immer noch die PLZ eingeben
- Bei Menüpunkt 1 bekommt der Benutzer die gleiche Ausgabe wie bei Teil 2, bei der Auswahl anderer Punkte kommt die Meldung
 - o Für diesen Menüpunkt ist noch keine Funktionalität vorhanden.

Teil 4: Backend erweitern und UI anpassen

- In der Lib wird die Klasse so erweitert, dass sie auf die verschiedenen Anforderungen reagieren kann. Für jede Art der Abfrage (einfach, stündlich, 3 Tage, 14 Tage) wird ein geeignetes Objekt zurückgegeben.
- Momentan wird immer noch mit zufälligen Daten gearbeitet.

- Im Konsolenprogramm werden alle Menüpunkte mit den entsprechenden Methoden in der Lib verbunden. Jeder Menüpunkt liefert nach Eingabe entsprechende (zufällige) Daten zurück.

Teil 5: Benutzereinstellungen

- Wenn das Programm gestartet wird, wird der Benutzer nach seinem Namen gefragt. Der Name wird lokal gespeichert und beim nächsten Aufruf der App wird der Benutzer mit Namen begrüßt.
- Bei jeder Menüwahl und Eingabe einer PLZ werden die zuletzt eingegebenen Daten ebenfalls lokal gespeichert.
- Beim nächsten Aufruf der App wird der Benutzer mit Namen begrüßt und die zuletzt gewählten Optionen werden ausgeführt, das heißt, der Benutzer bekommt eine Meldung, wie das Wetter an seinem zuletzt eingegebenen Ort gerade ist.

Teil 6: Favoriten

- Ein neuer Menüpunkt „Favoriten“ wird hinzugefügt
- Der Benutzer kann eine Liste aus Postleitzahlen anlegen, für die er bei Programmstart automatisch das Wetter angezeigt bekommt.
- Die Liste ist editierbar, das heißt, er kann neue Einträge machen und alte löschen.

Teil 7: Anbindung externen Wetterdaten

- Im Internet existieren verschiedene APIs, über die sich das Wetter abfragen lässt.
- Such dir eine API aus. Erstelle eine neue Klasse und frage die API über diese Klasse ab.

Teil 8: Verbindung mit der UI:

- Verbinde das Backend mit der Wetter-API. Der Benutzer bekommt jetzt Live-Wetterdaten.

Teil 9: Weitere APIs

- Binde mindestens eine weitere API in dein Programm ein.
- Dein Backend soll in der Lage sein, die APIs zu wechseln, wenn eine nicht erreichbar ist.
- Dein Backend soll in der Lage sein, aus mehreren APIs die Wetterdaten abzufragen und aus den Temperaturen den Durchschnittswert zu bilden.