

Declaratively Programming the Mobile Web with Mobl

Zef Hemel, Eelco Visser

Plan

1. Motywacje dla stworzenia Mobil
2. Mobilne aplikacje webowe, historia
3. Architektura Mobil
4. Model danych
5. Interfejs
 - 5a. Proste kontrolki
 - 5b. Zaawansowane kontrolki
 - 5c. Style
6. Podsumowanie - zalety i wady rozwiązania

Urządzenia mobilne a strony internetowe, problemy

- sprawność połączenia z internetem
- ograniczony ekran
- specyficzny rodzaj interakcji (zamiast myszy i klawiatury "pukanie" w ekran, przeciąganie elementów itp)
- reakcje na zmianę orientacji telefonu czy zmianę lokalizacji

Aplikacje natywne

- wiele platform: Android, iPhone, bada, Blackberry....
- ... a każda ma inne API, nietrywialne!

W zamian za to

- ładny wygląd
- szybkość działania
- bogactwo opcji

Pomysł

- proste, deklaratywne tworzenie programów (jak w HTML)... ale więcej możliwości, większa skalowalność, większa możliwość interakcji
- multiplatformowość... ale dobry wygląd/zachowanie na wszystkich urządzeniach

= HTML5 ?

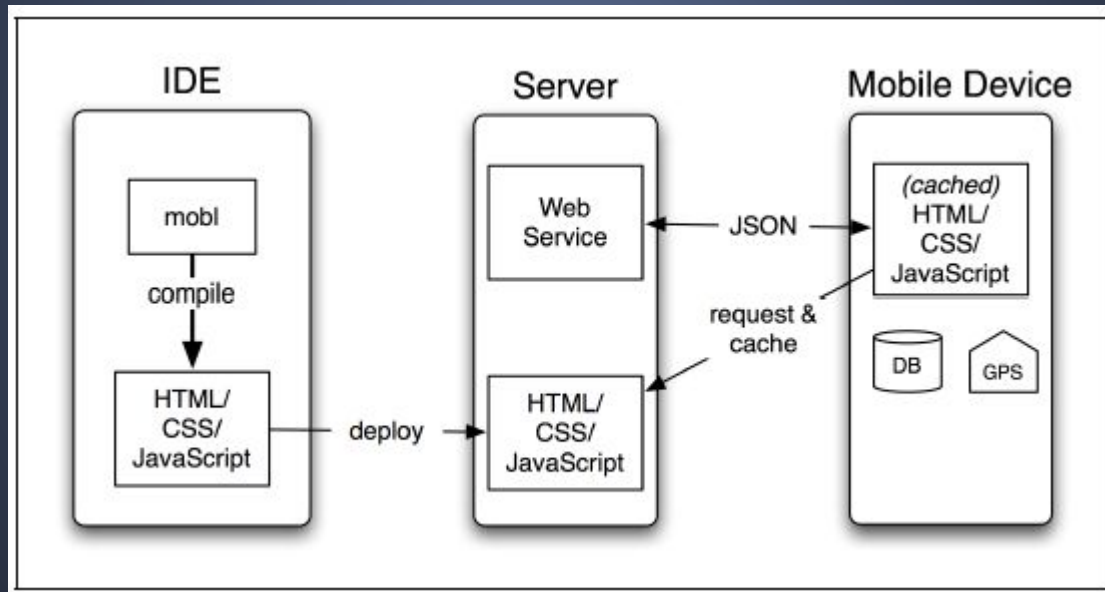
HTML5

- brak skalowalnej architektury (jak np. MVC)
- mieszanie HTML, CSS, JavaScript
- (HTML) brak możliwości wyodrębnienia abstrakcji
- (JavaScript) nienaturalna asynchroniczność

Rozwiązanie - Mob!

- wysokopoziomowy, deklaratywny język programowania dla aplikacji mobilnych
- MVC
- IDE (plugin do Eclipse)

Arhitektura aplikaciji Mobil



Zalety architektury Mobil

- spójność lingwistyczna
- separacja
- abstrakcje (języka i użytkownika)

Model danych

entity

- związane z instancjami w SQLite
- własności
 - skalarne np. String, Num, DateTime, Bool
 - Collection (sortowanie, filtrowanie, paginacja, fulltext-search; podobnie do LINQ)
- funkcje (logika)

Entity - przykład

```
entity Task {
name : String (searchable)
done : Bool
due : DateTime
category : Category (inverse: tasks)
tags : Collection<Tag>
(inverse: tasks)

function postpone(days : Num) {
    this.due = DateTime.create(
        this.due.getFullYear(),
        this.due.getMonth(),
        this.due.getDate() + days);
}

static function import(user :
String,
pw : String) {
    var tasksJSON =
    httpRequest("/export?user="
+ user + "&pw=" + pw);
    foreach(t in tasksJSON) {
        add(Task.fromSelectJSON(t));
    }
}
```

```
entity Category {
name : String
tasks : Collection<Task>
(inverse: category)
}

entity Tag {
name : String
tasks : Collection<Task> (inverse:
tags)
}
```

Interfejs użytkownika

screen, control

- argumenty
- kod HTML
- wyrażenia
- zmienne
- warunki
- pętle

Interfejs użytkownika, własności

- data binding

```
var name = "John"
```

```
<input type="text" value=name/>
```

```
<span body="Hello, "+name/>
```

- zamiast modyfikacji DOM -
reaktywny interfejs

Kontrolki - przykład

```
screen prompt(question : String) : String {
var answer = ""
  header(question) {
    button("Done", onclick={
      screen return answer;
    })
  }
  textField(answer)
}
screen root() {
  button("Ask", onclick={
    alert("Hello " + prompt("First name")
    + " " + prompt("Last name"));
  })
}
```

Kontrolki wyższych rzędów

czyli kontrolki z kontrolek

```
control tabSet(tabs : [(String,Control)], activeTab : String) {  
  list((tabName, tabControl) in tabs) {  
    block(onclick={ activeTab = tabName; },  
          style=activeTab==tabName ? activeTabButton : inactiveTabButton) {  
      label(tabName)  
    }  
  }  
  list((tabName, tabControl) in tabs) {  
    block(activeTab==tabName ? visibleTab : invisibleTab) {  
      tabControl()  
    }  
  }  
}
```

Style

również są parametryzowane

```
style mixin borderRadiusMixin($radius) {  
    -moz-border-radius: $radius;  
    -webkit-border-radius: $radius;  
    border-radius: $radius;  
}  
  
style buttonStyle {  
    color: $buttonTextColor;  
    borderRadiusMixin(5px);  
    ...  
}
```


Więcej o Mobl

- <http://www.mobl-lang.org>
 - przykłady
 - dokumentacja
 - IDE
 - dema
- "Mobl receives a lot of interest from industry."

Ograniczenia Mobil

- wydajność
- ograniczenia wynikające z użycia przeglądarki
 - wydajność
 - brak możliwości odczytu sensorów, niektórych zdarzeń itp
 - słaba obsługa multimedialnych
 - kiepskie user-experience

Podsumowanie

Mobl jest:

- elegancki
- multiplatformowy
- wygodny
- elastyczny

... przydatny?

Alternatywy

- RhoMobile
- Appcelerator
- PhoneGap
- Marmolade
- MoSync
- ...

Dziękuję za uwagę