

Ivana Savin

Razvoj JavaScript biblioteke za rad sa komponentama korisničkog interfejsa

Master rad

Novi Sad, 2019

Korišćene tehnologije

- Klijentska aplikacija
 - OldSchoolComponents biblioteka,
 - *jQuery*,
 - *Bootstrap* i
 - *axios*
- Serverska aplikacija
 - Java 1.8,
 - Spring Boot i
 - Google Gmail API

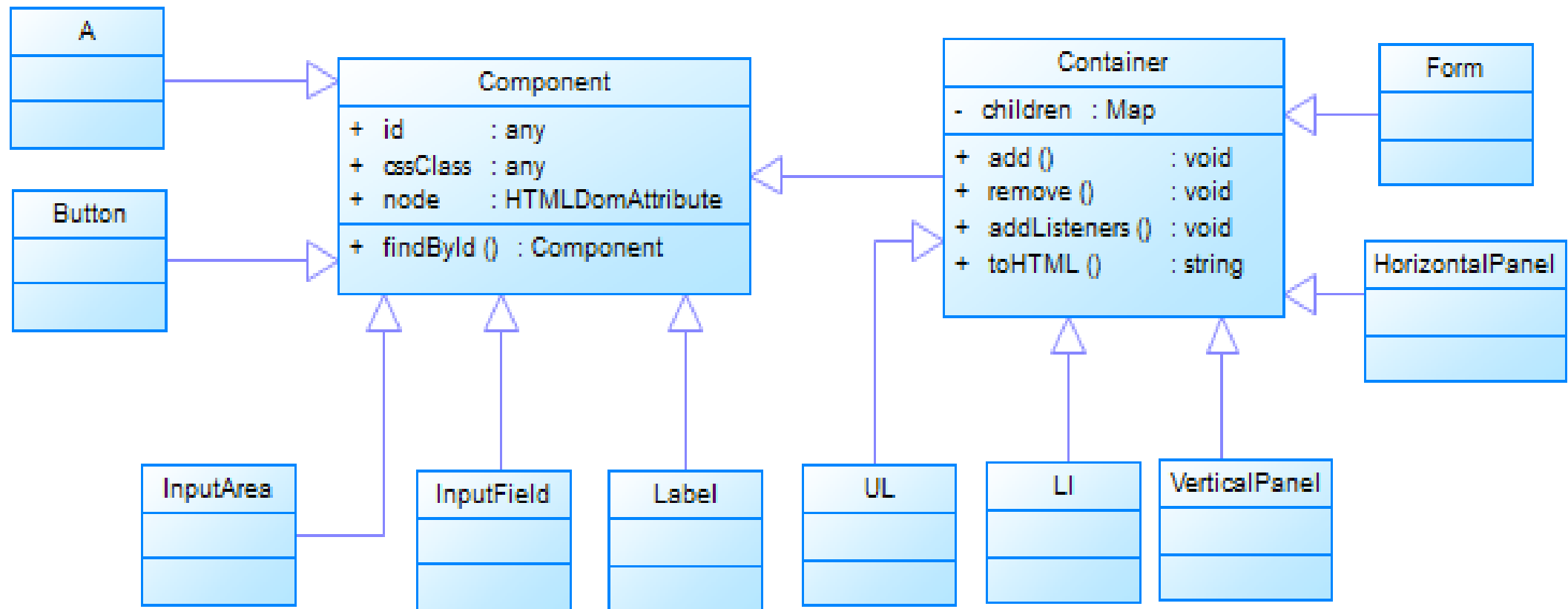
OldSchoolComponents biblioteka - Motivacija

- Omogućiti programeru da pravi web aplikaciju programski, upotrebom koda
- Deklarativno kreiranje komponenti korisničkog interfejsa i dodavanje na ekran

OldSchoolComponents biblioteka - Implementacija

- JavaScript biblioteka
- ECMAScript 6
 - Objektno-orijentisani pristup
 - Mogućnost definisanja JavaScript klasa
 - Objektno-orijentisani pristup prilikom nasleđivanja klasa

OldSchoolComponents biblioteka – Dijagram klasa



OldSchoolComponents biblioteka

- Klasa *Component*
- Osnovna komponenta
- Atributi:
 - *id* - identifikator,
 - *CSSclass* - atribut za prikaz stilova i
 - *node* - atribut koji predstavlja vezu sa DOM reprezentacijom komponente
- *constructor* metoda

OldSchoolComponents biblioteka

- *constructor* metoda

```
constructor(id, CSSclass) {  
    this.id = id;  
    this.CSSclass = CSSclass;  
    this.node = document.createElement("div");  
    this.node.id = id;  
    this.node.component = this;  
    document.body.appendChild(this.node);  
}
```

OldSchoolComponents biblioteka

- InputField klasa

```
export default class InputField extends Component {
  constructor(id, CSSclass, type, placeholder, value) {
    super(id, CSSclass);
    this.type = type;
    this.placeholder = placeholder;
    this.value = value;
  }

  tohtml() {
    return "<input id='" + this.id + "' type='" + this.type +
      "' class='" + this.CSSclass + "' placeholder='"
      + this.placeholder + "' value='" + this.value
      + "'></input>";
  }
}
```


OldSchoolComponents biblioteka

- Klasa *Container*
- Specijalizacija klase *Component*
- Komponenta koja sadrži ostale komponente
- Atributi:
 - *children* – mapa svih komponenti koje kontejner sadrži
- Metode:
 - *add* – dodaje komponentu u određeni kontejner i u DOM stablo,
 - *remove* – briše objektnu reprezentaciju i komponentu iz DOM stabla,
 - *addListeners* – dodaje sve akcije za komponentu,
 - *tohtml* – vraća string reprezentaciju komponente

OldSchoolComponents biblioteka

- *constructor* metoda

```
constructor(id, CSSclass) {  
    super(id, CSSclass);  
    this.children = new Map();  
}
```

OldSchoolComponents biblioteka

- *add* metoda

```
add(component) {  
    this.children.set(component.id, component);  
    component.setParent(this);  
  
    $('##' + component.id).remove();  
    $('##' + this.id).append(component.tohtml());  
    component.node = document.getElementById(component.id);  
  
    for (var child of this.children.values()) {  
        this.addListeners(child);  
    }  
}
```

OldSchoolComponents biblioteka

- *remove* metoda

```
remove(component) {  
    component.children.delete(this.id);  
    $('##' + this.id).remove();  
}
```

Google Gmail API

- Skup funkcija razvijen od strane Google-a
- Druge aplikacije mogu koristiti Google API kako bi poboljšale ili proširile svoja rešenja
- RESTful pristup funkcionalnostima
- Autorizovan pristup korisnikovim Gmail podacima

Google Gmail API

- Funkcionalnosti koje nudi Gmail API su:
 - pregled elektronske pošte,
 - manipulisanje elektronskom poštom,
 - kreiranje nove pošte,
 - slanje pošte i
 - upravljanje labelama.

Google Gmail API

- Resursi koje nudi Gmail API:
 - *message*,
 - *draft*,
 - *label*,
 - *history*,
 - *thread* i
 - *settings*.

Google Gmail API

- Tipičan proces rada sa Gmail API servisom:
 - autentikacija,
 - poziv API metode i
 - procesiranje resursa dobijenih u odgovoru.

SimpleMailApp

- Veb aplikacija za upravljanje elektronskom poštom
- Neke od funkcionalnosti koje aplikacija omogućava su:
 - pregledanje elektronske pošte,
 - kreiranje nove pošte,
 - slanje pošte,
 - odgovaranje na postojeću poštu,
 - brisanje pošte.

Pitanja, komentari, sugestije

Hvala na pažnji!