

Komponenty

SmartWeb framework pre uahenie vývoja webových aplikácií postavených na JavaScript knižnici React poskytuje niekoľko predpripravených komponentov s napojením na systém D2000 cez D2Api. Dodávané komponenty sa nachádzajú v adresári *components* SmartWeb frameworku a alej sú logicky rozlenené do adresárov podľa ich urenia. Niektoré komponenty sa dajú naimportovať alternatívne aj s CSS štýlmi - vtedy má názov súboru predponu "themed". "Otémované" verzie komponentov sú vlastné tie neotémované spolu s prednastavenými CSS štýlmi. Napríklad:

```
// otémovaná verzia komponentu alert
import alert from 'framework/components/alert/themedAlert'

// verzia komponentu alert bez CSS štýlov - neotémovaná, CSS je nutné rune naimportova
import alert from 'framework/components/alert/alert'
```

Dialógové okná (framework/components/alert)

Alert

Komponent *Alert* slúži na zobrazovanie dialógových okien. Komponent používa ako zdrojovú knižnicu [SweetAlert2](#) a jeho rozhranie je zhodné s rozhraním knižnice SweetAlert2. Komponent má aj našťýlovanú verziu s preddefinovanými CSS štýlmi.

```
// otémovaná verzia komponentu alert
import alert from 'framework/components/alert/themedAlert'
// verzia komponentu alert bez CSS štýlov - neotémovaná, CSS je nutné rune naimportova
// import alert from 'framework/components/alert/alert'

// príklad použitia
alert({title: 'Úspech', text: 'Podarilo sa to!', type: 'success'});
```

Autentifikované komponenty (framework/components/auth)

LoginForm

Komponent *LoginForm* predstavuje formulár, ktorým sa užívateľ prihlasuje do webovej aplikácie resp. systému D2000. Pre použitie *LoginForm* komponentu je v ňom potrebné zadať textové pole (html element `<input>`) s názvom `"j_username"` pre meno užívateľa, textové pole s názvom `"j_password"` pre heslo užívateľa a odosielať tlačidlo (element `<button type="submit">`). Samotná akcia prihlásenia je implementovaná na úrovni SmartWeb-u a po úspešnej autentifikácii je užívateľ presmerovaný na stránku *pages/index.html*.

```
import React from 'react'
import SimplePageContainer from 'framework/components/react/simplePageContainer'
import LoginForm from 'framework/components/auth/loginForm'

// Jednoduchý príklad použitia LoginForm komponentu
SimplePageContainer.renderComponent(class LoginPage extends React.Component {
  render() {
    return (
      <LoginForm {...this.props}>
        <input type="text" placeholder="Používateľské meno" required="" name="j_username"/>
        <input type="password" placeholder="Heslo" name="j_password"/>
        <button type="submit">Prihláši</button>
      </LoginForm>
    );
  }
});
```

LogoutLink

Komponent *LogoutLink* je jednoduchým odkazom (html element `<a>`), na kliknutie ktorého sa zobrazí potvrdzovací dialóg odhlásenia užívateľa. Po potvrdení je užívateľ z aplikácie odhlásený a presmerovaný na úvodnú (prihlasovaciu) stránku.

```
import React from 'react';
import LogoutLink from "framework/components/auth/logoutLink";

// Príklad použitia komponentu LogoutLink ako jednej z položiek menu
export default class PageHeader extends React.Component {
  render() {
    return (
      <div className="container-fluid">
        <div className="row">
          <div className="col-xs-4">
            <a href="#section1">Section 1</a>
          </div>
          <div className="col-xs-4">
            <a href="#section2">Section 2</a>
          </div>
          <div className="col-xs-4">
            <a href="#section3">Section 3</a>
          </div>
          <div className="col-xs-4">
            <LogoutLink>Logout</LogoutLink>
          </div>
        </div>
      </div>
    );
  }
};
```

Komponenty pre komunikáciu s D2000 (framework/components/d2)

DataContainer

Základný komponent SmartWeb Frameworku. Potomok komponentu *SimplePageContainer*, ktorý v sebe navyš zabauje rozhranie [D2Api](#) a automaticky nadväzuje a ukonuje pripojenie k D2000. Zároveň zabezpečuje podporu pre klávesové skratky na prepínanie režimu ladenia (CTRL+ALT+D) a rebundovania aplikácie (CTRL+ALT+R). Do ostatných komponentov sa propaguje ako atribút s názvom *d2*.

Názov atribútu	Typ	Povinný	Predvolená hodnota	Popis
connection	object			Obsahuje informácie o pripojení k D2000 ako názov aplikácie a meno prihláseného užívateľa. Atribút je nastavovaný automaticky.

TclScheme

Tento komponent slúži na zobrazenie D2000 schémy pomocou technológie tenkého klienta (TCL). Schéma bude na stránku vložená ako HTML element *<iframe>*. Atribúty komponentu sú:

Názov atribútu	Typ	Povinný	Predvolená hodnota	Popis
d2	object	áno		Odkaz na komponent DataContainer. Odovzdáva sa z rodičovského kontajneru.
datasource	string	áno		Názov D2000 objektu typu schéma, ktorá má byť zobrazená.

ValueComponent

Komponent zobrazujúci aktuálnu hodnotu D2000 objektu. Hodnota je zobrazovaná v HTML elemente **. Nasledujúca tabuľka popisuje atribúty komponentu:

Názov atribútu	Typ	Povinný	Predvolená hodnota	Popis
d2	object	áno		Odkaz na komponent DataContainer. Odovzdáva sa z rodičovského kontajneru.
datasource	string	áno		Názov D2000 objektu, ktorého hodnota má byť zobrazovaná.
returnFields	string[]			Zoznam atribútov hodnoty, ktoré majú byť spolu s hodnotou objektu zaslané. Možné atribúty sú popísané na stránke Serializácia dát medzi klientom a API rozhraniami .
description	string			Manuálne definovaný popis objektu - má prednosť pred popisom definovaným v D2000.

technicalUnits	string			Manuálne definované technické jednotku objektu - majú prednos pred definovanými v D2000.
numberFormat	string			Formát pre íselné hodnoty objektu v tvare pre knižnicu Numeral.js . Ak je definovaný, má prednos pred formátovanou hodnotou zaslanou z D2000.
dateFormat	string			Formát pre hodnoty typu absolútny as v tvare pre knižnicu Moment.js . Ak je definovaný, má prednos pred formátovanou hodnotou zaslanou z D2000.
missingDescription	string			Popis, ktorý sa použije v prípade, že nebol definovaný manuálne, ani v D2000.
missingFormattedValue	string			Formátovaná hodnota, ktorá sa použije v prípade, že objekt nemá platnú hodnotu.
missingTechnicalUnits	string			Technické jednotky, ktoré sa použijú v prípade, že neboli definované manuálne, ani v D2000.
className	string			Názov CSS triedy, ktorá bude aplikovaná na HTML element <code></code> zobrazujúci hodnotu objektu.
style	object			CSS štýl, ktorý bude aplikovaný na HTML element <code></code> zobrazujúci hodnotu objektu.

Komponenty dátumovníka (framework/components/datetime)

BootstrapDatePicker

Komponent výberovníka dátumu. Založený je na komponente [Bootstrap Datepicker](#), ktorého atribúty rozširuje o nasledovné:

Názov atribútu	Typ	Povinný	Predvolená hodnota	Popis
name	string			Unikátne meno pre HTML element <code><input></code> .
className	string			Názov CSS triedy, ktorá bude aplikovaná na HTML element <code><input></code> zobrazujúci vybraný dátum.
defaultValue	string date			Predvolená hodnota. Môže by zadaná ako text alebo ako objekt date.
disabled	bool		false	Zakáže alebo povolí modifikáciu.
enableKeyboardEdit	bool		true	Povolí alebo zakáže modifikáciu pomocou kláves.

Príklad použitia

```
// otémovaná verzia komponentu alert
import DatePicker from 'framework/components/datetime/themedBootstrapDatePicker'
// verzia komponentu bez CSS štýlov - neotémovaná, CSS je nutné rune naimportova
// import DatePicker from 'framework/components/datetime/bootstrapDatePicker'

// príklad použitia v JSX
<div className="row">
  <div className="col-md-2">
    <DatePicker defaultValue={new Date()} />
  </div>
</div>
```

BootstrapDateRangePicker

Komponent *BootstrapDateRangePicker* je rozšírením dátumovníka. Zobrazuje dva dátumy, ktorými sa definuje interval. Atribúty zabaleného komponentu [Bootstrap Datepicker](#) rozširuje o nasledovné:

Názov atribútu	Typ	Povinný	Predvolená hodnota	Popis
name	string			Unikátne meno pre HTML element <code><input></code> .
className	string			Názov CSS triedy, ktorá bude aplikovaná na HTML element <code><input></code> zobrazujúci vybraný dátum.
defaultFromValue	string date			Predvolená hodnota zaiatku intervalu. Môže by zadaná ako text alebo ako objekt date.
defaultToValue	string date			Predvolená hodnota konca intervalu. Môže by zadaná ako text alebo ako objekt date.
betweenDatesText	string		"do"	Text zobrazený medzi dátumami.
disabled	bool		false	Zakáže alebo povolí modifikáciu.

enableKeyboardEdit	bool		true	Povolí alebo zakáže modifikáciu pomocou kláves.
--------------------	------	--	------	---

Komponenty zoznamov (framework/components/list)

LazyList

Komponent *LazyList* implementuje zoznam, ktorý podporuje postupné načítavanie obsahu pri scrolovaní smerom nadol. Vo svojej základnej verzii zobrazuje obsah pomocou elementov `<div>`. Pre finálne použitie sa predpokladá preaženie metód *renderItem* a *render*, ktoré majú na starosti vykresovanie obsahu zoznamu. KomponentKomponent rozpoznáva nasledovné atribúty:

Názov atribútu	Typ	Povinný	Predvolená hodnota	Popis
loadHandler	function		function (size, callback) { callback([]); }	Funkcia, ktorá je volaná pri požiadavke na načítanie ďalších dát. Parameter <i>size</i> hovorí, maximálne koľko prvkov má byť načítaných a parameter <i>callback</i> je funkcia, ktorou budú načítané dáta doplnené na koniec zoznamu.
batchSize	number		10	Predvolený počet prvkov, ktorý je načítavaný v jednej dávke. Odovzdáva sa ako parameter do funkcie <i>loadHandler</i> .

Základné komponenty rozširujúce React (framework/components/react)

ConfigurableComponent

Tento komponent slúži ako wrapper pre komponenty, ktoré nie sú natívnymi React komponentami. Vytvorením potomka tohto komponentu a preažením metód *configureComponent*, *render* a *wrappedInstance* je možné jednoducho pridať podporu zobrazovania pomocou React frameworku.

RootElement

Komponent, ktorý nahrádza HTML element s `id = "root"` za zvolený React komponent. Tento komponent má len interné využitie.

SimplePageContainer

Základný kontajner web stránky. Zabezpečuje podporu pre klávesové skratky na prepínanie režimu ladenia (CTRL+ALT+D) a rebundovania aplikácie (CTRL+ALT+R). Jeho využitie je pre React stránky, ktoré nepotrebujú pripojenie k D2000. Na tvorbu web stránok, ktoré požadujú pripojenie k D2000 slúži potomok tohto komponentu *DataContainer*.

Tabuľkový komponent (framework/components/table)

DataTable a ThemedDataTable

Tabuľkový komponent *Table* v sebe zabauje knižnicu [DataTables](#). Rozhranie tejto knižnice rozširuje o nasledovné atribúty:

Názov atribútu	Typ	Povinný	Predvolená hodnota	Popis
className	string			Názov CSS triedy, ktorá bude aplikovaná na hlavný HTML element zobrazujúci tabuľku.
style	object			CSS štýl, ktorý bude aplikovaný na hlavný HTML element zobrazujúci tabuľku.
width	string			Šírka tabuľky.
height	string			Výška tabuľky.
renderFooter	bool		false	Príznak i názvy stpcov budú zobrazené aj v päťke tabuľky.

```

import React from 'react'
import {DataContainer} from '../custom/components'
import Table from 'framework/components/table/themedDataTable'
// verzia komponentu bez CSS štýlov - neotémovaná, CSS je nutné rune naimportova
// import Table from 'framework/components/table/dataTable'

// Príklad použitia komponentu ThemedDataTable, ktorý naíťavá dáta volaním RPC
DataContainer.renderComponent(class IndexPage extends React.Component {

  async loadData(data, callback, settings) {
    const d2Api = this.props.d2;
    const response = await d2Api.rpc("E.SmartWebApiTutorial", "GetPersonData", 1000, {
      type: 'record',
      structType: 'SD.Arr_Person',
      returnAs: 'vysledok',
      value: []
    }).call();

    callback({
      data: response.vysledok,
      recordsTotal: response.vysledok.length
    });
  }

  render() {
    return (
      <div>
        <Table className="table table-striped table-bordered dt-responsive nowrap dataTable" width="
100%"
          ajax = {(data, callback, settings) => {
            this.loadData(data, callback, settings);
          }}
          columns = {[
            {title: "Meno", data: "meno.value"},
            {title: "Vek", data: "vek.value"}
          ]}
          select={true}
          ref = {(component) => {
            this.table = component;
          }}/>
      </div>
    );
  }
});

```