

D2 Cloud

Predstavenie služby D2 Cloud

Správa a údržba webovej aplikácie a jej infraštruktúry zahŕňa aj inštaláciu a požiadavky na odbornosť vo sfére, ktorá nie je primárna pre priemyselnú firmu. Z tohto dôvodu je tu služba D2 Cloud, ktorá umožňuje priemyselným spoločnostiam jednoduchým a bezpečným spôsobom sprístupniť dáta a služby podnikovej aplikácie nasadenej na platforme D2000. Princíp fungovania služby je postavený na web moduloch platformy D2000 - Tenký Klient alebo Smart Web, ktoré sú prevádzkované na infraštruktúre (cloud) spoločnosti Ipesoft a prepojené s D2000 nasadenou u zákazníka. Zákazníkovi tak odpadá povinnosť spravovať infraštruktúru webovej aplikácie a prenecháva túto úlohu na spoločnosti Ipesoft. Spojenie modulov Tenký Klient alebo Smart Web s D2000 aplikáciou je realizované cez šifrované spojenie s obojstrannou autentifikáciou.

[blocked URL](#)

Výhody služby D2Cloud

Bezpečné a jednoduché publikovanie podnikových dát z D2000.

Používate systém D2000 nemusíte riešiť nasadenie a správu rozhrania s okolitým svetom, ale túto záležitosť prenecháte spoločnosti, ktorá má s touto správou skúsenosti. Zákazník/procesný inžinier/technolog/obchodník je tak odbremený od povinností, ktoré priamo nesúvisia s jeho podnikaním a zároveň môže spokojne využívať výhody a možnosti zabezpečeného online prístupu k real-time dátam, alarmom, procesným a obchodným informáciám.

Flexibilita nasadenia a dodatočné služby

Možnosti nasadenia a customizácie nasadenia modulov Smart Web a Tenký Klient sú veľmi široké. Zákazník má možnosť vybrať nasadenie jedného alebo oboch modulov, podľa potrieb svojej spoločnosti. Samozrejmosťou je možnosť sprístupniť web aplikácie týchto modulov napríklad na vlastnej firemnej doméne, zapnúť dodatočnú autentifikáciu používateľov cez klientské certifikáty alebo poveriť spoločnosť Ipesoft vývojom webových stránok, alebo funkcií na mieru svojím požiadavkám.

Šetrenie nákladov a jednoduchosť nasadenia

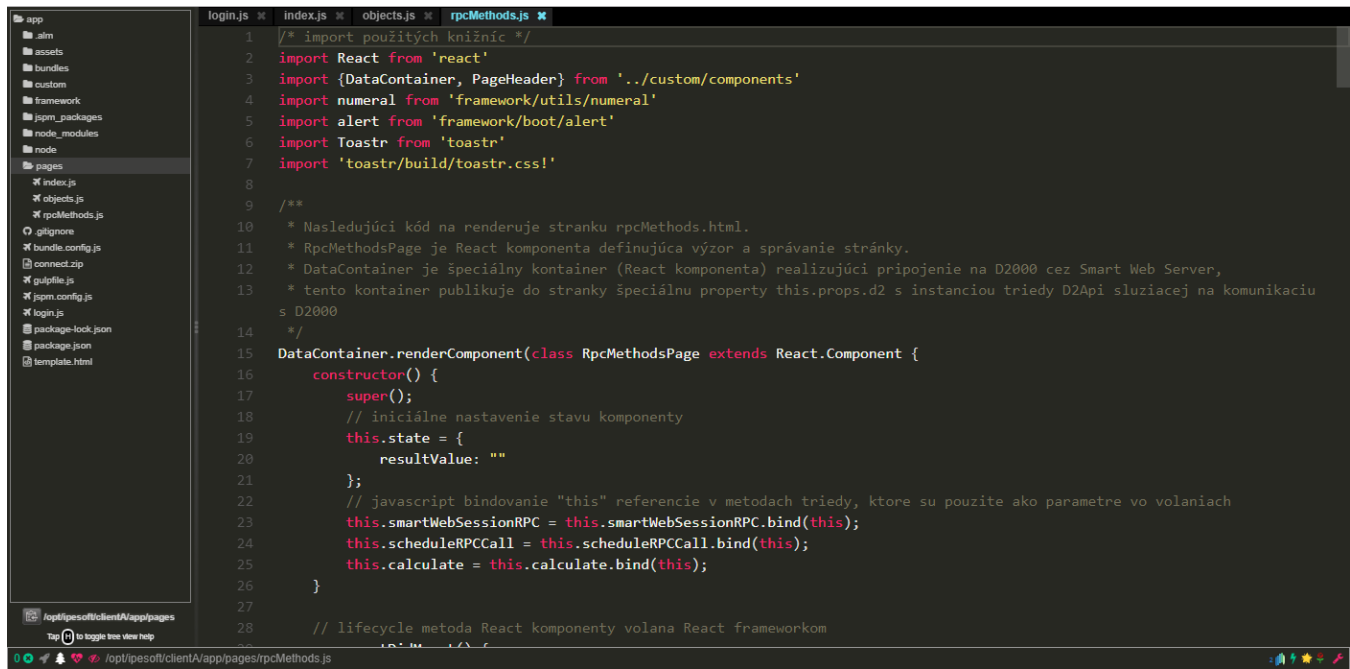
Na strane zákazníka nie je potrebné spravovať, konfigurovať a zabezpečovať infraštruktúru pre webovú aplikáciu. Zákazník taktiež nepotrebuje mať úzko špecializované odborné znalosti z oblasti webovej bezpečnosti a komunikácie. Konfigurácia spojenia D2000 so službou D2 Cloud je pre zákazníka veľmi jednoduchá, rýchla a zároveň bezpečná.

Postup pre využívanie služby D2 Cloud

1. Služba D2 Cloud je podporovaná od verzie D2000 V11.2. Pre používanie služby D2Cloud je nevyhnutné sa najskôr **zaregistrovať prostredníctvom formulára na portáli d2cloud.ipesoft.com**. V registrovanom formulári klient vyplní okrem kontaktných údajov aj informáciu o registrácii ktorého modulu (Smart WEB, Tenký klient - resp. obidva) má záujem.
2. Po odoslaní formulára bude následne **klienta kontaktovať poverený pracovník Ipesoftu**. Ten si vyžiada ašie upresňujúce informácie, potrebné pre zaregistrovanie požadovanej služby, ako napr. názov domény (url), pod akou sa bude služba publikovať. Zákazník si môže tento názov zvoliť sám pokiaľ má vlastnú doménu na ktorej chce prevádzkovať webovú aplikáciu. K dispozícii zdarma je prístup do aplikácie cez url: <https://d2cloud.ipesoft.com/<identifikátor aplikácie>>.
3. Po dohode povereného pracovníka Ipesoft a klienta, bude **v D2 Cloude vytvorená inštancia modulu Tenký Klient alebo Smart Web**.
4. O vytvorení príslušnej inštancie web aplikácie v D2 Cloude, **bude zákazník upovedomený mailom**, v ktorom mu budú zaslané dodatočné informácie o spôsobe ako prepojiť vnútropodnikovú D2000 s D2 Cloudom cez link na stiahnutie zašifrovaného balíka `connect.zip` s inštalátorom nastavujúcim šifrované spojenie medzi D2000 a D2 Cloudom.
5. Zákazník nainštaluje šifrovanú komunikáciu D2000 aplikácie so službou D2 Cloud. Postup inštalácie je popísaný samostatne na stránke [Nastavenie prepojenia D2000 aplikácie so službou D2 Cloud](#).
6. Ak je všetko nastavené správne, môžeme sa prístupovým menom a heslom do D2000 prihlásiť do vzorovej Smart Web aplikácie alebo Tenkého klienta na D2 Cloude.

Úprava zdrojových kódov web aplikácie pre Smart Web

Vytváranie a úprava zdrojových kódov webovej aplikácie postavenej na platforme Smart Web je možné aj použitím webového online editora [Alm Tools](#), ktorý je implicitne dostupný zákazníkovi D2 Cloud služby. Spôsob používania tohto online editora je popísaný v dokumentácii na stránke: <https://alm-tools.gitbooks.io/alm/content/>. Zobrazenie panelu so stromom súborov a adresárov sa realizuje cez skratku Ctrl+ \.



The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with folders like 'app', 'alm', 'assets', 'bundles', 'custom', 'framework', 'jspm_packages', 'node_modules', 'node', and 'pages'. The 'pages' folder is expanded, showing files like 'index.js', 'objects.js', 'rpcMethods.js', 'gitignore', 'bundle.config.js', 'connect.zip', 'gulpfile.js', 'jspm.config.js', 'login.js', 'package-lock.json', 'package.json', and 'template.html'. The 'rpcMethods.js' file is selected and its content is displayed in the code editor. The code is written in JavaScript and uses the React framework. It includes imports for 'React', 'DataContainer', 'PageHeader', 'numeral', 'alert', and 'Toastr'. The main component is 'RpcMethodsPage', which extends 'React.Component'. It has a 'constructor' method that initializes the state with 'resultValue' and binds several methods to 'this'. The 'render' method is not shown, but it is implied to render the 'PageHeader' component. The code is commented in Slovak, explaining the purpose of the component and the container.

```
1  /* import použitých knižnic */
2  import React from 'react'
3  import {DataContainer, PageHeader} from '../custom/components'
4  import numeral from 'framework/utils/numeral'
5  import alert from 'framework/boot/alert'
6  import Toastr from 'toastr'
7  import 'toastr/build/toastr.css!'
8
9  /**
10   * Nasledujúci kód na renderuje stránku rpcMethods.html.
11   * RpcMethodsPage je React komponenta definujúca výzor a správanie stránky.
12   * DataContainer je špeciálny kontajner (React komponenta) realizujúci pripojenie na D2000 cez Smart Web Server,
13   * tento kontajner publikuje do stránky špeciálnu property this.props.d2 s instanciou triedy D2Api slúžacej na komunikáciu
14   * s D2000
15   */
16   DataContainer.renderComponent(class RpcMethodsPage extends React.Component {
17     constructor() {
18       super();
19       // iníciaľne nastavenie stavu komponenty
20       this.state = {
21         resultValue: ""
22       };
23       // javascript bindovanie "this" referencie v metodach triedy, ktore su pouzite ako parametre vo volaniach
24       this.smartWebSessionRPC = this.smartWebSessionRPC.bind(this);
25       this.schedulerRPC = this.schedulerRPC.bind(this);
26       this.calculate = this.calculate.bind(this);
27     }
28   })
29
30   // lifecycle metoda React komponenty volana React frameworkom
```