

Univerzálne API rozhranie

Koncept univerzálneho API bol predstavený v kapitole [Univerzálne API pre prístup do D2000](#). Skôr než budú popísané jednotlivé rozhrania REST a Comet API je potrebné v nasledujúcej kapitole popísa spôsob serializácie hodnôt a parametrov, ktorý je pri oboch rozhraniach totožný.

Použiteľnosť jednotlivých API rozhraní

Na otázku prečo SmartWeb implementuje dve a nie iba jedno API rozhranie, treba mať odpoveď vo vhodnosti použitia jednotlivých API pre rôzne situácie.

D2000 je realtime SCADA systém, ktorý je schopný zaznamenáva zmeny veľkého počtu napr. meraných bodov. Na to aby o týchto zmenách bola notifikovaná aj webová, prípadne iná aplikácia, je potrebné implementovať komunikáciu typu [server-push](#), kedy server sám notifikuje klienta o zmene hodnoty objektu. V opačnom prípade je nevyhnutné, aby sa klient periodicky v nejakom časovom intervale pýtal servera či nastali zmeny hodnôt ním sledovaných objektov. Tento spôsob komunikácie je ale veľmi nevýhodný. Pri dlhšom časovom intervale môže nastať situácia, že hodnota objektu sa zmení tesne po odpovedi servera a klient nie je informovaný o tejto hodnote, až kým znova neskontroluje server. Na druhej strane ak je časový interval príliš krátky a hodnota sa mení sporadicky, môže dochádzať úplne zbytočne k známemu plytvaniu komunikacími prostriedkami. Z týchto dôvodov sú situácie kedy komunikáciu podporujúcu server-push naozaj využijeme. Spôsob server-push komunikácie ale nie je štandardizovaný, a preto bolo potrebné vybrať jednu z existujúcich technológií, ktoré boli k dispozícii. Ideálne takú ktorá bude podporovať robustnú server-push komunikáciu medzi webovou aplikáciou v prehliadači a serverom. Po analýze možností bola vybraná technológia postavená na [komunikácii konceptu Comet](#) implementovanom v robustnej Java knižnici [Cometd](#), s dlhou produkčnou históriou a [použitím v cloudovom riešení s viac ako 150 000 klientmi](#).

SmartWeb implementuje komunikáciu konceptu Comet predovšetkým kvôli získavaniu aktuálnych hodnôt a možnosti [volať z D2000 aj vzdialené RPC metódy](#) zaregistrované a vykonávané na klientovi. Keďže tieto prípady použitia nie sú vždy pre aplikáciu požadované, je možné použiť implementáciu jednoduchšie a štandardizované rozhranie REST API, ktoré je podmnožinou Comet API.

Autentifikácia v jednotlivých API rozhraniach

Ako už bolo spomenuté v kapitole [alšie funkcie SmartWeb platformy](#), spôsob autentifikácie jednotlivých rozhraní je rozdielny. REST API so svojou [HTTP-BASIC](#) autentifikáciou je skôr určené na komunikáciu pre ne-webových klientov, Comet API neautentifikovaného používateľa presmeruje na prihlasovací formulár (FORM autentifikácia) a preto je toto rozhranie momentálne prirodzene vhodné na komunikáciu webových aplikácií so serverom.



V prípade potreby, vie cez Comet API komunikovať aj iný klient ako web aplikácia. Jedinou požiadavkou je aby bol schopný sa autentifikovať cez poslanie špeciálnej HTTP požiadavky emulujúcej prihlásenie cez prihlasovací formulár. Alternatívnu ale komplikovanejšou možnosťou je zobraziť priamo prihlasovací formulár používateľovi v špeciálnom okne s vnoreným prehliadačom a po úspešnom prihlásení extrahovať autentifikované cookie pre ďalšie použitie v Comet API.