

# Identifikácia používateľa v RPC metóde

asto krát je potrebné vrátiť z RPC metódy iba tie dáta na ktoré má používateľ volajúci RPC metódu oprávnenie. Posielanie `userName` aktuálneho používateľa ako parametra RPC metódy nie je dostatočný spôsob zabezpečenia, pretože architektúra Smart Webu nevie zabrániť prihlásenému používateľovi zavolať povolenú RPC metódu s akýmikoľvek parametrami. To znamená, že používateľ by sa vedel identifikovať cez parameter `userName` ako niekto úplne iný. Z tohto dôvodu je potrebné identifikovať volaného používateľa spôsobmi popísanými nižšie.

Smart Web podporuje autentifikáciu cez D2000 používateľov alebo prostredníctvom aplikácie definovaných používateľov, kedy Smart Web server realizuje autentifikáciu cez volanie definovanej autentifikacej RPC metódy.

## Identifikácia D2000 používateľa v RPC metóde

V prípade, že Smart Web aplikácia realizuje autentifikáciu cez D2000 používateľov, je možné v akejkoľvek RPC metóde zavolať funkciu [GetUserObjName](#), ktorá vráti `userName` aktuálne prihláseného používateľa.

## Identifikácia aplikácie definovaného používateľa v RPC metóde

V prípade aplikácie definovaných používateľov je postup zložitejší a vyžaduje si vytvorenie kontajnera dát cez metódu [CNT\\_CREATE](#), kde kúrom k hodnotám kontajnera bude HOBJ prihlásenej session a hodnotou kúru `userName`. Kontajner sa bude plniť volaním [CNT\\_INSERT](#) v tzv. logon metóde, ktorú bude Smart Web volať pri každom úspešnom prihlásení. V RPC metóde je potom získanie používateľského mena realizované cez volanie [CNT\\_GETITEM](#). Vzorová implementácia tejto funkcionality je uvedená nižšie. RPC metódy, ktoré potrebujú identifikovať prihláseného používateľa, stať len zavolať metódu `getCurrentUserName`.

```

INT _CNT_sessions ;kontajner vsetkych aktivnych session vsetkych pouzivatelov

; metoda je zaregistrovana v Smart Web aplikacii ako "logOn" metoda,
; ktora sa vola po kazdej uspesnej autentifikacii
RPC PROCEDURE logOn(IN TEXT _UserName, BOOL _Ok)
    INT _hobj_session
    BOOL _bFound
    _Ok := @TRUE

    _hobj_session := %GetRPCCallerProcess()
    ; kontrola existencie session, nesmie existovat, opacny stav je chyba
    CNT_FIND _CNT_sessions, _hobj_session, _UserName, _bFound
    IF _bFound THEN
        LOGEX "ERR: uz existuje rovnaka SESSION pouzivателя: " + %IToStr(_hobj_session) PRIORITY _LOG_PRTY_ERROR
        _Ok := @FALSE
        RETURN
    ENDIF
    ; registracia volanie metody onSessionClosed pri "zmene session",
    ; t.j. bud standardne odhlasenie alebo akekolvek ine ukoncenie session
    _Ok := %OpenRefToObject(_hobj_session, @TRUE)
    IF !_Ok THEN
        LOGEX "ERR: neznama SESSION pre otvorenie: " + %IToStr(_hobj_session) PRIORITY _LOG_PRTY_ERROR
        RETURN
    ENDIF
    ON CHANGE (_hobj_session) GOTO onSessionClosed
    ; vlozenie mapovania "session HOBJ -> user name" do kontainera
    CNT_INSERT _CNT_sessions, _hobj_session, _UserName
END logOn

; zrusena session pouzivately (odhlasenim alebo inym dovodom zrusenia session pouzivately)
PROCEDURE onSessionClosed(IN INT _procValue, IN ALIAS _hobj, IN INT _row, _col)
    BOOL _bFound
    INT _hobj_session
    TEXT _userName
    _hobj_session := _hobj\HBJ

    CNT_FIND _CNT_sessions, _hobj_session, _userName, _bFound
    IF !_bFound THEN
        LOGEX "ERR: neznama SESSION bola ukoncena " + %IToStr(_hobj_session) PRIORITY _LOG_PRTY_ERROR
        RETURN
    ENDIF
    CNT_DELETE _CNT_sessions, _hobj_session
    %CloseRefToObject(_hobj_session)
    ON CHANGE (_hobj_session) GOTO onSessionClosed NONE
END onSessionClosed

; RPC metoda vrati meno aktualne prihlaseného používateľa
RPC PROCEDURE getCurrentUserName(TEXT _userName)
    INT _hobj_session
    BOOL _bFound
    _hobj_session := %GetRPCCallerProcess()
    CNT_FIND _CNT_sessions, _hobj_session, _userName, _bFound
    IF !_bFound THEN
        LOGEX "ERR: neznama SESSION bola ukoncena " + %IToStr(_hobj_session) PRIORITY _LOG_PRTY_ERROR
        RETURN
    ENDIF
END getCurrentUserName

BEGIN
    CNT_CREATE _CNT_sessions
END

```