

Dica técnica 13 - Sincronização de usuários via API

(01 nov 2023)

Função: permitir incluir, alterar e desabilitar usuários da versão 2 do SuperBI via API. Para isso, primeiro é necessário enviar uma requisição para a API de autenticação, para receber o token de acesso temporário; em seguida, uma nova requisição (agora para a API base) enviando os dados dos usuários.

Passo 1 – autenticação

Enviar uma requisição POST para a API de autenticação do SuperBI, com o sufixo `/authentication/logintoken` (p. ex.: <https://apiauthentication.bxbsoft.com/authentication/logintoken>), passando um JSON como parâmetro, contendo `company`, `username` e `password` (a senha deve ser em lower case e criptografada com algoritmo de hash MD5). O servidor responde com um JSON, com `result` (`true` ou `false`), `token` e `message` (se `result` for `false`, `message` contém a mensagem de erro gerada). O token gerado é válido por 5 minutos, e para uma chamada apenas.

É necessário realizar a autenticação com o usuário `Master` ou com usuário `Admin`. As mesmas regras de edição de usuários no SuperBI são aplicadas, ou seja, autenticando com o usuário `Master` é possível incluir/alterar qualquer usuário, e autenticando com usuário `Admin` não é possível incluir/alterar outros usuários `Admin`.

Ex. JSON enviado:

```
{
  "company": "BXBcompany",
  "username": "john",
  "password": " e10adc3949ba59abbe56e057f20f883e"
}
```

Ex. JSON recebido:

```
{
  "result": true,
  "message": "",
  "token": "e0c45d6d-ba92-4e48-b818-0e1b60fa002c"
}
```

Passo 2 – atualizando usuários

Enviar uma requisição POST para a API base do SuperBI, com o sufixo `/user/sync?token=TOKEN_RECEBIDO` (p. ex.: <https://apibase.bxbsoft.com/user/sync?token=e0c45d6d-ba92-4e48-b818-0e1b60fa002c>), passando um JSON como parâmetro (estrutura abaixo). O servidor responde com um JSON, com `result` (`true` ou `false`), `message` (se `result` for `false`, `message` contém a mensagem de erro gerada), `added`, `updated` e `disabled` com a quantidade de registros adicionados, atualizados e desabilitados.

Ex. JSON enviado:

```
{
```

```
"disable_others": false,
"users": [
  {
    "login": "olivia",
    "password": "af3fc89f7a449a8229ed76a8966fe6ca",
    "full_name": "Olivia Fox",
    "email": "olivia@company.com",
    "profile": "Sales",
    "license": "Viewer",
    "language": "en",
    "decimal_separator": ".",
    "initial_module": "Panels",
    "interval_skip_panels": 0,
    "lines_view": 20,
    "enable_user_config": true,
    "active": true
  },
  {
    "login": "john",
    "full_name": "John Smith",
    "email": "john@company.com",
    "profile": "Marketing",
    "license": "Personal admin",
    "language": "pt",
    "decimal_separator": ",",
    "initial_module": "Panels",
    "interval_skip_panels": 0,
    "lines_view": 20,
    "active": true
  },
  {
    "login": "robert",
    "active": false
  }
]
}
```

Propriedades:

Disable others: *True* para desabilitar outros usuários que não foram enviados na lista, exceto o usuário utilizado na autenticação e o usuário Master. Opcional. Se não enviado, assume *false*.

User - Login: Propriedade *Código (Login)* do usuário. Obrigatório.

User - Password: Propriedade *Senha* do usuário. Obrigatório para inclusão, opcional para edição. Regra: deve ser criptografada em MD5.

User - Full name: Propriedade *Nome completo* do usuário. Obrigatório para inclusão, opcional para edição.

User - Email: Propriedade *Email* do usuário. Obrigatório para inclusão, opcional para edição. Regra: deve ser um e-mail válido.

User - Profile: Propriedade *Perfil* do usuário. Obrigatório para inclusão, opcional para edição. Regra: deve ser o nome de um perfil cadastrado no SuperBI.

User - License: Propriedade *Edição da licença* do usuário. Obrigatório para inclusão, opcional para edição. Regra: deve ser uma das opções: *Professional, Professional admin, Personal, Personal admin, Viewer, Viewer admin* ou *Admin*.

User - Language: Propriedade *Idioma* do usuário. Obrigatório para inclusão, opcional para edição. Regra: deve ser uma das opções: *Def, Default, En, English, Pt* ou *Portuguese*.

User - Decimal separator: Propriedade *Separador decimal* do usuário. Obrigatório para inclusão, opcional para edição. Regra: deve ser uma das opções: *","* ou *."*.

User - Initial module: Propriedade *Módulo inicial* do usuário. Obrigatório para inclusão, opcional para edição. Regra: deve ser uma das opções: *Panels, Database* ou *Scheduler load*.

User - Interval skip panels: Propriedade *Intervalo para pular painéis (em segundos)* do usuário. Obrigatório para inclusão, opcional para edição.

User - Lines view: Propriedade *Num. linhas p/paginar a visão* do usuário. Obrigatório para inclusão, opcional para edição.

User - Enable user config: Propriedade *Habilitar 'Configurações do usuário'* do usuário. Obrigatório para inclusão, opcional para edição.

User - Active: Propriedade *Ativo* do usuário. Obrigatório para inclusão, opcional para edição.

Ex. JSON recebido:

```
{
  "result": true,
  "message": "",
  "added": 1,
  "updated": 2,
  "disabled": 1
}
```

Propriedades:

Result: *true* se operação executada com sucesso, *false* em caso de erro.

Message: em caso de erro retorna mensagem informativa.

Added: quantidade de usuários adicionados.

Updated: quantidade de usuários atualizados.

Disabled: quantidade de usuários desabilitados (usuários desabilitados via propriedade *disable_others* ou via lista *users*, alterando a propriedade *active*).

Opcional – validando login

É possível validar se um login é válido. Para isso, basta enviar uma requisição POST para a API de autenticação do SuperBI, com o sufixo */authentication/loginvalidation* (p. ex.: <https://apiauthentication.bxbsoft.com/authentication/loginvalidation>), passando um JSON como parâmetro, contendo *company*, *username* e *password* (a senha deve ser em lower case e criptografada com algoritmo de hash MD5). O servidor responde com um JSON, com *company_exists* (*false* se a empresa não existir), *user_exists* (*false* se o usuário não existir na empresa) e *password_check* (*false* se a senha estiver incorreta).

Ex. JSON enviado:

```
{
  "company": "BXBcompany",
  "username": "john",
  "password": "e10adc3949ba59abbe56e057f20f883e"
}
```

Ex. JSON recebido:

```
{
  "company_exists": true,
  "user_exists": true,

```

```
"password_check": true  
}
```