Leonardo Tadashi Kamaura

DATA Group

WEB ACCESS CONTROL

INTRODUÇÃO

O sistema Web Access Control é uma camada web para a biblioteca Access Control, que provê os meios para implementar um controle de acesso integrado entre sistemas. Ele é desenvolvido na linguagem Java usando o arcabouço Struts 2.

INTEGRAÇÃO DO CONTROLE DE ACESSO

A integração do controle de acesso entre sistemas é feita por meio de um web service baseado na arquitetura REST (Representational State Transfer). As entidades do modelo de controle de acesso são disponibilizados como recursos *RESTful*, permitindo a realização de operações CRUD, dentro de um ambiente integrado, por meio de uma API Web. Assim, diferentes sistemas instalados dentro de um ambiente compartilhado conseguem utilizar o mesmo sistema de controle de acesso.

DOCUMENTAÇÃO DA API:

A utilização da API se resume a realizar chamadas HTTP para uma URL do sistema Web Access Control que corresponde a uma operação que se deseja realizar. O sistema Web Access Control, por sua vez, ao receber a requisição, executa a operação solicitada e devolve um objeto JSON contendo o resultado.
As operações que podem ser realizadas são:

- Autenticar (non-RESTful):

URL: http://localhost:8080/webaccesscontrol/action/authenticate
Método: POST ou GET
Parâmetros:

```
   username          {login do usuário}
   password          {senha}
Saída: JSON
```

```
{
 "action":"authenticate",
 "returnCode":<0 para erro | 1 para succeso>,
 "message":"<mensagem relativa a ação executada>",
 "result":{
        "username":"<login do usuário>",
        "groups":[
                {
                        "group_id":<id de um perfil>
                },
                {
                        "group_id":<id de outro perfil>
                }
        ]
  }
}
```

Exemplo com cURL:

```
curl -i -H "Accept: application/json" -X POST -d "username=meulogin&password=1234"
http://localhost:8080/webaccesscontrol/action/authenticate
```

- Listar usuários (RESTful):

(Listar todos os usuários)
URL: http://localhost:8080/webaccesscontrol/users.json
Método: GET
Saída: JSON

```
[
     {
        "groups":[
                {
                        "group_id":<id de um perfil>,
                        "description":<nome de um perfil>
                },
                {
                        "group_id":<id de outro perfil>,
                        "description":<nome de outro perfil>
                },
                ...
        ],
        "id":<id do usuário>,
        "newPassword":"",
        "password":"<senha criptografada do usuário>",
        "username":"<login do usuário>"
       },
       {
        "groups":[],
        "id":<id de outro usuário>,
        "newPassword":"",
        "password":"<senha criptografada do outro usuário>",
        "username":"<login de outro usuário>"
       }, ...
]
```

Exemplo com cURL:

```
curl -i -H "Accept: application/json" -X GET
http://localhost:8080/webaccesscontrol/users.json
```

(Listar usuário com id = {id})
URL: http://localhost:8080/webaccesscontrol/users/{id}.json
Método: GET
Saída: JSON

```
{
  "groups":[
        {
              "group_id":<id de um perfil>,
              "description":<nome de um perfil>
        },
        {
              "group_id":<id de outro perfil>,
              "description":<nome de outro perfil>
        },
        ...
  ],
  "id":<id do usuário>,
  "newPassword":"",
  "password":"<senha criptografada do usuário>",
  "username":"<login do usuário>"
 }
```

Exemplo com cURL:

```
curl -i -H "Accept: application/json" -X GET
http://localhost:8080/webaccesscontrol/users/1.json
```

- Cadastrar usuário (RESTful):

URL: http://localhost:8080/webaccesscontrol/users
Método: POST
Parâmetros:

username	{login do usuário}
password	{senha}
selectedGroups	{id de um perfil} *pode ser repetido

Saída: JSON

```
{
 "action":"create_user",
 "returnCode":<0 para erro | 1 para succeso>,
 "message":"<mensagem relativa a ação executada>",
 "result":""
}
```

Exemplos com cURL:

```
curl -i -H "Accept: application/json" -X POST -d "username=meulogin&password=1234"
http://localhost:8080/webaccesscontrol/users

curl -i -H "Accept: application/json" -X POST -d
"username=meulogin&password=1234&selectedGroups=1&selectedGroups=2"
http://localhost:8080/webaccesscontrol/users
```

- Atualizar usuário (RESTful):

URL: http://localhost:8080/webaccesscontrol/users/{id}
Método: POST
Parâmetros:

_method	put
newPassword	{nova senha} *1
password	{senha atual}
username	{login do usuário}
selectedGroups	{id de um perfil} *pode ser repetido

Saída: JSON

```
{
 "action":"update_user",
 "returnCode":<0 para erro | 1 para succeso>,
 "message":"<mensagem relativa a ação executada>",
 "result":""
}
```

*1 - Caso os parâmetros *newPassword* e *password* sejam definidos, o sistema faz uma verificação da senha antiga antes de trocar para a nova. Se apenas o parâmetro *password* for definido, o sistema troca a senha diretamente.

Exemplos com cURL:

```
curl -i -H "Accept: application/json" -X POST -d
"username=meulogin&password=1234&newPassword=4321&_method=put"
http://localhost:8080/webaccesscontrol/users/1

curl -i -H "Accept: application/json" -X POST -d
"username=meulogin&selectedGroups=1&selectedGroups=2&_method=put"
http://localhost:8080/webaccesscontrol/users/1

curl -i -H "Accept: application/json" -X POST -d
"username=meulogin&password=1234&newPassword=4321&selectedGroups=1&selectedGroups=2&_m
ethod=put" http://localhost:8080/webaccesscontrol/users/1
```

- Deletar usuário (RESTful):

URL: http://localhost:8080/webaccesscontrol/users/{id}
Método: POST
Parâmetros:

_method	DELETE

Saída: JSON

```
{
 "action":"delete_user",
 "returnCode":<0 para erro | 1 para succeso>,
 "message":"<mensagem relativa a ação executada>",
 "result":""
}
```

Exemplo com cURL:

```
curl -i -H "Accept: application/json" -X POST -d "_method=DELETE"
http://localhost:8080/webaccesscontrol/users/1
```

- Listar grupos (RESTful):

(Listar todos os grupos)
URL: http://localhost:8080/webaccesscontrol/groups.json
Método: GET
Saída: JSON

```
[
    {
      "id":<id do perfil>,
      "description":"<nome do perfil>"
     },
     {
      "id":<id de outro perfil>,
      "description":"<nome de outro perfil>"
     }, ...
]
```

Exemplo com cURL:

```
curl -i -H "Accept: application/json" -X GET
http://localhost:8080/webaccesscontrol/groups.json
```

(Listar groupo com id = {id})
URL: http://localhost:8080/webaccesscontrol/groups/{id}.json
Método: GET
Saída: JSON

```
{
   "id":<id do perfil>,
   "description":"<nome do perfil>"
 }
```

Exemplo com cURL:

```
curl -i -H "Accept: application/json" -X GET
http://localhost:8080/webaccesscontrol/groups/1.json
```

- Cadastrar grupo (RESTful):

URL: http://localhost:8080/webaccesscontrol/groups
Método: POST
Parâmetros:

description	{nome do perfil}

Saída: JSON

```
{
 "action":"create_group",
 "returnCode":<0 para erro | 1 para succeso>,
 "message":"<mensagem relativa a ação executada>",
 "result":""
}
```

Exemplo com cURL:

```
curl -i -H "Accept: application/json" -X POST -d "description=meuperfil"
http://localhost:8080/webaccesscontrol/groups
```

- Atualizar grupo (RESTful):

URL: http://localhost:8080/webaccesscontrol/groups/{id}
Método: POST
Parâmetros:

_method	put
description	{nome do perfil}

Saída: JSON

```
{
 "action":"update_group",
 "returnCode":<0 para erro | 1 para succeso>,
 "message":"<mensagem relativa a ação executada>",
 "result":""
}
```

Exemplo com cURL:

```
curl -i -H "Accept: application/json" -X POST -d "description=outroperfil&_method=put"
http://localhost:8080/webaccesscontrol/groups/1
```

- Deletar grupo (RESTful):

URL: http://localhost:8080/webaccesscontrol/groups/{id}
Método: POST
Parâmetros:

_method	DELETE

Saída: JSON

```
{
 "action":"delete_group",
 "returnCode":<0 para erro | 1 para succeso>,
 "message":"<mensagem relativa a ação executada>",
 "result":""
}
```

Exemplo com cURL:

```
curl -i -H "Accept: application/json" -X POST -d "_method=DELETE"
http://localhost:8080/webaccesscontrol/groups/1
```