

Авторизация в СПИК

Авторизация в СКАУТ-Платформе через **СПИК** необходима для того, чтобы получить права на доступ к **Объектам мониторинга** и задать некоторые пользовательские настройки. Например, язык пользователя и часовой пояс.

Алгоритм авторизации:

- Формирование запроса авторизации **SpicAuthorizationRequest**.
- Вызов в сервисе **SpicAuthorizationService** метода входа в систему **Login (SpicAuthorizationRequest)**.
- Метод возвращает **SpicAuthorizationResponse**, содержащий результат открытия сессии, а именно:
 - уникальный идентификатор сессии (токен авторизации),
 - статус (удалось ли авторизоваться и аутентифицироваться),
 - время истечения сессии,
 - уникальный идентификатор пользователя,
 - имя пользователя.

Пример авторизации на языке JS.

В данном примере используется **HttpJsonRequestProvider**, код которого можно найти в сопровождающем файле `HttpJsonRequestProvider.js`.

```

var baseUrl = 'http://localhost:8081/spic/auth/rest/';
var loginServerUrl = baseUrl + 'Login';
var logoutServerUrl = baseUrl + 'Logout';
var requestProvider = new HttpJsonRequestProvider();
var authorizationToken = null;

function loginOnServer(request) {
    return response = requestProvider.POST(loginServerUrl,
        request);
}

function logoutOnServer(request) {
    return requestProvider.GETAuthorized(logoutServerUrl,
        null,
        authorizationToken);
}

function generateRequest() {
    return {
        Login: 'login',
        Password: 'password',
        TimestampUtc: '/Date(' + new Date().getTime() + ')/',
        TimeZoneOlsonId: 'Europe/Moscow',
        CultureName: 'ru-ru',
        UiCultureName: 'ru-ru'
    };
}

function login() {
    //?????????? ??????
    var request = generateRequest();

    //?????????????? ??????
    var response = loginOnServer(request);

    //?????????????? ?????? ??????????????
    authorizationToken = response.response.SessionId;
}

function logout() {
    logoutOnServer();
}

```

Пример авторизации на языке C#.

Сначала нужно создать заглушку к сервису **SpicAuthorizationSoapService** ([Создание заглушки для сервиса в среде Visual Studio](#)). Приведенный ниже метод выполняет авторизацию.

Здесь:

client – объект класса заглушки сервиса авторизации, сгенерированного на прошлом шаге.

Request – это сформированный класс, содержащий информацию о пользователе.

```

if (request == null) return null;
var client = GetAuthorizationClient();
if (client == null) throw new Exception();
var spicRequest = new SpicAuthorizationRequest
{
    CultureName = request.CultureName,
    Login = request.Login,
    Password = request.Password,
    TimestampUtc = request.TimestampUtc,
    TimeZoneOlsonId = request.TimeZoneOlsonId,
    UiCultureName = request.UiCultureName
};
var response = client.Login(spicRequest);
return response == null
    ? null
    : new AuthorizationResponse
    {
        ExpireDate = response.ExpireDate,
        IsAuthenticated = response.IsAuthenticated,
        IsAuthorized = response.IsAuthorized,
        SessionId = response.SessionId,
        UserId = response.UserId,
        UserName = response.UserName
    };

```

Выход из системы осуществляется вызовом **client.Logout()**;

При авторизации клиент должен послать в СПИК **SpicAuthorizationRequest** и получить **SpicAuthorizationResponse**.

Пример контракта **SpicAuthorizationRequest** в JS :

```

{
    "Login": "test_demo",
    "Password": "test_demo",
    "TimestampUtc": "/Date(1380174312983)/",
    "TimeZoneOlsonId": "Europe/Moscow",
    "CultureName": "ru-ru",
    "UiCultureName": "ru-ru"
}

```

Пример контракта **SpicAuthorizationResponse** в JS :

```

{
    "ExpireDate": "\\Date(1380203117863+0400)\\",
    "IsAuthenticated": true,
    "IsAuthorized": true,
    "SessionId": "a26fd966-dada-453f-adc8-359ac32c4b94",
    "UserId": 1075,
    "UserName": "????-?????????"
}

```

Результирующие классы для контрактов данных **SpicAuthorizationRequest** и **SpicAuthorizationResponse** для SOAP можно посмотреть в сгенерированных классах, сформированных после получения метаданных.

<<Назад
оглавлению

К

Далее>>