

OrgaNICEr



Dokumentation der REST-Abfragen

Um Daten über die WebUntis API abfragen zu können, muss man sich aus Sicherheitsgründen zuerst authentifizieren.

```
public static AuthData authUser(String username, String password) {
    JSONObject userdata = new JSONObject();
    try {
        userdata.put( name: "user", username);
        userdata.put( name: "password", password);
        userdata.put( name: "client", value: "Organicer");

        JSONObject standardParams = new JSONObject();
        standardParams.put( name: "id", value: "ID");
        standardParams.put( name: "method", value: "authenticate");
        standardParams.put( name: "params", userdata);
        standardParams.put( name: "jsonrpc", value: "2.0");

        HttpURLConnection httpURLConnection;

        httpURLConnection = (HttpURLConnection) new URL(URL_SCHOOLNAME).openConnection();
        httpURLConnection.setDoOutput(true);
        httpURLConnection.setRequestProperty("Content-Type", "application/json");
        httpURLConnection.setRequestProperty("Accept", "application/json");
        httpURLConnection.setRequestMethod("POST");
        httpURLConnection.connect();

        //Write
        OutputStream os = httpURLConnection.getOutputStream();
        BufferedWriter writer = new BufferedWriter(new OutputStreamWriter(os, charsetName: "UTF-8"));
        writer.write(standardParams.toString());
        writer.close();
        os.close();

        //Read
        BufferedReader br = new BufferedReader(new InputStreamReader(httpURLConnection.getInputStream(), charsetName: "UTF-8"));

        String line = null;
        StringBuilder sb = new StringBuilder();

        while ((line = br.readLine()) != null) {
            sb.append(line);
        }

        br.close();
        String result = sb.toString();
        JSONObject object = new JSONObject(result);
        object = object.getJSONObject("result");

        return new AuthData(object.getString( name: "sessionId"), object.getString( name: "klasseId"));
    } catch (JSONException | IOException ex) {
        return null;
    }
}
```

Bei der Authentifizierung ist entscheidend, dass die Parameter „user“, „password“ und „client“ mit übergeben werden. (siehe Screenshot)

Nach der Authentifizierung erhält man sowohl eine SessionID, als auch seine eigene KlassenID. Die „JSESSIONID“ wird bei weiteren Abfragen als im Header als „Cookie“ benötigt.

Hier ein Beispiel, wie weitere REST Abfragen nach dem Authentifizieren ablaufen werden:

The screenshot shows a REST client interface with the following details:

- Method:** POST
- Request URL:** <https://mese.webuntis.com/WebUntis/jsonrpc.do>
- Buttons:** A blue "SEND" button and a vertical ellipsis menu icon.
- Parameters:** A section header with an upward arrow.
- Headers Tab:** The "Headers" tab is selected, showing a table of request headers.
- Header Table:**

Header name	Header value	
content-type	application/json	X ✎
Cookie	JSESSIONID=DD627058B3A654E7FDAF4747D7CE42E6	X ✎ ?

```

private static JSONArray getData(SessionDataHelper dataHelper, String methodname) throws IOException, JSONException {
    return getData(dataHelper, methodname, params: null);
}

private static JSONArray getData(SessionDataHelper dataHelper, String methodName, String params) throws JSONException, IOException {
    HttpURLConnection httpURLConnection = (HttpURLConnection) new URL(URL_SCHOOLNAME).openConnection(); //TODO: Request must not be resetted.
    httpURLConnection.setDoOutput(true);
    httpURLConnection.setRequestProperty("Content-Type", "application/json");
    httpURLConnection.setRequestProperty("Accept", "application/json");
    httpURLConnection.setRequestProperty("Cookie", "JSESSIONID=" + dataHelper.getAuthData().getSessionId());
    httpURLConnection.setRequestMethod("POST");
    httpURLConnection.connect();

    JSONObject standardParams = new JSONObject();
    standardParams.put( name: "id", value: "ID");
    standardParams.put( name: "method", methodName);
    standardParams.put( name: "params", params);
    standardParams.put( name: "jsonrpc", value: "2.0");

    //Write
    OutputStream os = httpURLConnection.getOutputStream();
    BufferedWriter writer = new BufferedWriter(new OutputStreamWriter(os, charsetName: "UTF-8"));
    writer.write(standardParams.toString());
    writer.close();
    os.close();

    //Read
    BufferedReader br = new BufferedReader(new InputStreamReader(httpURLConnection.getInputStream(), charsetName: "UTF-8"));

    String line = null;
    StringBuilder sb = new StringBuilder();

    while ((line = br.readLine()) != null) {
        sb.append(line);
    }
    br.close();
    String result = sb.toString();

    JSONObject asJsonObject = new JSONObject(result);
    return asJsonObject.getJSONArray( name: "result");
}

```

Wenn Daten in Android abgefragt werden, muss man den „methodname“ übergeben, außerdem können Parameter übergeben werden. (überladene Methode) (siehe Screenshot)