

## “CreditReporting” REST/JSON service

### Interface #4: getCompanyReport2

ver. 19.05.2020

*Purpose:* In response to supplied order number, the interface returns all report files at once.

*Note:* Customer's contract record holds a Report Set - list of Products (report types) and formats, which by default should be included into finished report as attachments; so, in addition to ordered product, all files from the Report Set will be generated and applied to the report as well.  
Also, report may be supplemented by other files, applied manually or auto-generated, which all make the report complete.

Interface address : <https://online.igkservice.lv/rscr/getCompanyReport2>

Request method : POST

Contents:

1. Request structure
2. Response structure
3. Response status codes
4. Authentication
- Y. Source code of used classes

### 1. Request structure

#### 1.1. Request parameters:

Parameter name	Type	Required	Explanation
providerPartyID	string	yes	always "IGK-LV"
providerOrderReference	string	yes	IGK order reference

#### 1.2. Request sample:

```
{  
  "providerPartyID": "IGK-LV",  
  "providerOrderReference": "I-LV-200327201407.48"  
}
```

## 2. Response structure

### 2.1. Response parameters:

Parameter name	Type	Required	Explanation
getCompanyReport2Result	object	yes	
responseCode	string	yes	See "3.2. Application response codes"
responseText	string	yes	Details for "responseCode"
report2	object	no	
reportText	object []	no	
value	string	yes	Textual report
filename	string	yes	Filename for the value
reportBinary	object []	no	
value	string	yes	Binary report, Base64 encoded
filename	string	yes	Filename for the value
reportXML	object []	no	
value	string	yes	XML report
filename	string	yes	Filename for the value
reportJson	object []	no	
value	string	yes	Json report
filename	string	yes	Filename for the value

### 2.2. Response sample

```
{
  "getCompanyReport2Result": {
    "responseCode": "OK",
    "responseText": "OK",
    "report2": {
      "reportText": [{
        "value": "WARNING: C03-XML: BuyerNo has incorrect format: n/a -
starting-part length != 3.\n\r",
        "filename": "_messages.txt"
      }],
      "reportBinary": [{
        "value": "JVBERi0xLjQKJaqrRk0KNCAwIG9iago8PAovQ3JlYXRvciAoQXBh...",
        "filename": "I-LV-200327201407_48.pdf"
      }],
      "reportJson": [{
        "value": "{\"companyData\": {\"orderReference\": {\"iConsumerOrder...\",
        \"filename\": \"I-LV-200327201407_48.json\"
      }],
      "reportXML": [{
        "value": "<?xml version=\"1.0\" encoding=\"utf-8\"?>\n<document...\",
        "filename": \"I-LV-200327201407_48.xml\"
      }, {
        "value": "<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n<Report>...\",
        "filename": \"I-LV-200327201407_48-2.xml\"
      }
    ]
  }
}
```

### 3. Response status codes

#### 3.1. Http response status codes:

Status code	Description	Explanation
200	ok	Successful return
400	Bad request	Error in request, see "responseCode Text" for details
303	See other	Cannot process, see "responseCode Text" for details
500	Server Error	Service error, see "responseCode Text" for details

#### 3.2. Application response codes

responseCode / responseText	Explanation
OK / OK	Successful return
BadRequest / - No "authHeader" httpHeader. - Cannot unmarshal "authHeader" to JSON object. - AuthHeader: username is not set. - AuthHeader: password is not set. - ProviderPartyID: Mandatory value is not set. - ProviderOrderReference: Mandatory value is not set.	Incorrect request parameter value
UnknownProviderIDOrProviderOrderReference / - Unknown ProviderID: <partyID> - Unknown OrderReference <ref> for Provider <partyID>	Incorrect request parameter value
ReportNotAvailable / Order <ref> is not in Finished/FinishedNegative status, or no report is assigned.	Requested order is not in status that allows getting a report
ServerError / <program exception message>	Unexpected program error

### 4. Authentication

Authentication should be provided by the client in "authHeader" header of the request - as JSON presentation of AuthHeader object.

```
public class AuthHeader {
    String username, password;
    public String toJsonString() {
        return "{\"username\":\""+username+"\", \"password\":\""+password+"\"}";
    }
}
```

Example of the client code:

```
Client client = ClientBuilder.newClient();
WebTarget webTarget = client.target(BASE_URI);
WebTarget resource = webTarget.path("getCompanyReport2");
Invocation.Builder builder = resource.request(MediaType.APPLICATION_JSON);
builder.header("authHeader", _authHeader.toJsonString());
```

## Y. Source code of used classes

Please see file "rscr\_common-src.zip".

.end