

인증토큰 발급

REQUEST

URL

Parameter

Sample

<표 1-1> 유효기간 설정 코드

RESPONSE

Parameter

ResultDataSet

Table1

Sample

오픈스크랩 보관함 조회

REQUEST

URL

Parameter

Sample

RESPONSE

Parameter

ResultDataSet

Table

Sample

<표 2-1> 세금계산서 종류 구분 코드

오픈스크랩 원본 조회

REQUEST

URL

Parameter

Sample

RESPONSE

Parameter

ResultDataSet

Table

Sample

미리보기 양식 조회

Java

JavaScript

REQUEST

URL

Parameter

Sample

RESPONSE

Parameter

ResultDataSet

Table1

Sample

인증토큰 발급

스마트빌 홈페이지(<http://demo.smartbill.co.kr>)에서 발급받은 인증코드를 통해서 인증토큰을 발급받을 수 있으며, 발급받은 인증토큰은 목록조회나 XML원본조회 등 다른 API를 호출할 때 사용합니다.

필수 파라미터 값들을 담아 POST로 요청합니다. 요청에 대한 응답은 JSON객체로 RESPONSE 됩니다.

인증토큰은 원하는 유효기간을 설정하여 발급받을 수 있습니다.

REQUEST

URL

POST /sb-api/request **HTTP/1.1**

Host : demoapi.smartbill.co.kr

Content-type : application/json; charset=utf-8

Parameter

Name	Type(max)	Description	Required
MessageId	String(36)	트랜잭션 식별자	O
Signal	String(30)	고정 값 : GET_TOKEN	O
RequestTime	String(14)	요청시간 (YYYYMMDDHH24MISS)	O
SendComRegno	String(13)	송신자 사업자번호	O
AuthCode	String(40)	스마트빌에서 생성한 인증코드	O
SbId	String(10)	스마트빌 사용자 아이디	O
ExpirationDateCode	String(10)	유효기간 설정 코드 (하단의 표<1-1> 참조)	O
ExpirationDate	String(10)	사용자 지정 시 만료일 입력	X

Sample

```
curl -X POST -H "Content-Type: application/json"
http://demoapi.smartbill.co.kr/sb-api/request
-d "{ \"Signal\": \"GET_TOKEN\",
      \"AuthCode\": \"AAAABBBB11112222\",
      \"ExpirationDate\": \"2020-11-16\",
      \"RequestTime\": \"20201116035603\",
      \"SbId\": \"aaa1111\",
      \"SendComRegno\": \"111111111\",
      \"ExpirationDateCode\": \"D\",
      \"MessageId\": \"sbqa-20201116360001\"}"
```

<표 1-1> 유효기간 설정 코드

유효기간 코드	유효기간(일)	비고
D	1	Content
W	7	Content
H	15	Content
M	30	Content
Q	90	Content
R	180	Content
Y	365	Content
Z	영구 사용	Content
C	사용자 지정	'ExpirationDate'에 지정할 만료일 입력

RESPONSE

HTTP/1.1 200 OK

Content-Type : application/json; charset=utf-8

Parameter

Name	Type(max)	Description
MessageId	String(36)	요청 한 트랜잭션 식별자
Signal	String(30)	요청 한 고정 값 : GET_TOKEN
ResponseTime	String(14)	응답시간 (YYYYMMDDHH24MISS)
ResultCode	String(5)	처리 결과 코드 (성공 : 30000)
ResultMessage	String(2000)	처리 결과 상세 내역
ResultDataSet	DataSet	처리 결과 데이터

ResultDataSet

Name	Type	Description
Table1	Array	처리 결과 데이터 테이블1

Table1

Name	Type	Description
AUTH_TOKEN	String	인증토큰

Sample

```
{
  "MessageId": "sbqa-20201116360001",
  "Signal": "GET_TOKEN",
  "ConversationId": null,
  "ResponseTime": "20201116162117",
  "ResultCode": "30000",
  "ResultMessage": "정상적으로 처리되었습니다.",
  "ResultDataSet":
  {
    "Table1":
    [
      {
        "AUTH_TOKEN": "AAAABBBBccccddd1111222233334444="
      }
    ]
  }
}
```

오픈스크랩 보관함 조회

수집된 세금계산서 리스트를 조회합니다.

※수집된 데이터는 스마트빌 API서버에 3개월간 보관 후 자동삭제 됩니다.

REQUEST

URL

POST /sb-api/request **HTTP/1.1**

Host : demoapi.smartbill.co.kr

Content-type : application/json; charset=utf-8

Parameter

Name	Type(max)	Description	Required
MessageId	String(36)	트랜잭션 식별자	O
Signal	String(30)	고정 값 : OS_DTI_LIST	O
RequestTime	String(14)	요청시간 (YYYYMMDDHH24MISS)	O
SendComRegno	String(13)	송신자 사업자번호	O
AuthToken	String	인증토큰	O
ServiceCode	String(3)	고정 값 : DTI	O
SearchDate	String(1)	검색일자 구분 값 (작성일자:W, 발행일자:I, 전송일자:S)	O
SearchFromDate	String(10)	시작일자 (YYYY-MM-DD)	O
SearchToDate	String(10)	종료일자 (YYYY-MM-DD)	O
SearchComRegno	String(13)	검색 사업자번호	O
SupBuy_Type	String(2)	매출/매입 구분 값 (매출 : AR, 매입 : AP)	O

Sample

```
curl -X POST -H "Content-Type: application/json"
http://demoapi.smartbill.co.kr/sb-api/request
-d "{ \"Signal\": \"OS_DTI_LIST\",
      \"ServiceCode\": \"DTI\",
      \"SearchComRegno\": \"222222222\",
      \"SearchDate\": \"W\",
      \"SearchFromDate\": \"W2020-11-16\",
      \"SearchToDate\": \"2020-11-17\",
      \"AuthToken\": \"AAAABBBBccccddd1111222233334444=\",
      \"RequestTime\": \"20201117020043\",
      \"SupBuy_Type\": \"AR\",
      \"SendComRegno\": \"111111111\",
      \"MessageId\": \"sbqa-20201117020043\"}"
```

RESPONSE

HTTP1.1 200 OK

Content-Type : application/json; charset=utf-8

Parameter

Name	Type(max)	Description
MessageId	String(36)	요청 한 트랜잭션 식별자
Signal	String(30)	요청 한 고정 값 : OS_DTI_LIST
ConversationId	String(35)	참조번호 (null)
ResponseTime	String(14)	응답시간 (YYYYMMDDHH24MISS)
ResultCode	String(5)	처리 결과 코드 (성공 : 30000)
ResultMessage	String(2000)	처리 결과 상세 내역
ResultDataSet	DataSet	처리 결과 데이터

ResultDataSet

Name	Type	Description
Table	Array	처리 결과 데이터 테이블

Table

Name	Type	Description
ISSUE_ID	String	승인번호
SUPBUY_TYPE	String	매출/매입 구분 값 (매출 : AR, 매입 : AP)
DTI_WDATE	String	작성일자 (YYYY-MM-DDT00:00:00)
DTI_IDATE	String	발행일자 (YYYY-MM-DDT00:00:00)
DTI_SDATE	String	(국세청)전송일자 (YYYY-MM-DDT00:00:00)
TYPE_CODE	String	세금계산서 종류 구분 코드 (하단의 표<2-1> 참고)
DTI_TYPE	String	세금계산서 종류 구분 코드 (하단의 표<2-1> 참고)
TAX_DEMAND	String	청구방식 구분 코드 (영수:01, 청구:02)
SEQ_ID	String	세금계산서 관리번호
SUP_COM_REGNO	String	공급자 사업자번호
SUP_REP_NAME	String	공급자 대표자명
SUP_COM_NAME	String	공급자 회사명
SUP_EMAIL	String	공급자 이메일
BYR_COM_REGNO	String	공급받는자 사업자번호
BYR_REP_NAME	String	공급받는자 대표자명
BYR_COM_NAME	String	공급받는자 회사명
BYR_EMAIL	String	공급받는자 이메일
SUP_AMOUNT	String	공급가액 합계
TAX_AMOUNT	String	세액합계
TOTAL_AMOUNT	String	총 금액
TOTAL_CNT	String	데이터 총 건수

Sample

```
{
  "MessageId": "sbqa-20201117020043",
  "Signal": "OS_DTI_LIST",
  "ConversationId": null,
  "ResponseTime": "20201117140048",
  "ResultCode": "30000",
  "ResultMessage": "정상적으로 처리되었습니다.",
  "ResultDataSet":
  {
    "Table":
    [
      {
```

```

        "ISSUE_ID": "2020111641000008a11bb1cc",
        "SUPBUY": "AR",
        "DTI_WDATE": "2020-11-16T00:00:00",
        "DTI_IDATE": "2020-11-16T00:00:00",
        "DTI_SDATE": "2020-11-17T00:00:00",
        "TYPE_CODE": "0101",
        "DTI_TYPE": "0101",
        "TAX_DEMAND": "02",
        "SEQ_ID": "WX0001",
        "SUP_COM_REGNO": "1111111111",
        "SUP_REP_NAME": "임상원",
        "SUP_COM_NAME": "비즈니스온",
        "SUP_EMAIL": "test@test.com",
        "BYR_COM_REGNO": "2222222222",
        "BYR_REP_NAME": "김상원",
        "BYR_COM_NAME": "스마트빌",
        "BYR_EMAIL": "test@test.com",
        "SUP_AMOUNT": "100",
        "TAX_AMOUNT": "10",
        "TOTAL_AMOUNT": "110",
        "TOTAL_CNT": "2"
    }
}
{
    "ISSUE_ID": "2020111641000008a11bb2cc",
    "SUPBUY": "AR",
    "DTI_WDATE": "2020-11-16T00:00:00",
    "DTI_IDATE": "2020-11-16T00:00:00",
    "DTI_SDATE": "2020-11-17T00:00:00",
    "TYPE_CODE": "0201",
    "DTI_TYPE": "0201",
    "TAX_DEMAND": "01",
    "SEQ_ID": "WZ0002",
    "SUP_COM_REGNO": "1111111111",
    "SUP_REP_NAME": "임상원",
    "SUP_COM_NAME": "비즈니스온",
    "SUP_EMAIL": "test@test.com",
    "BYR_COM_REGNO": "2222222222",
    "BYR_REP_NAME": "김상원",
    "BYR_COM_NAME": "스마트빌",
    "BYR_EMAIL": "test@test.com",
    "SUP_AMOUNT": "200",
    "TAX_AMOUNT": "20",
    "TOTAL_AMOUNT": "220",
    "TOTAL_CNT": "2"
}
]
}
}

```

<표 2-1> 세금계산서 종류 구분 코드

세금계산서 분류(앞2자리)	설명	세금계산서 종류(뒤2자리)	설명
01	세금계산서	01	일반
02	수정세금계산서	02	영세율
03	계산서	03	일반 위수탁
04	수정계산서	04	수입
Content	Content	05	영세율 위수탁

오픈스크랩 원본 조회

수집된 세금계산서 원본을 조회합니다.

※수집된 데이터는 스마트빌 API서버에 3개월간 보관 후 자동삭제 됩니다.

REQUEST

URL

POST /sb-api/request **HTTP/1.1**

Host : demoapi.smartbill.co.kr

Content-type : application/json; charset=utf-8

Parameter

Name	Type(max)	Description	Required
MessageId	String(36)	트랜잭션 식별자	O
Signal	String(30)	고정 값 : OS_DTI_XML	O
RequestTime	String(14)	요청시간 (YYYYMMDDHH24MISS)	O
SendComRegno	String(13)	송신자 사업자번호	O
AuthToken	String	인증토큰	O
ServiceCode	String(3)	고정 값 : DTI	O
AuthCode	String(40)	스마트빌에서 생성한 인증코드	O
SupBuy_Type	String(2)	매출/매입 구분 값 (매출 : AR, 매입 : AP)	O
Issued	String[]	세금계산서 승인번호 배열 (1회 최대 100건)	O

Sample

```
curl -X POST -H "Content-Type: application/json"
http://demoapi.smartbill.co.kr/sb-api/request
-d "{ \"Signal\": \"OS_DTI_XML\",
      \"ServiceCode\": \"DTI\",
      \"AuthCode\": \"AAAABBBB11112222\",
      \"AuthToken\": \"AAAABBBBccccddd1111222233334444=\",
      \"RequestTime\": \"20201117020043\",
      \"SupBuy_Type\": \"AR\",
      \"IssueId\": [\"2020111641000008a11aa1bb\",
                    \"2020111641000008a22bb2cc\"],
      \"SendComRegno\": \"111111111\",
      \"MessageId\": \"sbqa-20201117020043\"}"
```

RESPONSE

HTTP/1.1 200 OK

Content-Type : application/json; charset=utf-8

Parameter

Name	Type(max)	Description
MessageId	String(36)	요청 한 트랜잭션 식별자
Signal	String(30)	요청 한 고정 값 : OS_DTI_LIST
ConversationId	String(35)	참조번호 (null)
ResponseTime	String(14)	응답시간 (YYYYMMDDHH24MISS)
ResultCode	String(5)	처리 결과 코드 (성공 : 30000)
ResultMessage	String(2000)	처리 결과 상세 내역
ResultDataSet	DataSet	처리 결과 데이터

ResultDataSet

Name	Type	Description
Table	ArrayList	처리 결과 데이터 테이블

Table

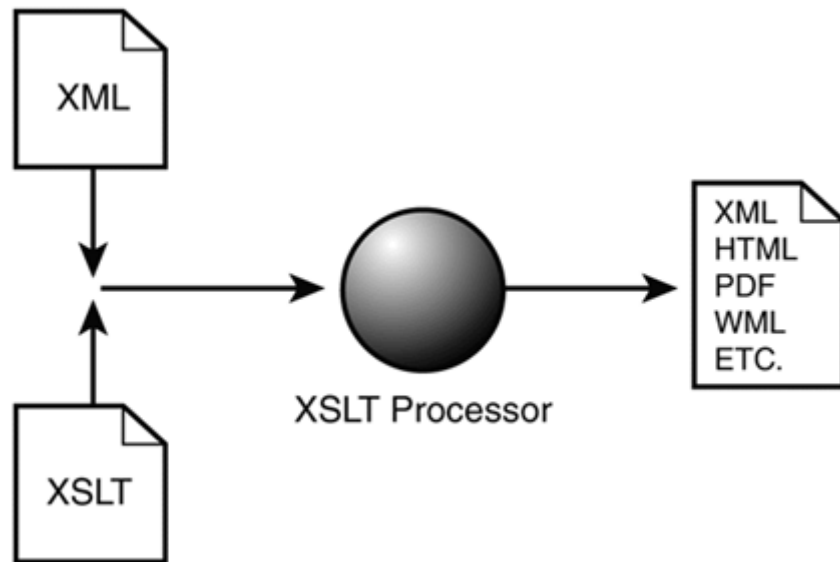
Name	Type	Description
ISSUE_ID	String	승인번호
DTI_MSG	String	세금계산서 원본 XML

Sample

```
{
  "MessageId": "sbqa-20201117020043",
  "Signal": "OS_DTI_XML",
  "ConversationId": null,
  "ResponseTime": "20201117140048",
  "ResultCode": "30000",
  "ResultMessage": "정상적으로 처리되었습니다.",
  "ResultDataSet":
  {
    "Table":
    [
      {
        "ISSUE_ID": "2020111641000008a11aa1bb",
        "DTI_XML": "<?xml version='1.0' encoding='UTF-8'>
<TaxInvoice>.....</TaxInvoice>",
      }
      {
        "ISSUE_ID": "2020111641000008a22bb2cc",
        "DTI_XML": "<?xml version='1.0' encoding='UTF-8'>
<TaxInvoice>.....</TaxInvoice>",
      }
    ]
  }
}
```

미리보기 양식 조회

세금계산서 데이터를 화면에 출력하기 위한 UI 템플릿을 요청합니다. 시스템 상에서 직접 세금계산서 양식을 구현하고자 할 때 사용하며 XML 형식으로 되어있는 전자세금계산서를 원하는 포맷에 맞추어 출력하는 XSLT라는 기술을 사용합니다. (www.w3schools.com/xml/xsl_intro.asp 참고)



미리보기 양식 조회 API를 호출하면 xsl문서를 String 형태로 리턴 받을 수 있는데 시스템 플랫폼에 적합한 라이브러리를 사용하여 세금계산서 원본 xml과 결합하여 출력 화면을 구현할 수 있습니다.

아래 간단한 java, javascript 샘플 소스를 첨부합니다.

Java

```
Source xmlSource = new StreamSource(new StringReader(xml));
Source xsltSource = new StreamSource(new StringReader(xsl));
TransformerFactory ft = TransformerFactory.newInstance();

StringWriter writer = new StringWriter();
StreamResult result = new StreamResult(writer);

Transformer trans = ft.newTransformer(xsltSource);
trans.transform(xmlSource, result);

String strResult = writer.toString();
```

JavaScript

```
// IE 브라우저
var signedXML; //원본XML을 할당하는 변수
var viewForm; //미리보기 양식을 할당하는 변수
var xml = new ActiveXObject("Microsoft.XMLDOM");
xml.async = false;
xml.loadXML(signedXML);
var xsl = new ActiveXObject("Microsoft.XMLDOM");
xsl.async = false;
xsl.loadXML(viewForm);
document.getElementById("VIEW").innerHTML = xml.transformNode(xsl);

// IE 이외 브라우저
var parser = new DOMParser();
var xml = parser.parseFromString(signedXML, "text/xml");
var xsl = parser.parseFromString(viewForm, "text/xml");
var xsltProcessor = new XSLTProcessor();
xsltProcessor.importStylesheet(xsl);
```

```
var res = xsltProcessor.transformToFragment(xml, document);
document.getElementById("VIEW").appendChild(res);
```

REQUEST

URL

POST /sb-api/request **HTTP/1.1**

Host : demoapi.smartbill.co.kr

Content-type : application/json; charset=utf-8

Parameter

Name	Type(max)	Description	Required
MessageId	String(36)	트랜잭션 식별자	O
Signal	String(30)	고정 값 : OS_DTI_XML	O
RequestTime	String(14)	요청시간 (YYYYMMDDHH24MISS)	O
SendComRegno	String(13)	송신자 사업자번호	O
AuthToken	String	인증토큰	O
ServiceCode	String(3)	고정 값 : DTI	O

Sample

```
curl -X POST -H "Content-Type: application/json"
http://demoapi.smartbill.co.kr/sb-api/request
-d "{ \"Signal\": \"PREVIEW_FORM\",
      \"ServiceCode\": \"DTI\",
      \"AuthToken\": \"AAAABBBBccccddddd1111222233334444=\",
      \"RequestTime\": \"20201117020043\",
      \"SendComRegno\": \"111111111\",
      \"MessageId\": \"sbqa-20201117020043\"}"
```

RESPONSE

HTTP/1.1 200 OK

Content-Type : application/json; charset=utf-8

Parameter

Name	Type(max)	Description
MessageId	String(36)	요청 한 트랜잭션 식별자
Signal	String(30)	요청 한 고정 값 : PREVIEW_FORM
ConversationId	String(35)	참조번호 (null)
ResponseTime	String(14)	응답시간 (YYYYMMDDHH24MISS)
ResultCode	String(5)	처리 결과 코드 (성공 : 30000)
ResultMessage	String(2000)	처리 결과 상세 내역
ResultDataSet	DataSet	처리 결과 데이터

ResultDataSet

Name	Type	Description
Table1	Array	처리 결과 데이터 테이블

Table1

Name	Type	Description
CONTENT	String	공급받는자용 매입 세금계산서
CONTENT	String	공급받는자용 매입 계산서
CONTENT	String	공급받는자용 위수탁 매입 세금계산서
CONTENT	String	공급받는자용 위수탁 매입 계산서
CONTENT	String	공급자용 매출 세금계산서
CONTENT	String	공급자용 매출 계산서
CONTENT	String	공급자용 위수탁 매출 세금계산서
CONTENT	String	공급자용 위수탁 매출 계산서
CONTENT	String	거래명세서(과세)
CONTENT	String	거래명세서(면세)
CONTENT	String	위수탁 거래명세서(과세)
CONTENT	String	위수탁 거래명세서(면세)

Sample

[illegible]