



API Guide

2019. 01. 14 v2.4

문서 버전

Ver	Date	History	작성자	비고
1.0	2018-03-29	최초 작성본	이봉상	
1.1	2018-03-31	예시 JSON 수정	이봉상	
1.2	2018-03-31	예시 JSON 수정	심상용	
1.3	2018-04-02	검색하기 API 추가	이봉상	
1.4	2018-04-06	Request Code, Client ID 추가	이봉상	
1.5	2018-05-29	시작하기 및 사용자 토큰 API 추가	이봉상	
1.6	2018-06-05	API 수정	이봉상	
1.7	2018-06-08	API 설명 추가	신지수	
1.8	2018-06-20	Web 서명하기 API 추가	윤태경	
1.9	2018-06-21	API 사용절차 페이지 추가	신지수	
2.0	2018-08-23	가져오기/내보내기 API 정의 수정	이봉상	
2.1	2018-09-20	시작하기 API 수정	이봉상	
2.2	2018-11-23	간단 계약 API 추가	이봉상	Export 부분 추가 고려*
2.3	2018-12-30	시작하기 API 수정, Web 서명하기 페이지 호출 수정	최윤창	
2.4	2019-01-07	간편 계약 API 수정(대면, 비대면)	심상용	

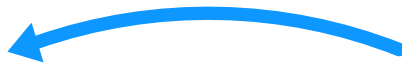
1. 사용자 토큰	
1.1. 사용자 토큰 API 설명	6
1.2. 예시 JSON	7
2. 가져오기	
2.1. 가져오기 API 설명	8
2.2. 예시 JSON	12
3. 내보내기	
3.1. 내보내기 API 설명	14
3.2. 예시 JSON	17
4. 필드 검색하기	
4.1. 검색하기 API 설명	20
4.2. 예시 JSON	21
5. 날짜별 검색하기	
5.1. 검색하기 API 설명	22
5.2. 예시 JSON	23
6. 시작하기	
6.1. 시작하기 API 설명	24
6.2. 예시 JSON	25
7. Web 서명하기 API 호출	
7.1. Web서명하기 API 설명	26
8. 간편 계약	
8.1. 간편 계약 API 설명	27
8.2. 예시 JSON	28

API 개요



eSignon
이용고객

- 검색하기 API
- 내보내기 API

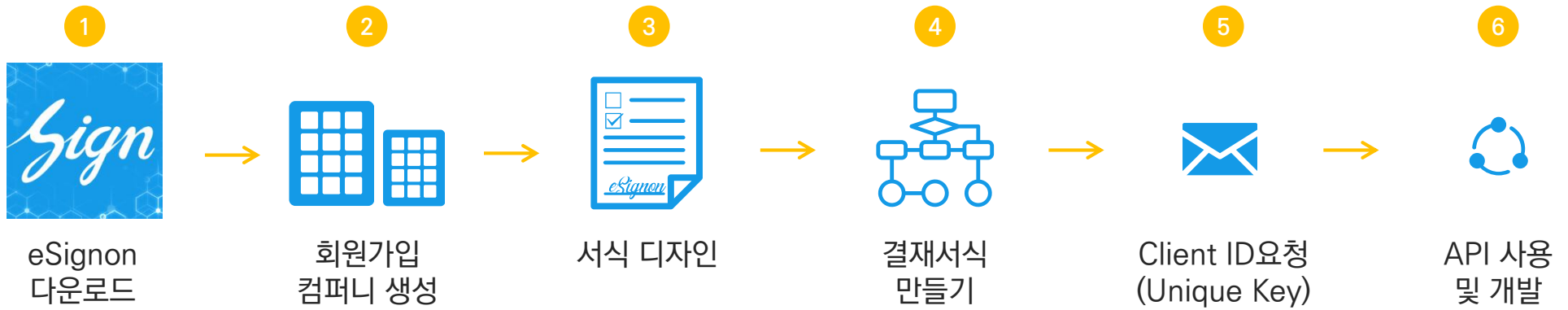


eSignon

- 가져오기 API
- 시작하기 API



API 사용절차



5, 6번은 서비스 플랜 구매 후
사용하실 수 있는 부분입니다.

1. 사용자 토큰(Start 할 경우에만 사용)

eSignon에서 제공하는 API를 호출하기 위해서는 반드시 사용자 토큰 값을 알아야 합니다. (사용자 토큰과 Client ID가 필수)
해당 API 호출 후 얻어진 사용자 토큰값을 다른 API를 호출시에 사용하게 됩니다.

URL 형식에서 {CompID} 부분에는 사용자 컴퍼니 ID값을 입력하고, Body 영역에서 client_id는 사용자에게 할당된 값을 입력합니다.

※ Client_id는 컴퍼니 별로 발행되며, eSignon팀에 요청하면 발급하여 알려드립니다.

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

- METHOD:** POST
- URL:** https://docs.esignon.net/api/guide/login (length: 40 bytes)
- QUERY PARAMETERS:** None
- HEADERS:** Content-Type: application/json
- BODY:** A JSON object with the following structure:

```
1 {
2   "header": {
3     "request_code": "1001Q",
4     "request_msg": "인증코드 요청을 합니다.",
5     "session_id": "thread_023"
6   },
7   "body": {
8     "comp_id": "guide",
9     "memb_email": "guide@esignon.net",
10    "memb_pwd": "xxxxxxxxxxxxxxxx",
11    "client_id": "xxxxxxxxxxxxxxxx"
12  }
13 }
14 }
15 }
```

• Request Parameter

Type	Contents
Method	POST
Scheme	https://docs.esignon.net/api/{CompID}/login
Content-Type	application/json
Body	<pre>{ "header": { "request_code": "1001Q", "request_msg": "인증코드 요청을 합니다.", "session_id": "thread_023" }, "body": { "comp_id": "{CompID}", "memb_email": "{사용자 이메일}", "memb_pwd": "{사용자 비밀번호}", "client_id": "{사용자 ID}" } }</pre>

11/11/2019

수신된 내용 중 사용자 토큰값은 access_token 입니다.

Response JSON

Response

Cache Detected - Elapsed Time: 146ms

Response

Cache Detected - Elapsed Time: 146ms

3] BODY ? pretty

pretty ▼

```
{
  header : {
    response_code : "1001A",
    result_code : "00",
    result_msg : "성공적으로 로그인되었습니다.",
    session_id : ""
  },
  body : {
    access_token : " ",
    comp_id : "guide",
    device_id : " ",
    expire_date : "2018-07-05 19:26:10",
    memb_email : "guide@esignon.net"
  }
}
```

lines nums

length: 406 bytes

2. 가져오기

외부에서 데이터를 가져오고자 할 때 사용하는 API 입니다.

Action이 시작될 때 호출 되며, Action 시작 시 필요한 정보를 불러와 eSignon에 데이터를 넣어줍니다.

가져오기 옆의 +를 Click -> 가져오기 이름 입력 -> 확인

* Response Parameter

파란색 표 안에 있는 입력 항목 모두 입력 후 호출(🚩)버튼을 누르면 실제 API가 호출되며, 호출된 API에서 넘겨 온 변수가 표시된다. 이때 표시된 변수 리스트 에서 도구(⚙️) 버튼을 클릭하여 문서에 매칭 해 주면 됩니다.

문서 1 < Process1 - Action1 >			
Workflow		API	
가져오기	+	Name 재직증명서	
• 재직증명서	⊗	URL http://esignon.support.net	
내보내기	+	Type 가져오기	Request Code 2002
		사용여부 <input checked="" type="checkbox"/>	Client ID eSignon
Request Parameter		Response Parameter	
• name 홍길동 ⚙️ ⊗		• koreaid 123456-789101 ⚙️	
		• address 서울특별시 OO ⚙️	
		• position 차장 ⚙️	

* 예시

재직증명서를 발급하기 위한 개인정보 요청

- **eSignon측**

사용자의 성명만 알고 있음(ex. 홍길동)

- **회사측(외부)**

사용자의 정보를 가지고 있음
(ex. 주민등록번호, 주소, 직책 등등)

- 1) Name = 해당 가져오기 API의 이름
- 2) URL = 통신 URL
- 3) Type = '가져오기' (고정)
- 4) 사용여부 = 해당 가져오기 API를 사용할지 여부
- 5) Request Code = 고객님의 정한 임의의 값
- 6) Client ID = eSignon Service ID 사용

2. 가져오기

‘+’ 버튼을 클릭하여 가져올 parameter를 추가합니다.
 톱니바퀴 버튼을 클릭하여 가져와서 보여주고자 하는 컨트롤을 클릭하여 매칭합니다.

문서 1
< Process1 – Action1 >
↑↓ ×

Workflow
API

가져오기	+	Name	재직증명서
• 재직증명서	×	URL	http://esignon.support.net
내보내기	+	Type	가져오기
		Request Code	2002
		사용여부	<input checked="" type="checkbox"/>
		Client ID	eSignon
		Request Parameter	Response Parameter
		• name 홍길동	• koreaid 123456-789101
			• address 서울특별시 OO
			• position 차장



• Request Parameter

* Test = 설정된 내용으로 가져오기를 시도한다.









No.	ID	VALUE
Default	wfuid	결재 ID
Default	wfluid	결재 Lib ID
Default	processid	프로세스 ID
Default	actionid	액션 ID
Default	useremail	유저 ID
1	name	홍길동

2. 가져오기

‘’ 버튼을 클릭하여 API를 테스트해볼 수 있습니다.

문서 1 < Process1 - Action1 >  

Workflow [API](#)

가져오기	+	Name	재직증명서
• 재직증명서		URL	http://esignon.support.net
내보내기	+	Type	가져오기
		Request Code	2002
		사용여부	
		Client ID	eSignon
		Request Parameter	<div>+</div> 
		• name	홍길동  
		Response Parameter	<div>• koreaid</div> 123456-789101 
		• address	서울특별시 OO 
		• position	차장 

• Response Parameter

외부에서 가져온 데이터가 채워진다.

No.	ID	VALUE
1	koreaid	123456-7891012
2	address	서울특별시 OO구 OO동
3	position	차장

2. 가져오기 예시 화면

값을 가져오면 아래와 같이 표시됩니다.

API

API

1 / 1

가져오기

+

재직증명서

×

Request Parameter

name

수정

Response Parameter

koreaid

수정

address

수정

position

수정

내보내기

+

재직증명서

성명	홍길동	주민등록번호	123456-7891012
주소	서울특별시 00구 00동 000번지		
직장명		직장전화번호	
직장주소		직위	차장
근무기간			

상기와 같이 재직하고 있음을 증명합니다.

년월일

2. 가져오기 예시 JSON

Request JSON Sample

: eSignon에서 고객님의 API를 호출하는 JSON 형식입니다. 관련 변수는 9페이지에서 설정한 변수들이 오게 됩니다.(아래 파랑색 부분)

```
{
  "header": {
    "request_code": "2002",      // 8페이지 화면에서 설정한 'Request Code'값
    "api_name": "재직증명서",    // 8페이지 화면에서 설정한 'Name'값
    "session_id": "0234444660"  // 고정된 값(Request, Response 간의 구분을 위한 값)-현재 사용하지 않음.
  },
  "body": {
    "wfluid": "123",
    "wfuid": "867",
    "processid": "1",
    "actionid": "1",
    "clientid": "eSignon",
    "useremail": "guid@esignon.net",
    "name": "홍길동"             // 9페이지 화면에서 추가한 것으로 고객님의 테이블에서 조회 할 Key값(문서에 사용자가 입력한 값)
  }
}
```

2. 가져오기 예시 JSON

Response JSON Sample

```
{
  "header": {
    "response_code" : "2002",      // 12페이지에서 받은 값을 넣어준다.(request_code값)
    "result_code" : "00",         // 00=성공, 10=실패
    "result_msg" : "처리되었습니다", // 고정된 값(성공="처리되었습니다." /실패="실패되었습니다.")
    "session_id": "0234444660"    // 고정된 값
  },
  "body": {
    "wfluid": "123",
    "wfuid": "867",
    "processid": "1",
    "actionid": "1",
    "clientid": "eSignon",
    "useremail": "guid@esignon.net",
    "name": "홍길동",
    "koreaid": "123456-7891012",    // 주민등록번호 (10페이지에서 추가한 항목)
    "address": "서울특별시 OO구 OO동 000번지", // 주소 (10페이지에서 추가한 항목)
    "position": "차장"             // 직위 (10페이지에서 추가한 항목)
  }
}
```

3. 내보내기

작성한 데이터를 내보내고자 할 때 사용하는 API입니다. 해당 Action이 완료 되었을 때 호출됩니다.
네트워크 불안(eSignon 서버와 카카오페이지 서버)으로 통신 실패 시 전달 내용을 eSignon서버에 내용을 저장 하고 있다가,
5분 스케줄러가 동작하여 성공 시 까지 호출합니다.

문서 1
< Process1 - Action1 >
↑↓ ×

Workflow		API	
가져오기	+	Name	휴가계
내보내기	+	URL	http://gw.jcone.co.kr
• 휴가계	⊗	Type	내보내기
		Request Code	2002
		사용여부	<input checked="" type="checkbox"/>
		Client ID	eSignon
		Request Parameter	Response Parameter
• reason	예비군 훈련날	⚙	⊗
• startdate	201712250000	⚙	⊗
• enddate	201712252359	⚙	
• type	ALL	⚙	
• public_yn	Y	⚙	

* 예시

예비군 훈련으로 인한 공가를 신청하는 휴가계 작성

- 1) Name = 해당 내보내기 API의 이름
- 2) URL = 통신 URL
- 3) Type = 내보내기 (고정)
- 4) 사용여부 = 해당 내보내기 API를 사용할지 여부
- 5) Request Code = 고객님이 정한 임의의 값
- 6) Client ID = eSignon Service ID 사용

3. 내보내기

‘+’ 버튼을 클릭하여 가져올 parameter를 추가합니다.
톱니바퀴 버튼을 클릭하여 가져와서 보여주고자 하는 컨트롤을 클릭하여 매칭합니다.

문서 1

< Process1 – Action1 >

↕ ×

Workflow

API

가져오기	+	Name	휴가계
내보내기	+	URL	http://gw.jcone.co.kr
• 휴가계	⊗	Type	내보내기
		Request Code	2002
		사용여부	<input checked="" type="checkbox"/>
		Client ID	eSignon
		Request Parameter	
		Response Parameter	
• reason	예비군 훈련날	⚙	⊗
• startdate	201712250000	⚙	⊗
• enddate	201712252359	⚙	
• type	ALL	⚙	
• public_yn	Y	⚙	


• Request Parameter

* Test = 설정된 내용으로 내보내기를 시도한다.

No.	ID	VALUE
Default	wfuid	결재 ID
Default	wfluid	결재 Lib ID
Default	processid	프로세스 ID
Default	actionid	액션 ID
Default	useremail	유저 ID
Default	workdatetime	결재 시간
Default	opinion	결재 의견
Default	worktype	승인/반려
1	reason	예비군 훈련날입니다.
2	startdate	20171225000000
3	enddate	20171225235959
4	type	ALL
5	public_yn	Y

3. 내보내기

‘’ 버튼을 클릭하여 API를 테스트해볼 수 있습니다.

문서 1		< Process1 - Action1 >		↕ ×		
Workflow		API				
가져오기	+	Name	휴가계			
내보내기	+	URL	http://gw.jcone.co.kr			
<ul style="list-style-type: none"> • 휴가계 	⊗	Type	내보내기	Request Code	2002	
		사용여부	<input checked="" type="checkbox"/>	Client ID	eSignon	
		Request Parameter		+		Response Parameter
		• reason	예비군 훈련날	⚙	⊗	
		• startdate	201712250000	⚙	⊗	
		• enddate	201712252359	⚙		
		• type	ALL	⚙		
		• public_yn	Y	⚙		

- **Response Parameter**

외부에서 응답에 해당하는 결과를 알려준다.

3. 내보내기 예시 JSON

Request JSON Sample

: eSignon에서 고객님의 API를 호출하는 JSON 형식입니다. 관련 변수는 15페이지에서 설정한 변수들이 오게 됩니다.(아래 파랑색 부분)

```
{
  "header": {
    "request_code": "2002",           // 14페이지 화면에서 설정한 'Request Code' 값
    "api_name": "휴가계",             // 14페이지 화면에서 설정한 'Name' 값
    "session_id": "02344444660"       // 고정된 값(Request, Response 간의 구분을 위한 값)-현재 사용하지 않음.
  },
  "body": {
    "wfuid": "637",
    "wfluid": "123",
    "processid": "1",
    "actionid": "1",
    "useremail": "abc@jcone.co.kr",
    "clientid": "eSignon",
    "workdatetime": "20171220123050", // 결재 요청일시(yyyymmddhh24miss)
    "opinion": "공가 요청합니다.",    // 결재자 의견
    "worktype": "CF",                 // CF=승인, RT=반려
    "reason": "예비군 훈련날입니다.", // 휴가사유
    "startdate": "20171225000000",    // 휴가 시작일시
    "enddate": "20171225235959",      // 휴가 종료일시
    "type": "ALL",                    // 휴가 종류 : AM=오전반차, PM=오후반차, ALL=종일
    "public_yn": "Y"                  // 공가 여부 : Y=공가임, N=공가아님
  }
}
```

3. 내보내기 예시 JSON

Response JSON Sample

: 17페이지에서 생성한 JSON 형식 데이터를 고객님의 받아 처리한 후, eSignon에 결과를 알려주는 JSON 형식입니다.
성공여부를 'header'의 'result_code'에 넣어준다. (00=성공, 10=실패)
성공할 경우 해당 건에 대해서 처리완료를 하고, 실패할 경우 eSignon 스케줄러에 의해서 호출하게 된다.

```
{
  "header": {
    "response_code": "2002",      // 17페이지에서 받은 값을 넣어준다.(request_code값)
    "result_code": "00",         // 00=성공, 10=실패
    "result_msg": "처리되었습니다", // 고정된 값(성공="처리되었습니다." /실패="실패되었습니다.")
    "session_id": "0234444660"   // 고정된 값
  },
  "body": {
    "wfluid": "637",
    "wfluid": "123",
    "processid": "1",
    "actionid": "1",
    "clientid": "eSignon",
    "useremail": "abc@jcone.co.kr",
    "workdatetime": "20171220123050", // 결재 요청일시(yyyymmddhh24miss)
    "opinion": "공가 요청합니다.",    // 결재자 의견
    "worktype": "CF",                 // CF=승인, RT=반려
    "result": "success"               // 결과(success=성공, fail=실패)
  }
}
```

3. 내보내기 예시 화면

값을 내보낼 때 아래와 같은 화면으로 표현됩니다.

API

API

가져오기

+

내보내기

+

• 휴가계

✕

Request Parameter

• reason

수정

• startdate

수정

• enddate

수정

• type

수정

• public_yn

수정

1 / 1

휴가계

작성일 :

부서 :

이름 :

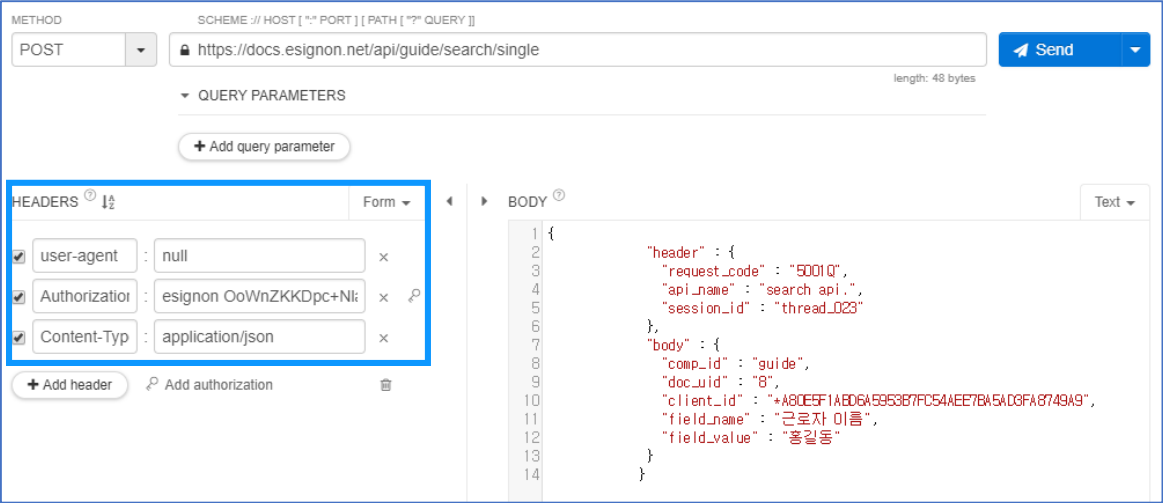
	담당	팀장	사업부장
결재			

휴가 종류	전일	<input checked="" type="checkbox"/> 공가 (결혼, 예비군훈련, 회갑 등)
휴가 사유	예비군 훈련날입니다.	
휴가 기간	201712250000 - 201712252359 (1 일간)	
비상연락처		

위와 같이 휴가를 신청하오니 허가하여 주시기 바랍니다.

4. 필드 검색하기

eSignon에서 제공하는 워크플로를 사용하여 완료된 워크플로우 데이터를 필드 이름과 값으로 조회할 때 호출합니다.
HEADERS 부분에 user-agent / Authorization / Content-Type 항목을 추가해야 합니다.
Authorization값에는 “esignon {access_token값}”형태이다. 중간에 빈칸이 삽입된다.



• Request Parameter

Type	Contents
Method	POST
Scheme	https://docs.esignon.net/api/{CompID}/search/single
user-agent	esignonapi
Authorization	esignon {사용자 토큰값}
Content-Type	application/json
Body	{ "header": { "request_code": "5001Q", "api_name": "search api.", "session_id": "thread_023" }, "body": { "comp_id": "{CompID}", "doc_uid": "{문서 ID}", "client_id": "{사용자 ID}", "field_name": "{필드 이름}", "field_value": "{필드 값}" } }

4. 필드 검색하기 예시 JSON

Response JSON

Response

Cache Detected - Elapsed Time: 132ms

200 OK

HEADERS [12]

BODY ②

pretty ▼

```
{
  header : {
    response_code : "5001A",
    result_code : "00",
    result_msg : "검색 완료되었습니다.",
    session_id : ""
  },
  body : {
    갑_대표자 : "이사인",
    갑_연락처 : "02-1234-5678",
    갑_주소 : "서울시 금천구 벚꽃로 234 에이스빌딩 1708호",
    갑_회사명 : "(주)이사인문",
    계약시작일 : "2018-03-13",
    계약종료일 : "2019-03-12",
    근로자 이름 : "홍길동",
    월급 : "월 2,500,000원",
    을_서명 : "",
    을_성명 : "",
    을_연락처 : "010-1234-5678",
    을_주소 : "",
    직무 이름 : "영업",
    직무이름 : "홍길동",
    직위 : ""
  }
}
```

lines nums

length: 558 bytes

5. 날짜별 검색하기

eSignon에서 제공하는 워크플로를 사용하여 완료된 워크플로우 데이터를 날짜별로 조회할 때 호출합니다.
HEADERS 부분에 user-agent / Authorization / Content-Type 항목을 추가해야 합니다.
Authorization값에는 “esignon {access_token값}”형태이다. 중간에 빈칸이 삽입된다.

METHOD

POST

SCHEME // HOST [":" PORT] [PATH ["?" QUERY]]

https://docs.esignon.net/api/guide/search/date

length: 46 bytes

Send

QUERY PARAMETERS

+

Add query parameter

HEADERS

Form

☒ user-agent

:

null

x

☒ Authorization

:

esignon OoWnZKKDpc+Nlk

x

☒ Content-Type

:

application/json

x

+

Add header

Add authorization

BODY

Text

```
1 {
2     "header" : {
3         "request_code" : "5002Q",
4         "api_name" : "search api.",
5         "session_id" : "thread_023"
6     },
7     "body" : {
8         "comp_id" : "guide",
9         "doc_uid" : "8",
10        "client_id" : "*A80DEF1ABD6A5953B7FC54AEE7BA5ACD8FA8749A9",
11        "start_date" : "2014-10-10",
12        "end_date" : "2018-10-10"
13    }
14 }
```

• Request Parameter

Type	Contents
Method	POST
Scheme	https://docs.esignon.net/api/{CompID}/search/date
user-agent	esignonapi
Authorization	esignon {사용자 토큰값}
Content-Type	application/json
Body	<pre>{ "header" : { "request_code" : "5002Q", "api_name" : "search api.", "session_id" : "thread_023" }, "body" : { "comp_id" : "{CompID}", "doc_uid" : "{문서 ID} ", "client_id" : "{사용자 ID} ", "start_date" : "{시작 날짜}", "end_date" : "{완료 날짜}" } }</pre>

5. 날짜별 검색하기 예시 JSON

Response JSON

Response

Cache Detected - Elapsed Time: 120ms

200 OK

HEADERS [12]

BODY

pretty

```
{
  header : {
    response_code : "5002A",
    result_code : "00",
    result_msg : "검색 완료되었습니다.",
    session_id : ""
  },
  body : {
    docinfo_list : [
      {
        memb_email : "guide@esignon.net",
        field_list : [
          {
            을_설명 : "홍길동",
            계약시작일 : "2018-03-13",
            직무 이름 : "영업",
            직무어름 : "홍길동",
            계약종료일 : "2019-03-12",
            근로자 이름 : "홍길동",
            을_주소 : "경기도 광명시 하안동 ",
            간_대표자 : "이사인",
            간_회사명 : "(주)이사인온",
            을_연락처 : "010-1234-5678",
            직위 : "과장",
            간_연락처 : "02-1234-5678",
            월급 : "월 2,500,000원",
            을_서명 : "3/13/2018 7:07:00 PM",
            간_주소 : "서울시 금천구 벚꽃로 234 에이스빌딩 1708호"
          }
        ],
        end_time : null,
        start_time : "2018-03-13 10:02:33",
        wfuid : "27"
      },
      { memb_email : "guide@esignon.net", field_list : [ { 을_설명 : "신지수", 계약시작일 : "2018-03-15",--},
      { memb_email : "guide@esignon.net", field_list : [ { 을_설명 : "김영남", 계약시작일 : "2018-03-13",--},
      { memb_email : "guide@esignon.net", field_list : [ { 을_설명 : "dhdhd", 계약시작일 : "2018-03-13",--}
    ]
  }
}
```

lines nums

length: 2 kilobytes

6. 시작하기

eSignon에서 제공하는 워크플로우를 시작할 때 호출합니다. 휴대폰 본인인증 기능이 포함되어 있습니다. HEADERS 부분에 user-agent / Authorization / Content-Type 항목을 추가해야 합니다.

Authorization값에는 “esignon {access_token값}” 형태로 중간에 빈칸이 삽입됩니다.

– access_token값은 1.사용자 토큰 페이지를 참고하여 로그인한 사용자의 토큰이 발생

version = 1.1.57 // 버전체크를 위해 추가된 내용으로 Default로 사용합니다.

memb_id_type // EMail로 보내는 경우 “EMail”, 휴대폰인 경우 “Mobile”(대소문자구분)

language // ko-KR(한글), en-US(영어), ja-JP(일본어) 중 선택입력(대소문자구분)

workflow_name //개별 workflow name을 지정하는 경우 사용합니다.

개별 workflow_name은 목록으로 표시되며 검색 조건으로 활용됩니다.

enable_mobile_cert // 휴대폰 본인인증을 사용 여부에 따라 “true/false” 선택합니다.

mobile_number // 휴대폰 본인인증에 사용할 휴대폰 번호를 입력합니다.

이메일로 보내는 경우에도 휴대폰 본인인증을 사용할 수 있습니다.

• Request Parameter

Type	Contents
Method	POST
Scheme	https://docs.esignon.net/api/{CompID}/startbulk
user-agent	esignonapi
Authorization	esignon {사용자 토큰값}
Content-Type	application/json; charset=utf-8
Body	<pre>{ "header": { "request_code": "5004Q", "api_name": "start api", "session_id": "S1001", "version": "1.1.57" }, "body": { "comp_id": "guide", "biz_id": "0", "memb_email": "guide@esignon.net", "workflow_lib_id": "58", "workflow_name": "가맹계약_api_test", "client_id": "**A80E5F1ABD6A5953B7FC54AEE7BA5AD3FA8749A9", "unset_player_list": [{ "memb_id_type": "EMail", "email": "hyo1@icone.co.kr", "name": "오효중", "language": "ko-KR", "workflow_name": "가맹계약_apiTest_오효중", "enable_mobile_cert": "false", "mobile_number": "", "field_list": [{ "doc_id": "17", "field_name": "상호명", "field_value": "제이씨원" }] }, { "memb_id_type": "Mobile", "email": "01035934047", "name": "오효중", "language": "ko-KR", "workflow_name": "가맹계약_apiTest_오효중_모바일", "enable_mobile_cert": "false", "mobile_number": "", "field_list": [{ "doc_id": "17", "field_name": "상호명", "field_value": "제이씨원_모바일" }] }], "comment": "" } }</pre>

6. 시작하기 예시 JSON

Response JSON

Response

200 OK

BODY ⓘ

```
{
  header : {
    session_id : "S1001",
    response_code : "5004A",
    result_code : "00",
    result_msg : "Work Flow bulk Sending이 시작됩니다.",
    version : "1.1.57"
  },
  body : {
    workflow_list : [
      {
        workflow_id : "780",
        unset_player_email : "hyo1@jcone.co.kr",
        unset_player_name : "오효중",
        result : "OK",
        workflow_name : "가맹계약_apiTest_오효중",
        token : "1D0FCzXAM8evywsPKKLh6Kq38Tf9ErPpmnMcXE7r%2Fic%3D",
        lang : "ko-KR"
      },
      { workflow_id : "780", unset_player_email : "01035934047", unset_player_name : "오효중", result : "OK",...}
    ],
    comp_id : "guide",
    biz_id : "0",
    memb_email : "guide@esignon.net",
    workflow_lib_id : "58"
  }
}
```

7. Web 서명하기 페이지 호출

eSignon에서 제공하는 워크플로우를 시작 후 Web 화면에서 서명하기 위한 화면을 호출하는 API입니다.

화면을 호출할때에는 javascript의 window.open 함수를 사용하여 POST 방법으로 호출하여야 하며, 서명이 완료된 후에는 호출한 페이지로 javascript event를 전달해줍니다.

• Request Parameter

Type	Contents
Method	POST
Scheme	https://docs.esignon.net/api/{CompID}/sign
parameters	
client_id(필수)	고객에게 전달된 unique key 값
token(필수)	6. 시작하기 후 전달받은 response의 body.workflow_id_list[n].token값
lang(옵션)	6. 시작하기 후 전달받은 response의 body.workflow_id_list[n].lang값의 앞에 두자리 ko-KR → ko , en-US → en
approval_pop(옵션)	true(결재 후 승인반려 팝업 호출) / false(결재 후 서명완료 아이콘 클릭시 프로세스 종료)
callback_fn (옵션)	true(결재 완료 후 서명하기 페이지를 호출한 페이지로 완료 이벤트 전달)

• 호출순서

1. 사용자 토큰 얻기

<https://docs.esignon.net/api/{CompID}/login>



6. 시작하기 API 호출

<https://docs.esignon.net/api/{CompID}/startbulk>



7. Web 서명하기 Page 호출

<https://docs.esignon.net/api/{CompID}/sign>

• 서명 완료 후 전달받을 이벤트 추가

- 서명하기 페이지를 호출한 페이지(고객 소스)에 javascript event listener 소스 추가, 결재가 성공하면 message listener event 발생

```
<script type="text/javascript">
window.addEventListener('message', function(e) {
  console.log(e.data);
});
</script>
```

8. 간편 계약 (비대면)

eSignon에서 제공하는 심플 워크플로우를 시작할 때 호출합니다.

HEADERS 부분에 user-agent / Authorization / Content-Type 항목을 추가해야 합니다.

Authorization값에는 “esignon {access_token값}” 형태로 중간에 빈칸이 삽입됩니다.

version = 1.1.58 // 버전체크를 위해 추가된 내용으로 Default로 사용합니다.

memb_id_type // EMail로 보내는 경우 “EMail”, 휴대폰인 경우 “Mobile”

workflow_name // workflow name을 입력합니다.

enable_mobile_cert // 휴대폰 본인인증을 사용 여부에 따라 “true/false” 선택합니다.

mobile_number // 휴대폰 본인인증에 사용할 휴대폰 번호를 입력합니다.

이메일로 보내는 경우에도 휴대폰 본인인증을 사용할 수 있습니다.

api_type // Export API 타입을 선택할 수 있습니다.

모든 단계에서 받으려면 ‘ALL’

시작과 마지막 단계에서만 받으려면 ‘StartAndEnd’를 선택해주세요.

• Request Parameter

Type	Contents
Method	POST
Scheme	https://docs.esignon.net/api/{CompID}/starsimple
user-agent	esignonapi
Authorization	esignon {사용자 토큰값}
Content-Type	application/json; charset=utf-8
Body	<pre>{ "header": { "request_code": "5005Q", "api_name": "start api", "session_id": "S1001", "version": "1.1.58" }, "body": { "comp_id": "guide", "biz_id": "0", "client_id": "A80E5F1ABD6A5953B7FC54AE7B5A03FA8749A9", "memb_email": "guide@esignon.net", "workflow_name": "api_test", "doc_id": "8", "language": "ko-KR", "player_list": [{ "field_owner": "1", "email": "guide@esignon.net", "id_type": "EMail", "name": "가이드", "language": "ko-KR", "enable_mobile_cert": "false", "mobile_number": "" }, { "field_owner": "2", "email": "01090381894", "id_type": "Mobile", "name": "심상용", "language": "ko-KR", "enable_mobile_cert": "false", "mobile_number": "" }] }, "field_list": [{ "doc_id": "8", "field_name": "근로자 이름", "field_value": "심상용" }], "export_api_info": { "api_type": "StartAndEnd", "url": "http://gw.jcone.co.kr:8090/espi", "enable": "true", "request_code": "2003Q", "client_id": "0234444660", "request_params": { { "param_id": "name", "param_value": "", "fields": { { "doc_id": "8", "field_name": "근로자 이름" } } }, { "param_id": "date", "param_value": "", "fields": { { "doc_id": "8", "field_name": "근로자 이름_1" } } }], "response_params": [], "comment": "" } }</pre>

Response JSON

Response

200 OK

HEADERS

pretty

Server: Apache-Coyote/1.1
Access-Control-Allow-Methods: GET, POST, PUT, DELETE, OPTIONS
Access-Control-Max-Age: 3600
Access-Control-Allow-Headers: Authorization, Content-Type
Access-Control-Allow-Origin: *
Access-Control-Expose-Headers: Content-Disposition, filename
Set-Cookie: JSESSIONID=33 B, session, HttpOnly
Content-Type: application/json; charset=UTF-8
Transfer-Encoding: chunked
Date: Mon, 07 Jan 2019 00:45:34 GMT +1s

▶ COMPLETE REQUEST HEADERS

BODY

```
{
  header : {
    session_id : "S1001",
    response_code : "5005A",
    result_code : "00",
    result_msg : "Work Flow가 시작됩니다.",
    version : "1.1.58"
  },
  body : {
    comp_id : "guide",
    biz_id : "0",
    memb_email : "guide@esignon.net",
    workflow_name : "api_test",
    token : "ZF4uahZpDBv1iMF0AucbCMAwWly2%2F9YIwuzqJslPOysan%2Fvd3iobw%3D%3D",
    lang : "ko-KR"
  }
}
```

lines nums

8. 간편 계약 (대면)

eSignon에서 제공하는 심플 워크플로우를 시작할 때 호출합니다.

HEADERS 부분에 user-agent / Authorization / Content-Type 항목을 추가해야 합니다.

Authorization값에는 “esignon {access_token값}” 형태로 중간에 빈칸이 삽입됩니다.

version = 1.1.58 // 버전체크를 위해 추가된 내용으로 Default로 사용합니다.

memb_id_type // EMail로 보내는 경우 “EMail”, 휴대폰인 경우 “Mobile”

workflow_name //workflow name을 입력합니다.

• Request Parameter

Type	Contents
Method	POST
Scheme	https://docs.esignon.net/api/{ComplD}/starsimpleface
user-agent	esignonapi
Authorization	esignon {사용자 토큰값}
Content-Type	application/json; charset=utf-8
Body	<pre>{ "header": { "request_code": "50070", "api_name": "start api", "session_id": "S1001", "version": "1.1.58" }, "body": { "comp_id": "guide", "biz_id": "0", "client_id": "A80E5F1ABD6A5953B7FC54AEE7BA5A3FA8749A9", "memb_email": "guide@esignon.net", "workflow_name": "api_test", "doc_id": "8", "language": "ko-KR", "field_list": [{ "doc_id": "8", "field_name": "근로자 이름", "field_value": "심상용" }], "customer_list": [{ "email": "01090381894", "id_type": "Mobile", "name": "심상용", "language": "ko-KR" }], "export_api_info": { "url": "http://gw.jcone.co.kr:8090/espi", "enable": "true", "request_code": "2003Q", "clientid": "0234444660", "request_params": [{ "param_id": "name", "param_value": "", "fields": [{ "doc_id": "8", "field_name": "근로자 이름" }] }, { "param_id": "date", "param_value": "", "fields": [{ "doc_id": "8", "field_name": "근로자 이름_1" }] }], "response_params": [] }, "comment": "" } }</pre>

8. 간단 계약 예시 JSON

Response JSON

Response

200 OK

HEADERS

Server: Apache-Coyote/1.1
Access-Control-Allow-Methods: GET, POST, PUT, DELETE, OPTIONS
Access-Control-Max-Age: 3600
Access-Control-Allow-Headers: Authorization, Content-Type
Access-Control-Allow-Origin: *
Access-Control-Expose-Headers: Content-Disposition, filename
Set-Cookie: JSESSIONID=33 B, session, HttpOnly
/
Content-Type: application/json;charset=UTF-8
Transfer-Encoding: chunked
Date: Mon, 07 Jan 2019 05:37:55 GMT +2s

▶ COMPLETE REQUEST HEADERS

pretty ▼

BODY

```
{
  header : {
    session_id : "S1001",
    response_code : "5007A",
    result_code : "00",
    result_msg : "Work Flow가 시작됩니다.",
    version : "1.1.58"
  },
  body : {
    comp_id : "guide",
    biz_id : "0",
    memb_email : "guide@esignon.net",
    workflow_name : "api_test",
    token : "Qy570Ie2sPqFJ5Tz%2FEJ9FLkiHjKvsA8M1XA7TB4HC6G4T8FUJuYp6w%3D%3D",
    lang : "ko-KR"
  }
}
```



“Contents is Everything, Everything is Contents”