

GUARDIAN

OpenAPI 명세서

거래 위험도 평가 API
요청·응답 및 연동 가이드

POST /api/v1/sites/{code}/api-risk-score

API 0.1.0 / OpenAPI 3.1.0

1개 경로 · 1개 작업 · 5개 고유 스키마

거래신호 전송과 동기 위험도 평가를 위한 단일 API 명세입니다.

이 PDF는 거래 위험도 평가 API 한 개와 해당 API가 사용하는 스키마만 수록합니다. PDF 첨부 JSON도 같은 범위로 제한됩니다. 목차 링크와 PDF 북마크로 상세 설명과 스키마를 확인하세요.

명세에 명시된 내용만 수록합니다. 실제 운영 주소·사이트 키·접근 권한은 운영 담당자에게 확인하세요.

원본 SHA-256: 313315e1fea91dad60c6928e2e057d28b2c5ae38033b947899b852856517e87b

문서 안내

서버와 인증

<https://api.fdsguard.co.kr>

가디언 API 서버

ApiKey: apiKey / header / x-api-key

읽는 방법

각 API에는 인증, 경로·쿼리·헤더 매개변수, 본문, 응답 코드를 표시합니다. S로 시작하는 스키마 번호를 클릭하면 부록의 상세 필드와 제약으로 이동합니다.

스키마에서 required는 필수 속성 이름 목록입니다. anyOf/oneOf/allOf는 각각 하나 이상/정확히 하나/전체 조건을 뜻합니다. additionalProperties=false는 정의되지 않은 속성을 금지합니다. 명세에 없는 인증 조건은 별도로 추정하지 않습니다.

참조 스키마 이름

Callback → [S005](#)

연동 가이드

1. 주소와 인증

GUARDIAN_BASE_URL=https://api.fdsguard.co.kr

GUARDIAN_SITE_CODE=발급받은 사이트 코드

GUARDIAN_API_KEY=발급받은 사이트 API 키

POST /api/v1/sites/\$GUARDIAN_SITE_CODE/api-risk-score

{code}는 GUARDIAN_SITE_CODE로 치환합니다. Content-Type: application/json과 x-api-key 헤더를 전송합니다. 개발 테스트에서는 본문에 mode: development를 명시하세요. API 기본값은 production입니다.

2. 거래만 전송 / 가맹점 포함 전송

transaction은 필수이며 merchant는 선택입니다. 가맹점 정보가 없다면 merchant 객체 전체를 생략합니다. 가맹점 정보를 보낼 때는 externalMerchantId와 merchantName을 반드시 포함합니다. 나머지 가맹점 속성은 선택입니다.

deviceId는 거래 간 동일한 단말기 식별자를 유지합니다. mid는 PG 채널 식별자, order_no는 주문 거래 번호, tid는 PG 거래 ID입니다. paidAt은 한국시간 yyyyMMddHHmmss 14자리의 유효한 일시입니다. amount는 0보다 커야 하며 KRW/JPY는 정수, USD/EUR는 소수 둘째 자리까지 허용합니다.

필수 거래 속성: paidAt, channel, amount, mid, order_no, tid, paymentType, paymentStatus, currency, deviceId. 각 자료형·길이·허용값은 상세 스키마에서 확인하세요. cardNo는 마스킹 형식만 허용합니다. 정의되지 않은 속성은 전송하지 마세요.

3. 멍등성과 재전송

멍등 키는 Idempotency-Key 헤더 > 본문 eventId > TID:결제상태 순서로 선택합니다. 헤더는 1~180자, 본문 eventId는 1~128자입니다. 본문 eventId는 호출자가 정하는 키이며 응답 eventId는 서버 이벤트 식별자입니다.

같은 사이트·mode에서 같은 키와 같은 본문을 재전송하면 중복 접수로 처리하고 추가 사용량을 소비하지 않습니다. 중복 응답은 현재 상태가 completed 또는 failed여도 HTTP 202입니다. 같은 키로 본문을 바꾸면 409 충돌이 발생합니다.

응답 유실·시간 초과로 결과가 불명확하면 최초 키와 본문 전체를 그대로 재전송하세요. 이때 paidAt도 바꾸지 않습니다. 새로운 거래를 테스트할 때만 새 식별자와 현재 시각을 사용하세요.

4. 취소 거래 연결

paymentStatus: CANCELED는 전체 취소입니다. 원 승인과 연결할 주문·단말기·채널 정보를 일관되게 보내고 cancellationLinkStatus와 originalTransactionId로 결과를 확인하세요. 부분 취소를 표현하는 별도 필드는 없습니다.

cancellationLinkStatus: not_applicable(연결 대상 아님), linked(연결됨), pending(원 승인 대기), ambiguous(후보 중복), mismatch(정보 불일치), duplicate(중복 취소).

5. 응답 해석과 오류 처리

200 completed는 거래·판정 저장이 커밋된 동기 완료입니다. score는 0~100, risk는 normal / review / high입니다. results에는 규칙별 hit / clear / unavailable / disabled 판정이 포함됩니다. merchantIdentityStatus는 provisional / identified / legacy_unknown 중 하나입니다.

202는 접수 또는 중복 결과입니다. status는 accepted / processing / completed / failed이며 항상 완료를 의미하지 않습니다. 전환 전 legacy 사이트는 기존 202 계약을 유지합니다. 응답의 statusUrl 필드는 계약에 포함되지만 별도 조회 API는 이

문서의 범위에 포함하지 않습니다.

401: API 키 확인. 403: 계약·인증 상태 확인. 409: 키/본문 충돌 해소. 422: issues의 path와 message를 확인하고 입력 수정. 429: Retry-After 초만큼 기다린 뒤 재시도. 503 또는 네트워크 오류: 지수 백오프를 적용하고 같은 키와 본문으로 재시도하세요.

거래만 전송하는 예제

아래 JSON을 request.json으로 저장합니다. paidAt과 거래 식별자는 실제 거래 값으로 바꾸세요. 재시도 시에는 최초 값을 유지합니다. 가맹점을 포함한 본문은 다음 API 상세 예제를 참고하세요.

```
{
  "transaction": {
    "paidAt": "20260710091200",
    "amount": 850000,
    "mid": "111",
    "order_no": "222",
    "tid": "T202607120001",
    "auth_no": "12345678",
    "paymentType": "CARD_BILL",
    "cardNo": "557042*****2076",
    "paymentStatus": "APPROVED",
    "currency": "KRW",
    "deviceId": "device-abc-123",
    "dailyCount": 12,
    "settlementDay": "D+1",
    "channel": "ONLINE"
  },
  "mode": "development",
  "eventId": "sample-order-001-approved"
}
```

cURL 요청 (셸 환경변수 설정 후 실행)

```
curl -X POST "$GUARDIAN_BASE_URL/api/v1/sites/$GUARDIAN_SITE_CODE/api-risk-score" \#
-H "Content-Type: application/json" \#
-H "x-api-key: $GUARDIAN_API_KEY" \#
-H "Idempotency-Key: sample-order-001-approved" \#
--data-binary @request.json
```

API 목차

POST /api/v1/sites/{code}/api-risk-score

단말기 거래신호를 동기로 평가합니다.

API 상세

POST /api/v1/sites/{code}/api-risk-score

단말기 거래신호를 동기로 평가합니다.

거래·판정 저장 커밋 후 200 completed를 반환합니다. merchant 선택, deviceId-mid-order_no 필수. mode 누락 시 production. CANCELED는 전체 취소이며 원 승인 연결 상태를 반환합니다. 기존 legacy 사이트는 전환 전 202 계약을 유지합니다. 멍등 키 우선순위는 Idempotency-Key 헤더, 본문 eventId, TID:결제상태 순서입니다. 사이트·mode 범위에서 동일 키/본문은 중복 접수이며 추가 사용량을 소비하지 않습니다. 현재 처리 상태가 completed/failed여도 중복 접수 응답 HTTP 상태는 202입니다.

인증: ApiKey

tags: ["PG callbacks"]

매개변수

code / path / 필수

schema: {"type": "string"}

Idempotency-Key / header / 선택

schema: {"type": "string", "minLength": 1, "maxLength": 180}

요청 본문 / 필수

Content-Type: application/json

상세 스키마: [S001 - 필드와 검증 조건 보기](#)

example

```
{
  "merchant": {
    "bizNo": "123-45-67890",
    "externalMerchantId": "mer_no",
    "merchantName": "루시전자 동탄점",
    "ownerName": "홍길동",
    "address": "경기도 화성시 동탄대로 100",
    "bizType": "전자제품 도소매",
    "phone": "010-1234-5678"
  },
  "transaction": {
    "paidAt": "20260710091200",
    "amount": 850000,
  }
}
```

```
"mid": "111",
"order_no": "222",
"tid": "T202607120001",
"auth_no": "12345678",
"paymentType": "CARD_BILL",
"cardNo": "557042*****2076",
"paymentStatus": "APPROVED",
"currency": "KRW",
"deviceId": "device-abc-123",
"dailyCount": 12,
"settlementDay": "D+1",
"channel": "ONLINE"
},
"mode": "production"
}
```

응답

200 / 동기 판정 완료 및 저장 결과

Content-Type: application/json

상세 스키마: [S002 - 필드와 검증 조건 보기](#)

202 / 접수 결과. 중복 요청은 기존 이벤트의 현재 상태를 반환합니다.

Content-Type: application/json

상세 스키마: [S003 - 필드와 검증 조건 보기](#)

401 / 인증 오류

Content-Type: application/json

상세 스키마: [S004 - 필드와 검증 조건 보기](#)

403 / 계약 정지 또는 미인증

Content-Type: application/json

상세 스키마: [S004 - 필드와 검증 조건 보기](#)

409 / 동일 먹등 키의 본문 충돌

Content-Type: application/json

상세 스키마: [S004 - 필드와 검증 조건 보기](#)

422 / 스키마 오류

Content-Type: application/json

상세 스키마: [S004 - 필드와 검증 조건 보기](#)

429 / 고객 합산 최근 60초 한도 또는 게이트웨이 IP 한도 초과

headers: {"Retry-After": {"schema": {"type": "integer"}, "description": "다시 요청하기 전 기다릴 초"}}

Content-Type: application/json

상세 스키마: [S004 - 필드와 검증 조건 보기](#)

503 / 내구성 저장 불가

Content-Type: application/json

상세 스키마: [S004 - 필드와 검증 조건 보기](#)

상세 스키마

S001 / Schema

\$

\$ref: "#/components/schemas/Callback"

S002 / Schema

\$

\$schema: "https://json-schema.org/draft/2020-12/schema" / type: "object" / required: ["eventId", "tid", "mode", "status", "duplicate", "score", "risk", "results", "deviceId", "merchantIdentityStatus", "cancellationLinkStatus", "originalTransactionId", "completedAt"] / additionalProperties: false

\$.eventId

type: "string"

\$.tid

type: "string"

\$.mode

type: "string" / enum: ["production", "development"]

\$.status

type: "string" / const: "completed"

\$.duplicate

type: "boolean"

\$.score

type: "integer" / minimum: 0 / maximum: 100

\$.risk

type: "string" / enum: ["normal", "review", "high"]

\$.results

type: "array"

\$.results[]

type: "object" / required: ["id", "name", "status", "score", "threshold", "reason", "version"] / additionalProperties: false

\$.results[].id

type: "string"

\$.results[].displayNumber

type: "integer" / exclusiveMinimum: 0 / maximum: 9007199254740991

\$.results[].name

type: "string"

\$.results[].status

type: "string" / enum: ["hit", "clear", "unavailable", "disabled"]

\$.results[].score

type: "number" / minimum: 0 / maximum: 100

\$.results[].observed

type: ["number", "string"]

`$.results[].threshold`

type: "number"

`$.results[].reason`

type: "string"

`$.results[].version`

type: "integer" / exclusiveMinimum: 0 / maximum: 9007199254740991

`$.deviceId`

type: "string"

`$.merchantIdentityStatus`

type: "string" / enum: ["provisional", "identified", "legacy_unknown"]

`$.cancellationLinkStatus`

type: "string" / enum: ["not_applicable", "linked", "pending", "ambiguous", "mismatch", "duplicate"]

`$.originalTransactionId`

`$.originalTransactionId.anyOf[0]`

type: "string" / format: "uuid" / pattern: "^([0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[1-8][0-9a-fA-F]{3}-[89abAB][0-9a-fA-F]{3}-[0-9a-fA-F]{12})00000000-0000-0000-0000-000000000000|ffffffff-ffff-ffff-ffff-fffffffffffff)\$"

`$.originalTransactionId.anyOf[1]`

type: "null"

`$.completedAt`

type: "string" / format: "date-time" / pattern: "^(?:[0-9]{4}-(?:[0-9]{2}-(?:[0-9]{2}-(?:[0-9]{2}-(?:[0-9]{2}-(?:[0-9]{2})))?)?)?)\$"

S003 / Schema

`$`

\$schema: "https://json-schema.org/draft/2020-12/schema" / type: "object" / required: ["eventId", "status", "duplicate", "tid", "mode", "statusUrl"] / additionalProperties: false

`$.eventId`

type: "string" / format: "uuid" / pattern: "^([0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[1-8][0-9a-fA-F]{3}-[89abAB][0-9a-fA-F]{3}-[0-9a-fA-F]{12})00000000-0000-0000-0000-000000000000|ffffffff-ffff-ffff-ffff-fffffffffffff)\$"

`$.status`

type: "string" / enum: ["accepted", "processing", "completed", "failed"]

`$.duplicate`

type: "boolean"

`$.tid`

type: "string"

`$.mode`

type: "string" / enum: ["production", "development"]

`$.statusUrl`

type: "string"

S004 / Schema

```
$
$schema: "https://json-schema.org/draft/2020-12/schema" / type: "object" / required: ["code", "message"] / additionalProperties: false
$.code
type: "string"
$.message
type: "string"
$.issues
type: "array"
$.issues[]
type: "object" / required: ["path", "message"] / additionalProperties: false
$.issues[].path
type: "array"
$.issues[].path[]
type: ["string", "number"]
$.issues[].message
type: "string"
```

S005 / Schema

```
$
$schema: "https://json-schema.org/draft/2020-12/schema" / type: "object" / required: ["transaction", "mode"] / additionalProperties: false
$.merchant
type: "object" / required: ["externalMerchantId", "merchantName"] / additionalProperties: false
$.merchant.externalMerchantId
type: "string" / minLength: 1 / maxLength: 160
$.merchant.merchantName
type: "string" / minLength: 1 / maxLength: 160
$.merchant.bizNo
type: "string" / minLength: 1 / maxLength: 32
$.merchant.ownerName
type: "string" / minLength: 1 / maxLength: 80
$.merchant.address
type: "string" / minLength: 1 / maxLength: 500
$.merchant.bizType
type: "string" / minLength: 1 / maxLength: 100
$.merchant.phone
type: "string" / minLength: 1 / maxLength: 30
$.transaction
type: "object" / required: ["paidAt", "channel", "amount", "mid", "order_no", "tid", "paymentType", "paymentStatus", "currency", "deviceId"] / additionalProperties: false
```

\$.transaction.paidAt

type: "string" / pattern: "^\\W\\d{14}\$"

\$.transaction.channel

type: "string" / enum: ["ONLINE", "OFFLINE"]

\$.transaction.amount

type: "number" / exclusiveMinimum: 0 / maximum: 999999999999

\$.transaction.mid

type: "string" / minLength: 1 / maxLength: 128

\$.transaction.order_no

type: "string" / minLength: 1 / maxLength: 128

\$.transaction.tid

type: "string" / minLength: 1 / maxLength: 128

\$.transaction.auth_no

type: "string" / minLength: 1 / maxLength: 40

\$.transaction.paymentType

type: "string" / enum: ["AUTH", "CARD_BILL", "CARD_READER", "MANUAL", "SMS"] / description: "결제수단 코드: AUTH=인증 결제, CARD_BILL=카드 빌링 결제, CARD_READER=카드 리더기 결제, MANUAL=수기 결제, SMS=문자 결제"

\$.transaction.cardNo

type: "string" / pattern: "^\\W\\d{4,6}\\W*{4,10}\\W\\d{4}\$"

\$.transaction.paymentStatus

type: "string" / enum: ["APPROVED", "CANCELED"]

\$.transaction.currency

type: "string" / enum: ["KRW", "USD", "JPY", "EUR"]

\$.transaction.deviceId

type: "string" / minLength: 1 / maxLength: 160

\$.transaction.dailyCount

type: "integer" / minimum: 0 / maximum: 9007199254740991

\$.transaction.settlementDay

type: "string" / minLength: 1 / maxLength: 20

\$.mode

default: "production" / type: "string" / enum: ["production", "development"]

\$.eventId

type: "string" / minLength: 1 / maxLength: 128