



# MiniProject\_LicensePlate

1.0.0 OAS 3.0

/v3/api-docs

차량 출입 기록 관리 서비스

Servers

http://localhost:8080 - Generated server url

Record API

차량 출입 기록 CRUD

^

PUT

/api/records

차량 출입 기록 수정(ADMIN)

^

AI 모델이 예측한 값이 틀린 경우, 관리자가 확인 후 수정, S3에 저장된 번호판 이미지 삭제, 재학습을 위해 차량 이미지 relearn 폴더로 이동

Parameters

Try it out

No parameters

Request body required

application/json

Example Value Schema

```
[
  {
    "recordId": 0,
    "licensePlate": "string"
  }
]
```

Responses

Code	Description	Links
200	<div>OK</div> <div>Media type<div>*/*</div><div>Controls Accept header.</div></div> <div>Example Value Schema<div>{}</div></div>	No links

POST /api/records 차량 출입 기록 저장

Flask 서버로 이미지 전달, AI 모델이 번호판 예측, S3에 이미지(차량, 번호판) 저장, 가장 정확도 높은 예측값 Record 인스턴스로 저장

Parameters

Try it out

Name	Description
<div>vehicleFile * required</div> <div>string(\$binary)</div> <div>(query)</div>	<div>파일 선택</div> 선택된 파일 없음

Responses

Code	Description	Links
200	<div>OK</div> <div>Media type<div>*/*</div><div>Controls Accept header.</div></div> <div>Example Value Schema<div>{   "record": {     "recordId": 0,     "modelType": "YOLO",</div></div>	No links

Code	Description	Links
	<pre>"licensePlate": "string", "accuracy": "string", "vehicleImage": "string", "plateImage": "string", "state": "string" }, "predictedList": [   {     "modelPredictResult": {       "modelType": "YOLO",       "predictedText": "string",       "accuracy": 0     },     "isEnrolled": "ENROLLED"   } ] }</pre>	

DELETE    /api/records    차량 출입 기록 삭제(ADMIN)    ^

Record 인스턴스 삭제(연결된 Image, PredictedPlate도 삭제됨), S3에 저장된 이미지 삭제

Parameters

Try it out

No parameters

Request body required

application/json

Example Value    Schema

```
[
  {
    "recordId": 0
  }
]
```

Responses

Code	Description	Links
200	OK	No links
	Media type <div><div>*/*</div></div> <div>Controls Accept header.</div>	
	Example Value    Schema	
	<pre>{}</pre>	

Code	Description	Links

GET

/api/records/plate/{plate} 차량 번호별 기록 조회

↗

차량 번호를 받아 기록을 조회

Parameters

Try it out

Name	Description
<b>plate</b> * required string (path)	<div>plate</div>

Responses

Code	Description	Links
200	OK  Media type <div>*/*</div> <div>Controls Accept header.</div> <div>Example Value Schema <pre>[ {   "recordId": 0,   "modelType": "YOLO",   "licensePlate": "string",   "accuracy": "string",   "vehicleImage": "string",   "plateImage": "string",   "state": "string" }]</pre></div>	No links

GET

/api/records/date/{start}/{end} 날짜별 기록 조회

↗

"yy-MM-dd" 형식의 날짜를 받아 기록 조회

Parameters

Try it out

Name	Description
<b>start</b> * required string (path)	<div>start</div>
<b>end</b> * required string (path)	<div>end</div>

Responses

Code	Description	Links
200	<div>OK</div> <div>Media type</div> <div><div>*/*</div></div> <div>Controls Accept header.</div> <div><div>Example Value</div><div>Schema</div></div> <div><pre>[   {     "recordId": 0,     "modelType": "YOLO",     "licensePlate": "string",     "accuracy": "string",     "vehicleImage": "string",     "plateImage": "string",     "state": "string"   } ]</pre></div>	No links

RefreshToken API

^

POST

/api/reissue

Access Token 재발행

^

Request header의 Refresh Token 검증 후, Access Token 재발행

Parameters

Try it out

Name	Description
<b>RefreshToken</b> * required string (header)	<div>RefreshToken</div>

Responses

Code	Description	Links
200	<div>OK</div> <div>Media type</div> <div><div>*/*</div></div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div><pre>{   "accessToken": "string" }</pre></div>	No links

Member API

^

POST

/api/members/sign-up

회원가입

^

아이디 중복 여부 검사 후, 비밀번호 인코딩하여 Role.USER로 저장

Parameters

Try it out

No parameters

Request body

required

application/json

Example Value

Schema

```
{
  "memberId": "string",
  "password": "string"
}
```

Responses		
Code	Description	Links
200	<div>OK</div> <div>Media type<div>*/*</div></div> <div>Controls Accept header.</div> <div><div>Example Value</div><div>Schema</div><div><pre>{  "accessToken": "string",  "refreshToken": "string"}</pre></div></div>	No links

POST

/api/members/sign-in 로그인

아이디, 비밀번호 검증 후, Refresh Token과 Access Token 발급

Parameters

Try it out

No parameters

Request body required

application/json

Example Value

Schema

```
{  "memberId": "string",  "password": "string"}
```

Responses		
Code	Description	Links
200	<div>OK</div> <div>Media type<div>*/*</div></div>	No links

Code	Description	Links
	Controls Accept header.	
	<b>Example Value</b> Schema	
	<pre>{   "accessToken": "string",   "refreshToken": "string" }</pre>	

# History API

GET	/api/histories 수정/삭제 기록 조회	
<b>Parameters</b>		<div>Try it out</div>
No parameters		
<b>Responses</b>		
<b>Code</b>	<b>Description</b>	<b>Links</b>
200	OK	No links
	Media type	
	<div>*/*</div>	
	Controls Accept header.	
	<b>Example Value</b> Schema	
	<pre>[   {     "historyId": 0,     "recordId": 0,     "memberId": "string",     "taskType": "DELETE",     "previousText": "string",     "currentText": "string",     "createdAt": "2023-12-05T04:48:39.359Z"   } ]</pre>	

<b>Schemas</b>	



```
Update {
  recordId [...]
  licensePlate [...]
}
```

```
ReissueResponseDto {
  accessToken [...]
}
```

```
ModelPredictResult {
  modelType [...]
  predictedText [...]
  accuracy [...]
}
```

```
MultiResponseDto {
  record RecordResponseDto {...}
  predictedList [...]
}
```

```
PredictedPlateDto {
  modelPredictResult ModelPredictResult {...}
  isEnrolled [...]
}
```

```
RecordResponseDto {
  recordId [...]
  modelType [...]
  licensePlate [...]
  accuracy [...]
  vehicleImage [...]
  plateImage [...]
  state [...]
}
```

```
SignUp {
  memberId* [...]
  password*  [...]
}
```

```
MemberResponseDto {
  accessToken  [...]
  refreshToken  [...]
}
```

```
SignIn {
  memberId*  [...]
  password*  [...]
}
```

```
HistoryResponseDto {
  historyId  [...]
  recordId  [...]
  memberId  [...]
  taskType  [...]
  previousText  [...]
  currentText  [...]
  createDate  [...]
}
```

```
Delete {
  recordId  [...]
}
```